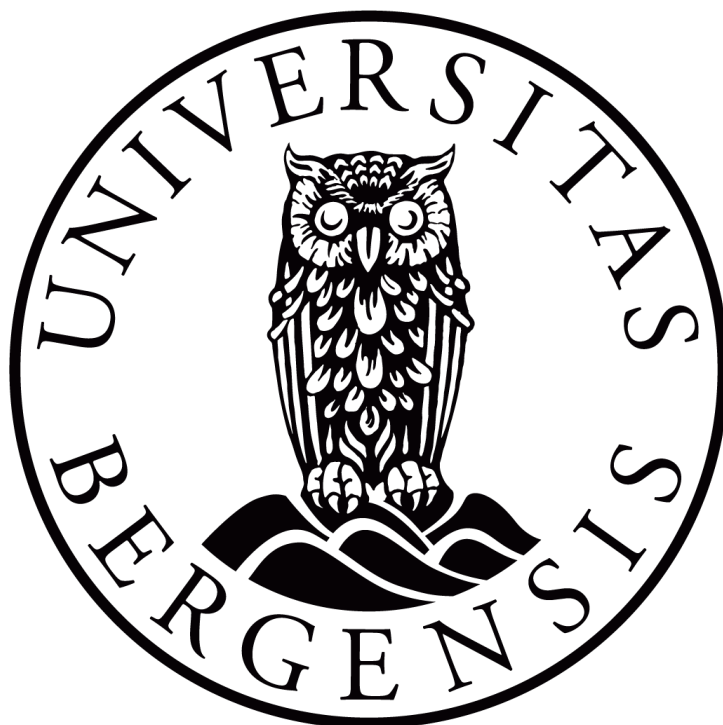


Status for innføring av algoritmisk tenking og programmering i matematikk

Ein kvantitativ studie av vidaregåande elevar i matematikk R1 sine meiningar, haldningar og erfaringar med innføringa av algoritmisk tenking og programmering i matematikk

Emil Heggstad Thune



Masteroppgåve i erfaringsbasert master i undervisning med fordjuping i matematikk.

Matematisk institutt

UNIVERSITET I BERGEN

Forord

Etter fire lærerike år ved Universitetet i Bergen avsluttar eg studiet mitt med denne masteroppgåva. Eg har både gleda meg og grua meg til å gjennomføre denne delen av studie. Både med tanke på arbeidsmengde og tidsklemma i ein travel kvardag, men også med tanke på at det er så mykje eg gjerne skulle ha forska på.

Algoritmisk tenking med programmering og matematikk har dei siste 15 åra vore noko av det som interesserer meg mest. Å kunne kombinere algoritmisk tenking og programmering med matematikk for å lære og forstå nye problemløysingsmetodar, omgrep og/eller samanhengar er utruleg interessant og gøy. Det var med stor glede, for min eigen del vertfall, då eg første gong høyrde at algoritmisk tenking og programmering skulle inn i matematikkfaga i den norske skulen. På bakgrunn av dette har eg valt å skrive om «Status for innføring av algoritmisk tenking og programmering i matematikk».

Studiet «Erfaringsbasert master i undervisning med fordjuping i matematikk» har gjeve meg fagleg påfyll, gode diskusjonar med både medstudentar og professorar, latter, frustrasjon og mykje meir. Det kjem til å bli eit sakn at eg ikkje lenger skal vere ein del av det fantastiske miljøet, som har vore på alle samlingane me har hatt.

Spesielt takk til Johan Lie, som har rettleia meg gjennom denne masteroppgåva. Me har hatt gode samtalar, og utveksla både idear og ytringar undervegs i prosessen.

Tusen takk for alt du har bidrege med.

Eg ynskjer også å takke alle mine kollegaer, venner og familie som har bidrege til dette prosjektet. Her er det fleire som har hjelpt meg undervegs og har påverka det endelege resultatet i ein positiv retning.

Dei to siste åra, som har gått til dette prosjektet, har vore travle både i heimen og på arbeid. Med korona overalt, og to små ungar i huset, har det vore alt anna enn ein normal kvardag. Derfor må eg rette ein stor takk til min sambuar og mine to små barn, som har halde ut med meg i denne perioden.

Emil Heggstad Thune

Bergen, 30.mai 2023

Samandrag

Denne masteroppgåva seier noko om korleis innføringa av algoritmisk tenking og programmering i matematikk har gått for elevar som har valt realfagsmatematikk i den vidaregåande skulen. Oppgåva fokuserer på kva elevane har av haldningar, meiningar, erfaringar og kompetanse knytt til algoritmisk tenking og programmering. For å svare på dette er det gjennomført ein kvantitativ spørjeundersøking med totalt 260 elevar i matematikk R1-klassar frå store delar av landet. Elevane har svara på spørjeundersøkinga i slutten av skuleåret, som betyr at dei har følgd LK20 i to heile skuleår. I analysen av data som er samla inn, viser det seg at elevane har fått ei grei forståing av fleire grunnleggande prinsipp i programmering og at omtrent halvparten av elevane klarer å løyse ei oppgåve som krev at dei kan setje seg inn i ein algoritme i eit program skrive i Python. Sjølv om elevane gjev tilbakemelding om at algoritmisk tenking og programmering er lite prioritert, så er dei fleste positive til kompetansen og har danna seg ei brei, men variert, forståing av kva som inngår i kompetansen. Elevane svarar at den mest vanlege metoden for å lære seg kompetansen i matematikkfaget er å lese og tolke kode frå eksempel i lærebok eller frå lærar.

Det er vidare sett på forskjellar i haldningar, meiningar, erfaringar og kompetanse mellom kjønn, prestasjonsnivå (i matematikkfaget), med/utan forkunnskapar frå ungdomsskulen og motivasjon for å lære seg algoritmisk tenking og programmering. Til slutt er det gjort ein korrelasjonsanalyse mellom variablane som er definert ut frå spørsmåla i undersøkinga.

Innholdsliste

Forord	2
Samandrag	3
Innholdsliste	4
1 Innleiing	6
1.1 Val av tema.....	6
1.2 Problemstilling og forskingspørsmål	6
1.3 Aktualitet og tidlegare forskning.....	7
2 Teori	9
2.1 Omgrep.....	9
2.2 Algoritmisk tenking og programmering i LK20	12
2.3 Lære algoritmisk tenking i matematikk	15
2.4 TIMSS Advanced 2015	18
2.5 Innføring av algoritmisk tenking og programmering i læreplanar i Europa (og Israel). 18	
2.6 Rammeverk for kompetansar innan algoritmisk tenking	19
3 Metodologi	24
3.1 Val av metode.....	24
3.2 Kvantitativ spørjeundersøking	25
3.3 Utval	32
3.4 Eksamensdata	34
3.5 Analyse av kvantitative data	35
3.6 Validitet.....	37
3.7 Reliabilitet	39
3.8 Etikk	39
3.9 Kritikk og feilkjelder.....	42
4 Resultat.....	43
4.1 Om informantane.....	43
4.2 Meiningar	45

4.3 Haldningar	48
4.4 Erfaringar	49
4.5 Kompetanse	52
4.6 L��rarunders��kinga	54
4.7 Data fr�� eksamen v��ren 2022	58
5 Analyse og dr��fting av inndelingar	62
5.1 K��nnforskjellar	62
5.2 H��gt presterande og l��gt presterande	65
5.3 Geografi.....	67
5.4 Med og utan forkunnskapar.....	68
5.5 Motivasjon for �� l��re algoritmisk tenking og programmering	71
5.6 Andre samanhengar.....	74
6 Diskusjon.....	76
6.1 Meininger og haldningar	76
6.2 Erfaringar	77
6.3 Kompetanse	78
6.4 Kva med l��rarane?.....	79
7 Avslutning	80
7.1 Status for innf��ring av algoritmisk tenking og programmering.....	80
7.2 Vegene vidare	81
8 Litteraturliste	82
9 Vedlegg	87
9.1 Elevunders��kinga	87
9.2 L��rarunders��kinga	95
9.3 Transkribert fokusgruppeintervju.....	100
9.4 Oppsummering av transkribering.....	105
10 Figurliste.....	106
11 Tabelliste	107

1 Innleiing

1.1 Val av tema

Innføringa av Kunnskapsløftet 2020 har vist seg å bli utfordrande. I mars 2020, då mykje av tida skulle gå til forarbeid med innføring av LK20, kom koronapandemien til Noreg for fullt. Med full nedstenging av samfunnet, vart det ei stor omvelting i den norske skulen. Skulane gjekk frå fysisk undervisning til full digital undervisning på nokre få døger. Denne store endringa og uvissa som kom med pandemien, gjorde antakeleg eit stort jafs i forarbeidet med innføringa av LK20.

Med ei tydeleggjering av algoritmisk tenking og programmering si rolle i matematikkfaga, var det mange matematikklærarar i landet som var, og fortsett er, usikre på korleis ein skulle få til dette. Fleire gav uttrykk for at det er vanskeleg å lære vekk noko, når ein ikkje kan det sjølv eingong. Sjølv om mange fekk opplæring og kurs, både lokalt, sentralt og gjennom andre instansar, med og utan studiepoeng, så var nok dei fleste spent på korleis det skulle gå med innføringa av algoritmisk tenking og programmering i matematikkfaga. Så, korleis har det gått med innføring av algoritmisk tenking og programmering i matematikkfaga? Dette er noko eg sjølv også er veldig nysgjerrig kring, og det er derfor eg ynskjer å utforske dette i denne masteroppgåva. Det nyskapande med mi forskning er at eg har vald å spørje elevane korleis dette har gått. I følge Cresswell (2014, s. 78) kan det å forske på noko som er lite forska på, og som samstundes får fram ei stemme til dei som ikkje har blitt høyrd, bidra til ny kunnskap til didaktisk forskning.

1.2 Problemstilling og forskingspørsmål

Målet med mitt masterprosjekt er å undersøke korleis det har gått med innføringa av algoritmisk tenking og programmering i realfagsmatematikkfaga i den vidaregåande skulen etter innføring av den nye læreplanen LK20. Målet er at elevane sjølve kan gje uttrykk for korleis innføringa har gått, samstundes som eg får anledning til å sjå eit større bilete basert på alle data eg har samla inn i prosjektet. Det har tidlegare vore gjort fleire datainnsamlingar knytt til læraren sitt perspektiv på dette temaet (Frantsen, 2019; Finstad, 2020; Hirt, 2020), men få har hatt fokus på elevane sine synspunkt og oppfatningar.

Eg ynskjer å forske på

(P1) **«Status for innføring av algoritmisk tenking med programmering som verktøy i matematikk i den vidaregåande skulen med fokus på realfagsmatematikkfaga»**

I tillegg til problemstillinga har eg laga meg tre forskarspørsmål knytt til dette.

- (F1) **Kva meiningar og haldningar har elevane knytt til algoritmisk tenking og programmering i matematikkfaga?**
- (F2) **Kva erfaringar har elevane gjort seg med algoritmisk tenking og programmering i matematikkfaga?**
- (F3) **Kva kompetanse har elevane tileigna seg i forhold til algoritmisk tenking og programmering?**

Ei god problemstilling er kjenneteikna ved at ein klarer å formulere eit presist spørsmål med klart og tydeleg språk og mål (Cresswell, 2012, s.66). Det er også viktig at den er avgrensa og at den er mogleg å forske på, noko eg meiner å ha fått til her.

For å finne svar på desse spørsmåla har eg samla inn kvantitative data ved hjelp av eit spørjeskjema som eg har distribuert til skular over heile landet. I tillegg har eg fått tak i statistikk frå eksamenssvar knytt til algoritmisk tenking og programmering frå Utdanningsdirektoratet. Dette er statistikk frå eksamen som vart gjennomført våren 2022, for privatistar.

1.3 Aktualitet og tidlegare forskning

Sidan LK20 berre nyleg har blitt innført, vil mine forskingsspørsmål vere av akademisk og didaktisk interesse for dei som arbeider med opplæring, spesielt i matematikk, i den vidaregåande skulen. Eg finn ingen fagartiklar som har sett på spørsmåla som eg stiller, og eg finn heller ingen statistikk frå det offentlege som er retta mot elev eller lærarar. Søket på dette er gjort blant anna på forskning.no, utdanningsforskning.no og udir.no.

Det ein finn av tidlegare forskning, er det som er gjort i tidlegare masteroppgåver. Her er det fleire som er interessert i lærarsynet på innføringa av algoritmisk tenking og programmering. Mathias Finstad har i si masteroppgåve i matematikkdidaktikk sett på «Kva er føremålet med å integrere algoritmisk tenking og programmering i matematikkfaget, og kva tenkjer matematikklærarane i den vidaregåande skulen om denne integreringa?» (Finstad, 2020). I denne oppgåva ser Finstad på korleis sju lærarar stiller seg til dei nye læreplanane med tanke på algoritmisk tenking og programmering. Studien er gjennomført som ein kvalitativ studie med intervju av lærarane.

I ei anna studie, som er gjort av Torbjørn Frantsen, er det fokus på «Å vere lærar i programmering utan å kunne programmere» (Frantsen, 2019). Oppgåva ser nærmare på korleis

lærarar ser på si eiga rolle i undervisning av programmering, utan sjølv å ha formell programmeringskompetanse. I tillegg ser oppgåva på «rolleutøving og forventingar om utbytte av undervisninga.» (Frantsen, 2019, s.5).

Ein studie som har fokus på elevperspektiv, og erfaringar knytt til algoritmisk tenking og programmering, er Stian Hirth si masteroppgåve om «Programmering som inngangsport til algoritmisk tenking» (Hirth, 2020). Hirth har gjennomført studien ved å nytte seg av ei spørjeundersøking før og etter at programmeringsundervisninga starta. I tillegg har forfattaren gjennomført eit forskingsintervju med fire av elevane. Elevane som vart forska på i denne studien var elevar frå matematikk 1P. Målet med masteroppgåva var å sjå på «deira erfaringar med programmering før undervisninga tek til, og deira forståing av programmering og syn på programmering etter at undervisninga som inkluderte programmering vart avslutta» (Hirt, 2020, «Samandrag»)

Olaug Andersen har i si masteroppgåve skrive om «Programmering si rolle i matematikkfaget» med elevperspektiv. I denne oppgåva har ho undersøkt elevar sine kjensler og tankar kring programmering si rolle i matematikkfaget. Dette er ein kvalitativ studie som er gjort av eigne elevar i ein matematikk 1T-klasse i skuleåret 2021/2022. Forfattaren konkluderer med at «..den brå innføringa av programmering for elevar på vidaregåande som ikkje har programmert før gjere at mange elevar ikkje likar programmering, fordi dei ikkje forstår det.» (Andersen, 2022, s.57). Det er likevel konkludert med at elevane uttrykker at programmering passar inn i matematikkfaget, og at dei ser koplinga mellom denne kompetansen og matematikk (Andersen, 2022, s.58). Omgrepet programmering er i denne masteroppgåva nært knytt til den algoritmiske tenkaren frå utdanningsdirektoratet, meir om denne i kapittel 2.2.1.

2 Teori

2.1 Omgrep

For at me skal ha same forståing av viktige og sentrale omgrep, vil eg i dette delkapittelet gje ei forklaring til desse. Viktige og sentrale omgrep i denne oppgåva er *computational thinking*, *algoritmisk tenking* og *programmering*.

2.1.1 Computational thinking

Omgrepet *Computational thinking* er definert og diskutert av fleire (Papert, 1980; Wing, 2006; Shute, 2017; Aho, 2010; Balanskat & Engelhardt, 2015; Grover & Pea, 2013). Papert er blant anna kjent for å vere den fyrste som brukte omgrepet *Computational Thinking* i boka si *Mindstorms: children, computers, and powerful ideas* frå 1980. Papert var også sentral i utviklinga av LOGO, eit programmeringsspråk tilpassa til bruk i utdanning.

«LOGO environments are not samba schools, but they are useful for imagining what it would be like to have a 'samba school for mathematics'»

(Papert, 1980, s.182)

Papert skildrar her LOGO som eit verktøy som kan skape 'samba'-skule i klasserommet, som forenkla kan tolkast som at det vil skape stor motivasjon og engasjement rundt matematikkproblem.

«Their vision of how to integrate computation thinking into everyday life was insufficiently developed. But there will be more tries, and more and more. And eventually, somewhere all the pieces will come together, and it will 'catch'»

(Papert, 1980, s.182)

Papert fekk heilt riktig i at dette har blitt populært, når det i dag har ein sentral plass i læreplanar over heile verda (Balanskat & Engelhardt, 2015).

I nyare tid har Wing skildra meir kva definisjonen av *Computational Thinking* er, og i sitatet under nemner Wing blant anna kompleksiteten i tenking på fleire nivå av abstraksjon samanlikna med å programmere ei datamaskin (eng: «to program a computer»).

«Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction. »

(Wing, 2006, s.34).

Balanskat & Engelhardt (2015) byggjer på same ide som Wing (2006), og skildrar *Computational Thinking* som «...is typically associated with coding and computer programming, but is more than that, involving ‘solving problems, designing systems, and understanding human behaviour’, according to the Carnegie Mellon University. » (Balanskat & Engelhardt, 2015, s.7)

Det at computational thinking er noko meir en koding og programmering er viktig for vidare definisjon av computational thinking. Å løyse gitte problem, lage system (og moglege algoritmar) og forstå menneskeleg oppførsel er også ein viktig del av computational thinking i følgje Carnegie Mellon University. Shute (2017) sin definisjon støttar opp under dette og definerer computational thinking på denne måten:

«We define CT as the conceptual foundation required to solve problems effectively and efficiently (i.e. algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts. »

(Shute et al., 2017, s.151).

Shute forklarar her computational thinking som noko grunnleggande for å kunne løyse problem effektivt og nøyaktig, og at ein kan *algoritmisk* løyse problem både med og utan datamaskiner. I tillegg skal løysinga på eit problem kunne generaliserast sånn at det kan brukast i fleire situasjonar. Aho (2010) ser også på computational thinking og algoritmar saman:

«We consider computational thinking to be the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms. »

(Aho, 2010, s.832)

Aho (2010) presiserer at computational thinking er ein tankeprosess, og at denne tankeprosessen skal lede fram til ei løysing som kan presenterast som ei stegvis løysing og ein algoritme.

Omsetjing av computational thinking til norsk har gitt oss omgrepet *algoritmisk tenking*. Dette omgrepet blir diskutert i neste delkapittel. I Sverige har dei vald å omsetje computational thinking til omgrepet *datalogiskt tänkande*. I artikkelen *Programmering från början* (Gratis i Skolan, 2018) blir denne definisjonen av datalogiskt tänkande gitt:

«(Datalogiskt tänkande) Det är en problemlösningsprocess för att beskriva, analysera, och lösa problem för att kunna ta hjälp av datorer. Det innebär att hitta mönster, generalisera (dra slutsatser efter vad vi vet), abstrahera (ta fram det viktiga) och skapa algoritmer för lösningar (skriva så datorerna förstår vad vi menar). »

(Gratis i Skolan, 2018)

Denne skildringa av datalogiskt tänkande er meir spissa inn mot at det er ein prosess for å kunne utnytte (sve: «ta hjelp av») datamaskiner. Elles er omgrepet definert mykje likt som *algoritmisk tenking* er definert av utdanningsdirektoratet (2019).

2.1.2 Algoritmisk tenking

Utdanningsdirektoratet har kome fram til ein definisjon av omgrepet *algoritmisk tenking*, og skildrar dette slik:

«*Algoritmisk tenkning innebærer å tilnærme seg problemer på en systematisk måte, både når vi formulerer hva det er vi ønsker å løse og når vi foreslår mulige løsninger. Litt forenklet kan vi si at det er 'å tenke som en informatiker' når vi skal løse problemer eller oppgaver.*»

(Utdanningsdirektoratet, 2019a, ingress)

Algoritmisk tenking er ifølgje Utdanningsdirektoratet ein metode for å løyse problem systematisk. Eg vil komme nærare inn på algoritmisk tenking slik det er skildra i læreplanane for realfagsmatematikk i kapittel 2.2.2.

Omgrepet algoritmisk tenking, som ofte blir samanlikna med computational thinking, er også likt definisjonen til Futschek (2006) av det engelske omgrepet *algorithmical thinking*. Algorithmical thinking blir forklart som ein samansett kompetanse, som blant anna består av å kunne analysere og spesifisere eit problem, finne grunnleggande handlingar for å løyse problemet, lage algoritme for å løyse problemet (ved å bruke dei grunnleggande handlingane), tenkje gjennom alle variantar av problemet og det å kunne optimalisere ein algoritme (Futschek, 2006, s. 160).

2.1.3 Programmering

«Computer programming is the process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve problems, and provide human interactivity. »

(Balanskat & Engelhardt, 2015, s.7)

Definisjonen av *Computational programming*, som ofte berre er oversett til programmering, har fleire like eigenskapar som deler av kompetansen algoritmisk tenking. Som ein kan lese ut frå definisjonen til Balanskat og Engelhardt, er programmering heile prosessen frå å utvikle og implementere alle delar av algoritmen og instruksjonane, sånn at datamaskina kan utføre og løyse eit gitt problem.

Ein artikkel skrevet av Lynnebakken (2018) viser skilnadar og koplingane mellom blant anna algoritmisk tenking, programmering og koding. Lynnebakken observerte eit vidareutdanningskurs for lærarar, som vart leia av to tilsette frå Universitetet i Oslo, Knut Mørken og Cathrine Tellefsen. Tellefsen var på dette tidspunktet leiar for utdanning av realfagslektorar ved UiO. Første leksjon for dagen var «Programmering er ikke det same som koding», og vert utdjupa ved å forklare at koding er berre ein liten del av programmeringa. «Viktigere er algoritmisk tenkning, måten en må tenke på for å oversette et problem uttrykt i ord til en logisk rekke av operasjoner som kan kodes inn i en datamaskin.» (Lynnebakken, 2018, 2.avsnitt). Vidare forklarar Tellefsen fordelane med programmering, og det å lage eit dataprogram, og trekker fram at «for å klare å programmere formelen, må elevene virkelig forstå likningen. ... kan elevene eksperimentere med formelen og for eksempel finne ut ...» (Lynnebakken, 2018, 11.avsnitt)

2.2 Algoritmisk tenking og programmering i LK20

Kunnskapsdepartementet kom i 2016 med ei tilråding til Stortinget (Meld. St. 28 (2015–2016)), som skulle vise seg å bli starten på fagfornyninga av Kunnskapsløftet i skulen, også kjend som Kunnskapsløftet 2020, eller berre LK20. Tilrådinga byggjer blant anna på NOU 2014: 7 *Elevens læring i fremtidens skole – Et kunnskapsgrunnlag* og Ludvigsenutvalget sin sluttrapport NOU 2015: 8 *Fremtidens skole: fornyelse av fag og kompetanser*, der djupnelæring er særleg lagt vekt på.

Matematikkfaget er sjølvstøtt nemnd fleire gonger i tilrådinga, men ein finn ikkje algoritmisk tenking eller programmering i denne tilrådinga. Det nærmaste ein kjem er

«Digitale beregningsverktøy medfører betydelige endringer i innholdet i skolefaget matematikk og andre realfag..»

(Meld. St. 28 (2015–2016), s.44)

I dette kan det ligge ei føring mot algoritmisk tenking og programmering, men det er ikkje noko tydeleg føring.

I den vidare prosessen vart det jobba med kjerneelementa i faga. Skisser vart delt, og lagt ut på høyring. Det kom så mange som 6700 tilbakemeldingar i denne prosessen (Utdanningsdirektoratet, 2021). Resultatet vart eit dokument som summerer kjerneelementa i faga. Dokumentet skulle gje «føringer for utforming av læreplaner for fag til LK20» (Kunnskapsdepartementet, 2018, s.1). I dette dokumentet finn me tydelege føringar mot algoritmisk tenking og programmering som ein del av matematikkfaget.

«Algoritmisk tenking er viktig i prosessen med å utvikle strategier og fremgangsmåter og innebærer å kunne bryte ned et problem i delproblem som kan løses systematisk»

(Kunnskapsdepartementet, 2018, s. 15)

I pressemeldinga (Regjeringa Solberg, 2018) kjem det også fram at «Programmering og algoritmisk tankegang blir også en del av faget.» under matematikkdelen. Så på dette tidspunktet er det ingen tvil om føringane for at algoritmisk tenking og programmering skal inn i læreplanane i matematikkfaga.

Den neste delen av arbeidet var utvikling av læreplanane i kvart einskilde fag. Denne omfattande prosessen engasjerte mange faglege personar kring heile landet. Totalt 14 000 innspel kom i denne runden (Utdanningsdirektoratet, 2021).

18. november 2019 var læreplanane fastsett og lansert av Kunnskapsdepartementet. I forhold til vidaregåande og studieforbereidande utdanningsprogram, vart det fastsett læreplanar for vg1 2.april 2020 (blant anna i matematikk). Vidare for vg2 og vg3 som vart fastsett i 2021 (blant anna i matematikk). (Utdanningsdirektoratet, 2021)

2.2.1 Den algoritmiske tenkjaren

Utdanningsdirektoratet skildrar «den algoritmiske tenkjaren», og forklarar algoritmisk tenking som «ein problemløysingsmetode» (Utdanningsdirektoratet, 2019).



(Utdanningsdirektoratet, 2019)

Figuren over, som er ei fornorska utgåve av Barefoot Computing (UK) sin figur av *The Computational Thinkers*, er ein oversikt over dei viktigaste nøkkelomgrepa og arbeidsmetodane som inngår i algoritmisk tenking (Barefoot Computing, 2022). Det står blant anna at algoritmisk tenking er å bryte ned komplekse problem til mindre, meir handsame delproblem (*dekomposisjon*). Vidare forklarar utdanningsdirektoratet dei ulike omgrepa og arbeidsmetodane rundt denne algoritmiske tenkjaren. Å vere ein algoritmisk tenkjar er krevjande, hen «...må vere systematisk og analytisk i sitt arbeid, men må vere skapande, eksperimenterande og open for alternative løysingar» (Utdanningsdirektoratet, 2019)

2.2.2 Kompetansemål i realfagsmatematikk for den vidaregåande skulen

I LK20 finn me algoritmisk tenking og programmering blant anna i begge dei to læreplanane for matematikk 1T, R1 og R2. I læreplan for 1T (MAT09-01, gyldig frå 01.08.2020) står det under *Kjerneelement* at

«Algoritmisk tenking er viktig i prosessen med å utvikle strategiar og framgangsmåtar for å løyse problem og inneber å bryte ned eit problem i delproblem som kan løysast systematisk.»

(Utdanningsdirektoratet, 2019b, s.2).

Programmering er nemnd under *Digitale ferdigheter* i læreplanen, og er skildra som at det «...inneber å kunne bruke grafteiknar, CAS, dynamisk geometriprogram og programmering til å utforske og løyse matematiske problem» (Utdanningsdirektoratet, 2019b, s.4).

Algoritmisk tenking og programmering dukkar også opp i eit spesifikt kompetansemål for matematikk 1T. «Mål for opplæringa er at eleven skal kunne formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering.» (Utdanningsdirektoratet, 2019b, s.5). Her kunne kanskje Utdanningsdirektoratet satt punktum etter algoritmisk tenking, sidan både problemløysingsstrategiar, digitale verktøy og programmering inngår i deira definisjon av algoritmisk tenking.

I læreplanen for realfagsmatematikk (MAT03-02) finn ein algoritmisk tenking i *Kjerneelement* for faget på same måte som for T-matematikk. «Algoritmisk tenking er viktig i prosessen med å utvikle strategiar og framgangsmåtar for å løyse problem og inneber å bryte ned eit problem i delproblem som kan løysast systematisk.» (Utdanningsdirektoratet, 2020a, s.2).

I matematikk R2 er algoritmisk tenking og programmering nemnd i eit spesifikt kompetansemål. Denne gongen er kompetansemålet retta inn mot integralrekning som eit spesifikt tema. «Mål for opplæringa er at eleven skal kunne utvikle algoritmar for å regne ut integral numerisk, og bruke programmering til å utføre algoritmane.» (Utdanningsdirektoratet, 2020a, s.6).

2.3 Lære algoritmisk tenking i matematikk

«To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability»

(Wing, 2006, s.33)

Wing (2006) peikar på algoritmisk tenking som noko alle elevar bør tileigne seg, på lik linje med dei grunnleggjande kompetansane lesing, skriving og rekning. Det at ansvaret for opplæring i algoritmisk tenking er lagt til matematikkfaget, er ikkje sjølvstøtt og har vore mykje diskutert.

Det er fleire spørsmål som går igjen blant matematikklærarar i forhold til å lære elevane opp i algoritmisk tenking i matematikkfaga. For eksempel undrar dei seg over korleis ein kan lære vekk noko ein ikkje har kompetanse i sjølv (Frantsen, 2019), korleis lærer ein seg algoritmisk tenking og korleis skal ein undervise i dette? Dei to siste spørsmåla har Hsu, Chang og Hung (2018) undersøkt. Dei har laga ein artikkel som prøver å gje svar på desse spørsmåla gjennom ein litteraturgjennomgang av forskingsartiklar på området.

Ein av studiane dei har sett på, er studien til Denning (2017) med tittel «Remaining trouble spots with computational thinking». I studien problematiserer Denning lærarane sine velkjende undervisningsmetodar, og at det difor er vanskeleg for dei å endre/tilpasse undervisninga til å fremje algoritmisk tenking (Hsu et al., 2018, s. 297). Denne velkjende undervisningsmetoden er skildra som opplæring i å forstå og bruke formlar, og at elevane skal «be able to come up with the correct answer in the examination» (Hsu et al., 2018, s.297).

