UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

Algoritmisk tenkning og kreativitet

En kvalitativ studie av kjennetegn ved algoritmisk tenkning og kreative framgangsmåter i matematikkfaget.

Yasmina Madi

Masteroppgave i matematikkdidaktikk, juni 2023

Lektorutdanning i naturvitenskap og matematikk, Matematisk institutt

Forord

En følelse av glede og tristhet fyller rommet idet slutten på lektorutdanningen nærmer seg. Å skrive denne masteroppgaven har vært en berikende reise, og jeg ønsker å uttrykke min takknemlighet til hver enkelt som har vært en del av den.

Først og fremst vil jeg rette en stor takk til min veileder, Jonas Oskarsson, ved Høyskolen i Innlandet. Din kompetanse og engasjement har vært uvurderlig i denne prosessen. Du har vært en kilde til inspirasjon og en viktig støttespiller gjennom hele reisen. Jeg er utrolig heldig som fikk gleden av å ha deg som veileder. Jeg ønsker også å takke lærerne og elevene som har gitt meg muligheten til å gjennomføre datainnsamlingen. Takk for deres interesse for mitt prosjekt og for det gode samarbeidet vi har hatt.

Videre vil jeg gjerne rette en takk til alle mine medstudenter. Jeg kommer til å savne våre lunsjpauser og de gode øyeblikkene vi har delt. Jeg vil også takke Nora og Magnus, som har vært ved min side. Takk til Mathilde, som har heiet på meg fra den andre siden av kloden. Deres vennskap og støtte er uerstattelig.

Til slutt vil jeg rette en stor takk til min familie. Jeg ønsker å takke mine kjære søstre, Alisa og Diana, for deres enorme støtte og oppmuntring i de krevende stundene. Takk for latterfylte øyeblikk og sene telefonsamtaler som har varmet hjertet mitt. Dere får meg til å tvile på om jeg virkelig er den eldste blant oss. En spesiell takk går til mine flotte foreldre, Yasser og Svitlana. Deres tro og kjærlighet har vært min største motivasjon. Ord kan ikke beskrive hvor mye dere betyr for meg. Denne masteroppgaven hadde aldri blitt til uten dere.

Denne oppgaven er dedikert i ære til min kjære venn og tidligere medstudent, Marcus Burkeland. Vi klarte det.

Yasmina Madi

1.juni, 2023

Sammendrag

Den raske teknologiutviklingen har ført til at det stadig stilles nye krav til befolkningens kompetanse og ferdigheter. Algoritmisk tenkning og kreativitet anses som essensielle ferdigheter i det 21.århundre. Gjennom fagfornyelsen har algoritmisk tenkning blitt synliggjort i læreplanverket LK20 etter Kunnskapsløftet 2020. I overordnet del av LK20 blir kreativitet framhevet som en vesentlig faktor som bidrar til å berike samfunnet. Til tross for at algoritmisk tenkning har fått mye oppmerksomhet de siste årene, er det fortsatt mangel på en universell definisjon, og usikkerhet knyttet til hvordan man observerer utviklingen av algoritmisk tenkning hos elever. Gitt betydningen av algoritmisk tenkning og kreativitet belyser derfor denne masteroppgave følgende problemstilling: *Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk hos R1 elever i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?*

Problemstillingen har blitt belyst gjennom en kvalitativ tilnærming. Datamaterialet består av videoopptak av elevsamtaler og deres tilhørende arbeid i matematisk problemløsning. Kjennetegn ved algoritmisk tenkning ble observert ut fra Csizmadia et al. (2015) sitt rammeverk. Videre ble Lithner (2008) sitt rammeverk brukt for å undersøke om framgangsmåter var av kreativ eller imitativ karakter.

Alle kjennetegn ved algoritmisk tenkning beskrevet i Csizmadia et al. (2015) sitt rammeverk ble observert i denne studien. *Algoritmebehandling, dekomposisjon* og *evaluering* var de mest frekvente kjennetegnene som kom til uttrykk. Videre ble det observert både kreative og imitative framgangsmåter. Studiens resultater antyder at kreative framgangsmåter kjennetegnes ved at elevene klarer å frigjøre seg fra kjente løsningsalgoritmer og metoder. De kreative framgangsmåtene viste også spor av faktorer som intuisjon og inkubasjon. Studiens funn indikerer at det er mulig i stor grad å påvirke hvilke kjennetegn ved algoritmisk tenkning som kommer til uttrykk, samt framdriften av kreative framgangsmåter gjennom utformingen og formuleringen av oppgaver. Resultatene antyder også at algoritmisk tenkning kan være nyttig for å legge grunnlaget for, og videreutvikle kreative idéer i matematikkfaget.

Innhold

Tabelliste

Figurliste

1 Innledning

1.1 Kompetanse for framtiden

Den teknologiske utviklingen har hatt en betydelig innvirkning på samfunnet (NOU 2020:2, 2020). I Norge har samfunnet endret seg mye gjennom flere industrielle revolusjoner, og endringene vi står overfor i dag omtales ofte som den fjerde industrielle revolusjonen (NOU: 2020:2, s.13). Sammen med disse endringene stilles det stadig nye krav til befolkningens ferdigheter, kunnskaper og kompetanser. Å forstå hvilken virkning teknologiutviklingen har på samfunnet er dermed essensielt for å kartlegge kompetansebehovene i framtiden (Selvik et al., 2016). En ekspert arbeidsgruppe oppnevnt av Utdanningsdirektoratet undersøkte hvilke ferdigheter og kompetanser norske elever burde ha for å møte framtidenes kompetansebehov i det 21.århundre (Sanne et al., 2016). Arbeidsgruppen utarbeidet rapporten *Teknologi og programmering for* alle, hvor algoritmisk tenkning ble løftet fram som en av de sentrale kompetansene for framtiden. Algoritmisk tenkning er sammenstilt med det engelske begrepet *computational thinking* og blir beskrevet som en viktig problemløsningsmetode (Sanne et al., 2016, s.26). Videre påpeker rapporten at elevene skal være rustet til å møte framtidens arbeidsliv ved å ha kompetanse til å håndtere utfordringer på en kreativ måte (Sanne et al., 2016, s.77). Kreativitet regnes som en viktig faktor i matematikkfaget fordi det kan bidra til å utvikle elevenes problemløsningsevner (Grover & Pea, 2013; Israel-Fishelson et al., 2021a; Israel-Fishelson et al., 2021b).

1.2 Bakgrunn for valg av tema og problemstilling

Algoritmisk tenkning har fått stor oppmerksomhet de siste årene og forskningen på algoritmisk tenkning er et fremvoksende felt som har vokst eksponentielt siden 2013 (Tekdal, 2021). Flere land har inkludert algoritmisk tenkning i læreplanen på ulike måter (Bocconi et al., 2018, s.1; Nordby et al., 2022, s.36). I Norge har matematikkfaget spesielt blitt tillagt ansvaret for å utvikle elevenes kompetanse i algoritmisk tenkning, noe som stiller nye krav til lærerne som underviser i matematikk (Kristensen & Kirfel, 2022, s.7). Til tross for at algoritmisk tenkning har fått en betydelig plass i skolen er det fremdeles uklart hva begrepet omfatter. Flere har gjort et forsøk på å definere algoritmisk tenkning og inkludere de ulike dimensjonene av begrepet, men likevel er det manglende konsensus om en entydig og universell definisjon (Bocconi et al., 2016, s.6; Brennan & Ressnik, 2012, s.1; Román-González et al., 2018, s.1; Selby & Woolard, 2014, s.1).