Blant rammeverk for skildring av kompetansen algoritmisk tenking er Brennan og Resnick (2012) ein av dei mest brukte (Ye et al., 2023). Skildring av dette rammeverket vil eg komme tilbake til i kapittel 2.6.

2.3.1 Faktorar som påverkar læring av matematikk og algoritmisk tenking

Det er eit utal faktorar som påverkar læring i skulen. I dette delkapittelet tek eg føre meg faktorar som kan påverke eleven si læring i matematikk, algoritmisk tenking og programmering. Når det gjeld læring i matematikk vil eg vise til «Attitudes, Beliefs, Motivation, and Identity in Mathematics Education» (Hannula et al., 2016). Dette er eit kapittel som er ein del av ei bok skriva og sett saman av Markku S. Hannula. Forfattaren har tatt kontakt med ekspertar på diverse områder knytt til påverknad av matematikkundervisning, og fått dei til å skriva eit samandrag knytt til si eiga forskning. Hovuddelen av kapittelet er delt inn i fem delar.

1. *Attitude* – oversett til haldning – skriva av Pietro Di Martino
2. *Student Self-efficacy Beliefs* – oversett til studenten sine oppfatningar – skriva av Marilena Pantziara
3. *Teacher Beliefs* – oversett til læraren sine oppfatningar – skriva av Qiaoping Zhang og Francesca Morselli
4. *Identity* – oversett til identitet – skriva av Einat Heyd-Metzuyanin, Sonja Lutovac og Raimo Kaasila
5. *Motivation* – oversett til motivasjon – skriva av James A. Middleton, Amanda Jansen og Gerald A. Goldin

Pietro Di Martino foreslår ein modell for haldningar beståande av tre dimensjonar som er med på å bestemme studenten sitt forhold til matematikk. Denne modellen består av ein emosjonell dimensjon, synet på matematikk og eiga oppfatning av å kunne lykkast, og at desse påverkar kvarandre (Hannula et al., 2016, s. 6). Vidare er det diskutert kompleksiteten av dette systemet, og at det er gjort meir forskning på feltet med denne modellen som utgangspunkt.

Marilena Pantziara startar i sitt delkapittel med å referere til Bandura si bok frå 1997 *Self-efficacy: The exercise of control* og viser til Bandura sin forklaring at *self-efficacy* som «beliefs in one's capabilities to organize and execute the courses of action required to produce given attainments» (Hannula et al., 2016, s. 7). Bandura forklarar her kjensla av meistringsevne (eng: *self-efficacy*) som studentens eiga oppfatning av i kva grad hen kan klare å utføre riktige steg for å prestere. Vidare i delkapittelet er det diskutert fleire relevante studie, som har sett på korleis studentane si eiga oppfatning endrar seg gjennom blant anna erfaringar og aldring, og at det er ei generell oppfatning blant forskarar på feltet at det er erfaringar frå faget som påverkar studenten si eiga oppfatning (Hannula et al., 2016, s. 9-11).

Zhang og Morselli summere *Teachers beliefs*, og startar med å konstatere at det ikkje er ei internasjonal aksept på ein definisjon av omgrepet *beliefs* (Hannula et al., 2016, s.11). Det er stor variasjon ved lærar sine oppfatningar, og kan på den eine sida vise til matematikk som noko statisk og formel-fokusert, medan på den andre sida eit dynamisk domene av kunnskap for å forstå og finne mønster (Hannula et al., 2016, s.12).

Heyd-Metzuyanım, Lutovac og Kaasila skriv om identitet og viser til at samanhengen mellom identitet og læring er kor delaktig studenten er (Hannula et al., 2016, s.14). Det å finne ut kva rolle ein har, og bestemme sin identitet, i klasserommet eller eit anna læringsmiljø, vil påverka korleis ein lærer i følgje fleire studie som er vist til (Hannula et al., 2016, s.15). På same måte får læraren sin eigen identitet, og denne kan forandre seg utifrå erfaringar og endringar knytt til yrket, sjølv om det er motvillig (Hannula et al., 2016, s. 16-17).

2.3.2 Samanheng mellom matematikk-kompetanse og læring av algoritmisk tenking og programmering

Studien til Pacheco et al. (2008) undersøker samanhengen mellom studentane sin matematiske kompetanse ved inntak til førsteåret på ei teknologilinje ved Universitetet i Coimbra og deira tileigna kompetanse i programmering i løpet av dette året. Pacheco et al. (2008) konkluderer med at studentar med låg problemløysingskompetanse relatert til matematiske metodar og modellar for problemløysing, er faktorar som påverkar deira evne til å tileigne seg kompetanse knytt til algoritmisk tenking og programmering.

«We concluded that the students that accessed the course with higher marks had better classifications in programming. »

(Pacheco et al., 2008, s.3)

Dei konkluderer med at dei med høgare karakterar i matematikk ved inntak hadde eit betre utgangspunkt til å lære seg programmering. Det kjem fram i studien at det er ein positiv korrelasjon på 0.373 mellom matematisk kompetanse ved inntak og den kompetansen dei viser i programmering i slutten av året.

2.4 TIMSS Advanced 2015

Sidan 1995 har TIMSS Advanced (Trends In International Mathematics And Science Study) samla inn data om trendar innan matematikk R2 og fysikk 2 i den vidaregåande skulen. Denne undersøkinga er gjennomført i 1995, 2008, 2015 og planlagd nye resultat i løpet av 2024. Grønmo, Hole og Onstad (2016) har satt seg grundig inn i resultatata til dei norske elevane i TIMSS Advanced 2015, og har summert sine analysar og funn i ei bok med tittelen «*Ett skritt fram og ett tilbake*».

Tidlegare rapportar frå både TIMSS, TIMSS Advanced og PISA har vist positiv korrelasjon mellom haldningar, sjølvoppfatning og faglege presentasjonar (Grønmo et. al, 2016, s. 87). Dette er noko eg også vil kunne sjå på i denne oppgåva. Om elevane «*Liker å lære matematikk*», som er ein samlevariabel på spørsmål om elevane sine haldningar og syn på faget, har 24% av elevane svart «*Veldig godt*», 55% «*Godt*» og 21% har svart «*Ikkje godt*» (Grønmo et al., 2016, s. 94). Gjennomsnittseleven uttrykker altså at hen likar godt å lære matematikk. I forhold til nytte av kva dei lærer i matematikk, har elevane svara alternativa «*Veldig nyttig*» og «*Nyttig*» med 40% og 43%, som viser at norske elevar ser nytten med matematikkopplæringa i matematikk R2 (Grønmo et al., 2016, s. 97). Det er også her ein korrelasjon mellom nytten eleven ser av kompetansen og fagleg prestasjonar (Grønmo et al., 2016, s.97).

Det er 38 % jenter og 62% gutar som har valt matematikk R2, og av alle jenter i vg3 er det berre 8,1 % som vel fordjuping i realfagleg matematikk (Grønmo et. al, 2016, s. 35).

2.5 Innføring av algoritmisk tenking og programmering i læreplanar i Europa (og Israel)

Sommaren 2015 gjennomførte «European Schoolnet» ei undersøking knytt til programmering i opplæringa i totalt 20 europeiske land og Israel (Balanskat & Engelhardt, 2015, s.5).

Rapporten til Balanskat og Engelhardt (2015) såg på spørsmål som «korleis er programmering integrert inn i læreplanar» og «korleis kan me vidareutvikle oss, slik at programmering, som er ein nøkkelferdighet, kan blomstre og stadig innovere det digitale samfunnet og økonomien».

Funna i rapporten er basert på eit spørjeskjema som er svarta på av utdanningsministrane til landa som deltok (Balanskat & Engelhardt, 2015, s.7).

Av dei 21 landa som deltok var det totalt 16 land, som på denne tida hadde integrert programmering i læreplanane for landet, regionen eller lokalt (Balanskat & Engelhardt, 2015, s.23). Noreg var ein av dei fem landa som ikkje hadde integrert programmering i sine læreplanar, på dette tidspunktet. Noreg er nemnt som ein av tre land, i tillegg til Belgia og Nederland, som ikkje hadde planar om å få programmering inn i læreplanane sine, men at dei hadde starta ein pilot i forhold til eit valfag i programmering i ungdomsskulen (Balanskat & Engelhardt, 2015, s.24). Dette er valfaget som er referert til i kapittel 4.1.2 (Utdanningsdirektoratet, 2016), og som er spurt om i spørjeskjemaet mitt til elevane under «Tidelegare erfaringar med programmering».

Rapporten viser også at land som Danmark, Estland, Finland, Frankrike og Spania har valt å integrere programmering i faget matematikk, medan andre land har det som eige fag (Balanskat & Engelhardt, 2015, s. 45).

2.6 Rammeverk for kompetansar innan algoritmisk tenking

For å kunne forske på algoritmisk tenking er det viktig med ei felles forståing av kva som inngår av kompetanse i omgrepet. Dette var ein av utgangspunkta når Brennan og Resnick starta utvikla av sitt rammeverk for algoritmisk tenking (Brennan & Resnick, 2012, s.1). Rammeverket deira byggjer blant anna på studie av elevar, som har brukt Scratch som verktøy for å lære seg algoritmisk tenking. Resultatet er ei tredeling av nøkkelkomponentar: *computational concepts* (konsept som programmerarar brukar), *computational practices* (praksis utviklaren nyttar under programmeringa), og *computational perspectives* (perspektiv som utviklaren forma om verda rundt dei og seg sjølv) (Brennan & Resnick, 2012, s. 3).

I tabellen på neste side er det ein oppsummering av nøkkelkomponenten *Computational concepts* til Brennan og Resnick, der eg har oversett kompetansane frå engelsk til norsk, og gjeve mi eiga tolking/forklaring av desse.

Nøkkelkomponent	Kompetanse	Tolking/forklaring
Computational concepts	Sekvensar (eng: sequences)	Rekkefølge på utføring av instruksjonar
	Løkker (eng: loops)	Køyring av same kodeblokker, fleire gonger
	Parallele instruksjonar (eng: parallelism)	Fleire sekvensar av kodeblokker blir køyrd samstundes
	Hending (eng: events)	Noko som blir køyrd som følgje av noko anna. For eksempel interaktivitet med knapp.
	Valsetning (eng: conditionals)	Utføring av bestemte kodeblokker basert på ei føresetnad.
	Operatorar (eng: operators)	Manipulering av tal og tekst ved hjelp av operasjonar.
	Data (eng: data)	Lagring, henting og endring av verdiar. For eksempel til ein variabel eller eit array.

Tabell 1: Sammenheng av nøkkelkomponenten *Computational concepts* (Brennan & Resnick, 2012, s. 3-6)

Desse kompetansane som er nemnt under *Computational concepts*, kan reknast som basiskompetanse for å kunne programmere og forstå kode. Med desse grunnleggande prinsippa på plass, og ei god forståing av desse, kan gjere at ein enklare utviklar sin kompetanse i algoritmisk tenking. Neste nøkkelkomponent i Brennan og Resnick sitt rammeverk er praksisar som vert nytta i algoritmisk tenking, som frå forfattarane er forklart som korleis ein lærer og ikkje berre kva (Brennan & Resnick, 2012, s.3). Under har eg oversett frå engelsk til norsk, og summera *computational practices*. Forklaringa er igjen mi eiga tolking av Brennan og Resnick si skildring av kompetansane.

Nøkkelkomponent	Kompetanse	Tolking/forklaring
Computational practices	Iterativ og stegvis løysing (eng: being incremental and iterative)	Å utvikle ei løysing krev fleire iterative prosessar, der ein tilpassar seg undervegs og byggjer vidare på det ein har fått til i førre prosess.
	Testing og feilsøking (eng: testing and debugging)	Sjeldan verkar alt på første forsøk, så det er viktig å ha metodar for å teste og feilsøke i løysinga.

	Gjenbruk og samansetting av delprogram (eng: reusing and remixing)	Å bruke delar av andre eller egne løysingar, og deretter sette saman til ei løysing av eit nytt problem.
	Abstraksjon og modularisering (eng: abstracting and modularizing)	Bryte ned problemet i mindre delar, løyse desse og lage modular som ein set saman til slutt.

Tabell 2: Sammen drag av nøkkelkomponenten *Computational practices* (Brennan & Resnick, 2012, s. 6-9)

Litt overraskande, i følge forfatarane, oppdaga Brennan og Resnick også ein tredje dimensjon av kompetansen algoritmisk tenking, nemleg *Computational Perspectives*. I tabellen under har eg igjen oversett frå engelsk til norsk, og lagt på mi tolking og forklaring av dei forskjellige kompetansane.

Nøkkelkomponent	Kompetanse	Forklaring
Computational perspectives	Uttrykkande (eng: expressing)	Med kompetanse i algoritmisk tenking kan ein uttrykke seg på nye måtar ved å for eksempel produsere eit program eller ein animasjon.
	Samarbeid (eng: connection)	Ser nytte av å bidra, dele, diskutere, lage, tilpasse og mykje meir saman med andre med interesse for algoritmisk tenking og for eksempel utvikling av program eller animasjon.
	Undring (eng: questioning)	Det å forstå og kunne stille spørsmål ved teknologi, verkemåten og bruksområda deira.

Tabell 3: Sammen drag av nøkkelkomponenten *Computational practices* (Brennan & Resnick, 2012, s. 10-11)

Denne siste nøkkelkompetansen gjer at ein kan uttrykke seg på ein ny måte ved at ein ser moglege bruksområde, tenkjer kreativt og blir betre av å arbeide og dele saman med andre. På dette nivået vil ein kunne ha god nytte av kompetansen algoritmisk tenking, og kan løyse problem og utfordringar med eit nytt verktøy som opnar for nye, enklare, raskare og meir effektive løysingar i fleire tilfelle.

2.6.1 Algoritmisk tenking i matematikk og realfag

Kompetansen algoritmisk tenking i matematikk og realfag kan delast i fire hovudkategoriar i følge Weintrop et al. (2016). Kvar av dei fire hovudkategoriane består av mellom fem og sju praksisar, som kan verke åtskilt, men som i realiteten heng tett saman med kvarandre (Weintrop

et. al, 2016, s. 134). Grunnen til at ein nyttar seg av omgrepet praksisar framfor *skills* eller *concepts* er for å understreke at ein ikkje berre treng ferdigheit, men også kunnskap om kvar enkelt praksis (Weintrop et. al, 2016, s.134). Det er ikkje nok å lære seg ein ferdigheit eller eit konsept, men ein må tileigne seg kompetanse og kunnskap til korleis og når ein kan nytte seg av dette.

Data practices «består av å samle inn, manipulere, lage nye, analysere og visualisere data» (Weintrop et. al, 2016, s.136, oversett frå engelsk). Sjølv om data lenge har vore ein sentral del av matematikkfaga og dei andre realfaga, så er det behov for ein meir og djupare forståing av behandling av store data, og ikkje minst vidareformidling av behandla data. Studentar som har tileigna seg denne kompetansen «will be able to use computational tools to produce visualizations that convey information gathered during analysis» (Weintrop et. al, 2016, s.136).

Modelling and Simulation Practices «kan gjere vitenskaplege konsept meir tilgjengelege og auke studentar si forståing av fenomenet» (Weintrop et al, 2016, s.137, oversett frå engelsk). Modellar i denne samanhengen er enkle representasjonar av eit fenomen som kan simulert ved hjelp av ei datamaskin. Den pedagogiske gevinsten er «... not just from students using existing models, but also from enabling students to design, build, and assess models of their own.» (Weintrop et. al, 2016, s.137). Studentane får ikkje berre kjennskap og forståing av eksisterande modellar, men dei vil også auke kompetansen sin til å utforme, lage og bruke eigne modellar.

Computational Problem Solving Practices «byggjer meir enn dei andre kategoriane på praksisar og strategiar frå informatikk, og har som mål å bruke dette feltet sitt bidrag til aktuelle realfagleg og matematisk arbeid i tillegg til at det er eit viktig verktøy for elevane å lære seg» (Weintrop et. al., 2016, s. 138, oversett frå engelsk). Det er vist til fleire studie som støttar at å utforske vitenskaplege og matematiske fenomen ved bruk av programmering, algoritmeutvikling og abstraksjonar kan hjelpe i å få ei djupare forståing av matematiske og vitenskaplege fenomen (Weintrop et. al, 2016, s.138).

Systems Thinking Practices er kompetansen som gjer at ein kan «forstå eit komplekst system og korleis dette endrar seg over tid» (Weintrop et. al, 2016, s. 140, oversett frå engelsk). Systemtenking (eng: systems thinking) analyser korleis delane av systemet verkar saman og som ein heilheit (Weintrop et. al, 2016, s.140). Det å ha eit analytisk overblikk på større system og forstå korleis delar samhandlar er ein viktig kompetanse, som er viktig å tileigne seg. Denne kompetansen er delt inn i *Investigating a Complex system as a whole, Understanding the*

Relationships within a System, Thinking in Levels, Communicating Information about a System og *Defining Systems and Managing Complexity* (Weintrop et. al, 2016, s.140-142). Nedanfor kan ein sjå ein oversikt over taksonomien til Weintrop et al. (2016).

Data practices	Modelling and Simulation Practices	Computational Problem Solving Practices	Systems Thinking Practices
Samle inn data	Nytte seg av berekningsmodellar for å forstå eit konsept	Dekomponera og klargjere for kalkulerbar løysing	Undersøkje eit komplekst system
Lage data	Nytte seg av berekningsmodellar for å finne og teste løysingar	Programmering	Forstå samanhengar i eit system
Manipulere data	Vurdere berekningsmodellar	Velje effektive algoritmar og berekningsverktøy	Tenkje på fleire nivå
Analysere data	Komme fram til berekningsmodellar	Vurdere fleire løysingar til problemet	Forklare eit system
Visualisere data	Konstruere berekningsmodellar	Utvikle delløysingar som kan brukast på nytt	Definera system og takle kompleksitet
		Utføre berekningsabstraksjonar, generalisere	
		Finne og rette feil	

Tabell 4: Oppsummering av taksonomien til Weintrop et. al. (2016)

Berekningsmodellar (eng: computational models) er meint som modellar som eignar seg til å utforske ved hjelp av algoritmar og numerisk kalkulasjon. Kompetansane over er blant anna nytta i denne masteroppgåva til å diskutere kva elevane meiner inngår i kompetansen algoritmisk tenking.

3 Metodologi

Eg vil i dette kapitlet skildre korleis problemstillinga blir forska på. Val av metode for å forske på ein problemstilling er viktig for å kunne finne mest mogleg ut om problemstillinga. Metoden som er valt er ei kvantitativ spørjeundersøking, og det blir argumentert for val av metode ved bruk av blant andre John. W. Cresswell (2012) som kjelde. Det blir også diskutert korleis ein utviklar eit godt spørjeskjema, kva målenivå kvart spørsmål kan ha og korleis ein kan analysere kvantitative data. I tillegg til å gjennomføre ei kvantitativ spørjeundersøking, er det samla inn data/resultat frå oppgåver, som vart gitt på eksamen våren 2022, som er relevante til å svare på problemstillinga.

Det er forsøkt å vere så transparent som mogleg frå val av metode til data er samla inn og analysert. Det er også diskutert validitet, reliabilitet og etikk knytt til innsamlinga av data. Avslutningsvis er det kommentert noko kritikk og feilkjelder til datainnsamlinga og analysen.

3.1 Val av metode

Fokuset for denne masteroppgåva er på korleis elevane sjølve har erfart «Status for innføring av algoritmisk tenking og programmering i matematikk i den vidaregåande skulen med fokus på realfagsmatematikkfaga». Det er fleire forskingsdesign som kan vere til hjelp her, men i utgangspunktet var planen min å først gjennomføre ei større undersøking på landsbasis og deretter gjennomføre intervju med bakgrunn i kva som kom fram i spørjeundersøkinga.

Grunnen til at eg ynskja å bruke ein blanda metode (eng: «mixed-methods») var at ved å bygge på den kvantitative delen, vil ein kunne finne meir ut av kva ein vil gå i djupna på i den kvalitative delen (Pring, 2015, s.73). Thorleif Lund (2012) trekkjer fram fire fordelar ved å velje blanda metode. To av argumenta er at ein kan få eit meir komplett bilete av problemstillinga og at ein ved like funn i dei to metodane, vil auke validiteten til resultata. (Lund, 2012, s.157)

På grunn av avgrensa tid og at eg allereie ved innsamling av data frå spørjeskjema og eksamensdata hadde veldig rikt datamateriale, så gjekk eg vekk frå blanda metode og intervjurundane. Dersom eg skal forske vidare på problemstillinga ønsker eg å gjennomføre ein intervjurunde for å få eit meir komplett bilete av problemstillinga, og kanskje får eg anledning til det i ein annan samanheng. Dette vil også bli foreslått som vidare forskning i kapittel 6.6 «Vegen vidare». Eg enda altså opp med ein kvantitativ metode, som har hovudgrunnlaget sitt frå ein kvantitativ spørjeundersøking i tillegg til resultat frå eksamen våren 2022.

3.2 Kvantitativ spørjeundersøking

«Survey research designs are procedures in quantitative research in which investigators administer a survey to a sample or to the entire population of people to describe the attitudes, opinions, behaviours, or characteristics of the population.»

(Cresswell, 2012, s.376)

Survey-design refererer til planlegginga og gjennomføringa av ei spørjeundersøking, som blir brukt til å skildre haldningar, meiningar og erfaringar i eit utval. Det er mange forhold ein må ta omsyn til når ein planlegg ei slik undersøking. Det er viktig å tenkje gjennom kva ein ynskjer å måle og korleis ein kan sikre at undersøkinga blir så nøyaktig og truverdig som mogleg. Cresswell forklarar at ein survey-studie kan skildre tendensar frå data, men ingen strenge forklaringar (eng: «rigorous explanations») (Cresswell, 2012, s.376).

Til å svare på mine forskingspørsmål, som undersøker meiningar (F1), haldningar (F1), erfaringar (F2) og kompetanse (F3), vil ei kvantitativ spørjeundersøking vere ein god metode til nettopp dette i følge Cresswell (2012).

- (F1) **Kva meiningar og haldningar har elevane knytt til algoritmisk tenking og programmering i matematikkfaga?**
- (F2) **Kva erfaringar har elevane gjort seg med algoritmisk tenking og programmering i matematikkfaga?**
- (F3) **Kva kompetanse har elevane tileigna seg i forhold til algoritmisk tenking og programmering?**

Det er to hovudtypar av survey-design, der den eine ser på status i eitt tidsrom og den andre har som mål å sjå på endringar over tid ved å gjennomføre same undersøking for eksempel kvart år med same gruppe (Cresswell, 2012, s.377). I denne masteroppgåva er målet å finne ein status på eit tidspunkt, som er satt til juni 2022 (meir om dette i kapittel 4.1). Når det er sagt, så hadde det vore veldig spennande å følgd opp med same undersøking til same gruppe (matematikk R1) i eit intervall på 2-3 år, sidan det potensielt kan vere stor utvikling etter kvart som elevane har meir erfaring frå algoritmisk tenking og programmering frå grunnskulen, men det fell utanfor rammene til denne masteroppgåva.

3.2.1 Målenivå

Det er viktig å tenkje gjennom korleis ein stiller spørsmåla og kva type svaralternativ ein har i ei kvantitativ undersøking, sidan dette vil påverke analysen. Det er to hovudkategoriar av svaralternativ der den eine er kategoriske svar og den andre som ein glidane (eng: continuous) skala (Cresswell, 2012).

«Categorical scales have two types: nominal and ordinal scales. Continuous scales (often called scale scores in computer data analysis programs) also have two types: interval/quasi-interval and ratio scales.»

(Cresswell, 2012, s. 165)

Den nominale skalaen har den enklaste forma for alternativ, og alternativa har ingen fast rekkefølge. Eksempel på spørsmål på nominalt nivå er for eksempel spørsmål om kjønn, fylke og om ein har programmeringserfaring. Det som kjenneteiknar svaralternativa er at dei er heilt uavhengige, og kan ikkje målast opp i mot kvarandre. Desse alternativa blir sett på som det lågaste analysenivået, og det er avgrensa til samanlikning av talet på svar i kategoriane og til å trekke slutningar, som at det er flest svar frå Oslo fylke (Healey, 2015). Når kategoriane blir laga er det tre kriterium som skal vere oppfylt i følgje Healey (2015). Det første er at det ikkje skal vere overlapp mellom kategoriane, det andre er at kategoriane skal dekke alle val og i tillegg til at vala må vere av lik type. Det siste kriterium skildrar Healey (2015) med at valalternativa må vere «truly comparable» (Healey, 2015, s. 12).

Ordinalnivå er på mange måtar lik nominalnivå, men desse svaralternativa kan bli rangert i ein rekkefølge, som for eksempel frå best til verst eller mest til minst (Cresswell, 2012; Healey, 2015). I motsetning til nominalnivå, så kan me i dette tilfelle samanlikna svaralternativa opp mot kvarandre, sidan me kan rangere dei. Ordinalnivå er likevel avgrensa med tanke på statistisk analyse, sidan det ikkje er ein eksakt talfesta forskjell mellom svaralternativa. «Strictly speaking, statistics such as the average, or mean ..., are not permitted with ordinal-level variables.» (Healey, 2015, s. 13).

I tredje (og fjerde) nivå er Cresswell (2012) og Healey (2015) litt forskjellige i forklaringa. Healey (2015) kallar det siste nivået for «interval-ratio», medan Cresswell (2015) deler det opp i to nivå med «interval and ratio scales». I dette nivået («interval scales») er det antatt like stor avstand mellom kvart svaralternativ, men det er godtatt praksis at ein skala som går frå heilt einig til heilt ueinig går under «interval-ratio» (Cresswell, 2012). «Ratio scales is a response

scale in which participants check a response option with a true zero and equal distances between units. » (Cresswell, 2012, s. 167). Healey (2015) forklarar at alle data målt på «interval-ratio»-nivået kan analyserast med alle statistiske teknikkar, og at ein kan bruke addisjon, multiplikasjon, gjennomsnitt, eller andre matematiske operasjonar (Healey, 2015, s. 13). I utarbeidinga av mine spørsmål og svaralternativ er det vurdert fortløpande i forhold til kva målenivå kvart spørsmål vil gje, meir om dette i neste delkapittel.

3.2.2 Førsteutkast til spørsmåla

Etter å ha bestemt meg for kven undersøkinga skulle gå til, starta arbeidet med å utvikle spørjeskjemaet. Cresswell (2012) skriv at for å utvikle eit spørjeskjema er det tre steg som er vanlege å utføre:

- 1) *Lag forskjellige typar spørsmål.* Blant anna spørsmål som er personlege, om haldningar, åtferdsspørsmål, sensitive spørsmål, lukka og opne spørsmål.
- 2) *Bruke ein strategi for å lage gode spørsmål.* Tydeleg språk, ingen overlapping av alternativ, og spørsmål som alle kan svare på.
- 3) *Utføre ein pilottest for undersøkinga.* Distribuere undersøkinga til ei lita gruppa og endre undersøkinga basert på tilbakemeldingar.

(Cresswell, 2012, s.385)

Dei personlege spørsmåla blir brukt til å hente inn bakgrunnsinformasjon om kven som har svart på undersøkinga. Av personlege spørsmål vart det satt opp spørsmål om kjønn, fylke, standpunktkarakter frå matematikk 1T, terminkarakter i matematikk R1 og tidlegare erfaringar med programmering. Blant desse spørsmåla er spørsmål om kjønn eit typisk spørsmål på nominalnivå, sidan alternativa «Mann», «Kvinne» og «Annet» dekkjer alle alternativ og ikkje er overlappande.

Grunnen til spørsmål om kjønn er for å kunne eventuelt finne forskjellar mellom kjønn i forhold til meiningar, haldningar og erfaringar. Demografiske forskjellar knytt til problemstillinga er også noko eg vurderte å sjå på, i tillegg til å få fram demografien til dei som har svart på undersøkinga. På bakgrunn av dette er spørsmål om fylke med i undersøkinga. Karaktergrunnlag frå 1T og terminkarakter i R1 er også med for å eventuelt finne forskjellar på meiningar, haldningar og erfaringar frå dei forskjellige karaktergrupperingane.

Alle spørsmål i undersøkinga er av typen lukka spørsmål, då desse blant anna gjev meir truverdige (eng: «reliable») svar enn opne spørsmål (Iarossi, 2006, s. 74). Med dette er det

meint at opne spørsmål må ofte tolkast i ei retning, medan lukka spørsmål ikkje opnar for tolking. I tillegg er det ofte lettare å svare på lukka spørsmål, då ein har eit avgrensa tal på alternativ. På den andre sida kan opne spørsmål gje meir «kjøtt på beinet», og eg tenkjer at det kan vere lurt å følgje opp med dette i vidare studie av problemstillinga.

Måling er «ofte definert som det å tillegg observasjonseiningar tal i samsvar med bestemte logiske reglar» (Lund og Christopher, 1999, s.16). Det er fleire variablar som er interessante å sjå på i forhold til meningar, erfaringar og haldningar, og nokre av desse vart formulert som «kva erfaring har du med ...» eller «kor motivert er du til å ...». For å måle dette vart det utarbeida ein skala, som er gjennomgåande i mange av spørsmåla. Skalaen går frå 1 til 7, og har med ein passende visualisering av kvart av alternativa gitt ved eit fjes. Fjeset er veldig positivt ved 7 og negativt ved 1, dette er lagt til for å hjelpe respondentane med å forstå kva verdiane 1 til 7 betyr som svar.