En litteraturanalyse gjort av Haseski et al. (2018) viser til hele 59 ulike definisjoner av algoritmisk tenkning i perioden fram til 2016 (referert i Nordby et al., 2022, s.36). Manglende konsensus rundt definisjonen av algoritmisk tenkning får konsekvenser i form av visse implikasjoner knyttet til lærerrollen. Hvordan kan man som lærer vurdere utviklingen av algoritmisk tenkning hos elevene uten en klar og presis definisjon? Hvilke kjennetegn skal man se etter? Og hvordan kan man skille mellom denne type tenkning fra matematisk tenkning? Videre er kreativitet et sentralt element i læreplanverket som skal fordre elevene til å skape noe nytt i møte med nye og ukjente problemer (Kunnskapsdepartementet, 2018). Men hvordan ser kreativitet ut i matematikkfaget? Hvilke karakteristiske trekk er det som kjennetegner kreative framgangsmåter? Noen studier antyder at algoritmisk tenkning kan stimulere kreative måter å løse problemer på (Balanskat & Engelhardt, 2015; Bråting et al., 2020; Nouri et al., 2020; OECD, 2019; Selvik et al., 2016). Samtidig er det begrenset med forskning som viser til hvorvidt det er en reell sammenheng mellom algoritmisk tenkning, og dermed er det vanskelig å trekke noen konklusjoner (Israel-Fishelson et al., 2021a, s.438).

Hensikten med denne masteroppgaven er å få bedre innsikt i konseptene algoritmisk tenkning og kreativitet i matematikkfaget. Ettersom både algoritmisk tenkning og kreativitet er omfattende temaer, har jeg valgt å begrense oppgaven gjennom følgende problemstilling: *Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk hos R1 elever i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?*

For å belyse problemstillingen har det blitt gjennomført en kvalitativ studie, hvor datamaterialet består av elevsamtaler og elevbesvarelser i arbeid med problemløsning. Gjennom en grundig analyse av kjennetegnene ved algoritmisk tenkning og kreative framgangsmåter i problemløsning, har denne masteroppgaven som mål å bidra til en dypere innsikt i hvordan disse konseptene kommer til uttrykk blant elever i matematikkfaget. Resultatene kan potensielt bidra til forskningsfeltet som omhandler sammenhengen mellom algoritmisk tenkning og kreativitet.

1.3 Oppgavenes oppbygning

Oppgaven er delt inn i 6 kapittel. I kapittel 1 introduseres temaet og bakgrunnen for valg av problemstillingen.

Kapittel 2 presenterer det teoretiske grunnlaget som er relevant for oppgavens problemstilling. Algoritmisk tenkning, kreativitet, matematisk kreativitet og andre relevante begreper og definisjoner blir redegjort. Csizmadia et al. (2015) sitt rammeverk for vurdering av utviklingen av algoritmisk tenkning blir presentert, i tillegg til Lithner (2008) sitt rammeverk for kreative og imitative resonnement.

Kapittel 3 omhandler studiens metodiske tilnærminger. Valg som ble foretatt i løpet av forskningsprosessen blir beskrevet og begrunnet. Her blir studiens vitenskapssyn, datainnsamlingsprosessen og analyseprosessen presentert. Til slutt blir kvaliteten på studien og etiske betraktninger diskutert.

I kapittel 4 presenteres resultatene fra analysen. Kapittelet er delt inn i to. I første del av kapittelet presenteres funn fra analysen av datamaterialet i henhold til Csizmadia et al. (2015) sine kjennetegn ved algoritmisk tenkning. Andre del av kapittelet presenterer funn fra analysen av elevenes framgangsmåter ut fra Lithner (2008) sitt rammeverk om kreative og imitative resonnement.

I kapittel 5 drøftes funn fra analysen i lys av det teoretiske grunnlaget fra kapittel 2.

Funnene fra oppgaven oppsummeres i kapittel 6. Kritisk refleksjon og veien videre blir presenter til slutt.

2 Teoretisk bakgrunn

Dette kapittelet presenterer det teoretiske grunnlaget for oppgaven. I første del av kapittelet følger en begrepsavklaring av algoritmisk tenkning. Deretter vil rammeverket utviklet av Csizmadia et al. (2015) for vurdering av utviklingen av algoritmisk tenkning bli presentert. Andre del av kapittelet består av en redegjøring av begrepet kreativitet, og hva matematisk kreativitet i problemløsning innebærer. Avslutningsvis presenterer kapittelet Lithner (2008) sitt rammeverk for observasjon av kreative og imitative resonnement.

2.1 Algoritmisk tenkning

2.1.1 Algoritmisk tenkning eller computational thinking?

Algoritmisk tenkning er den norske oversettelsen av det engelske begrepet *computational thinking* (Nordby et. al, 2022, s.37). Denne oversettelsen fører til noen implikasjoner. Når man prøver å oversette det engelske begrepet til norsk kan det oppstå språklige utfordringer. Historisk sett betegnet ordet *computer* en person som foretok matematiske utregninger enten for hånd eller kalkulator (Hannemyr, 2022). Ordet *computer* referer i dag til en elektronisk maskin som brukes til å behandle, lagre og dele data (Hannemyr, 2022). *Computational* er et adjektiv som betegner prosessen med å utføre beregninger ved hjelp av en matematisk formel eller en algoritme (Tallis, 2004, s.37). Ifølge Utdanningsdirektoratet (2019, s.1) er den norske oversettelsen en direkte oversettelse av *computational thinking*. Med andre ord har algoritmisk tenkning en-til-en korrespondanse med det engelske begrepet. Likevel kan denne oversettelsen anses å være noe problematisk. Under høringsutkastet til de nye læreplanene ble bruken av begrepet algoritmisk tenkning både kommentert og problematisert (Nordbakke, 2018, s.38). Ordet *algoritmisk*, avledet av *algoritme*, er ikke noe nytt i matematikkfaget. Divisjonsalgoritmen er et klassisk eksempel på en algoritme som mange elever kan (Kjøsnes, 1997, referert i Gjøvik & Torkildsen, 2019, s.31). Dermed kan algoritmisk tenkning gi assosiasjoner til standardalgoritmer i matematikk. I tillegg kan det forveksles med det engelske begrepet *algorithmic reasoning*. Denne assosiasjonen påpekes også i studien av Nordby et al. (2022, s.37) og Kravik et al. (2022, s.5). Den norske oversettelsen kan som Gjøvik & Torkildsen (2019, s.31) skriver, virke som et «[…] gufs fra fortiden», fordi matematikkundervisningen i dag verdsetter og bruker mer åpne og problemløsningsoppgaver der elevene ikke nødvendigvis kjenner til framgangsmåten på forhånd. Det er også en enighet i forskningslitteraturen om at algorithmic reasoning utgjør en del av algoritmisk tenkning (Bocconi et al., 2016, s.16; Csizmadia et al., 2015, s.6; Grover & Pea, 2013, s.41; Vinnervik & Bungum, 2022, s.392; Wing, 2006, s.33). Den norske oversettelsen ser dermed ut til å føre til en viss uklarhet i begrepsbruken, og mangel på et tilsvarende begrep på norsk for det engelske begrepet algorithmic reasoning.

På den andre siden antyder ordet *tenkning* til at det ligger noe mer bak begrepet. Tenkning referer til en rekke psykologiske aktiviteter og som utgjør essensen av det å være et menneske (Evans, 2017). Disse inkluderer mentale prosessen med å bearbeide, tolke, vurdere eller konstruere informasjon (Svartdal, 2018). Videre poengterer Svartdal at tenkning kan skje ubevisst og automatisk. Andre ganger er tenkningen bevisst og kjennetegnes ved at den er logisk, tid- og ressurskrevende. Eksempler på dette er når man skal løse et problem, ta en beslutning eller vurdere ulike muligheter. Slik avansert form for tenkning betegnes som resonnering og er en viktig del av problemløsning, som utgjør en viktig del av matematikkfaget (Solvang, 1991; Svartdal, 2018). Så om matematikkundervisning skal bidra til at elevene skal bli gode problemløsere og forberede dem på et samfunn og arbeidsliv i stadig endring (Utdanningsdirektoratet, 2020a), er det naturlig at ordet *tenkning* blir vektlagt både i læreplanen og i matematikkundervisningen. *Tenkning* kan dermed vise til at algoritmisk tenkning handler om mer enn bare standardalgoritmer. Som Kjøsnes (1997) poengterer så er det kun få elever som forstår divisjonsalgoritmen, til tross for at mange kan bruke den (referert i Gjøvik & Torkildsen, 2019, s.31). I denne sammenheng kan algoritmisk tenkning referere til at elevene skal vite hvordan en algoritme fungerer, men også *tenke over hvorfor* den fungerer.