Figur 1: Eksempel på standard svaralternativ brukt i undersøkinga

Eg vurderte om skalaen skulle «tvinge» elevane til å måtte velje positivt eller negativt svar, men landa på at eg ynskja å gje elevane alternativet på midten, som er nøytralt. Alle spørsmåla med svaralternativ illustrert ved hjelp av tal og tilhøyrande smilefjes, er vurdert til høgste målenivå, sidan dette er godtatt praksis (Cresswell, 2012), og dermed mogleg å behandle og analysere ved hjelp av fleire matematiske operasjonar. Spørsmål om karakter er også vurdert som høgste nivå, men her kan ein diskutere om IV burde telle som 0 eller om dei burde ignorerast i analysen når ein ser på for eksempel gjennomsnitt. Det er, som nemnt tidlegare, fleire spørsmål som er på nominalnivå, som for eksempel spørsmål om kjønn, fylke, programmeringserfaring, undervisningsmetodar, programmeringsspråk og liknande. I tillegg er kompetansespørsmåla (multiple-choice) på slutten på nominalnivå.

Spørjeundersøkinga vart utarbeida i verktøyet SurveyXact, som er eit lisensiert program for å lage og distribuere spørjeundersøkingar og samle inn og analysere data. Lisens til programmet er gitt til meg via Universitet i Bergen. I SurveyXact vart det laga eit spørjeskjema med alle relevante spørsmål, som eg sjølv hadde tenkt ut, og skala som vist over på dei spørsmåla der det var naturleg. I tillegg fekk nokre spørsmål fleirvalssvar, som gjer at elevane kan svare meir

enn eit alternativ. For eksempel på spørsmål om kva metodar som har blitt nytta i undervisning av algoritmisk tenking og programmering i matematikk, då det naturleg er blitt brukt fleire metodar, og derfor naudsynt å tillate fleire svar på same spørsmål.

Den siste delen av spørjeundersøkinga bestod av seks forskjellige oppgåver der elevane får presentert eit Python-program, og skal velje korrekt alternativ som skildrar kva som skjer i programsnittane. Her vart alternativa laga, så godt som råd, slik at det var alternativ som er riktige dersom eleven har ei eller fleire misoppfatningar. Denne delen er for å få ein indikasjon på kva elevane har tileigna seg av kompetanse innan algoritmisk tenking og programmering i matematikk.

3.2.3 Fokusgruppe-intervju

Etter å ha fått på plass fyrsteutkastet til ei spørjeundersøking, ynskja eg å få innspel frå dei som faktisk skulle svare på spørjeskjemaet til slutt, nemleg elevar frå faget matematikk R1. Eg fekk samla ei gruppe på fem elevar, som sjølve ynskja å stille, frå ein matematikk R1-klasse. Eg forklarte kva målet med fokusgruppeintervjuet var, og snakka generelt til heile klassen om kva som var gjort så langt i prosjektet. Fokusgruppeintervju kan bli brukt til å avklare ei felles forståing av spørsmål og omgrep i tillegg til å få nye innspel (Cresswell , 2012, s.218; Iarossi, 2006, s.11), noko som var intensjonen bak dette.

Under gruppeintervjuet vart det brukt lydopptak, som i etterkant vart transkribert og analysert. Lydopptaket vart sletta etter transkriberinga, og alle deltakarane vart anonymiserte i transkriberinga. Elevane har i transkriberinga fått alias Elev1, Elev2, og så vidare. Elevane kjende ikkje til spesifikt innhald i spørjeskjemaet, men var forberedt på å kome med innspel i forhold til problemstillinga mi.

Før eg rakk å stille første spørsmål, vart eg avbroten av ein ivrig elev som meina det kunne vere interessant å finne ut om elevane oppfatta algoritmisk tenking og programmering som vanskeleg. Eg bekrefte nikkande, og repeterte det eleven sa for å bekrefte at eg forstod eleven. Fokusgruppeintervjuet fortsette i same stil, og elevane var ivrige til å diskutere kva som var viktig å få med av spørsmål. I transkriberinga er det også brukt fargekoding for å fange opp/markere spørsmål etter kvart som dei kom fram. Lik farge i påfølgande linjer viser til at det er same spørsmål. Transkriberinga ligg som vedlegg.

Etter fargekoding, satt eg opp ei oppsummering av heile fokusgruppeintervjuet med fokus på spørsmåla som kom fram, og kategorisering av desse. Spørsmåla vart delt inn i fire kategoriar,

og spørsmåla vart plassert i desse. Her har eg tatt med den endelege oversikten etter fokusgruppeintervjuet:

Haldningar

1. Er du interessert i å lære om algoritmisk tenking og programmering?
2. Korleis er motivasjonen for å lære om algoritmisk tenking og programmering i matematikk?
3. Er det nyttig å lære om algoritmisk tenking og programmering?
4. Brukar du meir eller mindre tid på algoritmisk tenking og programmering enn andre kompetanseområde utanom skuletid?
5. Har du utforska programmering på eigen oppfordring i tillegg til det som blir brukt fagleg?
6. Trur du at du vil få bruk for kompetanse innan algoritmisk tenking og programmering etter vidaregåande?

Meningar

1. Synes du at algoritmisk tenking og programmering er vanskeleg?
2. Oppfattar du at læraren din har god nok kompetanse til å lære vekk algoritmisk tenking og programmering?
3. Kva kompetansar meina du er ein del av algoritmisk tenking og programmering?
 - a. Forslag frå elev: «Løyse problemstillingar med algoritme»
4. Meina du at algoritmisk tenking og programmering har blitt prioritert i undervisninga?
5. Korleis passer algoritmisk tenking og programmering inn med tema i matematikken?
6. Burde algoritmisk tenking og programmering vore eit eige obligatorisk fag?

Erfaringar

1. Visste du at du skulle lære algoritmisk tenking og programmering i matematikken?
2. Tidlegare erfaringar med algoritmisk tenking og programmering?
 - a. Frå barneskolen?
 - b. Programfag med programmering?
3. Har du erfart nokon bruksområde for algoritmisk tenking og programmering?
4. Korleis jobba de med algoritmisk tenking og programmering?
 - a. Les, kopier, tilpass
 - b. Video
 - c. Følg oppskrift
 - d. Utvid
5. Har du fått nok opplæring i algoritmisk tenking og programmering?
6. Har du fått nok tid til å jobbe med algoritmisk tenking og programmering?
7. Har du tileigna deg forståing av algoritmisk tenking og programmering?
 - a. Skala 1-7 (Ingen – Veldig mye)
8. Blir det brukt nok tid på programmeringseksempla i boka?

Kompetanse

1. Kan du lage program på eigenhand, eller er du avhengig av å kopiere koder frå læraren og tilpasse?
2. Til lærar: Formell kompetanse og erfaring i algoritmisk tenking og programmering.
3. TEST: print
4. TEST: løkke
5. TEST: if
6. TEST: Variablar

Med bakgrunn i oppsummeringa av fokusgruppeintervjuet, vidareutvikla eg spørjeskjemaet mitt, ved å legge til og tilpasse spørsmål som vart nemnt av elevane.

3.2.4 Pilot

Ein kritisk del av å gjennomføre ei kvantitativ spørjeundersøking er å gjennomføre ein pilot (Iarossi, 2006, s.86-87).

«The pilot test represents the only opportunity to verify this and the data collected will ultimately reflect any poorly defined question or concept.»

(Iarossi 2006, s. 87)

Pilottest er ein god måte å få testa om eit spørjeskjema har tydelege spørsmål og eventuelt avsløre dårleg formulerte eller uklare spørsmål. Dette vil bli fanga opp ved å sjå på data som er samla inn, og i samtale med deltakarane i etterkant av gjennomføringa. Pilottest er «the first live test of the instrument, as well as the last step in the finalization of the questions» (Iarossi, 2006, s. 87).

Piloten vart gjennomført i ein matematikk R1-klasse med 23 elevar der alle svara på spørjeskjemaet. Etter gjennomføringa vart resultata analysert, og deler av resultata vart presentert til klassen. Det vart satt av tid til å gå gjennom alle spørsmåla i etterkant for å avklare om det var nokon spørsmål som kunne misoppfattast eller som burde ha tydlegare språk. I denne dialogen kom det fram fleire forslag til endringar, det vart lagt til fleire alternativ og ein feil i spørjeskjemaet vart oppdaga. Tidsbruken varierte frå 5 minutt opptil like under 20 minutt, og elevane brukte i gjennomsnitt 12 minutt på å svare. Denne informasjonen vart brukt som ein del i distribueringa, ved at lærarane som skulle gjennomføre undersøkinga i sin klasse fekk vite at det ville ta omtrent 10 minutt å gjennomføre. Grunnen til at eg skreiv 10 minutt, var fordi

spørjeundersøkinga hadde forhåpentleg blitt enda tydelegare i spørsmålsformulering og alternativ, sånn at elevane ville bruke mindre tid enn i piloten.

Etter gjennomføring og analysing av piloten, vart spørjeskjema igjen tilpassa, og det nærma seg ferdig og klar for distribuering. Som ein siste kontroll, fekk eg alle mine realfaglege kollegaer til å gå gjennom spørsmålsskjemaet saman med meg. Her vart det berre små justeringar, som for eksempel at rekkefølge på spørsmåla vart endra og nokon få skrivefeil vart retta opp.

3.3 Utval

Cresswell (2012) forklarar at det er viktig å tenkje gjennom kven ein skal spørje og korleis ein kan sikre at undersøkinga blir representativ for målgruppa. I dette masterprosjektet har eg valt å spørje eit representativt utval for den vidaregåande skulen, og meir spesifikt matematikk R1-klassar. Målet med undersøkinga var å få svar frå minst tre klassar frå kvart fylke, som då ville vere eit representativt utval for R1-klassar på landsbasis. Grunnen til at eg har valt matematikk R1-klassar, er fordi det på tidspunktet for innsamling av data var dei som hadde lengst erfaring med algoritmisk tenking og programmering i den vidaregåande skulen.

*«The most popular and rigorous form of probability sampling from a population is simple random sampling. In **simple random sampling**, the researcher selects participants (or units, such as schools) for the sample so that any individual has an equal probability of being selected from the population. »*

(Cresswell, 2012, s.143)

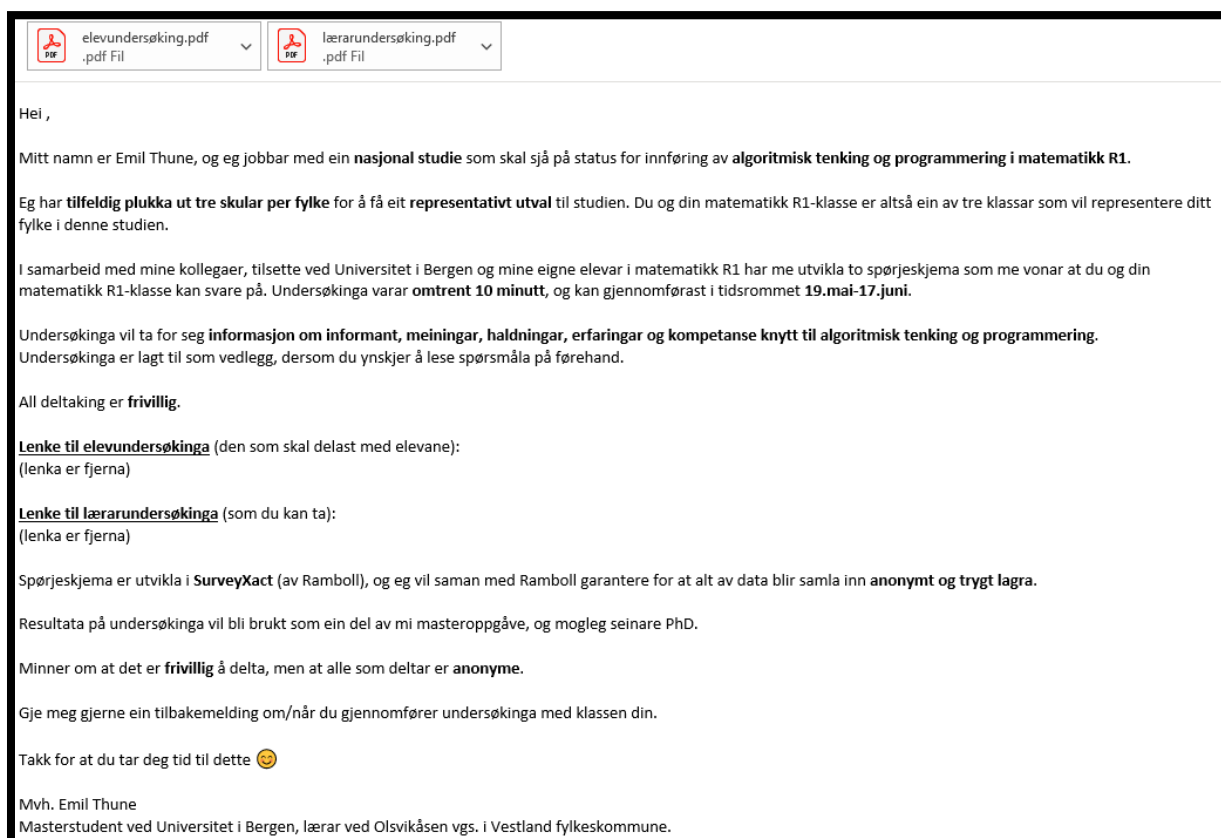
For å få eit mest mogleg tilfeldig utval, har eg tilfeldig plukka vidaregåande skular frå kvart fylke. Dersom det viste seg at skulen ikkje har matematikk R1, har eg trekt ut ein ny tilfeldig skule i same fylke. Utvalet mitt er forsøkt å vere ei «*simple random sampling*», som Cresswell skildrar i sitatet ovanfor. Kritikk til utvalet er diskutert i kapittel 3.9.

Cresswell (2012) kjem også med eit anslag på kor mange svar ein bør ha i ei spørjeundersøking og forklarar at «Approximately 350 individuals for a survey study, but this size will vary depending on several factors» (Cresswell, 2012, s.146).

I utsending av spørjeskjemaet tok eg eit val på berre å sende ut til offentlege skular. Som ein del av datainnsamlinga mi, hadde eg satt meg eit mål på å prøve å få svar frå minst tre vidaregåande skular i kvart fylke. Eg starta med å samle informasjon om alle vidaregåande skulane kategorisert etter fylke. Denne informasjonen vart henta direkte frå fylka sine egne

nettsider. Deretter trakk eg tilfeldig ut fire av skulane frå kvart fylke, som då skulle representere fylket sitt i denne studien. Grunnen til at eg valte å trekke ut fire skular, ikkje tre, var på bakgrunn av den erfaringa eg høyrde andre hadde gjort seg ved utsending av spørjeskjema på denne måten, som var at mindre enn halvparten svara på denne typen e-post.

Etter å ha trekt ut skulane, starte eg arbeidet med å lage to malar på e-post, som kunne sendast ut til skulane og faglærarane. I e-posten til skulane spurte eg om skulen tilbydde matematikk R1 i inneverande år, og at eg eventuelt ynskja kontaktinformasjon til faglærer i faget. E-posten som skulle gå til faglærer hadde som intensjon å innehalde alt av informasjon om studien min, og få fram tydeleg at det var frivillig og anonymt å delta. Lenkene til både lærarundersøkinga og elevundersøkinga var også med i denne e-posten i tillegg til pdf-filer, som innehald begge undersøkingane i sin heilheit. Utarbeiding av desse malane vart også kvalitetssikra ved at eg diskuterte dei med mine realist-kollegaer på arbeidskontoret.



Figur 2: Mal på e-post som vart sendt ut til faglærarane

Utsendinga av e-postar til skulane starta i midten av mai 2022, og siste innsamling vart gjort i midten av juni 2022. Grunnen til at dette tidspunktet for utsending vart valt, er fordi det ikkje var eksamen dette skuleåret og at eg då håpa på at faglærer og elevane hadde litt tid til overs i dette tidsrommet til å svare på undersøkinga.

Etter kvart som nokre skular meldte tilbake om at dei ikkje hadde matematikk R1, vart det fortløpande trekt ut ein ny tilfeldig skule frå same fylke. Dersom ein skule ikkje svara innan 7 dagar, vart det sendt ut ein ny e-post. I nokre få tilfelle vart også faglærer spora opp på andre måtar, som for eksempel via skulen si eiga nettside. Oversikten over skular med kontaktinformasjon og oversikt over faglærarar og kven som hadde sagt ja/nei vart lagra i eit dokument som var passordbeskytta. Figuren under viser ein liten del av oversikten, som vart brukt for å ha kontroll på alle skulane og status på svar.

Fylke	Skule1	Epost	Svar
Agder			nei
Innlandet			Ja
Møre og Romsdal			Ja
Nordland			Ja
Oslo			purra
Rogaland			ja
Troms og Finnmark			ja
Trøndelag			nei
Vestfold og Telemark			Ja
Vestland			ja
Viken			ja

Figur 3: Oversikt over svarstatus frå skulane

I tillegg til oversikten over skulane, vart det også laga ein tilsvarande oversikt over alle faglærarar etter kvart som eg fekk informasjon frå skulane.

3.4 Eksamensdata

I tillegg til mi eiga undersøking har eg undersøkt korleis det gjekk med eksamensresultata våren 2022, på dei oppgåvene som er relevante til problemstillinga. Eg sendte, like etter eksamen våren 2022 vart gjennomført, ei førespurnad til Utdanningsdirektoratet der eg spurte etter data knytt til sju oppgåver som omhandla algoritmisk tenking og programmering frå eksamen i matematikk 1T, R1, S1 og 2P.

Data vart gjort tilgjengeleg via ein sikker kanal, og derifrå lasta eg ned data og analyserte desse. Det er viktig å vere merksemd på at det berre er svar frå privatistar, sidan ordinær eksamen vart avlyst denne våren. Dette kan sjølvsagt påverke resultatet, og det kan argumenterast at det ikkje er eit representativt utval for matematikk-elevane i den vidaregåande skulen. Det er også stilt spørsmål ved om nokre av elevane melde seg opp i feil fagkode denne våren, då det både var mogleg å melde seg opp i læreplanen LK06 i tillegg til den nye LK20. Resultatet som vart gjort

tilgjengeleg for meg, var gitt som poeng per oppgåve per kandidat, og er analysert og summert i kapittel 4.7, og tatt med til vidare diskusjon.

3.5 Analyse av kvantitative data

Som nemnd, så er det SurveyXact som er brukt til innsamling av data. Denne programvara er også nytta i presentasjon av resultata og analysen. I tillegg er det brukt forskjellige metodar for analyse ved hjelp av Python-skript og rekneark (Excel). I denne delen ser eg på kva moglege analyser ein kan utføre utifrå kva data ein har samla inn. Det vil bli gitt ei forklaring til deskriptiv statistikk med sentralmål og spreingsmål, og til slutt kjem eg inn på slutningsstatistikk.

3.5.1 Deskriptiv statistikk

«How do we analyze data to describe trends?» (Cresswell, 2012, s. 183). Deskriptiv statistikk er forklart som analyse som kan skildra tendensar i gruppa som blir målt. Statistikken som blir nytta kan delast inn i tre kategoriar, der me har sentralmål, spreing og relativ plassering (eng: relative standing) (Cresswell, 2012).

Av sentralmål er modus, median og gjennomsnitt mykje brukt. Alle dei tre sentralmåla seier noko om kvar gruppa sine svar er sentrert rundt. Medan modus er det alternativet som har høgast frekvens, så er medianen verdien som er på midten etter at ein har sortert alle svar i stigande rekkefølge og gjennomsnittet seier noko om kva gruppa svara i gjennomsnitt. Alle desse sentralmåla er nytta i kapittel 4 og 5 for å kunne seie noko om gjennomsnittseleven, men også for å samanlikne grupperingar av elevane.

Av spreingsmål er variasjonsbredde og standardavvik vanlege statistiske mål. I kapittel 4 og 5 er desse spreingsmåla nytta til å seie noko om forskjellar blant alle kandidatane, men også for å samanlikne grupperingar av kandidatane. Variasjonsbredde gjev oss total spreing ved hjelp av differanse mellom minste og største registrerte verdi, medan standardavvik kan nyttast til å seie noko om korleis elevane er fordelt innanfor dette intervallet. Standardavvik kan brukast til å samanlikna to spørsmål med same svaralternativ eller to grupper som har svara på same spørsmål. Då vil lågare standardavvik bety at gruppa sine svar er meir samla enn ved høgare standardavvik.

For å seie noko om relativ plassering, nyttar ein seg av ein z-skår eller prosentilrangering. Her vil z-skåren «enables a researcher to compare scores from different scales.» (Cresswell, 2012, s. 187). Prosedyren er å ta alle målingar og trekke frå gjennomsnittet, og deretter dividere alle

med standardavviket. Z-skår er ikkje nytta i denne masteroppgåva, men kan nyttast ved samanlikning av same spørsmål frå undersøkinga mi og ei anna med forskjellige svaralternativ. Prosentilrangering er eit mål på kor mange prosent som er under ein gitt verdi. For eksempel kan dette nyttast til å finne ut kor mange som fekk karakteren 4 eller lågare i 1T, som er nytta i kapittel 4 og 5.

3.5.2 Slutningsstatistikk

Kva ein kan seie og generalisere utifrå analysen av data, blir kalla slutningsstatistikk og består av «prinsipper/begreper/metoder for å trekke optimale slutningar om parametere i populasjoner fra observatorer i utvalg, samt spesifisering av usikkerhet ved slike slutningar» (Lund & Christopher, 1999, s.66). Ein kan med andre ord nytte seg av matematiske metodar for å trekke ei slutning om ein observasjon av ei gruppe i utvalet, og støtte opp om slutninga ved å vise til matematisk berekna usikkerheit.

Cresswell (2012) nemner fleire matematiske berekningar som kan støtte opp om slutningar mellom to eller fleire variablar. Pearson sin korrelasjonstest er foreslått til samanlikning av to variablar av glidande svaralternativ, og for å samanlikne ein kategorisk variabel med ein variabel som har glidande svaralternativ er det nemnt t-test, variansanalyse og samvariasjon (Cresswell, 2012, s. 183).

Korrelasjonsanalyse voks fram siste halvdel av 1800-tallet, og Pearson sin formel for korrelasjon vart lagt fram allereie i 1895 (Cresswell, 2012).

«In presenting ideas about correlations, Pearson not only articulated the formula for a correlation, but he also presented concepts familiar to quantitative researchers today, such as the importance of sample size, the value of precise measurement, and the use of unbiased samples. »

(Cresswell, 2012, s. 339)

I dag er Pearson sin korrelasjonstest velkjend og mykje nytta i korrelasjonsforsking. Formelen er gitt som

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

der r er korrelasjonskoeffisienten, x_i er verdiane frå første gruppa, y_i er verdiane frå andre gruppa, \bar{x} er gjennomsnittet i første gruppe og \bar{y} er gjennomsnittet i den andre gruppa.

Korrelasjonskoeffisienten r vil få ein verdi mellom -1 og 1, og om ein ser på absoluttverdien så vil høgare verdi bety høgare sannsyn for at det er ein samanheng mellom variablane. Ved å kvadrere Pearson sin korrelasjonskoeffisient, får ein det som blir kalla for effektstorleik (Cresswell, 2012, s. 357).

Når ein har kalkulert korrelasjonskoeffisienten kan ei tolking av denne delast inn i fem grupper (Cresswell, 2012). Frå 0 til .20 er det ingen korrelasjon, og i intervallet frå .20 til .35 er det lite samanheng mellom variablane. I neste intervall frå .35 til .65 kan ein nytte seg av korrelasjonen i avgrensa slutningar, medan i .66 til .85 er det gode sjansar for at ein kan gjere statistisk gode slutningar mellom variablane. I det siste intervallet frå .85 til 1 er det klar korrelasjon, og Cresswell (2012) diskuterer om då variablane eigentleg sjekkar det same (Cresswell, 2012, s. 347).

T-test er ein statistisk test for å samanlikne to grupper sine gjennomsnitt, og dermed kunne seie noko om forskjellar mellom gruppene. T-test blir brukt som ein del av hypotesetesting for å avgjere om eit utsakn kan avvisast. Hypotesetesting tar utgangspunkt i den motsette påstanden av det me faktisk lurar på, og ender med anten å avvise eller godta denne påstanden. Det er totalt fem steg i hypotesetesting i følgje Cresswell (2012), og dei er «(a) identify a null and alternative hypothesis; (b) set the level of significance, or alpha level; (c) collect data; (d) compute the sample statistic; and (e) make a decision about rejecting or failing to reject the null hypothesis. » (Cresswell, 2012, s. 188)

T-testen resultere i to verdier, der den eine er t-verdien og den andre er «degrees of freedom», som er gitt ved $df = n - 1$ der n er talet på verdier til variabelen (Cresswell, 2012, s.190). T-testen gjev oss vidare ein p-verdi som skal testast opp mot ein førehandsdefinert alfa-verdi, og er alfa-verdien høgst så kan ein avvise null-hypotesen. I denne masteroppgåva blir p-verdien rekna ut og tolka direkte for å finne ut om det er ein samanheng mellom to grupper. Me antar at det er ingen forskjell mellom gruppene, og set signifikansnivået til å vere 0.05. Meir om bruken av t-test og p-verdien i kapittel 5.

3.6 Validitet

Kleven (2008) oppsummera og diskutera Shadish, Cook og Cambell (2002) sin inndeling av validitet i fire kategoriar. Dei fire kategoriane er *Construct validity*, *Statistical validity*, *Internal validity* og *External validity*.

«Construct validity» handlar om gyldigheita i å gå frå «what we have seen to what we call what we have seen» (Kleven, 2008, s.223), som eg tolkar som validering av slutningane me tar frå

indikatorane ein får ut frå data. I mine slutningar prøver eg å seie noko om korleis innføringa av algoritmisk tenking og programmering har gått. For å kunne ta gyldige slutningar har eg blant anna prøvd å få eit representativt utval for alle elevar i Noreg, som tok matematikk R1 våren 2022. Ved å samle inn data frå dei fleste fylka, er det tatt høgde for at det kan vere geografiske forskjellar. Skulane som har svara er også tilfeldig valt ut, sånn at det er eit mest mogleg eit representativt utval.

Om «Statistical validity» skriv Kleven (2008) at handlar om korrelasjon mellom variablar, og det å avgjere om korrelasjonen er triviell eller om den er av interesse for vidare tolking. For å kunne vurdere denne gyldigheita nemner Kleven (2008) bruk av statistiske metodar som signifikanstest og estimat av effekt for å avgjere om samanhengen er av betydning eller ikkje. I analysen av mine data er det brukt både signifikanstest og Pearson sin korrelasjonstest. Dette er med på å auke gyldigheita av samanhengar som er av betydning, og som dermed også gjev grunnlag for å eventuelt studere vidare.

Kleven (2008) definerer «Internal validity» som «validity of inferences from an observed covariation to a causal interpretation» (Kleven, 2008, s.227). Dette kan tolkast som at samvariasjonen ikkje treng å vere direkte, men at to variablar korrelerer fordi ein eller fleire andre variablar påverkar dei, noko Kleven (2008) skildrar som eit falsk forhold (eng: spurious relationship).

«However, covariance gives no basis for causal interpretation until alternative causal interpretations are eliminated or at least shown to be unlikely.»

(Kleven, 2008, s.227)

Samvariasjon i seg sjølv er ikkje nok for å seie om to variablar har ein årsaksamanheng, men ein må utelukka andre kausale tolkingar eller vise dei som usannsynlege.

Den siste kategorien er «External validity», som Kleven (2008) definerer som «validity of inferences from the context of the study to a wider context or to other contexts» (Kleven, 2008, s.229). Dette handlar om gyldigheita til slutningar frå studien sin kontekst og over i ein breiare kontekst eller ein ny kontekst.

Cresswell (2012) forklarar kva som kan vere lurt å undersøke når ein skal sjå på validitet: Val av riktig metode og instrument, sjå på tidlegare studie som har fått liknande resultat, sjå på kva intensjonen med instrumentet var i desse studia og korleis dei er tolka, og til slutt vurdere om det er gode bevis som knytte tolking til bruk (Cresswell, 2012, s.162).

3.7 Reliabilitet

Reliabilitet og validitet er «bound together in complex ways» (Cresswell, 2012, s. 159). Med dette meina Cresswell (2012) at validitet og reliabilitet kan vere mykje det same i ein samanheng, medan i ein annan kan dei vere heilt uavhengige. Vidare blir validitet forklara som noko større (eng: greater), medan reliabilitet er enklare å forstå sidan det er eit mål på konsistens.

«If scores are not reliable, they are not valid; scores need to be stable and consistent first before they can be meaningful. Additionally, the more reliable the scores from an instrument, the more valid the scores may be.»

(Cresswell, 2012, s.159)

For at me i det heile tatt skal kunne snakke om validitet, må me ha data som kan stolast på. Det betyr at data me samla inn, må komme frå kontrollerte forhold, der me sikrar konsistens. Meir kontrollerte forhold under innsamlinga vil gje betre reliabilitet, og som kan føre til betre validitet. Det er viktig å tenkje gjennom korleis innsamlinga av data blir gjort med tanke på korleis deltakarane blir presentert undersøkinga, utvikling av tydelege og klare spørsmål og at deltakarane er i same sinnstemning og tolka spørsmåla likt (Cresswell, 2012).