I en personlig korrespondanse skriver Utdanningsdirektoratet at computational thinking ble oversatt til algoritmisk tenkning i 2016 da læreplanen til forsøksvalgfaget programmering ble laget (Vedlegg 1). Oversettelsen ble diskutert i læreplangruppen på grunn av assosiasjonen med matematikkfaget og standardalgoritmer, men gruppen fant ingen bedre oversettelse som dekket begrepet godt nok. Oversettelser fra andre land ble også vurdert, inkludert det svenske *datalogisk tänkande*. Da fagfornyelsen skulle innføre algoritmisk tenkning i nye læreplaner, ble det inkludert i matematikkfaget, men også i naturfag, kunst og håndverk og musikk for å ytterlig poengtere begrepets omfang og brede definisjon. I tillegg ble algoritmisk tenkning lagt inn som del av kjerneelementet «Utforskning og problemløsning» for å vise til den store avstanden fra den konvensjonelle måten å bruke algoritmer på. Ettersom begrepet var allerede etablert det ble ikke vurdert å se på begrepet på nytt.

For å bidra til et ryddigere begrepsbruk og unngå misforståelser skal denne studien videre ta utgangspunkt i følgende:

- Til tross for en viss uklarhet i begrepsbruken, så har utdanningsmyndighetene i Norge valgt å sidestille begrepet algoritmisk tenkning med det engelske begrepet computational thinking (Utdanningsdirektoratet, 2019, s.1). Algoritmisk tenkning blir brukt i den norske litteraturen, og jeg vil dermed videre også bruke den norske oversettelsen algoritmisk tenkning som en nøyaktig oversettelse av computational thinking.

- Når det gjelder det engelske begrepet algorithmic reasoning så vil jeg benytte meg av Gjøvik & Torkildsen (2019) sin oversettelse. I det følgende vil *algoritmebehandling* referere til algorithmic reasoning. En definisjon av algoritmebehandling vil bli tydeliggjort i kapittel 2.1.4 ved hjelp av rammeverket til Csizmadia et al. (2015).

- En *algoritme* refererer til et sett med instruksjoner som gir en presis og fullstendig beskrivelse av framgangsmåten for å løse et problem (Solvang, 1991, s.134)

- *Problemløsning* er å søke etter handlinger som fører til en løsning av et ukjent problem (Solvang, 1991, s.135)

2.1.2 Begrepets opprinnelse

Begrepet algoritmisk tenkning anses å ha sin opprinnelse ifra matematikeren Seymour Papert (Bråting og Kilhamn., 2021, s.101; Shute et al., 2017, s. 143). Begrepet algoritmisk tenkning ble først brukt av Papert i boken *Mindstorms: Children, computerts, and powerful ideas* som ble gitt ut i 1980. Selve begrepet forekommer imidlertid kun én gang: «Their visions of how to integrate computational thinking into everyday life was insufficiently developed» (Papert, 1980, s.182). Ingen konkret definisjon av begrepet algoritmisk tenkning blir presentert i boken, og bruken av begrepet computational thinking i denne sammenhengen er snarere brukt på en tilfeldig måte (Lodi & Martini, 2021, s.980). I boken diskuterer Papert blant annet hvordan datamaskinen kan være med å påvirke evnen mennesker tenker og lærer på. Papert så nemlig et stort potensial ved bruk datamaskinen i læringssammenheng og refererte til datamaskinen som « […] the teaching machine» (Papert, 1980, s.3). Papert skriver blant annet at: «[…] the computer presence could contribute to mental processes not only instrumentally but in more essential, conceptual ways, influencing how people think even when they are far removed from physical contact with a computer» (Papert, 1980, s.4). Arbeid knyttet til datamaskinen vil med andre ord kunne bidra til mentale prosesser som vil påvirke hvordan mennesker tenker. Disse tankeprosessene ville også ha en overføringsverdi til andre sammenhenger. Videre mente Papert at programmeringsferdigheter er noe alle burde inneha, og hadde en visjon om at barn skulle lære seg å programmere fra tidlig alder (Papert, 1980, s.5). Han foreslo å bruke programmering som et verktøy for å legge til rette for en mer praktisk tilnærming til matematikkfaget, slik at elevene kunne å utvikle egne konstruksjoner eller mentale modeller av ellers abstrakte konsepter. På denne måten ville elevenes evne til å tenke på egenhånd utvikles: «And in teaching the computer how to think, children embark on an exploration about how they themselves think» (Papert, 1980, s.19). Selve veien fram til målet er viktig, og i denne sammenhengen er aktiv elevdeltakelse og samarbeid essensielle faktorer (Papert, 1980, s.126). Til tross for at teknologi spiller en essensiell rolle i realisering av Papert sin visjon, så er likevel hovedfokuset i boken på tankeprosessene som finner sted. Papert skriver at: «My interest is in universal issues of how people think and how they learn to think» (Papert, 1980, s.10).

Seymour Papert var forut sin tid og hans ideer fikk ikke noe stor innvirkning på den tiden boken ble utgitt (Bråting & Kilhamn, 2021, s. 172). En av årsakene som Bråting & Kilhamn påpeker er at digital teknologi ikke var en integrert del av menneskers hverdag på den tiden. Som et resultat var disse ideene både fjerne for mange, og vanskelige å implementere i praksis. Etter noen tiår fikk begrepet imidlertid ny oppmerksomhet, som bar preg av flere av Paperts ideer.

2.1.3 Algoritmisk tenkning i nytt søkelys

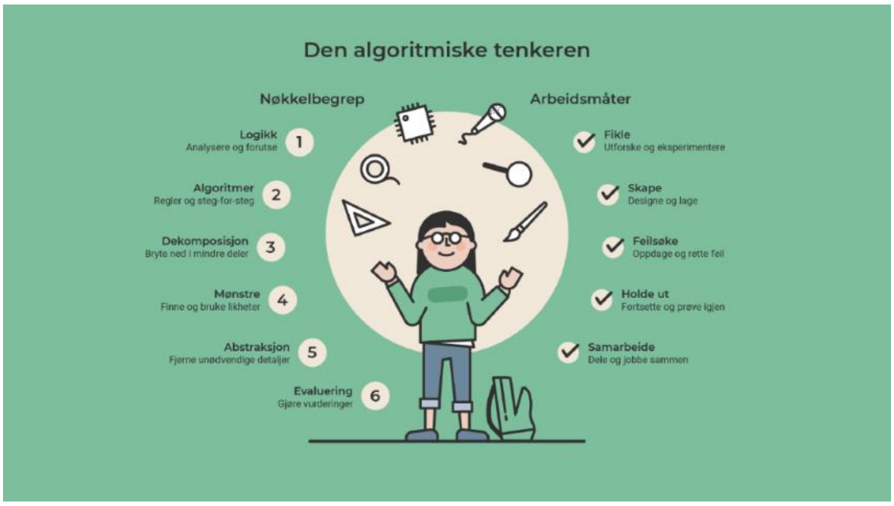
Omtrent tretti år senere fikk algoritmisk tenkning ny oppmerksomhet gjennom Jeannette Wing sin populærvitenskapelige artikkel i 2006 (Bråting & Kilhamn, 2021). Wing definerte algoritmisk tenkning som følger: «Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child’s analytical ability» (Wing, 2006, s.33).

I artikkelen framhever Wing (2006) betydningen av algoritmisk tenkning som en ferdighet som ikke bare er forbeholdt informatikere, men som alle kan ha nytte av å utvikle. Wing hevder videre at algoritmisk tenkning er en universal ferdighet på lik linje med blant annet lesing og skriving. Algoritmisk tenkning er heller ikke synonymt med det å tenke som en datamaskin. Tvert imot så framhever Wing (2006) den kreative siden som mennesker innehar når de løser problemer: «Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative» (Wing, 2006, s.34).

Videre i artikkelen knytter Wing algoritmisk tenkning til blant annet abstraksjon, dekomposisjon, algoritmebehandling, planlegging, nysgjerrighet og kreativitet. Disse komponentene ser ut til å framheve flere av prosessene som kan finne sted i problemløsning. Samtidig gir artikkelen en ganske bred beskrivelse av begrepet. Dette har resultert i at artikkelen har blitt utsatt for mye kritikk. Hemmendinger (2010) mente at Wing (2006) sin beskrivelse ikke var begrenset til algoritmisk tenkning alene (referert i Cansu, 2019, s.6). I følge Hemmendinger (2010) er flere av de nevnte komponentene gjenkjennelig i matematisk tenkning, noe som også støttes av Denning (2016) og Bueie (2019). Som svar på kritikken utarbeidet Cuny, Snyder og Wing en ny definisjon av algoritmisk tenkning: «Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing-agent» (Cuny et al. 2010, referert i Wing, 2011).

Wing (2011) bruker algoritmisk tenkning som et samlebegrep for den mentale aktiviteten som er viktig i prosessen med formuleringen og løsningen av et problem. I følge Shute et al. (2017, s.143) er denne definisjonen den mest siterte definisjonen av algoritmisk tenkning. To ting er verdt å merke seg her ifølge Bocconi et al. (2016, s.15). «Information-processing agent» refererer til at et problem kan løses enten av et menneske, datamaskin eller begge deler. Dermed kan algoritmisk tenkning tolkes som en tankeprosess uavhengig av teknologi. I undervisningssammenheng kan det derfor tenkes at lærere vil ha større frihet til å velge om de ønsker bruke programmering eller andre digitale verktøy for å utvikle algoritmisk tenkning hos elever. Det andre som følger av denne definisjonen er at algoritmisk tenkning er en prosess for å identifisere problemer, finne løsninger og representere disse på en strukturert måte uavhengig av konteksten eller typen problem. Dermed kan algoritmisk tenkning tolkes som en problemløsningsmetode for å tilnærme seg ulike typer problemer, ikke bare i matematikk, men også i andre sammenhenger (Bocconi et al., 2016, s.15). Wing (2011) presiserer videre at matematisk tenkning utgjør en del av algoritmisk tenkning. Ordene *problem* og *løsning* er likevel ikke begrenset til veldefinerte matematiske problemer og løsninger. Wing (2011) mener at algoritmisk tenkning omfatter også arbeidet med problemer fra den virkelige verden som har komplekse løsninger. Dette innebærer å ta hensyn til diverse betingelser, vilkår og feil som resultat av den uforutsigbare virkelige verden (Wing, 2011). Wing påpeker videre at algoritmisk tenkning, har en overføringsverdi til andre domener. Hvorvidt algoritmisk tenkning har en overføringsverdi til andre områder er forskningen imidlertid for sparsom til å kunne trekke konklusjoner (Lodi & Martini, 2021, s.883).

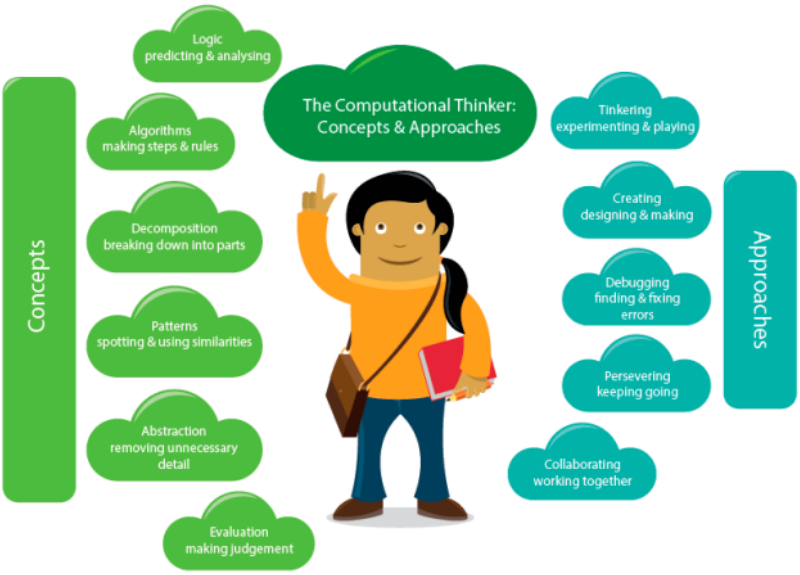
Basert på resultatene fra en litteraturanalyse gjennomført av Selby & Woollard (2013), viser det seg å være en konsensus i litteraturen om at algoritmisk tenkning er en tankeprosess som kjennetegnes av komponentene *abstraksjon* og *dekomposisjon*. Dette støttes også av resultatene fra en større litteraturanalyse gjort av Bocconi et al. (2016) og Haseski et al.(2018). Selby & Wollard (2013) konkluderte også med at *algoritmebehandling, evaluering* og *generalisering* er viktige komponenter i algoritmisk tenkning. Disse fem komponentene går også igjen i Angeli et al. (2016), Barr & Stephenson (2011), Csizmadia et al. (2015), Sanne et al. (2016), Shute et al. (2017) og Weintrop et al. (2016) sin beskrivelse av algoritmisk tenkning. Til tross for det store antallet definisjoner ser det dermed ut til at definisjonene bygger på noen karakteristiske fellestrekk. Dette er i overensstemmelse definisjonen som blir brukt av Utdanningsdirektoratet. Definisjonen er tilpasset Barefoot Computing (UK) sin modell, hvor algoritmisk tenkning er tydeliggjort som en problemløsningsmetode (Utdanningsdirektoratet, 2019, s.2). Ifølge Utdanningsdirektoratet innebærer algoritmisk tenkning å utvikle strategier og framgangsmåter for å tilnærme seg problemer på en systematisk måte. I dette ligger også en vurdering av hvilke problemer som kan løses med teknologi og hva som bør håndteres av mennesker. Figuren under er hentet fra Utdanningsdirektoratet sine sider som viser til viktige nøkkelbegrep og arbeidsmåter knyttet til algoritmisk tenkning (Figur 1).



Figur 1: Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019, s.2).

2.1.4 Rammeverk for algoritmisk tenkning: Csizmadia et al. (2015)

Det foreligger ingen generell enighet om hvordan man skal definere algoritmisk tenkning, men foreløpig ser det u til at flere hovedtrekk er sammenfallende. Flere rammeverk tar sikte på å beskrive og tydeliggjøre mange av de samme komponentene i algoritmisk tenkning. Ved å studere nøkkelbegrepene i «Den algoritmiske tenkeren» utviklet av Utdanningsdirektoratet, kan man observere at den er helt identisk med Computing at School sin modell utarbeidet av Csizmadia et al. (2015) i Figur 2.



Figur 2: The Computational Thinker (Csizmadia et al., 2015, s.8).

Computing at School (CAS) er en del av Barefoot Computing (Csizmadia et al., 2015, s.1). Ettersom den norske versjonen av modellen er en direkte oversettelse uten noen justeringer, vil denne studien ta utgangspunkt i det originale rammeverket utarbeidet av Csizmadia et al. (2015). Å ta i bruk rammeverket foreslått av Utdanningsdirektoratet (2019) vurderes til å skape utfordring i analyseprosessen, fordi det gir en lite konkret beskrivelse av komponentene som inngår i algoritmisk tenkning. Rammeverket til Csizmadia et al. (2015) er også kjent internasjonalt, og samsvarer godt med eksisterende rammeverk (Kravik et al., 2022, s.4). Rammeverk gir en detaljert og grundig beskrivelse av komponentene og arbeidsmåtene som inngår i algoritmisk tenkning. Derfor vurderes rammeverket til Csizmadia et al. (2015) som godt egnet til å vurdere kjennetegn ved algoritmisk tenkning. I rammeverket til Csizmadia et al. (2015, s.6) betegnes algoritmisk tenkning som en tankeprosess som bidrar til å løse problemer. Denne prosessen kjennetegnes av *algoritmebehandling*, *dekomposisjon*, *generalisering*, *abstraksjon og evaluering*.