Når det er samla inn data i denne masteroppgåva, har reliabilitet vore viktig for å sikre at data kan brukast til å trekke slutningar om indikasjonar frå datasettet. Som skildra tidlegare, så har eg hatt direkte kontakt med faglærarar i matematikk R1, som sjølve har distribuert spørjeundersøkinga til elevane via ei lenke. Reliabiliteten min ligg i at eg har kvalitetssikra at alle eg har kontakt med av faglærarane faktisk er faglærarar i matematikk R1, alle har fått same instruksjon i forhold til gjennomføring, anonymitet og friviljugskap. I spørjeundersøkinga er reliabiliteten auka ved at spørsmåla er utvikla over lengre tid, med både pilottest etterfølgt av analyse, gjennomgang og diskusjon av kvart spørsmål for å fange opp eventuelle ulike tolkingar og uklare spørsmål, og gjennomgang av alle spørsmål med kompetente kollegaer på fagområdet.

3.8 Etikk

Etikk i forskning er mykje diskutert, og Tangen (2014) har laga eit rammeverk som skal hjelpe forskarar med å finne ein balansegang mellom etikk og kvalitet i forskingsarbeid. På neste side forklarar Tangen (2014) korleis etikk ikkje treng å vere noko negativt for forskingsarbeid, men at det kan forsterka kvaliteten til forskinga.

«... ethical principles do not necessarily constrain or weaken the design and results of research. Sometimes ethics contributes to strengthening the quality of research, and vice versa. The interplay between ethics and quality of research may therefore be both risky and beneficial. »

(Tangen, 2014, s. 678)

Etikk er viktig for å bidra til styrking av kvaliteten til forskinga, og motsett. Kva er så etikk innan forskning knytt til undervisning? Tangen (2014) deler inn etikk i tre kategoriar: Den første er etikk i forskingsmiljøet, den andre handlar om å beskytte forskingskandidatane, og den siste er rolla og verdien til forskning på undervisning i samfunnet.

Innan etikk i forskingsmiljøet er det viktig med «forskarfrileik, individualitet, profesjonell integritet, og god forskingspraksis, inkludert respekt for kollega og studentar» (Tangen, 2014, s. 679, oversett frå engelsk). Tangen (2014) oppsummera dette med å kalle det for «kvaliteten av forskingsprosessen og resultatata» (Tangen, 2014, s.679, oversett frå engelsk).

Å beskytte forskingskandidatane inneber å beskytte dei mot skade og respektere deira integritet, sjølvvørnad og fridom, i tillegg til at dei og deira autonomi blir tatt vare på (Tangen 2014, s. 680). Når det gjeld rolla og verdien til forskning på undervisning i samfunnet forklarar Tangen (2014) at inkluderer relevansen for utdanningspraksis og verdien av forskinga til ulike grupper (Tangen, 2014, s. 680).

Tangen (2014) har laga ein oversikt som viser kva ein må tenkje på, i forhold til etiske spørsmål knytt til undervisningsforskning, og forklarar at den er meint som eit verktøy for forskarar på dette området. Tabellen (Tangen, 2014, s. 682) viser ein oversikt over dei tre hovudkategoriane, som skildra ovanfor, i tillegg til å vise korleis desse påverka kvarandre. Det er også tatt med ei kolonne for å vise til kva ein må tenkje på i dei forskjellige stega i forskinga, med inndeling i tre nivå der første steg er planlegging av forskinga, den andre er innsamling, analyse og rapportering, og til slutt informasjon, formidling og offentleg diskusjon.

For å beskytte elevane, gruppene eller institusjonane må ein under planlegginga tenkje gjennom eventuelle problem, og vurdere desse (Tangen, 2014). Det er i denne masteroppgåva blant anna diskutert om elevane kan seie noko om problemstillinga ved hjelp av ein kvantitativ undersøking, eller om det også burde involvere faglærarane og eventuelt eksamensresultat for å få eit betre grunnlag for å svare på dette. Metoden som er nytta er vurdert, og det er diskutert fordeler og ulemper ved denne. Det er også diskutert og argumentert for kvalitetssikring av

informasjon som er sendt ut, for å unngå misforståingar, og at det er tydeleggjort at det er frivillig og anonymisert med tanke på rekruttering av kandidatar.

Når det gjeld å sikre kvaliteten av forskinga under planlegginga, viser Tangen (2014) til relevans av problemstillinga, og kva som er nytten av den. Her er det viktig å aktualisere problemstillinga, og vise til for eksempel akademisk interesse. Det er også diskutert om nytteverdien direkte til praksis, vidare forskning og eventuelt politiske tiltak. Her vart det også diskutert konsekvensane av å undersøkje ei problemstilling, som ikkje naudsynt faglærer sjølv har kompetanse i og/eller ikkje har prioritert i undervisninga.

I innsamling og analysing av data nemner Tangen (2014) at ein må ta vare på eleven, og i mitt tilfelle også skulane, sin sjølvvørnad, autonomi, anonymitet, omdømme og at alle blir høyrd. Dette er tatt opp ved å tenkje gjennom kva data som blir samla inn, spesielt dei personlege spørsmåla. Av omsyn til skulane, og deira R1-klassar som har svara, er det ikkje sett på geografiske forskjellar, som vart vurdert til risikabelt med tanke på anonymitet som kunne føre til blant anna omdømme-problematikk. At datainnsamlinga er gjort i trygge omgivnader i klasserommet med faglærer, er med på å gje elevane gode og trygge rammer for gjennomføring.

For å sikre kvalitet i forskingsarbeidet nemner Tangen (2014) validitet og reliabilitet, som er diskutert i eigne delkapittel, i tillegg til truverd og kor transparent forskinga er. Gjennom heile masteroppgåva prøver eg å vere så transparent og truverdige som mogleg. Dette får eg til ved å få med viktige detaljar gjennom heile prosessen, frå utvikling av problemstilling og tilhøyrande forskingsspørsmål og heile vegen til avslutninga.

I publiseringa av denne masteroppgåva har eg diskutert og sikra anonymitet, og eg ser ingen negative konsekvensar som kan kome til nokon av klassane, elevane eller lærarane som har delteke. Eg meina sjølv at forskinga som er gjort kan nyttast på fleire nivå, blant anna direkte i praksis, til vidare forskning og som grunnlag for eventuelle politiske diskusjonar.

Tangen (2014) snakkar også om forventningar til forskar frå kandidatane, og stiller spørsmål som «How can their expectations be handled in a way that does not disturb the research process?» (Tangen, 2014, s. 687). Metoden som eg har brukt har gjort at få/ingen har følgd opp med forventningar, og at alle er blitt gjort merksemd på at deira bidrag går inn som ein del av ei masteroppgåve utan noko meir oppfølging.

3.9 Kritikk og feilkjelder

For å samle inn flest moglege svar, utan å tvinge nokon, vart det sendt ut e-post til faglærarar med ein førespurnad om ein kunne tenkje seg å delta i prosjektet ved å sette av tid til klassen for å svare på spørjeundersøkinga. At det i informasjonen i e-posten er skildra at studien skal undersøkje status i innføringa av algoritmisk tenking og programmering i matematikk R1, kan ha ei påverknad på kven som har valt å svare på denne. Det er mogleg at faglærarar rundt om i Noreg kvir seg for å la elevane svare, sidan mange ikkje er trygg i denne kompetansen sjølve (Frantsen, 2019). Dette kan igjen føre til at utvalet er mindre representative for heile populasjonen enn tenkt, og at det derfor ikkje er ei streng «simple random sampling» som Cresswell (2012) skildra.

Sjølv om faglærarane har fått same instruksjon for gjennomføring av spørjeundersøkinga, har eg ingen garanti for at instruksjonane er gitt korrekt. Dersom forskjellige rammer er lagt for gjennomføringa, som for eksempel tidspunkt, tidsfrist og meir, så kan dette få utslag i deltakarane sine svar.

Det er lagt ned mykje arbeid i spørjeskjema med spørsmål, forklaringar og alternativ, men det vil alltid vere ein risiko for at ein eller fleire oppfattar spørsmåla eller alternativa forskjellig. Det viktige er at det er gjort grundig arbeid, for å sikre at flest moglege har same oppfatning av spørsmåla og tilhøyrande svaralternativ.

Med 260 svar vil ein kunne få eit godt inntrykk av elevar i matematikk R1 sine haldningar, erfaringar og kompetanse, men det ville sjølvsagt gjort resultatet enda betre om fleire hadde svara. I tillegg hadde det vore betre om alle fylka svara, for å betre representasjonen til populasjonen.

Etter datainnsamlinga har eg kritisert spørjeskjemaet mitt for å ha lite fokus på faktisk kompetanse i algoritmisk tenking på høgare nivå. Om eg skulle gjort spørjeundersøkinga på nytt, ville eg prøvd å fokusere meir på det som Brennan og Resnick (2012) kallar «computational concepts» og «computational perspectives». Som eg diskuterer seinare har eg med alle dei tre kategoriane, men i mindre grad dei to sistnemnde.

4 Resultat

Datainnsamlinga mi består av 260 elevsvar på ei spørjeundersøking i matematikk R1, og seksten lærarsvar på ei eiga lærarundersøking i matematikk R1. I tillegg viser eg til relevante oppgåver og resultat på desse frå eksamen våren 2022 i matematikkfaga 1T, 1P, 2P, S1 og R1.

4.1 Om informantane

Spørjeundersøkinga vart svara på av 19 klassar/grupper i matematikk R1 i slutten av skuleåret 2021/22, og eit ukjent tal på elevar fekk anledning til å svare (meir om dette i kap.3.2.5). Totalt har 260 elevar svara på undersøkinga, som er litt under anbefalinga på 350 deltakarar til Cresswell (2012). Av dei 260 har 245 svara på alle spørsmåla, medan femten har svara på delar av spørsmåla. For skuleåret 2021/2022 var det 6952 som valde matematikk R1 ved offentlege skular i Noreg (Utdanningsdirektoratet, u.å.). Det vil seie at omtrent 3,5% av alle elevar i denne gruppa har vore med i undersøkinga.

Fordelinga av elevar geografisk kan ein sjå i tabellen under.

	Prosent	Respondentar
Agder	0,0%	0
Innlandet	10,8%	28
Møre og Romsdal	21,5%	56
Nordland	11,5%	30
Oslo	8,5%	22
Rogaland	16,9%	44
Troms og Finnmark	0,0%	0
Trøndelag	2,7%	7
Vestfold og Telemark	8,1%	21
Vestland	8,8%	23
Viken	11,2%	29
Total	100,0%	260

Tabell 5: Oversikt over geografisk fordeling blant deltakarane

Elevane som har svara på undersøkinga er fordelt over heile landet, med unntak av Agder og Troms og Finnmark som ikkje er representerte.

Kjønnsfordelinga er 57% (147) menn, 42% (110) kvinner og 1% (3) annet, noko som tyder på at utvalet er representativt for populasjonen, sidan det stemmer med Utdanningsdirektoratet sine tal som indikerer at 55% er registrert som menn i matematikk R1 (Utdanningsdirektoratet, u.å.).

4.1.1 Tidlegare karakterar

Elevane har gode karakterar frå både 1T og R1, med gjennomsnitt på omtrent 4,5 frå matematikk 1T og omtrent 4 frå matematikk R1. Under er det ein oversikt over korleis elevane har oppgitt sine karakterar frå dei to matematikkfaga.

	Gjennomsnitt	Modus	Median
Karakter 1T	4,54	5 (109)	5
Karakter R1 (1.termin)	3,99	4 (88)	4

Tabell 6: Statistikk frå karakterar i matematikk 1T og R1 (1.termin)

Elevane i utvalet har fått ein gjennomsnittskarakter på 4,54 i matematikk 1T, og omlag 85% (220) av elevane har fått karakterane 4 eller betre. Resultatet viser at karakteren 5 er den mest vanlege karakteren i matematikk 1T, medan det i fyrste termin i matematikk R1 er 4 som er den vanlegaste karakteren. Her kjem ein prosentvis fordeling av karakterane i matematikk 1T.

	Prosent	Respondentar
IV	1,2%	3
1	0,4%	1
2	2,7%	7
3	11,2%	29
4	25,4%	66
5	41,9%	109
6	17,3%	45
Totalt	100,0%	260

Tabell 7: Oversikt over standpunktkarakter får matematikk 1T

I første termin av matematikk R1 har elevane i utvalet fått 3,99 i gjennomsnittskarakter. Dette er omtrent ein halv karakter lågare enn utvalet svara at dei fekk i standpunktkarakter i matematikk 1T. Ein av tre elevar har fått karakteren 4, og omtrent 85,3% (222) har fått karakterane 3, 4 eller 5. Minner om at alt av data er sjølvrapportert, så det er ingen garanti for at dette stemmer med faktiske karakterar.

I tabellen under er tilsvarende tabell for karakterar i matematikk R1 for fyrste termin.

	Prosent	Respondentar
IV	0,8%	2
1	1,5%	4
2	6,2%	16
3	22,3%	58
4	33,8%	88
5	29,2%	76
6	6,2%	16
Totalt	100,0%	260

Tabell 8: Oversikt over terminkarakter i matematikk R1

4.1.2 Tidlegare erfaring med programmering

Av 260 elevar har 18,5% (48) erfaring frå programmering som valfag frå ungdomsskulen, som stemmer bra med det Hirth (2021) fant i ein matematikk 1P klasse (21%). I tillegg er det 11,6% (30) som svart at dei har programfaga informasjonsteknologi 1 eller 2 samstundes som dei har matematikk R1.

I programmering som valfag frå ungdomsskulen skal elevane blant anna "... kunne sette seg inn i og forklare eigen og andre sin kode.", "... kunne bruke minst eit tekstbasert programmeringsspråk.", "...bruke grunnleggjande prinsipp i programmering, som variablar, løkker, vilkår og funksjonar.", "... analysere problem, gjere dei om til delproblem og forklare korleis nokre av delproblema kan løysast ved programmering", og til slutt skal dei "...kunne utvikle og feilsøke dataprogram som løyser definerte problem, inkludert kontrollering eller simulering av fysiske objekt.". (Utdanningsdirektoratet, 2016)

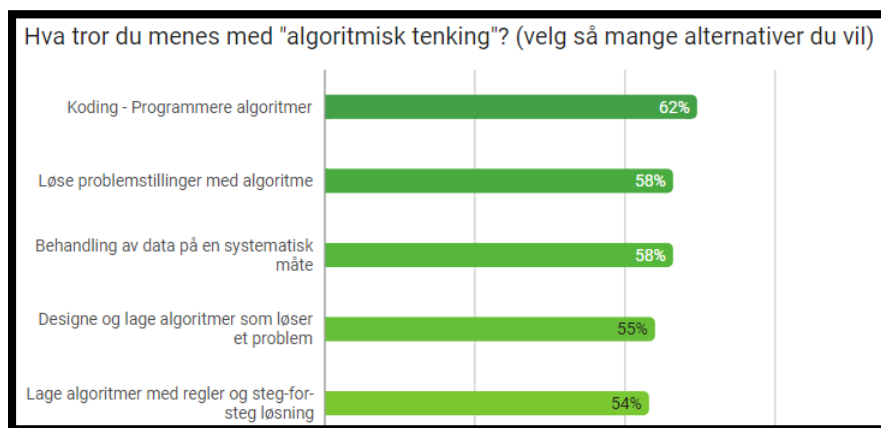
Elevane som har svara på undersøkinga var det siste kullet som hadde forsøkslæreplanen i programmering som valfag. Denne læreplanen er erstatta med ein ny vedtatt læreplan (Utdanningsdirektoratet, 2020b) som i hovudtrekk er lik forsøkslæreplanen frå 2016.

4.2 Meininger

I dette delkapittelet ser eg på resultat som er knytt til elevane sine meininger om kva algoritmisk tenking er, om dei føler at dei har fått god opplæring i kompetansen, kompetansen sin relevans for matematikkfaget og eigenvurdering av tileigna kompetanse.

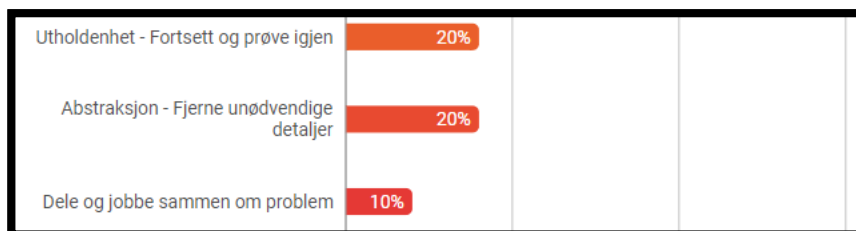
4.2.1 Kva er algoritmisk tenking?

På spørsmål om kva elevane sjølve oppfattar at algoritmisk tenking er, svara 62% at det er «Koding - programmering av algoritmar», som Weintrop (2016) skildrar som ein av «Computational Problem Solving Practices». Dei fire neste alternativa med høgast svarprosent var «Behandling av data på ein systematisk måte» (58%), «Løyse problemstillingar med algoritmar» (58%), «Designe og lage algoritmar som løyser eit problem» (55%) og «Lage algoritmar med reglar og steg-for-steg løysing» (54%), som alle kan koplast med Weintrop (2016) og/eller Brennan og Resnick (2012) sin definisjon av kompetansen algoritmisk tenking.



Figur 4: Topp 5 val for kva elevane meina algoritmisk tenking er.

Den lågaste svarprosenten finn me på alternativet «Dele og jobbe saman om problem» med berre 10%, som Brennan og Resnick (2012) har som ein del av høgste nivå av algoritmisk tenking. Dei to neste med lågast svarprosent, begge på 20%, er alternativa «Uthalden – Fortsette og prøve igjen» og «Abstraksjon – Fjerne unødvendige detaljar», som er med i Barefoot Computing (2022) og Brennan og Resnick (2012) sin oppsummering av kva kompetansen algoritmisk tenking er.



Figur 5: Dei tre alternative for å skildre algoritmisk tenking med lågast svarprosent

4.2.2 Opplæring i algoritmisk tenking og programmering

Elevane svara nokså spreidd på spørsmåla om algoritmisk tenking og programmering er prioritert, om det er vanskeleg å lære seg og om det er gitt god opplæring.

	Gjennomsnitt	Modus	Median	Standardavvik
Prioritert	3,4	4 (75)	3	1,3
Vanskeleg	3,5	3 (64)	3	1,5
God opplæring	4,1	5 (64)	4	1,5

Tabell 9: Opplæring i algoritmisk tenking (resultat)

På spørsmål om eleven oppfattar at opplæring i algoritmisk tenking og programmering har blitt prioritert i matematikken, svarar elevane i gjennomsnitt ein verdi på 3,4 av 7. I dette spørsmålet er 1 skildra som svært lite og 7 som svært mykje. Det tyder at elevane har svart at det er prioritert litt mindre, då ein gjennomsnitt på 4 ville vore «normalt prioritert». Det kjem også fram at det er 7% som svara svært lite, medan mindre enn 1% svara svært mykje.

Elevane meiner at algoritmisk tenking og programmering er ein utfordrande kompetanse å tileigne seg, når dei svara med ein gjennomsnittsverdi på 3,48. I dette tilfelle er det 9% som svara at dei synes det er svært vanskeleg, og omtrent 3% svara at det er svært enkelt. Det verkar i tillegg som at dei fleste er passe nøgd med opplæringa dei har fått i algoritmisk tenking og programmering i 1T og R1, med ein gjennomsnittsverdi på 4,05.

4.2.3 Relevans til matematikkfaget

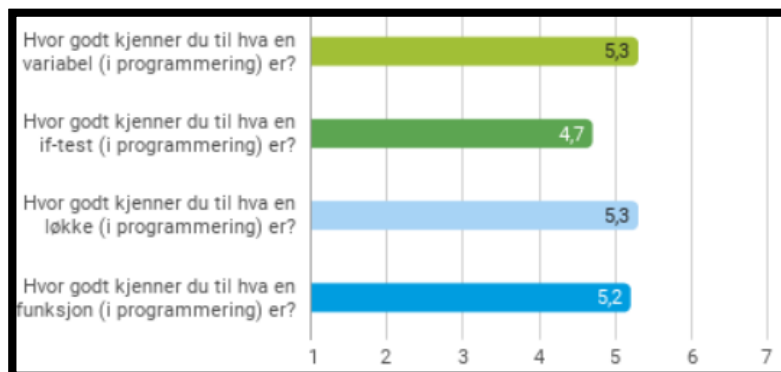
Elevane svarar at algoritmisk tenking og programmering passar greitt inn med tema som blir gjennomgått i matematikk, med ein gjennomsnittleg verdi på 4,32. Av dei som er positive til opplæringa finn me omtrent 44% av alle elevane (dei som har svart minst 5). Med standardavvik på 1,5 er det litt spreining, og dei fleste (66%) svara 3, 4 eller 5.

På spørsmål om algoritmisk tenking og programmering burde vere eit eige fag, svara elevane med stor spreining, med standardavvik på 2,0, på alle verdiane frå og med 1 til og med 7. Det verkar som dei fleste ikkje har tatt stilling til dette spørsmålet.

4.2.4 Eigenvurdering av tileigna kompetanse

I denne delen svarar elevane på spørsmål knytt til eiga oppfatning av deira tileigna kompetanse i grunnleggjande prinsipp til programmering. Minner om at skalaen går frå 1 som er «Kjenner ikkje til» og 7 som er «Trygg på bruk».

Elevane svarar at dei har god kontroll på variablar (5,3), løkker (5,3) og funksjonar (5,2), men litt mindre kontroll på vilkår/if-test (4,7).



Figur 6: Eigenvurdering av kompetanse i grunnleggjande prinsipp

Totalt sett meiner elevane at dei har god kontroll, og over 55% svarar 6 eller 7 på spørsmål om kompetanse knytt til variablar. Det er nesten ingen som svara 1, med einaste unntak i spørsmålet om valsetningar der 15% svara 1, eller «Kjenner ikkje til». Ein statistisk sumering er gitt av kompetansane i tabellen under. Her kan ein sjå at det er bra spreieing blant elevane med standaravvik frå 1,5 til 2,2, og differansen av modus og median er to for både valsetningar og funksjonar. Dette tyder på at det er mange som meina at dei kan kompetansen godt, men at det også er fleire som meina at dei ikkje kan kompetansen noko særleg.

	Gjennomsnitt	Modus	Median	Standardavvik	Variasjonsbredde
Variabel	5,3	7 (89)	6	1,8	7
If-test	4,7	7 (73)	5	2,2	7
Løkke	5,3	7 (80)	6	1,7	7
Funksjon	5,2	7 (71)	5	1,5	7

Tabell 10: Kompetanse i grunnleggjande programmering (resultat)

4.3 Haldningar

I denne delen ser eg litt på motivasjonen til kandidatane, og om dei meina at algoritmisk tenking og programmering er ein nyttig kompetanse å tileigne seg. Sidan det truleg også er ein korrelasjon mellom det å sjå nytten av denne kompetansen og faglege prestasjonar, som Grønmo et al. (2016) seier om matematikk, så er dette eit viktig spørsmål å ha med for å seie noko om status i innføring av algoritmisk tenking og programmering.

4.3.1 Motivasjon

Motivasjonen for å lære algoritmisk tenking og programmering har ein gjennomsnittsverdi på 4,1, som viser ein litt over middels haldning til denne kompetansen. Det er 25% som svarar 1

eller 2, som altså er svært lite eller lite motivert. I andre enden er det like mange (25%) som gir skåren 6 eller 7. I matematikk er motivasjonen høgare, og her har elevane i gjennomsnitt gjeve ein verdi på 4,95, noko som indikerer at elevane i matematikk R1 har ein god motivasjon for å lære matematikk. Det er heile 42% som gir skåren 6 eller 7.

	Gjennomsnitt	Modus	Median	Standardavvik
Motivasjon matematikk	4,95	6 (79)	5	1,4
Motivasjon algoritmisk tenking og programmering	4,1	5 (56)	4	1,8

Tabell 11: Motivasjon til matematikk og algoritmisk tenking (resultat)

Det er fleire som er meir motiverte til å lære seg matematikk, og gruppa er meir samla om svaret sitt om dette, enn i motivasjon til å lære seg algoritmisk tenking og programmering.

4.3.2 Nyttig kompetanse

På spørsmål om elevane ser på algoritmisk tenking og programmering som ein nyttig kompetanse, svara 67% at dei meiner det er ein nyttig kompetanse (5, 6 eller 7). Gjennomsnittsskåren er 5,05, noko som tydar på at elevane ser på algoritmisk tenking og programmering som nyttige kompetansar å tileigne seg. Berre 2% gir skåren 1, og ser på dette som heilt unyttig. På liknande spørsmål svara elevane i ein matematikk 1P klasse ein gjennomsnittsskår på 4,04 (justert frå skala frå 1 til 6, over til 1 til 7, for samanlikning) at dei er positive til nytteverdien av å lære programmering i matematikkundervisninga (Hirth, 2021, s. 56). Det ser ut til å samsvare med mitt funn, men at realfagselevane som har eit år lengre erfaring med algoritmisk tenking og programmering er endå meir positive til nytteverdien.

4.4 Erfaringar

Eg vil i dette delkapittelet vise til resultat som fell under kategorien «Erfaringar», som både er erfaringar eleven har gjort seg, men også kva arbeidsvanar elevane har tileigna seg, kva arbeidsmetodar som er nytta i undervisninga og om elevane har oppdaga bruksområde til algoritmisk tenking og programmering.

4.4.1 Programmeringsspråk

Elevane fekk i denne delen spørsmål om kva programmeringsspråk dei kjenner til, og kunne i dette spørsmålet velje fleire alternativ. Python er valt av 99% av elevane, og berre 2 elevar svara at dei ikkje kjenner til nokon av programmeringsspråka som er nemnt blant alternativa til dette spørsmålet. Ut frå dette er det tydeleg at Python har blitt «valt» som ein standard for tekstbasert

programmering i den vidaregåande skulen. Av andre språk som er verdt å nemne, så har 36% svara at dei kjenner til JavaScript, og dette har nok sin forklaring i at dette er det mest vanlege språket i valfaga informasjonsteknologi 1 og 2, og mogleg også nytta i valfag programmering på ungdomsskulen. Elles er Java (21%), C++ (22%), C# (15%) og andre (15%) dei fire alternativa som har ein relativt høg svarprosent.

4.4.2 Arbeidsvanar og eiga prioritering

Noko som kan påverka arbeidsvanar og prioritering av ein kompetanse, er om elevane tykkjer det er kjekt å arbeide med kompetansen. På spørsmål om elevane har likt å jobba med algoritmisk tenking og programmering i matematikk 1T og R1, så svara elevane i gjennomsnitt 3,8 som er omtrent midt mellom svært lite og svært mye. Det er berre 7% som svarar svært mye, og 10% som svarar svært lite. I tillegg er standardavviket på 1,7, som tyder på god spreining blant elevane sine svar.

På spørsmål om elevane har utforska programmering meir enn det som står i læreboka og det læraren har gått gjennom, svara 35% at dei ikkje har gjort noko i det heile tatt. Totalt er gjennomsnittet på 3,04, som tyder på at dei fleste arbeida svært lite med kompetansen utanom det som står i læreboka og læraren vel å gå gjennom. Det er likevel 16% som svara 5 og som dermed dreg opp gjennomsnittet og standardavviket.

I neste spørsmål, som heng saman med førre spørsmål, blir elevane spurt om dei har brukt meir/mindre tid på algoritmisk tenking og programmering enn andre fagkompetansar utanom skuletid. Her vil 4 gje ein indikasjon på at dei arbeidar lika mykje med denne kompetansen som dei andre fagkompetansane. I gjennomsnitt svara elevane 3,05, som tyder på at dei prioriterer denne kompetansen mindre enn andre fagkompetansar. Det er berre 21% (51) som svara at dei arbeider meir med denne kompetansen enn andre kompetansar utanom skuletida.

	Gjennomsnitt	Modus	Median	Standardavvik
Kjekt	3,8	4 (53)	4	1,7
Utforska på eigenhand	3,0	1 (88)	2	2,0
Tidsbruk	3,1	1 (59)	3	1,7

Tabell 12: Arbeidsvanar og eiga prioritering (resultat)

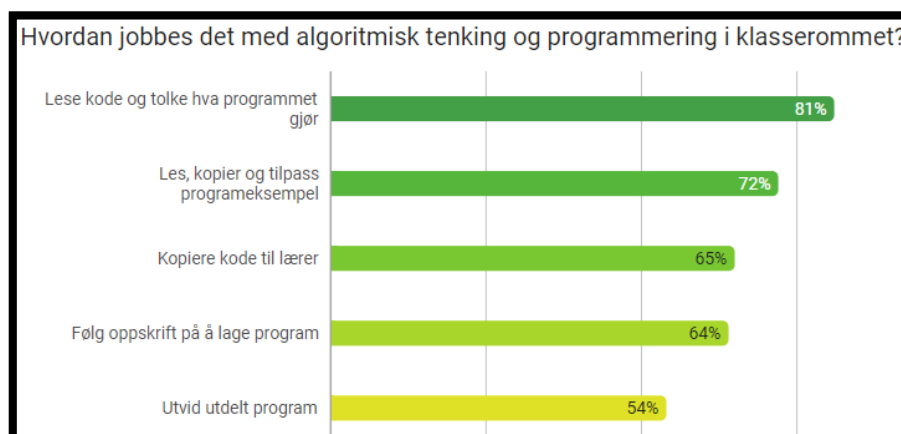
Dei tre variablane kan samla tyde på at elevane ikkje har fått noko særleg interesse for algoritmisk tenking og programmering, med både eiga utforsking og tidsbruk godt under 4 (som kan tolkast som «passe»). Samstundes er det ein del variasjon, med standardavvik frå 1,7 til

2,0, som seie oss at det er også fleire som har fått interesse for kompetansen og vel å prioritere og bruke tid på denne.