Algoritmebehandling handler om å finne en løsning på et problem gjennom å tydelig definere en algoritme. Det innebærer evnen til å formulere regler og trinnvise instruksjoner på en strukturert og oversiktlig måte. Algoritmebehandling handler også om å følge og bruke kjente algoritmer, når en handling skal utføres gjentatte ganger i lignende problemer. På denne måten trenger ikke elevene å måtte starte på nytt hver gang. Ifølge Csizmadia et al. (2015, s.14) kan algoritmebehandling observeres mer konkret når elevene:

- Formulerer instruksjoner som skal følges i en gitt rekkefølge
- Formulerer instruksjoner som bruker aritmetiske og logiske operasjoner
- Skriver instruksjoner som gjentar en handling
- Formulerer instruksjoner som lagrer, flytter eller manipulerer informasjon som blir tatt imot
- Skriver et sett med instruksjoner som kan følges av et menneske eller en datamaskin.
- Danner en algoritme for å teste en hypotese

Dekomposisjon er evnen til å bryte ned et komplekst problem i mindre komponenter, slik at problemet blir mer overkommelig. Hver komponent utgjør et delproblem som kan bli forstått og løst individuelt (Csizmadia et al., 2015, s.8). Dette kan observeres når elevene:

- Deler opp et problem ved å formulere mindre delproblemer
- Løser delproblemene direkte, eller bryter disse ned igjen i mindre delproblemer
- Kombinerer løsningene på delproblemene sammen for å løse det originale problemet

Csizmadia et al. (2015, s.8) assosierer *generalisering* med å identifisere mønstre, likheter og sammenhenger. Å kjenne igjen mønstre innebærer at tidligere kunnskap og erfaringer blir aktivert. Det påpekes at spørsmål som «Likner dette på et problem jeg allerede har løst?» og «Hvordan er dette annerledes?» er viktige i generaliseringsprosessen. Følgende trekk kjennetegner generalisering:

- Å identifisere mønstre og fellestrekk
- Å tilpasse løsninger slik at de kan gjelde for tilsvarende problemer
- Å overføre ideer og løsninger mellom ulike problemområder

Abstraksjon er prosessen med å redusere overflødige detaljer slik at et problem blir mer forståelig og håndterbart (Csizmadia et al., 2015, s.7). Prosessen går ut på å identifisere og fokusere på de viktigste elementene i en situasjon ved å ignorere unødvendige detaljer for å løse problemet. Et eksempel på en abstraksjon er kartet av undergrunnen i London. Kartet inneholder kun den informasjonen som er nødvendig for å navigere i undergrunnen. Utfordringen med abstraksjon er å velge ut de relevante detaljene uten å gå glipp av viktig informasjon. Abstraksjonsprosessen innebærer blant annet:

- Forenkling ved fjerning av unødvendige detaljer
- En god og nyttig representasjon
- Representasjon som skjuler den fullstendige kompleksiteten til en situasjon
- Identifisering av relasjoner mellom abstraksjoner

Evaluering handler ifølge Csizmadia et al. (2015, s.7) om å kontrollere og sikre at en løsning er tilfredsstillende og faktisk løser problemet. Løsningen kan være en algoritme, et system eller en prosess. En rekke egenskaper ved løsningen må evalueres for å avgjøre om løsningen er god. Rammeverket til Csizmadia et al. (2015, s.15) foreslår fjorten kjennetegn på evalueringsprosessen som man kan observere i klasserommet. Noen av disse innebærer at elevene:

- Vurderer om en løsning passer dets formål
- Tester løsninger og tolker resultatene
- Vurderer effektiviteten og brukervennligheten til løsningen
- Bruker strenge argumenter for å begrunne hvorfor løsningen fungerer
- Går gjennom løsningen steg for steg

2.1.5 Algoritmisk tenkning og matematisk tenkning

Denne delen av teksten tar sikte på å klargjøre forskjellene og likhetene mellom algoritmisk og matematisk tenkning. Matematisk tenkning refererer til prosessen med å bruke matematiske ferdigheter og konsepter for å løse matematiske problemer (Sneider et al., 2014, sitert i Shute et al., 2017, s.145). En vanlig oppfatning i faglitteraturen er at matematisk tenkning består av tre deler: forståelse av matematiske konsepter, prosesser knyttet til problemløsning, og resonnering og bevisførsel (Shute et al., 2017, s.145; Wu & Yang, 2022, s.8). Som Shute et al. (2017) påpeker, hevder også Voogt et al. (2015) at algoritmisk tenkning er en egen prosess som skiller seg fra matematisk tenkning. Algoritmisk tenkning anses som en problemløsningsmetode for å løse komplekse problemer som kan anvendes på en rekke faglige områder, ikke bare innen matematikk. På den andre siden anses algoritmisk tenkning som en viktig del av matematisk tenkning, da det gir en måte å tilnærme seg matematiske problemer og finne løsninger på en strukturert og systematisk måte (Kallia et al., 2021). I en matematisk kontekst bruker elevene flere ferdigheter som inngår i algoritmisk tenkning, slik som abstraksjon, dekomposisjon og mønstergjenkjenning (Kallia et al., 2021, s.20). Mønstergjenkjenning er videre en forløper til generalisering, som ifølge Mason (1996, s.65) er grunnlaget for matematisk tenkning.

Funnene i faglitteraturen antyder at både algoritmisk tenkning og matematisk tenkning er viktige kognitive prosesser i problemløsning, men at de ikke er synonymer til tross for mange likheter (Kallia et al., 2021, s.22). Både algoritmisk tenkning og matematisk tenkning ser ut til å være viktige i matematisk problemløsning som kan støtte og forbedre hverandre. Likevel ser det ut til at algoritmisk tenkning anses som et bredere begrep enn matematisk tenkning (Barr & Stephenson, 2011; Selby & Woollard, 2013; Shute et al., 2017; Sneider et al., 2014; Weintrop, 2016; Wing, 2011). Kallia et al. (2021, s.22) argumenterer for at algoritmisk tenkning kan betraktes som et paraplybegrep som er tilpasningsdyktig og fleksibelt til ulike kontekster og bruksområder. Det er imidlertid et tema som det er gjort begrenset forskning på, og litteraturen understreker behovet for ytterligere forskning for å bedre forstå forholdet mellom algoritmisk tenkning og matematisk tenkning (Israel-Fishelson et al., 2021b).

2.2 Kreativitet

2.2.1 Å tenke utenfor boksen

Ordet kreativitet stammer fra det latinske ordet «creare» som betyr å skape eller å lage (Store norske leksikon, u.å.). Å tenke utenfor boksen regnes som et grunnleggende trekk ved kreativitet. Det innebærer evnen til å komme opp med nye ideer og løsninger for å skape noe originalt og innovativt, eller gjøre endringer på eksisterende ideer (Kaufmann, 2006, s.12; Torrance 1974, referert i Israel-Fishelson et al., 2021b, s.929; Liljedahl & Sriraman, 2006, s.18). Kreativ idéutvikling referer ofte til ukonvensjonelle måter å tenke på og kjennetegnes ved at tankene er ukontrollerte og usystematiske (Kaufmann, 2006, s.21). En situasjon som ofte blir referert til som et eksempel på dette, er når kjemikeren Kekule, halvveis sovnet foran peisen og så et drømmebilde av en slange som bet seg selv i halen. Denne opplevelsen inspirerte ham til å oppdage den såkalte benzenringstrukturen i kjemi, som viser at kjemiske strukturer kan være sirkulære og ikke bare lineære (Kaufmann, 2006, s.31). En kreativ idé kan dermed komme overraskende og uforventet, for eksempel i form av inntrykk og forestillinger. Den trenger heller ikke å basere seg på strengt rasjonelle eller logiske slutninger. Av den grunn regnes kreativ tenkning som lite krevende, sammenlignet med annen type tenkning som følger logiske slutninger og forløper på en systematisk og kontrollert måte (Kaufmann, 2006, s.27). En vanlig oppfatning tidligere var at kreativitet også var begrenset til utvalgte individer som var spesielt begavet (Bicer et al., 2020, s.1; Kaufmann, 2006, s.7; Lithner, 2008, s.267). Forståelsen av kreativitet som noe uforutsigbart og ukontrollert kan by på utfordringer i undervisningssammenheng. Hvordan kan man for eksempel legge til rette for undervisning som fremmer kreativitet, når kreative tankeprosesser er spontane og tilfeldige? Om man videre tar utgangspunkt i at kreativitet er en iboende og medfødt evne, kan det bety at evnen til å være kreativ er i liten grad avhengig av ytre faktorer. Dersom en forfølger en slik oppfattelse av kreativitet betyr det at elever ikke kan trenes opp til å være kreative tenkere. Dette står i strid med ønsket om at undervisningen skal bidra til at elevene blir mer kreative (Utdanningsdirektoratet, 2017)

Til tross for at kreativitet i lang tid har blitt sett på som en iboende evne, har denne oppfatningen imidlertid endret seg. Denne endringen kan tilskrives den økende forskningen og studiene på kreativitet (Israel-Fishelson et al., 2021b). Det har vært en sterkt økende forståelse for at sosiale miljøer både i oppvekst og i utdanningsinstitusjoner har en betydelig innvirkning på kreativiteten. Forskning i dag støtter dermed tanken om at kreativitet hos elever kan bli påvirket av omgivelsene og faglige premisser (Amabile & Pillemer, 2012; Reiter-Palmon et al., 2009 referert i Israel-Fishelson et al., 2021b, s.930; Kaufmann, 2006, s.22; Nadjafikhah et al., 2011, s.2). Kreativitet blir ikke lenger sett på som et fenomen basert på kun medfødte egenskaper og individuelle faktorer, men heller som en ferdighet som kan læres, øves opp og forbedres (Hsiao et al., 2014, referert i Israel-Fishelson et al. 2021b, s.930).

2.2.2 Matematisk kreativitet

Forskningen omkring kreativitet fikk spesielt en sterk oppblomstring i 1960-årene, og blir i dag omtalt som sentral ferdighet i det 21.århundre (Israel-Fishelson et al., 2021a, s.418; Kaufmann, 2006, s.8). I mange år har kreativitet blitt assosiert med kunstfag og litteratur (Bicer, 2020, s.1), men i dag anses kreativitet til å være en viktig del i tekniske og naturvitenskapelige fag, inkludert matematikkfaget (Bicer, 2020, s.1; Grégoire, 2016; Israel-Fishelson et al., 2021a; Israel-Fishelson et al., 2021b; Sanne et al., 2016; Selvik et al., 2016) Men hva kjennetegner matematisk kreativitet? Og hvilke arbeidsmåter er det som fremmer kreative tankeprosesser i matematikkfaget?

I litteraturen blir matematisk kreativitet ofte beskrevet som et fenomen som kjennetegnes av flere faktorer. Sriraman (2009) gjennomførte en studie av tankeprosesser til fem matematikere for å studere kreativitet i matematikkfeltet. Resultatene fra studien indikerte at den kreative prosessen til matematikerne generelt fulgte den firetrinns Gestalt-modellen som består av forberedelse, inkubasjon, opplysning og verifisering (Sriraman, 2009, s.13, egen oversettelse). Studien konkluderte med at sosial interaksjon, visuelle uttrykksformer og intuisjon var blant de vanlige kjennetegnene av matematisk kreativitet. Videre har Haylock (1987) identifisert fleksibilitet og evnen til divergent tenkning som to sentrale aspekter innen matematisk kreativitet. Med fleksibilitet refererer Haylock til evnen til å frigjøre seg fra kjente algoritmer og metoder, og til å tilnærme seg et problem på alternative måter. Divergent tenkning, på sin side, handler om å kunne generere mange forskjellige løsninger på et problem. Dette støttes også av Chamberlin & Moon (2005) som mener at matematisk kreativitet observeres når man finner en ikke-standard løsning på et problem, og at divergent tenkning er et av de mest fremtredende beskrivelsene av matematisk kreativitet. Laycock (1970) sin beskrivelse definerer også matematisk kreativitet som en evne til å analysere et gitt problem fra forskjellige perspektiver, komme opp med flere ideer og velge en passende metode for å håndtere nye matematiske problemer (referert i Nadjafikhah et al., 2011, s.286).

Matematisk kreativitet begrenser seg imidlertid ikke til ukjente og nye matematiske problemer. Å løse et gammelt problem på en ny måte kan ifølge Ervynck (1991) betraktes som et eksempel på matematisk kreativitet. Videre skiller Ervynck mellom tre ulike stadier for utvikling av matematisk kreativitet. Han påpeker at matematisk kreativitet ikke kan oppstå i vakuum. Utvikling av matematisk kreativitet må dermed foregå innenfor en viss kontekst der individets tidligere erfaringer kan hjelpe med å ta et steg i en ny retning. Det første stadiet (Stage 0) innebærer bruk av kjente regler, formler eller definisjoner, uten å nødvendigvis ha kjennskap til det underliggende teoretiske grunnlaget bak dem. I andre stadiet (Stage 1) anvender man algoritmer og prosedyrer for å utføre beregninger og matematisk operasjoner. Disse stadiene regnes ikke som kreative i seg selv, men de skal danne et passende miljø for at kreativ utvikling skal skje. Man kan trekke paralleller mellom disse to stadiene og det første trinnet i Gestalt-modellen som handler om en forberedende fase. I den forberedende fasen bruker matematikere vanligvis god tid på å innhente informasjon og undersøke konteksten rundt problemet, for at den kreative prosessen skal kunne finne sted (Sriraman, 2009, s.21). Matematiske prosedyrer som involverer memorering eller bruk av kjente løsningsalgoritmer uten å forstå sammenhengen mellom disse, stiller ifølge Stein et al. (1996, s.461) lave kognitive krav til elevene. Dette er i samsvar med Ervynck (1991, s.42) sin påstand om at kreativitet spiller en viktig rolle i avansert matematisk tenkning. Ifølge Ervynck (1991) er det i det tredje stadiet (Stage 2) at sann matematisk kreativitet kan oppstå. Framgangsmåten er ikke begrenset til bruk av kjente regler og prosedyrer, og tilnærming til problemet er basert på intuitive, plausible og rimelige hypoteser. Veien til målet innebærer å ta flere valg basert på disse hypotesene, noe som kan føre til nye innsikter eller nyskapende måter å løse problemer på. Disse stegene omtaler Ervynck (1991, s.43) som «non-algorithmic decision making», og det er i en slik prosess matematisk kreativitet kan fremtrede. Denne beskrivelsen har flere likhetstrekk med det som Haylock (1987) betegner som fleksibilitet.

Videre kan matematisk kreativitet ofte bli betraktet som et domene kun forbeholdt profesjonelle matematikere (Nadjafikhah et al., 2011, s.286). Grégorie (2016, s.27) løfter fram spørsmålet om elevene er i stand til å være kreative, om man skal forfølge tankegangen om at kreativitet innen matematikk handler om å skape nye løsninger på matematiske problemer. Sriraman (2009, s.26) hevder at studenter kan være kreative hvis de engasjerer seg i utfordrende matematiske problemer. Det argumenteres med at elevene bør få muligheten til å tilnærme seg matematiske problemer slik det gjøres blant profesjonelle matematikere. Dette innebærer å la elevene arbeide selvstendig med tilsynelatende vanskelige problemer som ikke har en klar løsningsalgoritme, og gjerne over en lengre periode for å stimulere deres matematiske kreativitet. Hensikten er ikke nødvendigvis at studentene produserer originalt arbeid som vesentlig utvider kunnskapsområdet eller åpner opp nye spørsmål, men at de kan finne uvanlige og innsiktsfulle løsninger på gitte problemer. Meningen er at elevene skal få følelsen av å oppdage matematiske sammenhenger på egenhånd. Dette kan innebære å formulere spørsmål som gjør at gamle problemer kan betraktes fra en ny vinkel. En slik tilnærming til å håndtere matematiske problemer regnes som roten til matematisk kreativitet, og gjennom denne tilnærminger vil elevene engasjere seg i prosessen med kreativ problemløsning (Nadjafikhah & Bakhshalizadeh, 2001, referert i Grégorie, 2016, s.25). Denne oppfatningen støttes også av Sanne et al. (2016, s.19) som beskriver prosessen fra idé til en ferdig løsning som en kreativ problemløsningsprosess. Oppgaver som legger til rette for at problemløseren utforsker flere muligheter under problemløsningsprosessen regnes som en forutsetning for at kreativ prosess skal finne sted. Når det er flere veier til målet og mange valg som skal tas, blir elevene oppfordret til å skape nye ideer og tenke utenfor boksen. Kaufmann (2006, s.16) mener også at det må være rom for åpen og søkende problemløsning for å stimulere til kreativitet, og framhever *reformulering* av et problem som en viktig nøkkeloperasjon i kreativ problemløsning (Kaufmann, 2006, s.41). Ved å omformulere et problem kan man betrakte problemet fra et nytt perspektiv og dermed få tilgang til nye muligheter for å løse problemet. Denne prosessen kan forklares nærmere ved å forestille at tenkning foregår i et slags problemrom (problem space), som inkluderer alle mulige løsninger innenfor en bestemt fortolkning av problemet (problemrammen). Ofte blir problemløseren værende i samme *rom* under problemløsningsprosessen. Kreativ problemløsning kjennetegnes også ved at man er mer tilbøyelig for å utforske andre rom.