4.4.3 Arbeidsmetodar i klasserommet

På spørsmål om elevane har fått nok tid til å jobbe med algoritmisk tenking og programmering, så svara elevane i gjennomsnitt 3,64, og standardavvik på 1,5. Det er totalt 70% (178) som svara skåren 4 eller mindre. Skåren 1 er «ingen tid» og 7 er «god tid», som tyder på at kandidatane gjev uttrykk for at det er gitt lita tid til å arbeide med algoritmisk tenking og programmering. Det at kompetansen har blitt lite prioritert, kan skuldast at matematikk 1T og R1 «er travle frå før, kor det er pensum som krev hardt arbeid» (Finstad, 2020, s. 111).

Noko anna som er interessant og finne ut av, er kva metode som er nytta når elevane har arbeida med algoritmisk tenking og programmering i klasserommet. I dette spørsmålet kan elevane velje fleire svaralternativ. Svaralternativet med høgast svarprosent på 81% (207) er metoden «lese kode og tolke kva programmet gjer», som Weintrop (2016) nemner som ein kompetanse som inngår i algoritmisk tenking («Systems Thinking Practices»). Ein litt utvida metode til dette er «Les, kopier og tilpass programeksempel» som kjem like bak med 72% (184). Vidare er «Kopiere kode til lærer» på 65% (166), «Følgje oppskrift på å lage program» på 64% (162) og «Utvide utdelt program» på 54% (138) vanlege metodar innanfor opplæring i algoritmisk tenking og programmering.



Figur 7: Arbeidsmetodar for algoritmisk tenking og programmering i klasserommet

Dei neste alternativa på lista kjem heilt nede på 17% («Algoritmisk tenking utan PC») og 15% («Bruk av opplæringsvideoar»). I tillegg har 26% svara at ein metode som blir brukt er å la elevane arbeide med dette på eigenhand («Overlate til elevane»).

Finstad (2020) har sitert ein lærar som skildrar korleis lærebøkene har lagt opp til algoritmisk tenking og programmering, og som meiner at «det derimot ser ut som rein opplæring i Python-programmering, kor ein lærer ved å imitere dømer i boka» (Finstad, 2020, s. 101).

4.4.4 Bruksområde

Elevane har erfart nokon bruksområde for algoritmisk tenking og programmering, som kjem fram gjennom ein gjennomsnittsverdi på 3,66. I dette spørsmålet var 1 ingen område, og 7 veldig mange område. Berre 14% svara at dei ikkje kjenner til noko område der denne kompetansen kan brukast. Det er positivt at elevane ser bruksområde for kompetansen, då dette vil kunne motivere elevane til å lære seg kompetansen.

4.5 Kompetanse

I denne delen har elevane fått spørsmål som testar kompetanse innan grunnleggjande prinsipp innanfor algoritmisk tenking og programmering. På kvart spørsmål har elevane fått mellom 5 og 10 svaralternativ. I fyrste spørsmål, der ein testar kompetanse knytt til kommandoen «print», svara 71% (174) korrekt. Det er 13% (33) som har delvis riktig ved at dei har svara «Print vil skrive ut verdiane til a, b og c», men har ikkje fått med at det er summen av dei. Resterande (16%) har svara feil. Under er programmet som elevane skulle svare utifrå.

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7 b = a + 2
8 a = a + 11
9 c = b ** 2
10
11 print(a + b + c)
```

Figur 8: Test av forståing av kommandoen print

I neste spørsmål blir elevane testa i om dei har tileigne seg god forståing av variablar. Til dette spørsmålet svara 85% (208) riktig. Elevane viser god kontroll på variablar, og dei operasjonane som blir utført i programmet. Same programeksempel vart nytta til dette spørsmålet, men her var spørsmålet om kva verdier variablane a, b og c hadde etter at programmet var køyrt.

Elevane viser også god kontroll på valsetningar (if-testar), med 78% (192) som svara korrekt på spørsmål knytt til dette. Programmet er satt opp med både if, elif og else som tvinger elevane til å gå gjennom heile programmet. Dei siste 12% svara jamt fordelt på dei fire andre

svaralternativa, som alle var feil. Programmet som vart nytta i dette spørsmålet kan ein sjå under.

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7
8 if a < 3:
9     a = 0
10 elif a < 7:
11     a = 1
12 elif a > 5:
13     a = 2
14 else:
15     a = 9
```

Figur 9: Program med vilkår

I det neste spørsmålet blir elevane sin kompetanse i løkker testa. Svarprosenten på korrekt svar er redusert betydeleg i forhold til kompetansen vist innan print, variablar og vilkår, men likevel er det fortsett 44% (107) som har svara korrekt. For fyrste gong har fleire samla seg på eit anna alternativ, nemleg 3, som 24% (59) har svara. Som ein kan sjå av programmet under, så er det fleire måtar å kome fram til 3 på. For eksempel kan ein ta utgangspunkt i «range(3)», og gjette berre med utgangspunkt i dette at a blir 3. Elles vil eleven mogleg legge merke til at det blir lagt til totalt 3 til variabelen a i løpet av løkka, og om ein gløymer at a i utgangspunktet var 1, så vil svaret bli 3.

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 1
7 for i in range(3):
8     a += i
```

Figur 10: Program som testar kompetanse knytt til løkker

Eit anna konsept, og som blir rekna som ein grunnleggjande kompetanse i programmering, er korleis funksjonar i programmering fungerer. Det neste spørsmålet testar om elevane har

forståing av ein enkel funksjon. Av alle elevar er det totalt 50% (122) som har svara korrekt. 18% har oppgitt at dei trur programmet vil gje ei feilmelding, medan 14% har ignorert multiplikasjonen i funksjonen og har svara at funksjonen vil returnere a, b og c (ikkje produktet av dei).

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def fun(a,b,c):
7     return a * b * c
8
9 fun(1,2,3)
```

Figur 11: Program som testar forståing av funksjon

I det siste eksempelet, som kombinerar grunnleggande prinsippar i programmering med algoritmisk tenking og matematisk kompetanse, er tanken å nærma seg korleis ei eksamensoppgåve kan sjå ut. Som me skal sjå i neste delkapittel 4.6, der eg vil komme inn på eksamensoppgåver gitt våren 2022 i matematikk for vidaregåande elevar, er typisk fyrste oppgåve nettopp det å forstå kva som er meint med programmet.

Elevane klarar seg bra, og totalt har 52% (128) av elevane svara korrekt. Det er 29% (71) som har vore så ærlege at dei har svara at dei «har ikkje peiling».

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def f(x):
7     return 2*x**2 + x -5
8
9 def k(f, a, delta_x):
10    return (f(a + delta_x) - f(a - delta_x)) / (2*delta_x)
11
12 k(f, 1, 1E-8)
```

Figur 12: Programeksempel for å finne tilnæringsverdi av derivert i eit gitt punkt

4.6 Lærarundersøkinga

Sidan berre 16 lærarar har valt å svare på spørjeundersøkinga har eg valt å unngå data om kjønn og geografi i denne delen. Det er likevel av interesse å sjå på kva lærarane til dei som har svara på spørjeskjemaet svarar på spørsmål om eiga utdanning, motivasjon, erfaring og definisjon av

algoritmisk tenking. I tillegg blir også lærarane spurte om eigen kompetanse i algoritmisk tenking og programmering og har fått dei same oppgåvene som elevane fekk.

4.6.1 Formell utdanning og motivasjon for å lære elevane

Lærarane har svara at dei stort sett har formell utdanning innan algoritmisk tenking og programmering, med 25% (4) som svara at dei har minst 60 studiepoeng og 56 % (9) mindre enn 60 studiepoeng. Berre 3 lærarar (19%) svara at dei har ingen relevant utdanning knytt til algoritmisk tenking og programmering.

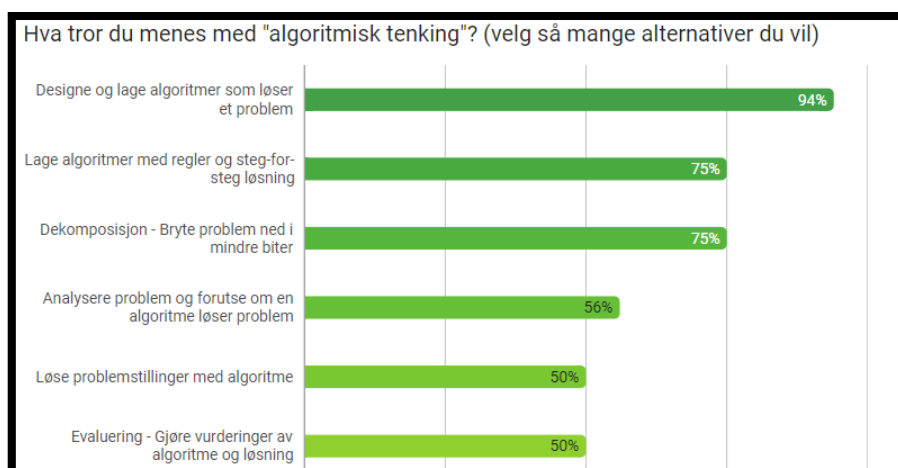
Når det gjeld motivasjon for å lære elevane algoritmisk tenking og programmering svara lærarane ein gjennomsnittsverdi på 5,38, som tydar at lærarane er sjølve motiverte til å lære elevane opp i denne kompetansen. Som eg nemnde i kapittel 3.9, så er det ein sjanse for at dei som har valt å svare på spørjeundersøkinga er nettopp dei som er meir motiverte og trygge på å lære vekk kompetansen.

100% av lærarane svart at dei kjenner til Python som programmeringsspråk, og fleire kjenner til C++ (56%), Java (50%), andre (50%) og JavaScript (44%).

4.6.2 Kva er algoritmisk tenking?

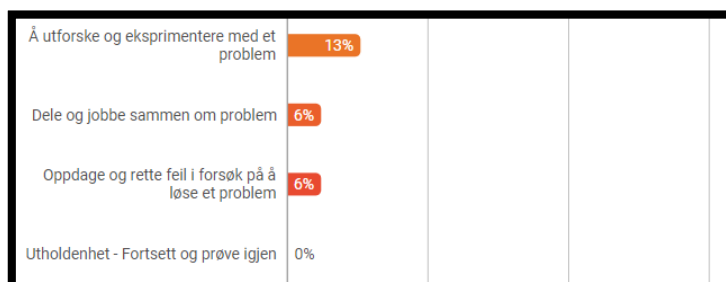
94% (15) av lærarane meiner at algoritmisk tenking blant anna er å «Designe og lage algoritmar som løyser eit problem», på tilsvarande spørsmål svara 55% av elevane det same. Av alle alternativ som minst 50% har valt finn me «Lage algoritmar med reglar og steg-for-steg løysing» (75%), «Dekomponering – Bryte problem ned i mindre bitar» (75%), «Analysere problem og sjå om ein algoritme løyse problemet» (56%), «Løyse problemstillingar med algoritmar» (50%) og «Evaluering – Gjere vurderingar av algoritmar og løysing» (50%).

Kompetansane som lærarane knytt til algoritmisk tenking finn me igjen i både Brennan og Resnick (2012), Weintrop (2016) og i «den algoritmiske tenkjaren» til Utdanningsdirektoratet (2019).



Figur 13: Lærarane si tolking av algoritmisk tenking

«Koding - Programmere algoritmar» som 44% av lærarane svarta var elevane sitt toppval med 62%. Dei tre svaralternativa som færrest av lærarane valte er «Uthald – Fortsette og prøve igjen» (0%), «Oppdage og rette feil i forsøk på å løyse eit problem» (6%) og «Dele og jobbe saman om problem» (6%), som alle er nemnd under som arbeidsmåtar i Utdanningsdirektoratet (2019) si skildring av «den algoritmiske tenkjaren».



Figur 14: Lærarane sine tre minst populære val til algoritmisk tenking

4.6.3 Algoritmisk tenking og programmering i matematikkfaga

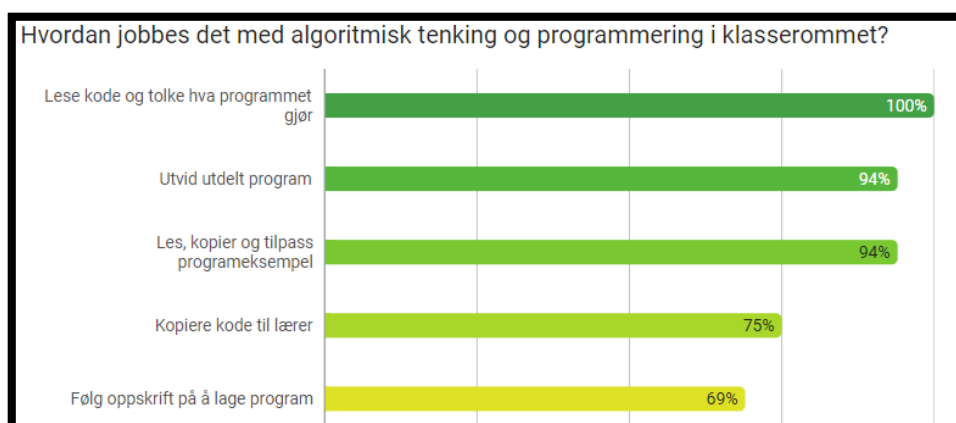
Lærarane melder om at dei prioriterer og meiner at det passar inn å ta med algoritmisk tenking og programmering i matematikk med ein gjennomsnittsverdi på 4,25 (på begge spørsmåla). Sjølv om lærarane meiner at dette passer inn i faget, så uttrykker dei også at denne kompetansen kunne vore eit eige fag med ein gjennomsnittsverdi på 5 (63% svarta meir enn 4). I ei anna undersøking av Finstad (2020) svarar fem av sju lærarar dei er positive til å inkludere programmering og algoritmisk tenking i matematikkfaga totalt sett, noko som samsvarar med mitt funn.

På spørsmål om algoritmisk tenking og programmering er ein nyttig kompetanse, så svara lærarane ein gjennomsnittsverdi på 5,69. Ingen av lærarane har svara mindre enn 4 på dette spørsmålet, som viser at dei er einige om at det er ein nyttig kompetanse å tileigne seg.

I forhold til kva bruksområde algoritmisk tenking og programmering kan brukast til, så svara lærarane i gjennomsnitt 5,19, som kan seie oss at alle lærarar ser ein del bruksområde.

4.6.4 Arbeidsmetodar

Alle (100%) melder om at dei brukar metoden å «lese kode og tolke kva programmet gjer» som metode. I tillegg svara nesten alle (94%) at å «utvida utdelt program» og «les, kopier og tilpass programeksempel» er vanlege undervisningsmetodar. Å «kopiere kode til lærer» (75%) er også ein populær læringsmetode, og til slutt å «følgje oppskrift på å lage program» (69%).



Figur 15: Arbeidsmetodar for algoritmisk tenking og programmering

Under er ei samanlikning av elevsvar og lærarsvar, med differanse i prosentpoeng. Me legg merke til at det er dei same fem alternativa som har høgast relativ frekvens hjå elevane, som hjå lærarane.

	Lese kode og tolke kva programmet gjer	Utvide utdelt program	Les, kopier og tilpass programeksempel	Kopiere kode til lærer	Følgje oppskrift på å lage program
Lærarsvar	100%	94%	94%	75%	69%
Elevsvar	81%	54%	72%	65%	64%
Differanse	19 pp.	40 pp.	22 pp.	10 pp.	5 pp.

Tabell 13: Undervisningsmetode (resultat)

Det er spesielt stort sprik mellom svara på «Utvide utdelt program», og det er mogleg at elevane har ei anna (smalare) tolking enn lærarane på dette alternativet. Den minst brukte metoden som er oppgitt av lærarane er «Opplæringsvideoar» med berre 13%, som elevane har svart 15% på.

4.6.5 Kompetanse

Lærarane svara om eigen kompetanse, innan grunnleggande prinsipp i programmering, at dei er trygg på bruk av variablar (6,62), vilkår (6,69), løkker (6,62) og funksjonar (6,38).

I resultatata på tilsvarende program og alternativ som elevane vart testa i, får lærarane gode resultat. Lærarane har 100% riktig på spørsmål knytt til programeksempel med print, variablar, vilkår og funksjonar, men omtrent 1 av 5 har svara feil på oppgåva som testar kompetanse knytt til løkker. Lærarane som har svara feil i oppgåva har tatt med ein ekstra runde i løkka, og dermed lagt til 3 til slutt, og dermed fått 7 som svar (4 var riktig).

Alle lærarane har også svara riktig på den siste oppgåva, som var eit eksempelprogram på å finne ein tilnæringsverdi til den deriverte i eit gitt punkt.

4.7 Data frå eksamen våren 2022

I tillegg til å samle inn egne data, har eg også fått tak i data frå eksamen våren 2022. Data eg har fått utlevert er resultat på alle deloppgåver, som testa kompetansar innan algoritmisk tenking og programmering. Desse data er gjort tilgjengeleg for meg i direkte samarbeid med Utdanningsdirektoratet. Alt av data er anonymisert og trygt overført til meg, og deretter analysert av meg.

Det er viktig å understreke at alle som tok eksamen våren 2022 er privatistar, og dette vil sjølvsagt kunne påverke resultatet på eksamen totalt og på einskilde oppgåver. Det er i tillegg ein overgangsfase mellom to læreplanar (LK06 og LK20), som kan ha ført til forvirring ved oppmelding til eksamen med tanke på at det var to fagkodar til same fag.

Eg vil i dette delkapittelet gå gjennom dei relevante oppgåvene i matematikk 1T, 1P, 2P, S1 og R1 frå eksamen våren 2022. Oppgåvene blir diskutert i tillegg til at resultatata blir lagt fram.

4.7.1 Eksamen i matematikk 1T, 1P og 2P

I matematikk 1T, 1P og 2P er det berre ei oppgåve på kvar av dei tilhøyrande eksamenssett, som kjem direkte i kategorien algoritmisk tenking og programmering. Alle oppgåvene er gitt på del 1, utan hjelpemiddel, og er av typen «kva vil eleven finne ut av?» og «kva blir resultatet?».

```

1 startverdi = 2000
2 verdi = startverdi
3 vekstfaktor = 1.05
4 år = 0
5
6 while verdi < startverdi * 2:
7     verdi = verdi * vekstfaktor
8     år = år + 1
9
10 print(verdi)
11 print(år)

```

Figur 16: Oppgave 5 (del 1), eksamen 1P våren 2022

```

1 beløp = 0
2 vekstfaktor = 1.02
3 innskot = 20000
4 år = 0
5
6 while beløp < 500000:
7     beløp = beløp + innskot
8     beløp = beløp * vekstfaktor
9     år = år + 1
10
11 print(år)
12 print(beløp)

```

Figur 17: Oppgave 4 (del 1), eksamen 2P våren 2022

Til oppgåvene over krev ein at elevane har grunnleggande prinsipp i programmering på plass, i tillegg til å vere trena i lesing av program. Den fyrste algoritmen (frå eksamen i 1P) skal finne ut kor mange år det tar før eit startbeløp har dobla seg, gitt ei fast årleg rente. Den andre algoritmen finn kor lang tid det tar før ein har nådd eit visst beløp med fast årleg sparing og fast årleg rente (eksempelet frå 2P) er relevant og ei ganske vanleg problemstilling i matematikk 2P. Elevane i 1P (1931 elevar) og 2P (191 elevar) har i gjennomsnitt fått i underkant av 1 poeng (0,892 for 1P og 0,969 for 2P) av totalt 3 poeng. Ein poengfangst på omtrent 30%.

I matematikk 1T fekk elevane omtrent same, men litt betre, resultat på oppgåva som er oppstilt under. Dei fekk totalt 1,04 av 3 poeng i gjennomsnitt, med ein poengfangst på omtrent 35%.

```

1 def f(x):
2     return x ** 2 # Definerer funksjonen f gitt ved f(x) = x ^ 2
3
4 x = 1
5
6 while f(x) <= 400:
7     print(f(x))
8     x = x + 1

```

Figur 18: Oppgave 4 (del 1) eksamen 1T våren 2022

Oppgåva som vart gitt i 1T testar grunnleggande prinsipp i programmering. I tillegg til at ein må kunne lese at programmet utfører ein algoritme for å skrive ut alle kvadrattal som er mindre eller lik 400.

4.7.2 Eksamen i matematikk S1 og R1

I matematikk S1 vart det gjeve to oppgåver knytt til algoritmisk tenking og programmering, oppgåve 6 i del 1 og oppgåve 4 i del 2. Oppgåve 6 er av same typen som for vg1, der elevane skal forklare kva programmet skal finne ut av og ei oppfølgingsoppgåve der ein skal sjølv finne teoretisk riktig sannsyn.

```

1 from random import randint # Importerer funksjonen randint(a, b). Denne gir
2                             # eit tilfeldig heiltal frå og med a til og med b.
3
4 N = 1000000
5 gunstige = 0
6
7 for i in range(N):          # Gjentar N gonger
8     a = randint(1, 6)
9     b = randint(1, 6)
10    if a + b == 9:
11        gunstige = gunstige + 1
12
13 print(gunstige/N)

```

Figur 19: Oppgåve 6 (del 1) frå eksamen i S1 våren 2022

I oppgåve 4 på del 2, skal elevane sjølve lage eit program for å simulere eit terningspel.

Oppgåve 4

I eit spel kastar du tre terningar. Du multipliserer saman augetalet på terningane. Dersom dette produktet er større enn 100, vinn du.

Bestem sannsynet for å vinne ved å køyre simuleringar. Hugs å vise korleis du kjem fram til svaret.

Figur 20: Oppgåve 4 (del 2) frå eksamen i S1 våren 2022

Det var berre 54 elevsvar på desse oppgåvene, og resultatet hamna på omtrent 0,5 poeng av 3 moglege i oppgåva på del 1, og i gjennomsnitt 0,333 poeng av 3 moglege på del 2.

I matematikk R1 finn ein flest poenggivande oppgåver innanfor det som er algoritmisk tenking og programmering. I del 1 er det liknande problemstilling som for matematikk S1, «Kva skjer når koden blir køyrd?», «Kva ynskjer eleven å finne ut av?» og «Finn teoretisk riktig svar.».