2.2.3 Inkubasjon

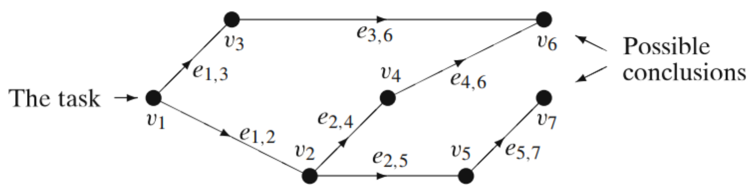
Som nevnt innledningsvis kan kreative ideer framstå som en plutselig innsikt, uten noe klar kontinuitet eller følge av logiske slutninger. Et annet nærliggende eksempel på dette er når man bevisst forsøker å finne løsning på et problem, uten suksess, og deretter setter problemet til side. Plutselig ramler løsningen ned i hodet på oss, gjerne når vi holder på med helt andre ting enn å arbeide med det aktuelle problemet (Kaufmann, 2006, s.43). Denne pausen fra problemet kalles *inkubasjon.* Inkubasjon regnes som en viktig nøkkeloperasjon i kreativ problemløsing (Kaufmann, 2006; Sriraman, 2009). Hvorvidt inkubasjonsfenomenet er reelt har vært et mye omdiskutert tema (Kaufmann, 2006, s.45), men funn fra nyere forskning støtter imidlertid teorien om at inkubasjon er en viktig del i den kreative prosessen i matematisk problemløsning (Sriraman, 2009). Det finnes to hovedforklaringer på dette fenomenet som blir beskrevet av Kaufmann (2006, s.46). Den første teorien foreslår at det skjer en aktiv bearbeiding av problemet på det ubevisste plan i inkubasjonstiden, mens man på det bevisste plan ikke har tilgang til disse prosessene. Denne teorien støttes også av Sriraman (2009, s.23) sin studie. Ifølge den andre teorien hjelper inkubasjonstiden å se problemet fra et nytt perspektiv. Om man tilnærmer seg et problem fra en uheldig vinkel, kan man blir fiksert på det samme sporet. Ved å ta pause fra problemet får man avstand fra denne fikseringen. Dette kan hjelpe med å betrakte problemet fra andre innfallsvinkler, noe som kan føre til mer fruktbare løsninger. Denne forklaringen er analog med Laycock (1970) sin beskrivelse av matematisk kreativitet som evnen til å tilnærme seg et gitt problem fra forskjellige perspektiver. Uansett hvilken teori man tar utgangspunkt i, vil i så fall konklusjonen fremdeles den samme: inkubasjonstiden kan ha en potensiell effekt på den kreative prosessen.

2.2.4 Rammeverk for å observere kreative resonnement: Lithner (2008)

Til tross for mange ulike definisjoner, er det enighet i litteraturen om at den matematiske kreative prosessen oppfyller noen sentrale kriterier (Israel-Fishelson et al., 2021a, s.418). Disse kriteriene innebærer evnen til å generere et stort antall ideer, utvide en eksisterende idé, og unngå fikseringer på en bestemt løsningsstrategi (Bergqvist & Lithner, 2005, s.8; Israel-Fishelson et al., 2021a, s.418). Kjennetegne på matematisk kreativitet blir betraktet på en mer systematisk måte innenfor rammeverket til Lithner (2008). Dette rammeverket har blant annet blitt brukt i prosjektet «Elevstrategier» som er en del av EU-prosjektet KeyCoMath (navnet står for Developing Key Competences by Mathematics Education) (Andersen & Brun, 2015, s.2).

Derfor anses rammeverket til å ha høy validitet og dermed godt egnet for å observere kreative framgangsmåter i denne studien.

Lithner (2008, s.257) påpeker at begrepet *resonnement* ofte brukes i matematikken med en antagelse om en felles forståelse av hva begrepet innebærer. I rammeverket til Lithner blir resonnement definert som en sekvens av tanker som resulterer i at man produserer påstander og trekker konklusjoner i problemløsning. Et resonnement kan både referere til en tankeprosess, resultatet av den eller begge deler. Et resonnement trenger heller ikke nødvendigvis å basere seg på formell logikk. Påstandene og konklusjonene kan til og med være feilaktige, så lenge det er noe form for fornuftige grunner (for den som resonnerer) som støtter det. En resonneringssekvens kan visualiseres som en sti i en graf (Figur 3). Hvert hjørne (v_n) representerer ulike deloppgaver en elev befinner seg ved. Kantene ($e_{n,m}$) mellom hjørnene representerer strategivalg som eleven tar. For hvert strategivalg blir ny kunnskap dannet, eller tidligere forkunnskaper blir hentet fram. Strategivalget fører til et nytt hjørne som representerer en ny deloppgave. Hver gang eleven beveger seg til et nytt hjørne, blir oppgaven delvis løst og en ny deloppgave oppstår. Denne framgangsmåten kan føre til forskjellige stier, avhengig av de ulike strategivalgene eleven tar. Lithner (2008) påpeker at det finnes alltid resonnementer i form av argumenter bak ethvert strategivalg, men at dybden i argumentasjonen kan variere. Videre kan de ulike stiene føre til like eller ulike svar. I lys av Lithner (2008) sitt rammeverk er *svaret* definert som konklusjonen på det en oppgave etterspør, mens *løsningen* er veien frem til konklusjonen. Videre i denne studien blir derfor ordet løsning brukt synonymt med framgangsmåte.



Figur 3: Resonneringssekvenser representert som stier i en graf (Lithner, 2008, s.258).

Lithner (2008) skiller videre mellom to typer matematiske resonnement: *kreativ matematisk fundert resonnering* (CMR, for Creative mathematically founded reasoning) og *imitativ resonnering* (IR, for Imitative Reasoning). Mer konkret oppfyller CMR disse kriteriene:

i) Originalitet: En ny (for den som resonnerer) resonneringssekvens blir skapt, eller en glemt sekvens blir gjenkalt
ii) Plausibilitet: Det finnes argumenter som begrunner strategivalget og/eller strategiimplementeringen som begrunner hvorfor konklusjonen er sann eller troverdig.
iii) Matematisk forankring: Argumentene er forankret i matematiske egenskaper og løsningen er verifisert med matematikk.
iv) Fleksibilitet: eleven fikserer seg ikke/låser seg ikke til en løsningsmetode

Resonneringen som ikke er kreativ betegnes som IR. Det innebærer å bruke memorerte prosedyrer, kjente mønstre eller etablerte metoder for å løse et problem. Denne typen resonnement kjennetegnes ved at strategien er basert på at eleven kan gjenkalle et fullstendig svar eller en løsningsalgoritme umiddelbart. Tilsvarende innebærer strategiimplementeringen utelukkende at eleven bare noterer svaret eller følger den gjenkalte løsningsalgoritmen. Lithner (2008) påpeker at all løsning av oppgaver bygger delvis på IR, men som en overordnet strategi er den bare begrenset til noen oppgaver. For eksempel bruker elevene IR når skal gjenkalle hvor mange dl utgjør en liter eller når de gjengir hvert steg i en bevisføring. Denne type resonnement beskrives gjerne som overfladisk, da det legges lite tanke i hvordan man kommer fram til svaret, i tillegg til mangel på nødvendig matematisk forankring (Lithner, 2008, s.262).