```

1 def f(x):
2     return x/(1+x**2) # Definerer funksjonen f(x)=x/(1+x^2)
3
4 x = 0
5 h = 0.001
6 while f(x) <= f(x+h):
7     x = x+h
8
9 print(x)

```

Figur 21: Oppgåve 5 (del 1) eksamen i matematikk R1 våren 2022

I denne oppgåva (sjå figur over) må eleven kjenne til grunnleggande prinsipp i programmering og funksjonsdrøfting knytt til rasjonale funksjonar, og i tillegg kjenne til algoritmen for å finne eit toppunkt numerisk når ein kjenner til monotonieigenskapane til ein funksjon. Programmet

vil altså finne ein tilnærming (med ein feilmargin mindre enn 0.001) til eit toppunkt (berre x-verdi) for funksjonen som er skildra i funksjonen $f(x)$.

I oppgåve 5 på del 2 må eleven utvikle ein eigen algoritme for å finne ut om tre punkt dannar ein rettvinkla trekant, og deretter lage eit program som utfører algoritmen. Oppgåva set krav til kompetanse innan algoritmisk tenking i tillegg til at eleven må ha grunnleggande prinsipp i programmering på plass.

Oppgåve 5

Gitt tre punkt $A(a,b)$, $B(c,d)$ og $C(e,f)$.

- a) Beskriv ein algoritme som du kan bruke til å avgjere om $\triangle ABC$ er ein rettvinkla trekant.
- b) Skriv ein kode basert på algoritmen du beskreiv i oppgåve a). Input skal vere koordinatane a , b , c , d , e og f . Output skal vere ein av følgjande tekstar:
 - Punkta dannar ein rettvinkla trekant.
 - Punkta dannar ikkje ein rettvinkla trekant.

Figur 22: Oppgåve 5 (del 2) frå eksamen i matematikk R1 våren 2022

Av 82 elevsvar er det gitt omtrent 0,7 poeng i gjennomsnitt på oppgåva frå del 1, der ein maks kunne få 4 poeng. I del 2 er det gitt i gjennomsnitt 0,95 poeng av maksimalt 5 poeng. Dette gjev ein gjennomsnittleg poengfangst på 18% av totalt moglege.

5 Analyse og drøfting av inndelingar

I dette kapittelet vil eg sjå nærare på forskjellar mellom forskjellige inndelingar, som blant anna kjønn, prestasjon, tidlegare erfaringar og motivasjon. I analysen er det nytta gjennomsnitt som hovudindikator ved samanlikning av gruppene og kva dei har svara. Det er også nytta t-test for å finne ein p-verdi som kan seie noko om signifikante forskjellar mellom to grupper. Når t-test er nytta er det valt p-verdi til 0,05 som grense for signifikant forskjell, og nullhypotesen er at det er ingen forskjell. Eg har i tillegg nytta meg av Pearson sin korrelasjonskoeffisient for å sjå på samvariasjon og korrelasjon mellom variablar.

5.1 Kjønnforskjellar

I spørjeundersøkinga var det 147 menn og 110 kvinner, i tillegg til 3 som identifiserer seg som annet. Dette er ei fordeling på 57,2% menn og 42,8% kvinner, som er omtrent den same fordelinga som i TIMSS Advanced frå 2015, som analyserte matematikk R2 elevar, der det var 62% menn og 38% kvinner (Grønmo et al., 2016, s. 35). I dette delkapittelet vil eg sjå nærare på kjønnforskjellar, og har valt å sjå vekk frå dei 3 som identifiserer seg som annet.

5.1.1 Karakterar frå matematikk 1T og 1.termin R1

Karaktersnitt frå matematikk 1T er signifikant høgare (p-verdi på 0.0003) blant kvinnene som svarte 4,85 mot mennene sitt gjennomsnitt på 4,39. Altså, omtrent ein halv karakter betre i gjennomsnitt blant kvinnene. I 1.termin for matematikk R1 er forskjellen mindre, og kvinnene har her svara 4,18 mot mennene sitt gjennomsnitt på 3,85. Det er likevel ein signifikant forskjell mellom kjønna, med ein kalkulert p-verdi på 0,0117.

5.1.2 Tidlegare erfaringar

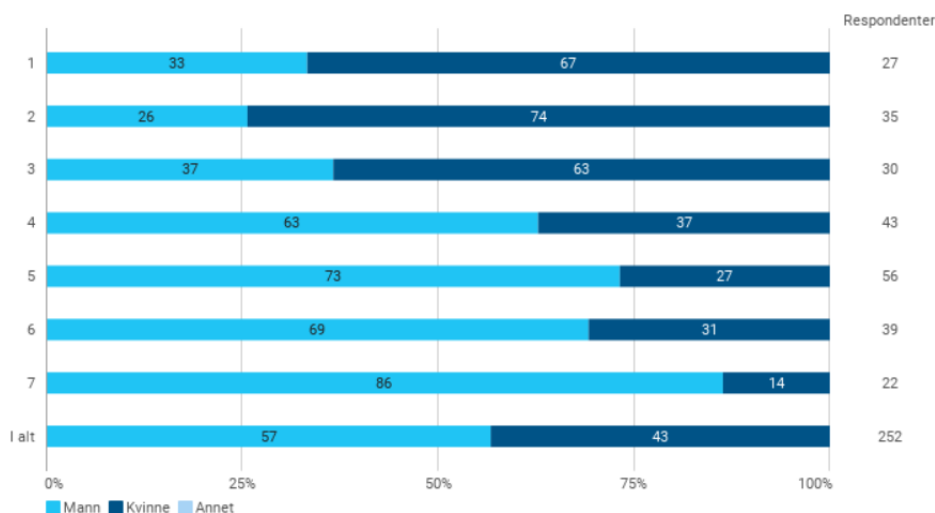
Det er 35 menn og 12 kvinner som har valt programmering som valfag i ungdomsskulen. Det er i tillegg 23 menn som har valt IT1, medan berre 3 kvinner har svart at dei har valt faget.

Med dette som utgangspunkt så har mennene meir erfaring med algoritmisk tenking og programmering enn kvinnene, og mogleg tidlegare interesse for kompetansen. Når det er sagt, så har dei fleste 192 av 260 svara at dei har ingen erfaring frå før av.

5.1.3 Motivasjon og arbeidsvanar

Motivasjonen for matematikk er lika høg hjå begge kjønn, og det er ingen signifikant forskjell (p-verdi på 0,9243). I gjennomsnitt svara begge kjønn at motivasjonen til å lære seg matematikk er på 4,93. Ser me på motivasjon for å lære seg algoritmisk tenking og programmering, er det

derimot stor forskjell (p-verdi på 0,0001). Kvinnene svarta i gjennomsnitt 3,29, medan mennene svart 4,67. Dette er ein tydeleg indikasjon om at kvinner generelt har lågare motivasjon enn menn for å lære seg algoritmisk tenking og programmering. Som ein kan sjå av figuren under, så er det kvinnene som svarta oftast 1, 2 og 3, medan mennene har høgst frekvens for 5, 6 og 7.



Figur 23: Motivasjon for å lære seg algoritmisk tenking og programmering

På spørsmål om algoritmisk tenking og programmering er ein vanskeleg kompetanse å tileigne seg, svarta også kvinnene lågare enn mennene. I gjennomsnitt svarta kvinnene 2,9, medan mennene svarta 3,9. I dette spørsmålet er 1 «svært vanskeleg» og 7 «svært enkelt», som tyder på at mennene synes at det er enklare å tileigne seg kompetanse enn kvinnene.

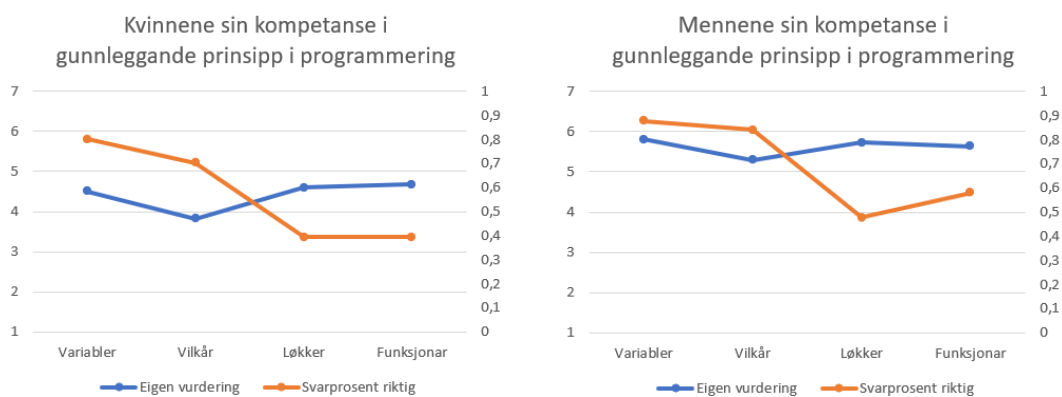
Resultata viser også at menn brukar meir tid på å lære seg denne kompetansen enn kvinnene. Mennene svarta 3,31, medan kvinnene 2,70 på spørsmål om dei brukar meir tid på denne kompetansen samanlikna med andre kompetansar i andre fag. Mennene svarta også betydeleg høgare på spørsmål om dei har utforska programmering meir enn det som står i læreboka og det som læraren går gjennom. Her svarta mennene 3,74 og kvinnene svarta 2,05, som viser ein svært signifikant forskjell (p-verdi på 0,0001).

Både tidsbruk og meir utforsking innan ein kompetanse, heng ofte saman med motivasjonen for å lære seg nettopp denne kompetansen. Mine utrekningar viser at det er ein korrelasjon på 0,641 mellom utforsking på eigenhand og motivasjon, som kan tyde på at det er ein samanheng mellom desse i følgje Cresswell (2012). Det er også 0,669 i korrelasjon mellom kor vanskeleg ein synes at kompetansen er å lære seg og motivasjonen for å lære seg kompetansen.

5.1.4 Kompetanse

Når det gjeld grunnleggande prinsipp i programmering, vurderer kvinnene sine eigen kompetansar til eit gjennomsnitt på 4,51 om variablar, 3,82 om vilkår, 4,60 om løkker og til slutt 4,68 om funksjonar. Til samanlikning vurderer mennene sin kompetanse til å vere 5,81 om variablar, 5,29 og vilkår, 5,73 om løkker og 5,64 om funksjonar.

Faktisk kompetanse innanfor grunnleggande prinsipp viser at kvinnene har ein korrekt svarprosent på 80,8% om variablar, 70,2% om vilkår, 39,4% om løkker og 39,4% om funksjonar. For mennene er tilsvarande korrekt svarprosent på 87,7% om variablar, 84,1% om vilkår, 47,8% om løkker og 58% om funksjoner. I figurane under er det gjort ein samanlikning mellom eigen vurdering av kompetanse og korrekt svarprosent.



Figur 24: Kvinnene og mennene sin kompetanse samanlikna med eigen vurdering av kompetanse

Det er ikkje eit direkte samanlikningsgrunnlag, men ein kan likevel sjå ein tendens til at begge kjønn undervurderer eigen kompetanse innan variablar og vilkår, medan begge har overvurdert eigen kompetanse knytt til løkker og funksjonar. Eller så kan ein forklaring vere at oppgåvene om variablar og vilkår var enklare enn dei som testa løkker og funksjonar.

I den siste kompetanseoppgåva svara 40,4% av kvinnene korrekt og 60,9% av mennene. Denne oppgåva likna meir på ei eksamensoppgåve, som me såg i kapittel 4.7, og gjev oss, saman med avsnittet over, ein indikasjon på at mennene har tileigna seg meir kompetanse innan grunnleggande prinsipp i programmering og algoritmisk tenking enn kvinnene.

5.1.5 Oppsummering av kjønnsforskjellar

	Menn (gj.snitt)	Kvinne (gj.snitt)	SD (Menn)	SD (Kvinner)	P-verdi
Karakter 1T	4,39	4,85	1,08	0,85	0,0003
Karakter R1 (1.termin)	3,88	4,22	1,16	0,94	0,0117
Motivasjon matematikk	4,94	4,93	1,45	1,43	0,9243
Motivasjon algoritmisk tenking og programmering	4,67	3,29	1,64	1,72	0,0001
Utforska meir enn det som står i læreboka	3,75	2,05	2,05	1,46	0,0001

Tabell 14: Oppsummering av kjønnsforskjellar

5.2 Høgt presterande og lågt presterande

I denne seksjonen deler eg elevane inn i to grupper, dei høgt presterande (karakter 5 eller 6) og dei lågt presterande (karakter 1, 2 eller 3). Eg har i denne samanheng tatt utgangspunkt i karakter får 1.termin i matematikk R1. Det er 35,4 % (92) som er i den høgt presterande gruppa, og 30,7% (80) som er i den lågt presterande gruppa.

5.2.1 Omgrepet algoritmisk tenking

Forholdet mellom dei to gruppene er 23 til 20, som eg har brukt i analysen i forskjellane mellom synet til desse gruppene på kva algoritmisk tenking er. I figuren under har eg brukt fargekodar for å markere forskjellar mellom dei to gruppene, i forhold til kor mange i kvar gruppe som har svara dei forskjellige alternativa til kva algoritmisk tenking er. Gul farge indikerer ingen/liten forskjell på gruppene, medan rød indikerer at dei lågt presterande har fleire som har valt alternativet og til slutt indikerer grøn at dei høgt presterande har fleire som har valt alternativet.

	Å utforske og eksperimentere med et problem	Design og lage algoritmer som løser et problem	Oppdage og rette feil i forsøk på å løse et problem	Utholdenhet - Fortsett og prøve igjen	Dele og jobbe sammen om problem	Analysere problem og forutse om en algoritme løser problem	Lage algoritmer med regler og steg-for-steg løsning	Dekomposisjon - Bryte problem ned i mindre biter	Mønstre - Å finne og bruke likheter	Abstraksjon - Fjerne unødvendige detaljer	Evaluering - Gjøre vurderinger av algoritme og løsning
Høg	36	62	31	20	9	46	54	41	49	13	37
Lav	32	34	27	18	10	33	47	30	35	20	28

Figur 25: Prestasjon og omgrepet algoritmisk tenking

Av dei elleve første alternativa (av totalt sytten) på kva algoritmisk tenking er, så er fem av dei omtrent like mykje svara av begge gruppene. Berre ein av desse elleve er valt oftare blant dei

lågt presterande, og det er «Abstraksjon – Fjerne unødvendige detaljar», som er ein viktig del av algoritmisk tenking (Utdanningsdirektoratet, 2019; Shute et al., 2017).

Me legg også merke til at det er heile fem alternativ som er relativt hyppigare svara av dei høgt presterande elevane. Det var i tillegg seks alternativ til, som dei høgt presterande svara relativt hyppigast. Til saman er fordelinga at elleve alternativ er flest svara av høgt presterande, fem omtrent likt og berre eit alternativt er svara oftare blant dei med låg prestasjon. Dette indikerer at elevar, som er høgt presterande i matematikkfaget, har ei meir utvida forståing av omgrepet algoritmisk tenking.

5.2.2 Motivasjon og haldningar

Motivasjonen for å lære matematikk er, ikkje overraskande, høgare hjå dei høgt presterande, med ein gjennomsnittleg skår på 5,60 mot dei lågt presterande som skåra 4,36. Når det gjeld motivasjonen for å lære algoritmisk tenking og programmering er forskjellane mindre (p-verdi på 0,1218), men fortsett er det dei høgt presterande som er mest motivert med 4,52 samanlikna med ein motivasjonsskår på 4,08 hjå dei lågt presterande.

På spørsmål om algoritmisk tenking og programmering passar inn med tema i matematikkfaga, svara dei høgt presterande ein gjennomsnittskår på 4,64 og dei lågt presterande svara 4,37. Det er liten forskjell i haldninga til at algoritmisk tenking og programmering passar inn i matematikkfaga, og ingen signifikant forskjell mellom gruppene (p-verdi på 0,2195). Om kompetansen er nyttig, er begge gruppene einige i at den er det, og svara 5,35 for høgt presterande og 5,21 for lågt presterande.

Når det gjeld eiga utforsking knytt til algoritmisk tenking og programmering, er det nesten ingen forskjellar å finne. Om elevane har brukt meir/mindre tid på algoritmisk tenking og programmering enn andre fagkompetansar utanom skuletida, så skåra dei høgt presterande 3,26 og dei lågt presterande på 3,13. På spørsmål om elevane har utforska programmering meir enn det som står i læreboka og læraren har gått gjennom, skåra begge gruppene heilt likt på 3,24. Her kan me få ein indikasjon på at det å sette av tid til og jobbe meir med kompetansen enn det som blir gjennomgått i undervisninga er uavhengig av tidlegare prestasjonar i matematikkfaget (p-verdiar på 0,6151 og 0,9637).

5.2.3 Kompetansar

I tabellen under er det lista opp prosentvis riktige svar for kvar av gruppene i dei forskjellige grunnleggande prinsippa i programmering og oppgåva om den deriverte.

Kompetanse	print()	Variablar	Vilkår	Løkker	Funksjonar	Derivert
Høgt presterande	81%	92%	85%	48%	59%	64%
Lågt presterande	56%	75%	74%	41%	48%	47%

Tabell 15: Tileigne kompetanse gruppert etter karakter i 1.termin matematikk R1

Dei høgt presterande i matematikkfaget, presterer også høgt på oppgåvene som testar kompetanse innan grunnleggande prinsipp i programmering og algoritmisk tenking. Dei lågt presterande utmerka seg positivt innan forståing av variablar og vilkår, men negativt spesielt med tanke på eksempelet med print(). Utifrå dette resultatet har me ein indikasjon på at dei som presterer høgt i matematikkfaget, også presterer betre innan algoritmisk tenking og bruk av programmering.

5.2.4 Oppsummering av forskjellar mellom høgt og lågt presterande elevar

	Høgt prest. (gj.snitt)	Lågt prest. (gj.snitt)	SD (Høg)	SD (Lågt)	P-verdi
Motivasjon matematikk	5,60	4,38	1,04	1,44	0,0001
Motivasjon for å lære algoritmisk tenking og programmering	4,52	4,09	1,75	1,81	0,1218
Passar algoritmisk tenking og programmering i matematikk	4,64	4,36	1,53	1,48	0,2195
Utforska meir enn det som står i læreboka	3,24	3,24	1,96	1,99	0,9637
Brukt meir tid enn på andre fagkompetansar	3,26	3,13	1,53	1,70	0,6151

Tabell 16: Oppsummering av forskjellar mellom høgt og lågt presterande elevar

5.3 Geografi

I utgangspunktet hadde eg tenkt å analysere geografiske forskjellar med ein inndeling i nord (Trøndelag, Nordland og Finnmark og Troms), vest (Møre og Romsdal, Vestland og Rogaland), øst (Innlandet, Viken og Oslo) og Sør (Agder og Vestfold og Telemark). På bakgrunn av talet på svar per fylke, vart dette problematisk både med tanke på anonymiteten til klassane som har valt å svare og at nokre av grupperingane vart veldig få.

5.4 Med og utan forkunnskapar

I denne delen vil eg dele inn elevane i to grupper, der den eine gruppa er dei som har tatt programmering som valfag frå ungdomsskulen, og den andre er dei som har ingen erfaring. Det er 18,5% (48) som har svara at dei har hatt programmering som valfag, og 73,5% (191) som har svara at dei har ingen forkunnskapar.

5.4.1 Kompetanse

Om me denne gongen startar med kompetansen, og spørsmåla som testar grunnleggande kompetanse i algoritmisk tenking og programmering, så er det ein tydeleg indikasjon i resultatata at det hjalp elevane å ha forkunnskap frå ungdomsskulen. På alle spørsmål svara dei med forkunnskapar betre enn dei som var utan forkunnskapar.

Kompetanse	print()	Variablar	Vilkår	Løkker	Funksjonar	Derivert
Med forkunnskap	88%	91%	91%	56%	67%	60%
Utan forkunnskap	66%	84%	74%	42%	43%	49%

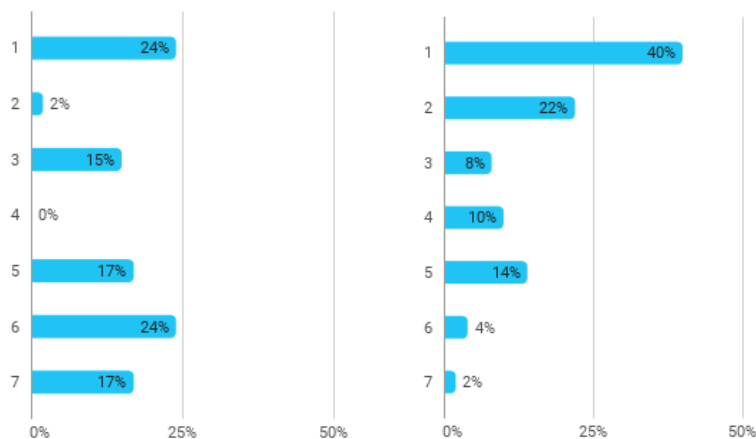
Tabell 17: Forkunnskapar og kompetanse i grunnleggande algoritmisk tenking og programmering

Elevane med forkunnskapar svara 4,62 på spørsmål om algoritmisk tenking og programmering er ein vanskeleg kompetanse å tileigne seg, medan dei utan forkunnskap svara 3,18. Med ein signifikant forskjell (p-verdi på 0,0001) i svar på om algoritmisk tenking og programmering er ein vanskelege kompetanse å tileigning seg, så kunne ein likevel ha forventa enda større forskjell i resultatata frå tabellen over.

5.4.2 Motivasjon og haldningar

Det er liten/ingen forskjell i haldningar kring prioritering og tidsbruk i desse gruppene. Motivasjon for å lære seg matematikk er heilt lik i begge grupper med ein skår på 4,96. Ser ein derimot på motivasjon for å lære seg algoritmisk tenking og programmering, så er forskjellen ganske stor med ein skår på 5,26 for dei med forkunnskap og berre 3,70 for dei utan forkunnskap.

Elevane med tidlegare erfaring svarar at dei ser fleire (4,48) bruksområde for algoritmisk tenking og programmering enn dei utan tidlegare erfaring (3,34). Når det gjeld utforsking er det også stor forskjell, og dei med forkunnskap svara 4,33 på spørsmål om dei har utforska programmering meir enn det som står i læreboka og det læraren har gått gjennom. Gruppa utan forkunnskap svara i stor grad at dei har utforska lite og har gitt ein gjennomsnittsskår på 2,58.



Figur 26: Utforsking, til venstre med forkunnskap og utan forkunnskap til høgre

Figuren over viser at elevane med forkunnskap har anten fått sansen for programmering og blitt veldig nysgjerrig eller dei har lite/ingen interesse. I gruppa utan forkunnskap er det flest som svara 1, og over 60% svara 1 eller 2.

Det er ein tydeleg haldningsforskjell mellom gruppene om dei ser på algoritmisk tenking og programmering som ein nyttig kompetanse. Elevane med forkunnskap svara med ein gjennomsnittskår på 5,91 (av 7), medan dei utan forkunnskap svara ein skår på 4,77. Heile 39% (18) av dei med forkunnskap svara 7 på at kompetansen er nyttig, tilsvarende svara berre 16% (31) det same blant dei utan forkunnskap.

Om algoritmisk tenking og programmering passar med tema i matematikkfaga, svara gruppa med programmering som valfag frå ungdomsskulen ein skår på 5,16 medan dei utan forkunnskap svara 4,08. Dette indikere at det kan vere enklare for dei med forkunnskapar å sjå samanhengar mellom tema i matematikken og kompetansen som kjem med algoritmisk tenking og programmering.

5.4.3 Omgrepet algoritmisk tenking

Det er små forskjellar når det kjem til kva dei to gruppene ser på som ein del av omgrepet algoritmisk tenking. Tabellen under viser ein oversikt over kor stor prosentdel (og talet på elevar i parentes) frå kvar av gruppene har svara på dei forskjellige alternativa.

	Med forkunnskapar	Utan forkunnskapar
Å utforske/eksperimentere med problem	28,3%	40,2%
Feilsøking	43,5%	30,7%
Analysere problem og forutse algoritme	58,7%	46,0%
Koding – Programmere algoritmer	54,3%	64,6%
Automatisere problemløsing	52,2%	36,5%

Tabell 18: De største forskjellene som de med og uten forkunnskaper mener at algoritmisk tenking er

Det er tre alternativ som dei med forkunnskapar har svara meir enn ti prosentpoeng meir enn dei utan, og på same måte er det to alternativ der meir enn ti prosentpoeng av dei utan forkunnskapar har svara enn dei med. Det er med andre ord lite som tyder på at dei med forkunnskapar har eit meir utvida syn på kva kompetansen algoritmisk tenking er.

5.4.4 Oppsummering av forskjellar mellom dei med/utan forkunnskapar

	Med fork. (gj.snitt)	Utan fork. (gj.snitt)	SD (Med)	SD (Utan)	P-verdi
Vanskeleg å tileigne seg kompetansen algoritmisk tenking og programmering	4,59	3,18	1,47	1,45	0,0001
Motivasjon for å lære algoritmisk tenking og programmering	5,26	3,70	1,45	1,75	0,0001
Motivasjon for å lære matematikk	4,96	4,96	1,59	1,42	0,9761
Bruksområde for algoritmisk tenking og programmering	4,48	3,34	1,81	1,72	0,0001
Algoritmisk tenking og programmering er ein nyttig kompetanse	5,91	4,77	1,20	1,62	0,0001

Tabell 19: Oppsummering av forskjellar mellom dei med/utan forkunnskapar

5.5 Motivasjon for å lære algoritmisk tenking og programmering

Dette delkapittelet ser på forskjellar blant dei som har gitt høg skår på motivasjon for å lære algoritmisk tenking og programmering, og dei som har gjeve låg skår. Eg har valt å dele dei inn i høg motivasjon, som tilsvara skår på 6 eller 7, og låg motivasjon er då dei som har gitt skår 1 eller 2. I gruppene er det like stor del av alle respondentar med 25% (63) i kvar av dei. Det er, som me også såg ein tendens til i analysen av kjønnsforskjellar, ein klar skeivfordeling mellom kjønn i dei to gruppene med låg og høg motivasjon. I gruppa med høg motivasjon finn me 73% (46) menn og i gruppa med låg motivasjon er det 71% (44) kvinner. Dette er ein indikasjon på at menn er meir motivert til å lære seg algoritmisk tenking og programmering, som me også såg i kapittel 5.1.

5.5.1 Matematikk

Dei med høg motivasjon for å lære algoritmisk tenking og programmering har også høg motivasjon for å lære matematikk, sjølv om den ikkje er like høg med ein gjennomsnittskår på 5,72. Dei med låg motivasjon for algoritmisk tenking og programmering har mykje høgare motivasjon for å lære seg matematikk, enn algoritmisk tenking og programmering, med 4,27 i gjennomsnitt, og 74% (46) svara ein skår på 4 eller meir. Det er likevel tendensar til ein samanheng mellom motivasjon for algoritmisk tenking og motivasjon for matematikk, med ein korrelasjonskoeffisient på 0,375.

Når det gjeld karakterar frå matematikk 1T har dei med låg motivasjon for algoritmisk tenking og programmering ein høgare gjennomsnittskarakter på 4,66 samanlikna med dei med høg motivasjon som har i gjennomsnitt 4,53. Dette snur når ein samanliknar karakterane frå 1.termin i matematikk R1, der dei med låg motivasjon har ein gjennomsnittskarakter på 3,9 og dei med høg motivasjon har 4,3. Det er heller ingen signifikant forskjell mellom gruppe på motivasjon for å lære seg algoritmisk tenking og programmering og matematikk, med p-verdiar på 0,4702 og 0,0622.

5.5.2 Algoritmisk tenking og programmering

Gruppa med høg motivasjon har svara med høgast prosentdel på alle alternativ som vart oppgitt i spørsmål om kva ein meina med algoritmisk tenking. Dette indikerer at dei med høg motivasjon for å lære seg algoritmisk tenking og programmering har eit breiare syn på kva algoritmisk tenking er. Figuren under viser prosentdel elevar frå dei to gruppene som har svart på dei 12 første alternativa.

	Høg motivasjon	Låg motivasjon
Å utforske/eksperimentere med problem	49,2%	34,9%
Designe og lage algoritmer	68,3%	47,6%
Feilsøking og rette feil	49,2%	23,8%
Utholdenhet – Prøve igjen	25,4%	15,9%
Dele og jobbe sammen om et problem	12,7%	4,8%
Analysere problem og forutse algoritme	55,6%	39,7%
Lage algoritmer med steg-for-steg løsning	58,7%	54,0%
Dekomposisjon – Bryte ned problem	54,0%	23,8%
Mønstre – Å finne og bruke likheter	71,4%	33,3%
Abstraksjon – Fjerne unødige detaljar	31,7%	12,7%
Evaluerings – Gjøre vurdering av algoritme	46%	28,6%
Koding – Programmere algoritmer	65,1%	55,6%

Tabell 20: Algoritmisk tenking kryssa med motivasjon for å lære algoritmisk tenking og programmering

Elevane med høg motivasjon er generelt meir positive til alle spørsmål knytt til læring, tidspunkt, prioritering og andre haldningar. Dette viser at elevane med høg motivasjon har fått ei meir utvida forståing av kva algoritmisk tenking som kompetanse omfattar. På spørsmål om ein har fått god opplæring svara dei med høg motivasjon ein gjennomsnittskår på 4,8 mot 3,3 frå dei med låg motivasjon. Om opplæringa innan algoritmisk tenking og programmering har blitt prioritert svara dei med høg motivasjon ein skår på 4,0, medan dei med låg motivasjon svara 3,0.

Forskjellane blir større når elevane sjølve skal vurdere om algoritmisk tenking og programmering passar inn i tema i matematikkfaga, om det er ein vanskeleg kompetanse å tileigne seg og om det er ein nyttig kompetanse. På spørsmål om kor godt algoritmisk tenking og programmering passar inn, svara dei med låg motivasjon ein gjennomsnittskår på 3,0, og dei med høg motivasjon svara heile 5,6. Om det er ein vanskeleg kompetanse å tileigne seg svara dei med låg motivasjon at det er vanskeleg (2,2), medan dei med høg motivasjon svara at det er lettare (5,0). I det neste spørsmålet, om elevane ser nytte av kompetansen, er det enda større forskjell med 3,4 frå dei med låg motivasjon, og 6,5 frå dei med høg motivasjon. Dette er tydeleg indikasjon på at dei med høg motivasjon ser nytten av å lære seg algoritmisk tenking og programmering, i tillegg til at dei er generelt meir positive i svara sine elles.

Elevane med høg motivasjon ser også fleire bruksområde der kompetansen kan nyttast, med ein gjennomsnittskår på 5,2, medan elevane med låg motivasjon ikkje ser fullt så mange område, med ein gjennomsnittskår på 2,1.

5.5.3 Prioritering av og utforsking av algoritmisk tenking og programmering

Elevane med høg motivasjon uttrykker at dei likar å arbeide med algoritmisk tenking og programmering, og har svar ein skår på 5,6 på spørsmål om dei har likt å arbeide med dette. Blant dei med låg motivasjon er gjennomsnittskåren 2,1, som kan tyde på at fleire i denne gruppa ikkje har likt å arbeide med dette. Dei med høg motivasjon gjev også uttrykk for at dei har arbeida meir (4,4) med algoritmisk tenking og programmering utanom skuletida enn dei som har låg motivasjon (2,6).

På spørsmål om dei har utforska programmering meir enn det som står i læreboka og læraren har gått gjennom, er det ein tydeleg indikasjon på at motivasjon bidreg til utforskingslyst. Med ein gjennomsnittskår på 1,7 frå dei med låg motivasjon mot dei med høg motivasjon som svar heile 5,1. Her kan ein legge merke til at standardavviket for dei med høg motivasjon er på 1,57, som tyder på at det er større forskjellar i gruppa, men at dei fleste svara mellom 4 og 6. Pearson sin korrelasjonskoeffisient er rekna ut til å vere 0,641 for alle kandidatane, som vil seie at det er gode sjansar for ein samanheng mellom motivasjon og utforskingslyst (Cresswell, 2012).

5.5.4 Kompetanse i algoritmisk tenking og programmering

Om eigen kompetanse svara dei med høg motivasjon at dei meina dei har høgare kompetanse enn dei med låg motivasjon innanfor alle områder i grunnleggande prinsipp i programmering, noko som også stemmer frå resultatet. Dei med høg motivasjon svara om eigen kompetanse ein gjennomsnittskår frå 6,0 til 6,5, medan dei med låg motivasjon svar frå 3,4 til 4,6. Figuren under oppsummera prosentvis riktig på oppgåvene, som testa grunnleggande prinsipp i programmering og algoritmisk tenking, og seier noko om faktisk kompetanse.

Kompetanse	print()	Variablar	Vilkår	Løkker	Funksjonar	Derivert
Høg motivasjon	84%	97%	92%	51%	77%	75%
Låg motivasjon	64%	75%	69%	36%	33%	36%

Tabell 21: Motivasjon og kompetanse i grunnleggande algoritmisk tenking og programmering

For alle kandidatar er det rekna ut ein positiv korrelasjon (0,395) mellom motivasjon til å lære seg algoritmisk tenking og programmering og kompetansen ein tileignar seg. Dette er noko som også rapportar frå TIMSS, TIMSS Advanced og PISA har konkludert med (Grønmo et al., 2016, s. 87).

5.5.5 Oppsummering av forskjellar mellom dei med høg/låg motivasjon for å lære seg algoritmisk tenking og programmering

	Høg motivasjon (gj.snitt)	Låg motivasjon (gj.snitt)	SD (Høg)	SD (Låg)	P-verdi
Motivasjon for å lære matematikk	5,72	4,27	0,96	1,6	0,0001
Karakter 1T	4,53	4,66	1,15	0,83	0,4702
Karakter R1 (1.termin)	4,27	3,92	1,14	3,92	0,0622
Algoritmisk tenking og programmering passar i matematikkfaget	5,56	2,97	1,19	1,25	0,0001
Bruksområde for algoritmisk tenking og programmering	5,24	2,14	1,29	1,33	0,0001
Utforska på eigenhand	5,13	1,68	1,57	1,24	0,0001

Tabell 22: Oppsummering av forskjellar mellom dei høg/låg motivasjon for å lære algoritmisk tenking og programmering

5.6 Andre samanhengar

I denne delen tar eg føre meg andre samanhengar som kan vere greitt å ta med, og som ikkje har blitt nemnt i tidlegare delkapittel. Her er det nytta Person sin korrelasjonskoeffisient saman med Cresswell (2012) sin tolking av koeffisienten.

5.6.1 Karakter i R1

Det ser ut som at karakter i R1 (1.termin) har liten/ingen korrelasjon med haldningar og erfaringar med algoritmisk tenking og programmering. Det er likevel ein mogleg samanheng mellom karakteren i R1 og faktisk kompetanse i algoritmisk tenking, der eg har fått ein korrelasjonskoeffisient på 0,290. Pacheco et al. (2008) kom i sin studie fram til liknande resultat då dei fant ein korrelasjonskoeffisient på 0,373 mellom matematikk-karakter ved inntak til høgare studie og kompetanse tileigna i programmering i løpet av eit introduksjonskurs

førsteåret. Dette støtter opp under påstanden om at dei med god matematisk kompetanse vil tileigne seg programmeringskompetanse raskare.

Eg har kome fram til ein korrelasjonskoeffisient på 0,119 mellom motivasjon for å lære seg algoritmisk tenking og programmering og karakter frå R1, som kan tyde på at kompetanse i matematikk ikkje spelar inn på motivasjonen for å lære seg algoritmisk tenking og programmering.

Korrelasjonskoeffisienten mellom karakter i R1 og utforsking på eigenhand er 0,002, og er dermed tydeleg uavhengig i følgje Creswell (2012). Vidare er det også liten/ingen samanhengen mellom karakteren i matematikk R1, og om elevane ser på algoritmisk tenking og programmering som nyttig, med korrelasjonskoeffisient på 0,044.

5.6.2 Nyttig kompetanse

Det er ein positiv korrelasjon (0,360) mellom eleven sitt syn på nytten av kompetansen og fagleg prestasjon i oppgåvene som vart gitt i siste delen av spørjeskjemaet. Dette samsvarar med tidlegare funn frå blant anna TIMSS Advanced 2015 (Grønmo et al., 2016, s. 97)

Det ser også ut som at det ein samanheng mellom det å sjå på kompetansen som nyttig og det å bruke tid til å utforske kompetansen på eigenhand (korrelasjonskoeffisient på 0,539).

6 Diskusjon

I dette kapitlet vil eg drøfte problemstillinga ved hjelp av empiri, teori og tidlegare forskning. Eg vil oppsummera korleis eg har svara på mine forskings spørsmål, tolke funna mine, og diskutere kva mine funn kan ha å seie for praksis og vidare forskning.

6.1 Meininger og haldningar

I denne masteroppgåva har eg svara på «**Kva meininger og haldningar har elevane knytt til algoritmisk tenking og programmering i matematikkfaga?**» (F1).

Elevane i matematikk R1 har variert forståing av kva kompetansar og arbeidsmetodar som ligg under omgrepet algoritmisk tenking. Mange av elevane svarar fleire av kompetansane som både Weintrop (2006) og Brennan og Resnick (2012) har med i sine definisjonar av algoritmisk tenking. Det er derimot få som svarar andre kompetansar og arbeidsmetodar som Barefoot Computing (2022) og Brennan og Resnick (2012) seier er ein del av algoritmisk tenking. For eksempel er det berre 20% av elevane som svarar at uthalden og abstraksjon er kompetansar som inngår i kompetansen algoritmisk tenking. Så er det kanskje litt overraskande at «berre» 62% svarar at «Koding – programmering av algoritmar». Ein kan på bakgrunn av dette diskutere om elevane har tileigna seg nok kompetanse om sjølve omgrepet algoritmisk tenking og kva kompetansar og arbeidsmetodar som ligg under denne kompetansen.

Elevane som har svara at dei har fått karakter 5 eller 6 svarar i stor grad at fleire kompetansar inngår som ein del algoritmisk tenking, enn dei med karakter 1, 2 eller 3. Litt overraskande svarar dei same gruppene nokså likt på spørsmål om eiga utforsking, tidsbruk, nytte og om kompetansen passar inn i matematikkfaget. Det ser ut som at matematisk kompetanse ikkje påverkar elevane sin prioritering av algoritmisk tenking og programmering, og heller ikkje det å sjå på kompetansen som nyttig eller at den passar inn i matematikkfaget.

Høg motivasjon for å lære seg algoritmisk tenking og programmering er tett knytt saman med meir positive haldningar og meininger om å lære seg denne kompetansen. Elevane med høg motivasjon har også, på same vis som dei høgt presterande, ei breiare forståing av kva kompetansar og arbeidsmetodar som inngår i algoritmisk tenking. Elevane med høg motivasjon ser god nytte av kompetansen og svarar i gjennomsnitt 6,5 på dette spørsmålet. Motivasjon og læring (faktisk tileigna kompetanse) har positiv korrelasjon, og sjølv om det ikkje nødvendigvis er direkte korrelasjon, så er det gode indikasjonar på at høg motivasjon for algoritmisk tenking fører til meir læring.

Elevane meiner at algoritmisk tenking og programmering er litt lite prioritert (3,4 av 7 gjennomsnitt), og uttrykker at dei synes det er ein vanskeleg kompetanse å tileigne seg. Elevane meiner at det er gitt god opplæring (4,1 av 7 i gjennomsnitt) når det er gitt undervisning i kompetansen, som er ein føresetnad for læring.

Medan motivasjonen for å lære seg matematikk er omtrent på 5 av 7, så er motivasjonen for å lære seg algoritmisk tenking og programmering på omtrent 4 av 7. Dette tyder på at motivasjonen for å lære seg algoritmisk tenking og programmering er lågare enn for å lære seg matematikk blant elevar i matematikk R1. Dette viser også studien til Andersen (2022), som skriv at «Elevane i denne studia likar den tradisjonelle matematikken betre enn programmering.» (Andersen, 2022, s 34). Mennene er ein god del meir motiverte enn kvinnene, og det skil berre 0,26 i forskjell mellom motivasjon for matematikk og algoritmisk tenking for mennene. Grunnen til at elevane har lågare motivasjon for algoritmisk tenking og programmering kan vere at lærarane sjølve har liten erfaring med kompetansen (Frantsen, 2019), kompetansen er lite prioritert, at elevane får dårlege eksempel på bruk av kompetansen og at elevane har for liten erfaring med kompetansen endå (Andersen, 2022).

Dei fleste melder om at dei synes algoritmisk tenking og programmering passar inn i matematikkfaget med 4,32 i gjennomsnittskår. At elevane ser at kompetansen passar inn, kan gjere at dei kan integrere dette inn i matematikkfaget som ein problemløysingstrategi. Elevane til Andersen (2022) svara på tilsvarende spørsmål at «det gjev meining å bruke programmering til det meste i matematikkfaget» (Andersen, 2022, s.37). Det å sjå nytteverdien av eit fag eller ein kompetanse er viktig for det faglege utbytte (Grønmo et al., 2016). I mi undersøking meina 67% av elevane at kompetansen er nyttig, som er eit godt grunnlag for eit godt fagleg utbytte. Det kan sjølv sagt bli betre, men det er eit godt utgangspunkt for å få elevane engasjerte i å lære algoritmisk tenking og programmering.

6.2 Erfaringar

Eg har også svara på «**Kva erfaringar har elevane gjort seg med algoritmisk tenking og programmering i matematikkfaga?**» (F2).

Det er tydeleg at det er programmeringsspråket Python, som er det føretrekte i matematikk på vidaregåande skule. I undervisninga melder dei fleste elevane at dei har oppfatta algoritmisk tenking og programmering som noko kjekt. Sjølv om dei fleste har gitt tilbakemelding om at kompetansen er kjekk til tider, så er det omtrent ein tredjedel som svarar at dei aldri har utforska

algoritmisk tenking og programmering på eigenhand. Elevane melder også om at dei brukar mindre tid på denne kompetansen samanlikne med andre fagkompetansar.

Elevane gjev uttrykk for at dei har fått litt lita tid med å lære seg kompetansen i klasserommet, noko som også Andersen (2022, kap.4.2.2) har fått fram i sine intervju. Når lærar har jobba med algoritmisk tenking og programmering, er det fire arbeidsmetodar som er nemnt av omtrent to tredeler eller meir. Den fyrste, med 81%, er «Lese kode og tolke kva programmet gjer». Den andre, med 72%, er ei meir utvida form av den førre, og er «Les, kopier og tilpass programeksempel». Elevane melder også om at «kopiering av kode til lærer» er ein vanleg arbeidsmetode. Dei melder også om at «Følg oppskrift på å lage program» er ein vanleg arbeidsmetode, med 64%. Det er fleire elevar (26%) som svarar at dette er «Overlate til elevane».

Dei aller fleste, 86% av elevane, kjenner til minst eit bruksområde for algoritmisk tenking og programmering. Her kan det vere spennande å følgje opp med ein studie som ser på kva elevane svara av bruksområde, og om det er nokon bruksområde som går igjen. Kanskje elevane svara generelt, som for eksempel «i matematikk» eller «som ein problemløysingsstrategi», eller har dei sett at det kan brukast til konkrete oppgåver som «numeriske metodar», «databehandling», «løyse komplekse matematiske problem», «finne tilnærmingar», «kunstig intelligens» og så vidare.

6.3 Kompetanse

Til slutt har eg svara på «**Kva kompetanse har elevane tileigna seg i forhold til algoritmisk tenking og programmering?**» (F3).

Elevane sjølve meiner i stor grad at dei er trygg i bruk av dei grunnleggande prinsippa til programmering, og har svara om variablar (5,3), valsetningar (4,7), løkker (5,3) og funksjonar (5,2). Det er likevel god spreining i svarfordeling på alle kompetanse, men modus er 7 for alle dei fire grunnleggande prinsippa.

Elevane viser god kompetanse i grunnleggande prinsipp innan programmering, og viser god forståing (mellom 71 og 85 % riktig) for print-kommando, variablar og valsetningar. Det er derimot dårlegare resultat på løkker (44%) og funksjonar (50%), som kan tyde på at desse har vore nytta i mindre grad og/eller er meir komplekse å lære seg. Mennene viser meir kompetanse enn kvinnene i alle kompetansane som er testa, men det er ikkje store forskjellar. Det er over halvparten av elevane (52%), som viser kompetanse innan algoritmisk tenking og programmering, når dei skal lese og tolke eit meir avansert eksempel. Det er tydeleg at fleirtalet

av elevane har jobba med liknande oppgåver i undervisninga, og dermed er kome godt i gang med å lære seg algoritmisk tenking og programmering.

For dei elevane som valte programmering som valfag i ungdomsskulen, har det gitt utslag på kompetansen som er tileigna etter matematikk R1. Dei skårar mellom 7 og 24 prosentpoeng betre på alle kompetansar. I tillegg er desse elevane meir motiverte for å lære seg algoritmisk tenking og programmering, enn dei som ikkje har erfaring frå ungdomsskulen. Dei er også generelt meir positive til kompetansen i forhold til å sette av tid, utforske på eigenhand, sjå bruksområde og sjå nytten av den.

6.4 Kva med lærarane?

Lærarane melder om høg motivasjon (5,38) for å lære elevane algoritmisk tenking og programmering. Kompetansen «designer og lager algoritmar som løyser eit problem» er i følgje dei aller fleste av lærarane ein viktig del av algoritmisk tenking. Lærarane svarar elles mykje likt som elevane på kva algoritmisk tenking består av, og blant anna er «uthalden» utelete frå alle lærarane.

Dei fleste er positive til algoritmisk tenking og programmering, og melder om at det er ein nyttig kompetanse, relevant til matematikkfaget og at dei ser fleire bruksområde for kompetansen. Igjen er lærarane og elevane nokså einige, når alle lærarane svarar at «lese kode og tolke kva programmet gjer» blir brukt som arbeidsmetode. Elles nemner lærarane lesing av kode, kopiering av kode og tilpassing av kode som vanlege arbeidsmetodar i arbeid med algoritmisk tenking og programmering.

I spørsmål om eigen kompetanse svarar lærarane at dei er trygg i bruk, med over 6.38 i gjennomsnitt, for alle kompetansane. Resultatet av kompetansetesten viser at 100% av lærarane svarar riktig på alle oppgåvene, med unntak av oppgåva som testar løkker, der ein av fem svarar feil. Kan dette tyde på at lærarane ikkje har så dårleg kompetanse som fleire studie (Finstad, 2020; Frantsen 2019) påpeikar som ein utfordring i innføringa av algoritmisk tenking og programmering?

7 Avslutning

7.1 Status for innføring av algoritmisk tenking og programmering

Hovudmålet med masteroppgåva var å finne ut av «**Status for innføring av algoritmisk tenking med programmering som verktøy i matematikk i den vidaregåande skulen med fokus på realfagsmatematikkfaga**» (P1).

Etter to år med algoritmisk tenking og programmering i matematikk 1T og R1, har elevane tileigna seg kompetanse om grunnleggande prinsipp innan programmering og gjort seg erfaringar med kva algoritmisk tenking er og kva ein kan bruke denne kompetansen til. Elevane har fått gjort seg opp meiningar og haldningar til kompetansen, og dei fleste ser at dette kan vere ein nyttig kompetanse å ta med seg inn i matematikkfaga.

Her kjem ei kort oppsummering av funn mellom grupperingar blant alle kandidatane:

- Mennene er meir motiverte til å lære algoritmisk tenking og programmering, og har høgare kompetanse enn kvinnene i grunnleggande prinsipp i programmering og algoritmisk tenking.
- Mennene er meir interesserte i algoritmisk tenking og programmering, og svara generelt meir positivt på tidsbruk, eiga utforsking og meir knytt til kompetansen.
- Det er små forskjellar i haldningar og meiningar knytt til algoritmisk tenking og programmering mellom dei høgt presterande i matematikkfaga og dei lågt presterande.
- Kompetansenivået til ein elev i matematikk har ein innverknad på tileigning av kompetanse i algoritmisk tenking og programmering.
- Det er ein tydeleg fordel å ha tidlegare erfaring frå kompetansen når ein startar med opplæringa i algoritmisk tenking og programmering i matematikkfaga.
- Elevane med erfaringa frå programmering i ungdomsskulen er meir positive til algoritmisk tenking, og ser fleire bruksområde og nytten av å lære seg kompetansen.
- Dei med høg motivasjon for lære seg algoritmisk tenking og programmering vel å bruke meir tid på å lære seg kompetansen, og er generelt meir positive til spørsmål som går på haldningar og erfaringar med algoritmisk tenking og programmering.
- Elevar som ser på algoritmisk tenking og programmering som nyttig, vel å bruke meir tid på å utforske og lære seg dette, som også viser seg i fagleg kompetanse i siste del av spørjeskjemaet.

7.2 Veggen vidare

Det hadde vore interessant å undersøkt nærmare kva bruksområde elevane har oppdaga for algoritmisk tenking og programmering. Det kunne for eksempel vore spennande med ei kartlegging av bruksområde som elevane meiner er relevante gjennom ein kvalitativ studie med intervju av elevane som svarar at dei ser fleire bruksområde. Motivasjonen og målet for å gjennomføre ein studie på dette, kan vere å opplyse elevar og lærarar om kva bruksområde som kompetansen kan nyttast innan.

I analysedelen kjem det fram forskjellar mellom kjønna, og spesielt er motivasjonen for å lære seg algoritmisk tenking og programmering sprikande. Kva som er årsaka til dette, kan vere nyttig kunnskap for både elevar, lærarar og andre i vidare utvikling av opplæringa i kompetansen.

Det kunne vore spennande å sjå meir på algoritmisk tenking utan PC, altså opplæring i algoritmisk tenking utan bruk av PC. Kanskje kunne ein fått til ein studie som såg på læringseffekten av to eller tre undervisningsopplegg innan algoritmisk tenking, for å undersøkje forskjellar i kva kompetanse elevane sit igjen med etter opplegga.

I forhold til eksamen, kunne det ha vore interessant å følgje eksamensresultata på oppgåvene som er meint å løyse ved algoritmisk tenking og programmering, og korleis resultata utviklar seg dei neste åra. Eventuelt kunne ein analysert kva kvar av oppgåvene krev av elevane, eller kva som er føremålet med oppgåvene. Her kunne ein også ha snakka med dei som faktisk lagar oppgåvene, som kunne gjeve lærarane og andre innsyn i korleis dei tenkjer når dei lagar eit eksamenssett.

Som eg har nemnt tidlegare, så hadde vore veldig interessant å følgd opp denne studien med ei ny runde med spørjeundersøkinga om nokon år. Det er interessant å sjå endring over tid på denne kompetansen då den fortsett er veldig fersk i skulen, og at elevane i den vidaregåande skulen etter kvart vil få meir erfaring med algoritmisk tenking og programmering før dei tek til på den vidaregåande skulen.

Noko anna som har vore mykje diskutert, er om algoritmisk tenking og programmering burde vere eit eige fag. Finnes det internasjonale studie som allereie har sett på dette, eller kan ein bruke noko liknande som eg har nytta meg i denne studien, til å samanlikne elevar i den vidaregåande skulen som har hatt algoritmisk tenking og programmering som eige fag med dei som har integrert det i matematikkfaga.

8 Litteraturliste

- Aho, A. V. (2012). Computation and Computational thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Aminah, N., Sukestiyarno, Y. L, Wardono, W. & Cahyono, A. N. (2022). Computational thinking process of prospective mathematics teacher in solving diophantine linear equation problems. *European Journal of Education Research*. 11(3). 1495-1507. <https://doi.org/10.12973/eu-jer.11.3.1495>
- Andersen, O. (2022). *Programmering si rolle i matematikkfaget: Frå eit elevperspektiv* [Masteroppgåve]. Universitetet i Tromsø.
- Balanskat, A. & Engelhardt, K. (2015) Computing our future: Computer programming and coding – Priorities, school curricula and initiatives across Europe. *European Schoolnet*.
- Barfoot Computing (2022). *The Computational Thinkers* [Poster] <https://www.barefootcomputing.org/resources/computational-thinking-poster>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association (pp. 1–25).
- Cresswell, J.W. (2012). *Educational research: planning, conducting, and evaluating quantitative and qualitative research* (4th edition ed.). Pearson Education.
- Finstad, M. (2020). *Algoritmisk tenking og programmering i matematikk: Ein kvalitativ studie av ni matematikklærarar sitt syn på integreringa av algoritmisk tenking og programmering i matematikkfaget* [Masteroppgåve]. Universitetet i Bergen.
- Frantsen, T. (2019). *Å vere lærar i programmering utan å kunne programmere*. [Masteroppgåve]. OsloMet - Storbyuniversitetet
- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. In Springer Berlin Heidelberg (pp. 159-168). https://doi.org/10.1007/11915355_15
- Grover, S. & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>

- Grønmo, L. S., Hole, A., & Onstad, T. (2016). *Ett skritt fram og ett tilbake*. Cappelen Damm Akademisk/NOASP. ISSN 978-82-02-55029-5
- Hannula, M.S. *et al.* (2016). Attitudes, Beliefs, Motivation, and Identity in Mathematics Education. In: Attitudes, Beliefs, Motivation and Identity in Mathematics Education. ICME-13 Topical Surveys. Springer, Cham.
https://doi.org/10.1007/978-3-319-32811-9_1
- Healey, J. F. (2015). *Statistics: A tool for social research*. (10. utg.). Cengage Learning.
- Hirth, S. (2020). *Programmering som inngangsport til algoritmisk tenking: Ein mixed methods studie av IP-elevars erfaringar, forståing og syn knytt til programmering i matematikkundervisninga* [Masteroppgåve]. Universitetet i Bergen.
- Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
<https://doi.org/https://doi.org/10.1016/j.compedu.2018.07.004>
- Khine, M. S. (2018). Computational Thinking in the STEM Disciplines : Foundations and Research Highlights (1st ed. 2018., p. 1 online resource (XVII, 325 p. 122 illu)). Springer International Publishing : Imprint: Springer.
- Kleven, T. A. (2008). *Validity and validation in qualitative and quantitative research*. Nordisk Pedagogik, Vol. 28, pp. 219–233. Oslo. ISSN 0901-8050.
- Kunnskapsdepartementet. (2018). *Kjernelementer i fag*.
<https://www.regjeringen.no/contentassets/3d659278ae55449f9d8373fff5de4f65/kjerneelementer-i-fag-for-utforming-av-lareplaner-for-fag-i-lk20-og-lk20s-fastsatt-av-kd.pdf>
- Iarossi, G. (2006). *Power of Survey Design : A User's Guide for Managing Surveys, Interpreting Results, and Influencing Respondents*. World Bank Publications.
<http://ebookcentral.proquest.com/lib/bergen-ebooks/detail.action?docID=459526>
- Lund, T. & Christophersen, K.A. (1999). *Innføring i statistikk*. Universitetsforlaget AS.
- Lund, T. (2012). *Combining Qualitative and Quantitative Approaches: Some Arguments for Mixed Methods Research*. *Scandinavian Journal of Educational Research* 56(2): 155-165

- Lynnebakken, H. (2018, 11.okt). Slik kan programmering endre skolen. *Titan.uio.no*.
Forskningsnyheter om realfag og teknologi.
<https://www.titan.uio.no/naturvitenskap/2018/slik-kan-programmering-endre-skolen.html>
- Meld. St. 28 (2015-2016). *Fag – fordypning – Forståelse – En fornyelse av Kunnskapsløftet*.
Kunnskapsdepartementet.
<https://www.regjeringen.no/no/dokumenter/meld.-st.-28-20152016/id2483955/>
- NOU 2014: 7 (2014). *Elevenes læring i fremtidens skole – Et kunnskapsgrunnlag*.
Utredning fra et utvalg oppnevnt ved kongelig resolusjon 21.juni 2013.
<https://www.regjeringen.no/no/dokumenter/NOU-2014-7/id766593/>
- NOU 2015: 8. (2015). *Fremtidens skole – Fornyelse av fag og kompetanser*.
Utredning fra et utvalg oppnevnt ved kongelig resolusjon 21.juni 2013.
<https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/>
- Pacheco, A., Gomes, A., Henriques, J., de Almeida, A. M., & Mendes, A. J. (2008, 12.juni).
Mathematics and programming: Some studies. CompSysTech '08. In Proceedings of
the 9th International Conference on Computer Systems and Technologies and
Workshop for PhD Students in Computing (p. V-15).
<https://doi.org/10.1145/1500879.1500963>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*.
New York: Basic Books.
- Pring, R. (2015). *Philosophy of Educational Research* (3rd edition).
London: Bloomsbury Publishing UK
- Regjeringa Solberg. (2018, 26.juni). *Fornyelse innholdet i skolen*. Pressemelding. Nr: 132-18.
<https://www.regjeringen.no/no/dokumentarkiv/regjeringen-solberg/aktuelt-regjeringen-solberg/kd/pressemeldinger/2018/fornyelse-innholdet-i-skolen/id2606028/?expand=factbox2606064>
- Shadish, W.R., Cook, T.D. & Campbell, D.T. (2002). *Experimental and Quasi-experimental Designs for Generalized Causal Inference*. Boston, MA: Houghton Mifflin Company.

- Shute, V. J., Sun, C. & Asbell-Clark, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22(2017), 142-158.
<https://doi.org/10.1016/j.edurev.2017.09.003>
- Tangen, R. (2014). Balancing Ethics and Quality in Educational Research—the Ethical Matrix Method. *Scandinavian Journal of Educational Research*, 58(6), 678-694.
<https://doi.org/10.1080/00313831.2013.821089>
- Utdanningsdirektoratet. (2016, 1.august). *Forsøkslæreplan i valgfag programmering (PRG1-01)*. Gjeld berre for skular som han fått innvilga forsøk med programmering som valfag frå 1.8.2016. <https://data.udir.no/kl06/prg1-01.pdf>
- Utdanningsdirektoratet. (2019a, 27.mars). *Algoritmisk tenking*.
<https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2019b, 15.november). *Læreplan i matematikk fellesfag vg1 teoretisk (MAT09-01)*. Fastsett som forskrift av Kunnskapsdepartementet. Læreplanverket for Kunnskapsløftet 2020. <https://data.udir.no/kl06/v201906/laereplaner-1k20/MAT09-01.pdf?lang=nno>
- Utdanningsdirektoratet. (2020a, 26.mai). *Læreplan i matematikk for realfag (MAT03-02)*. Fastsett som forskrift av Utdanningsdirektoratet 26.mai 2020 etter delegasjon i brev av 13.september 2013 fra Kunnskapsdepartementet med hjemmel i lov 17.juli 1998 nr,61 om grunnskolen og den vidaregåande opplæringa (opplæringslova) § 3-4 første ledd.
<https://data.udir.no/kl06/v201906/laereplaner-1k20/MAT03-02.pdf?lang=nob>
- Utdanningsdirektoratet. (2020b, 12.juni). *Læreplan i valgfaget programmering (PRG01-02)*. Fastsett som forskrift av Kunnskapsdepartementet 12.juni 2020.
<https://data.udir.no/kl06/v201906/laereplaner-1k20/PRG01-02.pdf?lang=nob>
- Utdanningsdirektoratet. (2021, 20.september). *Slik ble læreplanene utviklet*.
<https://www.udir.no/laring-og-trivsel/laereplanverket/fagfornyelsen/slik-ble-lareplanene-utviklet/>
- Utdanningsdirektoratet. (u.å.). *Fagvalg i videregående skole – elever*. Hentet 27.januar 2023 frå <https://www.udir.no/tall-og-forskning/statistikk/statistikk-videregaende-skole/fagvalg-i-videregaende-skole/fagvalg-vgs/>

- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
<https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49 (3), 33-35.
<https://dx.doi.org/10.1145/1118178.1118215>
- Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: a systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10(1).
<https://doi.org/10.1186/s40594-023-00396-w>

9 Vedlegg

9.1 Elevundersøkinga

Om undersøkelsen

Takk for at du tar deg tid til å svare på spørreundersøkelsen.

Undersøkelsen vil ta omtrent 8-10 minutt, og resultatene vil bli publisert som en del av min masteroppgave sommeren 2023.

Undersøkelsen er delt i fem deler.

Del 1: Samler inn **informasjon om deg** som informant.

Del 2: Spørsmål om dine **meninger** om algoritmisk tenking og programmering.

Del 3: Spørsmål om dine **holdninger** til algoritmisk tenking og programmering.

Del 4: Spørsmål om dine **erfaringer** med algoritmisk tenking og programmering.

Del 5: Spørsmål som skal **teste tilegnet kompetanse** innen algoritmisk tenking og programmering.

Minner om at ingen av opplysningene kan knyttes mot deg som informant. Jeg garanterer for din anonymitet.

Takk igjen for at du deltar.

Beste hilsen, Emil Thune, Student ved Universitetet i Bergen og lærer ved Olsvikåsen vgs.

Om informanten

I denne delen ber jeg deg om å fylle inn korrekte opplysninger om deg selv.

Minner igjen om at dataene ikke kan knyttes mot deg som person.

Velg kjønn

Mann

Kvinne

Annet

I hvilket fylke ligger skolen du går på?

Agder

Innlandet

Møre og Romsdal

Nordland

Oslo

Rogaland

Troms og Finnmark

Trøndelag

Vestfold og Telemark

Vestland

Viken

Hvilken karakter fikk du i standpunkt i 1T?

IV 1 2 3 4 5 6

Hvilken karakter fikk du i 1.termin i R1?

IV 1 2 3 4 5 6

Har du (eller har du hatt) programmering som valgfag, IT1 eller IT2?

- Programmering som valgfag fra ungdomsskolen
- IT1
- IT2
- Ingen tidligere erfaring fra fag

Meninger

I denne delen vil du få spørsmål om dine meninger til algoritmisk tenking og programmering i matematikk-fagene. Minner igjen om at ingen av dine svar kan knyttes mot deg som person.

Hvor motivert er du til å lære programmering og algoritmisk tenking?


1 2 3 4 5 6 7

Hva tror du menes med "algoritmisk tenking"? (velg så mange alternativer du vil)

- Å tenke som en datamaskin
- Abstraksjon - Fjerne unødvendige detaljer
- Å utforske og eksperimentere med et problem
- Designe og lage algoritmer som løser et problem
- Lage algoritmer med regler og steg-for-steg løsning
- Evaluering - Gjøre vurderinger av algoritme og løsning
- Automatisere problemløsning
- Koding - Programmere algoritmer
- Behandling av data på en systematisk måte
- Mønstre - Å finne og bruke likheter
- Dele og jobbe sammen om problem
- Utholdenhet - Fortsett og prøve igjen
- Analysere problem og forutse om en algoritme løser problem
- Oppdage og rette feil i forsøk på å løse et problem
- Dekomposisjon - Bryte problem ned i mindre biter
- Løse problemstillinger med algoritme
- Løse problemer ved å lage et program som løser problemet for oss

Har du fått god opplæring i programmering i 1T og R1? 1 er svært dårlig og 7 er svært god



Har opplæring i programmering og algoritmask tenking blitt prioritert i matematikken? 1 er svært lite og 7 er veldig mye



Er programmering og algoritmask tenking en vanskelig kompetanse å tilegne seg? 1 er svært vanskelig og 7 er svært enkelt



Hvor godt passer algoritmask tenking og programmering inn med temaene i matematikkfagene? 1 er svært dårlig og 7 er svært godt



Bør programmering og algoritmask tenking være et eget fag i vgs?
1 er helt uenig, og 7 er helt enig



I de neste spørsmålene er 1 "Kjenner ikke til", og 7 er "trygg på bruk"

Hvor godt kjenner du til hva en variabel (i programmering) er?



Hvor godt kjenner du til hva en if-test (i programmering) er?



Hvor godt kjenner du til hva en løkke (i programmering) er?



Hvor godt kjenner du til hva en funksjon (i programmering) er?



Holdninger

I denne delen ønsker jeg dine ærlige holdninger om matematikk-faget, algoritmisk tenking og programmering som en del av matematikk-faget. Minner igjen om at ingen av dine svar kan knyttes mot deg som person.

I de neste spørsmålene er 1 definert som "Svært lite" og 7 som "Svært mye"

Hvor motivert er du til å lære matematikk?



Har du i løpet av 1T og R1 likt å jobbe med "algoritmisk tenking"?



Programmering og algoritmisk tenking er en nyttig kompetanse



Erfaringer

I den neste delen kan du ta utgangspunkt i at "algoritmisk tenking" er definert som "...grunnlaget som kreves for å løse problemer effektivt (dvs. algoritmisk, med eller uten hjelp fra datamaskiner) med løsninger som kan gjenbrukes i forskjellige sammenhenger."

Hvilke programmeringsspråk kjenner du til?

- R
- C#
- Andre
- Python
- JavaScript
- C++
- Java
- Ingen

Var du forberedt på at du skulle lære deg algoritmisk tenking og programmering i 1T og R1?

- Nei
- Ja
- Litt
- Vet ikke

Har du brukt mer/mindre tid på algoritmisk tenking og programmering enn andre fagkompetanser utenom skoletiden? 1 er mye mindre og 7 er mye mer



Har du utforsket programmering mer enn det som står i læreboken og det læreren har gått gjennom? 1 er ingenting og 7 er veldig mye



Har du erfart noen bruksområder for programmering og algoritrisk tenking? 1 er ingen områder og 7 er veldig mange områder



Hvordan jobbes det med algoritrisk tenking og programmering i klasserommet?

- Overlatt til elevene
- Les, kopier og tilpass programeksempel
- Utvid utdelt program
- Kopiere kode til lærer
- Følg oppskrift på å lage program
- Algoritrisk tenking uten PC
- Lese kode og tolke hva programmet gjør
- Tutorials (videoer)
- Har ikke jobbet med det

Har du fått nok tid til å jobbe med algoritrisk tenking og programmering? 1 er lite/ingen tid og 7 er god tid



Har du tilegnet deg forståelse av hva algoritrisk tenking er? 1 er ingen og 7 er veldig god



Programmeringskompetanse (med algoritrisk tenking)

I disse spørsmålene vil du bli spurt om å forstå programmene, og svare på spørsmål relatert til programmene.

Hvilke verdier har variablene a, b og c etter programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7 b = a + 2
8 a = a + 11
9 c = b ** 2
```

- a har verdi 5.5, b har verdi 9 og c har verdi 16
- a har verdi 5.5, b har verdi 9 og c har verdi 81
- a har verdi 7, b har verdi 9 og c har verdi 16
- a har verdi 18, b har verdi 9 og c har verdi 16
- a har verdi 18, b har verdi 9 og c har verdi 81
- a har verdi 18, b har verdi a2 og c har verdi 4
- Jeg har ikke peiling

Hva gjør kommandoen print i programmet?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7 b = a + 2
8 a = a + 11
9 c = b ** 2
10
11 print(a + b + c)
```

- Print vil skrive ut verdiene til a, b og c
- Print sender summen til utskrift på lokal printer
- Print blir brukt til å utføre matematiske beregninger. Her vil den summere verdiene til a, b og c
- Print vil lage en ny variabel med verdiene til det som er gitt i parentesene
- Print vil la brukeren skrive inn svar på matematisk utregning
- Print skriver ut summen av verdiene til a, b og c

Hvilken verdi vil a ha etter programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7
8 if a < 3:
9     a = 0
10 elif a < 7:
11     a = 1
12 elif a > 5:
13     a = 2
14 else:
15     a = 9
```

- 2
- 0
- 9
- 7
- 1

Hva er verdien til a når programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 1
7 for i in range(3):
8     a += i
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Hva skjer i linje 9? (fun(1,2,3))

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def fun(a,b,c):
7     return a * b * c
8
9 fun(1,2,3)
```

- Et funksjonskall som vil returnere produktet av 1, 2 og 3, som blir 6.
- Et funksjonskall som vil returnere variablene a, b og c
- Et funksjonskall som sjekker om tallene kan multipliseres
- Et funksjonskall som vil returnere 123
- Et funksjonskall som vil returnere 1*2*3 som tekst
- Kommandoen fun vil gi deg et tilfeldig tall av de tallene du oppgir
- Her vil programmet gi en feilmelding

Hva skal funksjonen k gjøre? Skriv kort.

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def f(x):
7     return 2*x**2 + x - 5
8
9 def k(f, a, delta_x):
10     return (f(a + delta_x) - f(a - delta_x)) / (2*delta_x)
11
12 k(f, 1, 1E-8)
```

- Finne en tilnærming til nærmeste nullpunkt
- Finne en tilnærmingsverdi for deriverte til funksjonen f
- Finne integralet til en funksjon
- Finne en tilnærmingsverdi til neste verdi i funksjonen f
- Finne toppunktet til f
- Har ikke peiling

9.2 Lærerundersøkinga

Om undersøkelsen

Takk for at du tar deg tid til å svare på spørreundersøkelsen.

Undersøkelsen vil ta omtrent 5-6 minutt, og resultatene vil bli publisert som en del av min masteroppgave sommeren 2023.

Undersøkelsen er delt i fem deler:

Del 1: Samler inn **informasjon om deg** som informant.

Del 2: Spørsmål om dine **meninger** om algoritmisk tenking og programmering.

Del 3: Spørsmål om dine **holdninger** til algoritmisk tenking og programmering.

Del 4: Spørsmål om dine **erfaringer** med algoritmisk tenking og programmering.

Del 5: Spørsmål som skal **teste tilegnet kompetanse** innen algoritmisk tenking og programmering.

Minner om at ingen av opplysningene kan knyttes mot deg som informant. Jeg garanterer for din anonymitet.

Takk igjen for at du deltar.

Beste hilsen, Emil Thune, Student ved Universitetet i Bergen og lærer ved Olsvikåsen vgs.

Om informanten

I denne delen ber jeg deg om å fylle inn korrekte opplysninger om deg selv.

Minner igjen om at dataene ikke kan knyttes mot deg som person.

Velg kjønn

- Mann
- Kvinne
- Annet

I hvilket fylke ligger skolen du går på?

- Agder
- Innlandet
- Møre og Romsdal
- Nordland
- Oslo
- Rogaland
- Troms og Finnmark
- Trøndelag
- Vestfold og Telemark
- Vestland
- Viken

Meninger

I denne delen vil du få spørsmål om dine meninger til algoritmisk tenking og programmering i matematikk-fagene.

Minner igjen om at ingen av dine svar kan knyttes mot deg som person.

Hvor motivert er du til å lære elevene i programmering og algoritmisk tenking?



Hva tror du menes med "algoritmisk tenking"? (velg så mange alternativer du vil)

- Designe og lage algoritmer som løser et problem
- Lage algoritmer med regler og steg-for-steg løsning
- Evaluering - Gjøre vurderinger av algoritme og løsning
- Å tenke som en datamaskin
- Dele og jobbe sammen om problem
- Dekomposisjon - Bryte problem ned i mindre biter
- Behandling av data på en systematisk måte
- Automatisere problemløsning
- Å utforske og eksperimentere med et problem
- Koding - Programmere algoritmer
- Løse problemer ved å lage et program som løser problemet for oss

- Mønstre - Å finne og bruke likheter
- Oppdage og rette feil i forsøk på å løse et problem
- Analysere problem og forutse om en algoritme løser problem
- Abstraksjon - Fjerne unødvendige detaljer
- Utholdenhet - Fortsett og prøv igjen
- Løse problemstillinger med algoritme

Har du prioritert opplæring i programmering og algoritmisk tenking i matematikken? 1 er svært lite og 7 er veldig mye



Hvor godt passer algoritmisk tenking og programmering inn med temaene i matematikkfagene? 1 er svært dårlig og 7 er svært godt



Bør programmering og algoritmisk tenking være et eget fag i vgs? 1 er helt uenig, og 7 er helt enig



I de neste spørsmålene er 1 "Kjenner ikke til", og 7 er "trygg på bruk"

Hvor godt kjenner du til hva en variabel (i programmering) er?



Hvor godt kjenner du til hva en if-test (i programmering) er?



Hvor godt kjenner du til hva en løkke (i programmering) er?



Hvor godt kjenner du til hva en funksjon (i programmering) er?



Holdninger

I denne delen ønsker jeg dine ærlige holdninger om matematikk-faget, algoritmisk tenking og programmering som en del av matematikk-faget. Minner igjen om at ingen av dine svar kan knyttes mot deg som person.

I de neste spørsmålene er 1 definert som "Svært lite" og 7 som "Svært mye"

Hvor motivert er du til å lære vekk programmering og algoritmisk tenking til bruk i matematikkfagene?



-
-
-
-
-
-
-

Programmering og algoritmisk tenking er en nyttig kompetanse



Erfaringer

I den neste delen kan du ta utgangspunkt i at "algoritmisk tenking" er definert som "...grunnlaget som kreves for å løse problemer effektivt (dvs. algoritmisk, med eller uten hjelp fra datamaskiner) med løsninger som kan gjenbrukes i forskjellige sammenhenger."

Hvilke programmeringsspråk kjenner du til?

- Andre
- R
- C++
- JavaScript
- Python
- Java
- C#
- Ingen

Har du erfart noen bruksområder for programmering og algoritmisk tenking? 1 er ingen områder og 7 er veldig mange områder



Hvordan jobbes det med algoritmisk tenking og programmering i klasserommet?

- Følg oppskrift på å lage program
- Les, kopier og tilpass programeksempel
- Tutorials (videoer)
- Kopiere kode til lærer
- Algoritmisk tenking uten PC
- Overlatt til elevene
- Lese kode og tolke hva programmet gjør
- Utvid utdelt program
- Har ikke jobbet med det

Programmeringskompetanse (med algoritmisk tenking)

I disse spørsmålene vil du bli spurt om å forstå programmene, og svare på spørsmål relatert til programmene.

Hvilke verdier har variablene a, b og c etter programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7 b = a + 2
8 a = a + 11
9 c = b ** 2
```

- a har verdi 18, b har verdi 9 og c har verdi 16
- a har verdi 5.5, b har verdi 9 og c har verdi 81
- a har verdi 7, b har verdi 9 og c har verdi 16
- a har verdi 18, b har verdi 9 og c har verdi 81
- a har verdi 18, b har verdi a2 og c har verdi 4
- a har verdi 5.5, b har verdi 9 og c har verdi 16
- Jeg har ikke peiling

Hva gjør kommandoen print i programmet?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7 b = a + 2
8 a = a + 11
9 c = b ** 2
10
11 print(a + b + c)
```

- Print vil skrive ut verdiene til a, b og c
- Print vil lage en ny variabel med verdiene til det som er gitt i parentesene
- Print blir brukt til å utføre matematiske beregninger. Her vil den summere verdiene til a, b og c
- Print skriver ut summen av verdiene til a, b og c
- Print vil la brukeren skrive inn svar på matematisk utregning
- Print sender summen til utskrift på lokal printer

Hvilken verdi vil a ha etter programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 7
7
8 if a < 3:
9     a = 0
10 elif a < 7:
11     a = 1
12 elif a > 5:
13     a = 2
14 else:
15     a = 9
```

- 0
- 9
- 1
- 7
- 2

Hva er verdien til a når programmet er kjørt?

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 a = 1
7 for i in range(3):
8     a += i
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Hva skjer i linje 9? (fun(1,2,3))

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def fun(a,b,c):
7     return a * b * c
8
9 fun(1,2,3)
```

- Et funksjonskall som vil returnere produktet av 1, 2 og 3, som blir 6.
- Et funksjonskall som vil returnere variablene a, b og c
- Et funksjonskall som sjekker om tallene kan multipliseres
- Et funksjonskall som vil returnere 123
- Et funksjonskall som vil returnere 1*2*3 som tekst
- Kommandoen fun vil gi deg et tilfeldig tall av de tallene du oppgir
- Her vil programmet gi en feilmelding

Hva skal funksjonen k gjøre? Skriv kort.

```
1 # -*- coding: utf-8 -*-
2 """
3 @author: emithu
4 """
5
6 def f(x):
7     return 2*x**2 + x -5
8
9 def k(f, a, delta_x):
10     return (f(a + delta_x) - f(a - delta_x)) / (2*delta_x)
11
12 k(f, 1, 1E-8)
```

- Finne en tilnæringsverdi for deriverte til funksjonen f
- Finne integralet til en funksjon
- Finne en tilnærming til nærmeste nullpunkt
- Finne toppunktet til f
- Finne en tilnæringsverdi til neste verdi i funksjonen f
- Har ikke peiling

9.3 Transkribert fokusgruppeintervju

Emil: «Da har eg med meg fem elevar frå R1-klassen. Eg skal prøva no å få innspel i forhold til kva spørsmål som er relevante for å finne ut om holdningar, meiningar og erfaringar med algoritmisk tenking og programmering i matematikkfaget. Da lurar eg på om de har nokon.. ke spørsmål berr eg stille for å finne ut om holdningar, meiningar og erfaringar med algoritmisk tenking egentlig. For å finne ut korleis det har gått. Eg kan begynne med eit spørsmål om (elev avbryt)»

Elev1: «Eg kan begynne. Det e kanskje litt lurt å finne ut om folk syns det er vanskelig, eller om det lett i forhold til ka matte di ha. Og om de var forberedt på det di fikk da.

Emil: «Vanskegrad i forhold til det dei var forberedt på? (Elev nikkar). Ja, fint. Er det andre?

Elev2: «Kanskje om de har hatt noe fra barneskolen... noe lignende tidligere?»

Emil: «Så kanskje ta med tidligere erfaring fra programmering?»

Elev3: «Om de er avhengige av å kopiere kode som lærar har skrive, eller klare skrive sjølv?»

Emil: «Om dei kan lage egne program, eller avhengig av å kopiere kodesnutta?»

Elev3: «Det stemmer for meg hvertfall»

Emil: «Ja, hvertfall på dette tidspunktet

Elev2: «Kanskje hvor god lærar er til å kode?»

Emil: «Så...Kompetanse til lærar?»

Elev2: «Ja, lurte på kompetanse i fag...»

Emil «Kompetanse til lærar da?» «Da kan eg ta både korleis elev oppfatta det, og faktisk kva kompetanse lærar har?» « Det kan vere at lærer har kompetanse, sjølv om dei ikkje får det fram.»

Elev4: «Kanskje om de vet ka algoritmisk tenking betyr?»

Emil: «Ja, så kva algoritmisk tenking betyr? Ja.» (elev nikkar) «Er det eit lett spørsmål? Ka er algoritmisk tenking?» «Kunne de svart på det no?» (Elevane ser usikre ut, og ein seier forsiktig nei.) «Forslag til korleis ein kunne forklart det?»

Elev2: «Løse problemstilling med algoritme»

Emil: «Ja, så løyse problemstilling med algoritme». «Men kva er då ein algoritme?»

Elev2: «Sånn liningssett og så vidare.»

Emil: «Okei, så å sette opp ligningssett»

Elev1: «Kan jo spørre om det er motiverende.. i forhold til... egentlig litt vanskegrad og sånt. Om det blir for vanskelig i forhold til det vi har hatt om. Om det passer inn med de andre temaene?»

Emil: «Ja, motivasjon i forhold til å lære matematikk»

Elev1: «Ja, det er jo litt sånn nytt å liksom ta matematikk som egentlig kan gjerest på PC sant? Det blir jo litt sånn koding og så logaritmer og sånt, som kanskje gjøres for hånd, men det går jo an å gjøre ting på pc»

Emil: «Ja»

Elev1: «Og sånn bruksområde og sånn. Sånn du vet eventuelt kor vi kan bruke det vi lærer. Som også vil spille inn på motivasjon om vi ikke vet hvor vi kan bruke det og sånt.»

Emil: «Korleis skal eg formulere spørsmålet da?» «På hvilket område mener du at algoritmisk tenking kan bli brukt... eller programmering kan bli brukt?» «Ke område kan det bli brukt til da? Har de noko erfaringar no?»

Elev4: «Sikkert bli brukt til utregning om man har tenkt å gå... eller lage noe av det»

Emil «Ja, hvertfall lage modeller» «Ja, ...»

Elev1: «Trukje vi får bruk for programmering, siden vi ikke startet med det fra tidlig alder. Vi vet ikke hvordan det er å prgrammere... di oppgavene vi får er jo bare å si hva koden kommer til å gjøre. Men di som går i barneskolen kommer sikkert til å lage program. Men trukje vi kommer til å få bruk for det.»

Emil: «Ja, så det er meir sånn les, kopi og tilpass»

Elev2: «Vi er jo prøvekaniner. Prøv å feil med kode i andre fag. Vi har jo begynt med koding i kjemi og fysikk. Det står hvertfall i boken da. Den nye boken vi har fått. Det er meir kodar. Eg vetkje om det har vert før?»

Emil: «Neida, det er nytt.»

Elev1: «Men di bøkene skal jo videre flere kull. Så di som går i barneskolen kan sikkert programmere mere selv. I stedetfor å bare si ka koden gjør.»

Emil: «Korleis kan eg eventuelt stille spørsmål i forhold til kva erfaringar elevane har gjort seg da?» «Me snakka om at det var nyttig? Som spørsmål.» «Er det andre spørsmål eg kan få fram erfaringar som elevane har gjort seg med algoritmisk tenking og programmering?»

Elev3: «Kordan vi jobbar med oppgåver? Programmering eller komplekse oppgaver.»

Emil: «Ja, så kossen type oppgåver?»

Elev4: «Kor mye tid bruker vi på det her? Viss man bare går kjapt gjennom det sant, så er det fort gjort å ikke få det med seg. Siden det er helt nytt, så burde man jo bruke lengre tid på det.»

Emil «Ja, kan eg da spørre om antall timer liksom?» «Eller, berre sånn der...»

Elev3: «Kanskje spørre om det de har lært er nok? Føler di at di har fått nok tid til å jobbe?»

Elev1: «Kanskje bare nok til å lære? Om di har prioritert det? Det er jo overalt i boken. Men når de spør om oppgaver som skal gjøre i Python, så gjør eg ikkje det, så hopper eg over oppgavene.» «Så vi liksom prioriterer det ikke så mye»

Emil: «Så det kan eg spørre både til elev og lærer da? Om korleis dei har prioritert?» «Lyddopptaket er i gang, sånn at du veit det» (Til ny elev som kom for seint) «Ehm.. Me har vore innpå litt forskjellig type holdningsspørsmål til algoritmisk tenking og programmering. Erfaringar. Så har me prøvd å få ned litt om kompetansenivå til lærer og korleis han formidla dette. Så eit spørsmål gjekk på om elevane ser kva område han kan utnytte dette til. Er det andre spørsmål som du kan komme med?» «Ka e det viktigaste spørsmålet eg må stille for å finne ut om detta har blitt inført for alle elevane i norge?»

Elev5: «Eg sliter med å formulere det. Da eg sittar i koden, så kan eg ikkje sånn... Eg kan det ikkje særleg. Eg må søke opp.. ka kommando bruker eg til det. Så legger vi det inn. I prosjektet no, så kopierte vi det du gjort, så gjorde vi endringer vi trengte og fikk det til å funke. Så vi kan det ikkje, men vi forstår det godt nok til å endre på.. til å kopiere det riktig.»

Emil: «Ja, eg trur me var innom det. Korleis det blir brukt, og ja.. mange har erfaring at kopiere og tilpasse til bruk.»

Elev5: «Ja, så spørsmålet eg tenkte på da.. Har du forståelse på koding og dette som går på algoritmisk tenking og programmering?»

Emil: «Ja, så skala fra 1-7, med surt fjes til blidt fjes. Kan det funke?»

Elev5: «Ja, det kan funke»

Emil: «Ville de svart sånn 4-5 kanskje?»

Elev5: «Ja, sånn cirka.»

Elev1: «Eg tror det virka litt høyt. Kanskje 3-4»

Elev5: «Ja for vi har veldig mye om alle andre tingene, så har vi veldig lite koding i forhold til hvor mye det er å lære om det. »

Elev1: «For eksempel om eg hadde fått spørsmål om å kode alt fra prosjektet. NEI. Det hadde eg ikke. Det var veldig mye nytt som kom, ikke sant. Så det var veldig veldig veldig mye lærerikt. Lærte mye nytt.»

Emil: «Ja, så da er det måten vi jobbar på som var spørsmål tidligere her. Men så var det også spørsmål om, dei som kjem etter vil ha meir erfaring. Til neste år erfaring fra 10ende, så 9ende, osv. Noko som vil ha noko å seie.»

Elev5: «Så spørsmålet er jo. Er du interessert? Ser du bruksområdet? Forstår du ka tankegang du må ha? Eller kordan formidle informasjon til systemet, for at det skal forstå det videre?»

Emil «Ja, så da er vi inne på koding, den siste biten.» «Oppbyggingen av eit program da.. kossen skulle me.. starta til slutt... i prosjekt hadde de... les inn data... ehm... sortere data i x og y-verdi.. lag modell... plott modell.. det er jo liksom stega. Om de hadde fått ei anna oppgåve no, hadde de klart å sortere stega?»

Elev5: «Ja, om det hadde vert samme oppgave.»

Emil: «Så om det hadde vore lignande. Men ka om det hadde vore: Finn eit nullpunkt til denne funksjonen her.»

Elev5: «Da måtte eg såtte og tenkt litt. Det hadde tatt en stund å prøvd meg ut for det. Det hadde ikke berre vert sånn: Så klart er det den eg må bruke..»

Elev2: «Eg kunne klart å tatt i bruk hjelpemiddel fra nettet, men vi hadde ikke hatt kunnskapen til å sitte der på en prøve å klare det sjølv»

Emil: «Ja, så det hadde ikkje tatt ti minutt.» «Men ein time, så hadde du fått det til?»

Elev4: «Man kan jo spørre noe om kommandoene. Om det er nok læring til å forstått kommandoene. Sånn i forhold til print, og alle di der.»

Emil: «Ja, eg tenkte å ta med dei fire konseptane. I forhold til løkke, me har vore inne på det at koden blir kjørt flere ganger, det var løkke. Så hadde me noko som heiter if-setning, da kunne me sjekke nokon verdiar. Og så skal eg spørre om variablar tenkte eg. Forskjell på ein variabel i programmering og i matte. I programering meir konstant enn i matte.»

Elev5: «Ja, det er vanskelig for oss med variabler. Det er helt nytt.»

Emil: «Ja, da har me igjen eit minutt. Er det noko som har komt på nokon spørsmål?»

Elev5: «Syns du at det brukes nok tid på det? Du kommer nok til å se en sammenheng mellom syns du det er interessant og bruksområde. De to, og det brukes nok tid på det. Det kan vere noe annet også. Har du sett på det utenom skolen. Har du sett på det på egen oppfordring?»

Emil: «Bruk av tid utenom skole på dette temaet da.» «Ja, ...»

Elev5: «Ka spørsmål har du no?»

Emil: «Mange..» (latter) «Har fått mange gode innspel» «Er det nokon andre som har noko meira no? Det var hvertfall det om nytten, som eg hvertfall skal ha med.»

Elev1: «Om det burde vert et egent obligatorisk fag. Siden det så stort. Det blir kanskje det i framtiden. Det er jo overalt i boken.»

Emil: «Me valgt Aschehoug fordi det var nettopp mange eksempler i den boken. Mulig for mange for dykk. Førstemann ut.»

Elev1: «Vi bruker ikke nok tid på det. «

Emil: «Ka kan grunnen til at me ikkje brukar mest tid på det da?»

Elev5: «At me har så mye annet å lære fra før.»

Elev1: «Ingen erfaring med det. At vi ikkje skal bli vurdert på dette, men på løsing av oppgaver. Vil endre seg... De vil jo gjøre de oppgavene som jeg ville hoppet over.»

Elev5: «Vi er veldig grunnleggende. Kan sammenlignes med gloseprøve...»

Emil: «Så pugging.»

Elev5: «Så ingen lange tekster. Eg har en ting til. Det er litt sånn omformulering. Tror du at du kommer til å bruke dette. Tror du at du kommer til å bruke programmering eller koding i jobben din? Algoritmisk tenking?»

Emil: «Ja, etter videregående eller endt utdanning. Ja, da stoppar eg opptaket.»

9.4 Oppsummering av transkribering

Introduksjon frå meg:

«Eg skal prøva no å få innspel i forhold til kva spørsmål som er relevante for å finne ut om holdningar, meiningar og erfaringar med algoritmisk tenking og programmering i matematikkfaget»

Holdninger

1. Er du interessert i å lære om algoritmisk tenking og programmering?
2. Hvordan er motivasjonen for å lære om algoritmisk tenking og programmering i matematikk?
3. Er det nyttig å lære om algoritmisk tenking og programmering?
4. Bruker du mer eller mindre tid på algoritmisk tenking og programmering enn andre kompetanseområder utenom skoletid?
5. Har du utforsket programmering på egen oppfordring i tillegg til det som blir brukt faglig?
6. Tror du at du vil få bruk for kompetanse innen algoritmisk tenking og programmering etter videregående?

Meninger

1. Synes du at algoritmisk tenking og programmering er vanskelig?
2. Oppfatter du at læreren din har god nok kompetanse til å lære vekk algoritmisk tenking og programmering?
3. Hvilke kompetanser mener du er en del av algoritmisk tenking og programmering?
 - a. Forslag fra elev: «Løse problemstillinger med algoritme»
4. Mener du at algoritmisk tenking og programmering har blitt prioritert i undervisningen?
5. Hvordan passer algoritmisk tenking og programmering inn med de andre temaene i matematikken?
6. Burde algoritmisk tenking og programmering være et eget obligatorisk fag?

Erfaringer

1. Visste du at du skulle lære algoritmisk tenking og programmering i matematikken?
2. Tidligere erfaringer med algoritmisk tenking og programmering?
 - a. Fra barneskolen?
 - b. Programfag med programmering?
3. Har du erfart noen bruksområder for algoritmisk tenking og programmering?
4. Hvordan jobbes det med algoritmisk tenking og programmering?

- a. Les, kopier, tilpass
- b. Video
- c. Følg oppskrift
- d. Utvid
5. Har du fått nok opplæring i algoritmisk tenking og programmering?
6. Har du fått nok tid til å jobbe med algoritmisk tenking og programmering?
7. Har du tilegnet deg forståelse av algoritmisk tenking og programmering?
 - a. Skala 1-7 (Ingen – Veldig mye)
8. Blir det brukt nok tid på programmeringseksemplene i boka?

Kompetanse

1. Kan du lage program på egenhånd, eller er du avhengig av å kopiere koder fra læreren og tilpasse?
2. Til lærar: Formell kompetanse og erfaring i algoritmisk tenking og programmering.
3. TEST: print
4. TEST: løkke
5. TEST: if
6. TEST: Variabler

10 Figurliste

Figur 1: Eksempel på standard svaralternativ brukt i undersøkinga	28
Figur 2: Mal på e-post som vart sendt ut til faglærarane.....	33
Figur 3: Oversikt over svarstatus frå skulane	34
Figur 4: Topp 5 val for kva elevane meina algoritmisk tenking er.	46
Figur 5: Dei tre alternative for å skildre algoritmisk tenking med lågast svarprosent	46
Figur 6: Eigenvurdering av kompetanse i grunnleggjande prinsipp	48
Figur 7: Arbeidsmetodar for algoritmisk tenking og programmering i klasserommet	51
Figur 8: Test av forståing av kommandoen print	52
Figur 9: Program med vilkår	53
Figur 10: Program som testar kompetanse knytt til løkker	53
Figur 11: Program som testar forståing av funksjon	54
Figur 12: Programeksempel for å finne tilnæringsverdi av derivert i eit gitt punkt	54
Figur 13: Lærarane si tolking av algoritmisk tenking	56
Figur 14: Lærarane sine tre minst populære val til algoritmisk tenking	56
Figur 15: Arbeidsmetodar for algoritmisk tenking og programmering.....	57
Figur 17: Oppgåve 5 (del 1), eksamen 1P våren 2022	59
Figur 16: Oppgåve 4 (del 1), eksamen 2P våren 2022	59
Figur 18: Oppgåve 4 (del 1) eksamen 1T våren 2022.....	59

Figur 19: Oppgave 6 (del 1) frå eksamen i S1 våren 2022	60
Figur 20: Oppgave 4 (del 2) frå eksamen i S1 våren 2022	60
Figur 21: Oppgave 5 (del 1) eksamen i matematikk R1 våren 2022.....	60
Figur 22: Oppgave 5 (del 2) frå eksamen i matematikk R1 våren 2022	61
Figur 23: Motivasjon for å lære seg algoritmisk tenking og programmering	63
Figur 24: Kvinnene og mennene sin kompetanse samanlikna med eigen vurdering av kompetanse	64
Figur 25: Prestasjon og omgrepet algoritmisk tenking	65
Figur 26: Utforsking, til venstre med forkunnskap og utan forkunnskap til høgre.....	69

11 Tabelliste

Tabell 1: Sammendrag av nøkkelkomponenten Computational concepts (Brennan & Resnick, 2012, s. 3-6)	20
Tabell 2: Sammendrag av nøkkelkomponenten Computational practices (Brennan & Resnick, 2012, s. 6-9)	21
Tabell 3: Sammendrag av nøkkelkomponenten Computational practices (Brennan & Resnick, 2012, s. 10-11)	21
Tabell 4: Oppsummering av taksonomien til Weintrop et. al. (2016).....	23
Tabell 5: Oversikt over geografisk fordeling blant deltakarane.....	43
Tabell 6: Statistikk frå karakterar i matematikk 1T og R1 (1.termin).....	44
Tabell 7: Oversikt over standpunkt karakter får matematikk 1T	44
Tabell 8: Oversikt over terminkarakter i matematikk R1	45
Tabell 9: Opplæring i algoritmisk tenking (resultat).....	47
Tabell 10: Kompetanse i grunnleggande programmering (resultat)	48
Tabell 11: Motivasjon til matematikk og algoritmisk tenking (resultat)	49
Tabell 12: Arbeidsvanar og eiga prioritering (resultat).....	50
Tabell 13: Undervisningsmetode (resultat)	57
Tabell 14: Oppsummering av kjønnsforskjellar	65
Tabell 15: Tileigne kompetanse gruppert etter karakter i 1.termin matematikk R1	67
Tabell 16: Oppsummering av forskjellar mellom høgt og lågt presterande elevar	67
Tabell 17: Forkunnskapar og kompetanse i grunnleggande algoritmisk tenking og programmering	68

Tabell 18: De største forskjellene som de med og uten forkunnskaper mener at algoritmisk tenking er	70
Tabell 19: Oppsummering av forskjellar mellom dei med/utan forkunnskapar	70
Tabell 20: Algoritmisk tenking kryssa med motivasjon for å lære algoritmisk tenking og programmering	72
Tabell 21: Motivasjon og kompetanse i grunnleggande algoritmisk tenking og programmering	73
Tabell 22: Oppsummering av forskjellar mellom dei høg/låg motivasjon for å lære algoritmisk tenking og programmering	74