Videre i rammeverket følger ytterligere underinndelinger av IR. Ettersom denne studien kun er begrenset til å undersøke hvorvidt elevenes framgangsmåter er av kreativ karakter, vil disse derfor ikke bli betraktet, men anbefales for videre lesing.

3 Metode

Dette kapittelet beskriver og begrunner de metodiske tilnærmingene og valg som ble foretatt i løpet av forskningsprosessen. For å besvare problemstillingen: *Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?* ble det gjennomført en kvalitativ studie som er forankret i et fenomenologisk og hermeneutisk vitenskapssyn. Videoopptak av elevsamtaler, skjermopptak av elevarbeid på datamaskinen, og besvarelser på papir utgjør studiens empiri. Kapittelet presenterer 1) studiens vitenskapssyn, 2) metodisk tilnærming 3) beskrivelse av datainnsamlingsprosessen, 4) analysemetode, 5) studiens kvalitet og 6) etiske retningslinjer.

3.1 Studiens vitenskapssyn

Forskning handler om å skaffe seg pålitelig informasjon om verden rundt oss (Christoffersen & Johannessen, 2012, s.19). Et vitenskapssyn er fortolkningsrammen som danner et utgangspunkt for den forståelsen forskeren utvikler i løpet av forskningsprosessen (Thagaard, 2013, s. 14). Fenomenologisk vitenskapssyn kjennetegnes ved at man er opptatt av hvordan fenomener og situasjoner oppleves (Tjora, 2020, s.258). En sentral tanke innen fenomenologien er å forstå fenomener gjennom menneskene som studeres, og å beskrive omverdenen slik den erfares av dem (Thagaard, 2013, s.20). Dette skyldes en underliggende antagelse om at virkeligheten er slik mennesker oppfatter den (Christoffersen & Johannessen, 2012, s.99; Kvale & Brinkmann, 2009, referert i Thagaard, 2013, s.40;). Hermeneutisk vitenskapssyn fokuserer videre på tolkninger av fenomener og menneskers handlinger (Thagaard, 2013, s. 41). Dette innebærer å tillegge fenomener og handlinger en spesiell mening gjennom å utforske et dypere og underliggende meningsinnhold enn det som umiddelbart er innlysende (Thagaard, 2013, s. 41). Et eksempel på en hermeneutisk tilnærming vil være å gi en *tykk* beskrivelse av det som observeres og om hva de personene som blir studert kan ha ment med sine handlinger. Hermeneutikk legger vekt på at det ikke finnes en absolutt eller ordentlig sannhet, men at fenomener og handlinger kan tolkes på forskjellige måter, avhengig av forskerens forforståelser (Christoffersen & Johannessen, 2012, s.22). Dette betyr at tolkningene kan variere og at det ikke nødvendigvis finnes en riktig tolkning. Ifølge Geertz (1973), vil overbevisningskraften avhenge av hvordan forskeren argumenterer for at en tolkning er den mest korrekte gitt en kontekst, noe som gjerne vil være basert på tidligere forskning (referert i Thagaard, 2013, s.41).

I korte trekk kan man si at fenomenologisk vitenskapssyn handler om å beskrive og forstå den umiddelbare erfaringen, mens hermeneutisk vitenskapssyn handler om å tolke og forstå betydningen bak handlinger og erfaringer i en mer generell forstand. I denne studien legges elevenes opplevelser, handlinger og erfaringer til grunn for analysen, som deler av en mer omfattende helhet. Derfor har jeg valgt en kombinasjon av fenomenologisk og hermeneutisk vitenskapssyn, fordi disse supplerer hverandre på en hensiktsmessig måte i konteksten av denne studien.

3.2 Metodisk tilnærming

For å frembringe pålitelig kunnskap og informasjon om fenomenet som studeres trenger man redskaper som tjener dette formålet. En forskingsmetode er en samlebetegnelse på alle midler som kan hjelpe å forstå et fenomen (Bjørndal, 2017, s.30). I hovedsak dreier disse metodene seg om å følge en bestemt vei og har som mål å få informasjon om et fenomen, og hvordan denne informasjonen skal analyseres og tolkes (Christoffersen & Johannessen, 2012, s.16). Det kan være vanlig å skille mellom *kvantitative* og *kvalitative* forskningsmetoder. Ifølge Bjørndal (2017, s.122) er det viktig å vurdere hvilken metode som best egner seg for å besvare den aktuelle problemstillingen. Både Bjørndal (2017) og Christoffersen & Johannessen (2012) påpeker at et skille mellom kvantitative og kvalitative metoder er hvordan de forholder seg til tall. I anvendelsen av kvantitative metoder, fokuserer man ofte på en presis tallfesting av data fra et større utvalg. Dette gir et godt utgangspunkt for generalisering. Ved bruk av kvalitative metoder er man derimot mindre opptatt av streng tallfesting. I stedet søker man å oppnå en dypere forståelse av fenomenet som undersøkes basert på et begrenset utvalg. Kvalitative metoder gir også mer rom for ustrukturerte og fleksible tilnærminger, noe som tillater spontanitet, tilpasninger og endringer i takt med økt innsikt i det studerte fenomenet.

I denne studien ønsker jeg å studere kjennetegn ved algoritmisk tenkning og kreative framgangsmåter i elevers arbeid med matematikkoppgaver/matematisk problemløsning. Ettersom fortolkninger av elevenes ytringer, handlinger og opplevelser har en sentral plass i denne studien, ble valget av en kvalitativ metode ansett som mest hensiktsmessig. Den kvalitative metoden gir gode muligheter for relevante tolkninger av data og rike beskrivelser av et begrenset utvalg elever, noe som kan gi meg en dypere innsikt i og større forståelse for elevenes tenkning. I tillegg knyttes kvalitative metoder til fenomenologi og hermeneutikk, noe som er i samsvar med studiens vitenskapssyn (Thagaard, 2013, s.14).

3.2.1 Videopptak

Kvalitative metoder innebærer en rekke datainnsamlingsmetoder som gjenspeiler den virkeligheten som undersøkes. En av disse er bruk av videoopptak (Christoffersen & Johannessen, 2012, s.71). Videopptak sikrer dokumentasjonen av både bevegelse, kroppsspråk og tale. Valg av en slik datainnsamlingsmetode gir dermed mer detaljert og nøyaktig beskrivelse av situasjonen på en mer oversiktlig måte. En annen fordel ved bruk av videopptak er at det gir muligheten til å observere situasjonen flere ganger, noe som uerfarne observatører kan ha nytte av (Bjørndal, 2017, s. 126).

Bruk av video kan imidlertid påvirke deltakernes atferd. Filmingen kan ha en skremmende effekt og påvirke deltakernes vilje eller evne til å gi informasjon, spesielt dersom sensitiv informasjon tas opp (Christoffersen & Johannessen, 2012, s.71). Det kan være flere faktorer som spiller inn. Thagaard (2013, s.86-87) beskriver at frykten for eksponering er en faktor som kan påvirke dem som blir observert. Noen kan være bekymret for hvordan de blir oppfattet av andre, i frykt for å bli dømt eller kritisert basert på hva de sier eller hvordan de opptrer. Andre kan bli nervøs av å være foran kamera, spesielt om dette er en uvant situasjon for dem. En annen ulempe som Christoffersen & Johannessen (2012) påpeker ved bruk av video, er at det er mer tidskrevende å bearbeide innsamlede data. Det kan dermed være vanskeligere å presentere data på en oversiktlig og ryddig måte.

I denne studien benyttet jeg meg av videopptak av deltakernes pc-skjerm for å samle inn data. Før selve datainnsamlingen ble deltakerne bedt om å opprette og starte et møte i videotelefoniprogrammet/videokonferanseplattformen Zoom. På denne måten fikk jeg en nøyaktig gjengivelse av deltakernes samtaler hvor det var enkelt å avgjøre hvilke ytringer som tilhørte hvem. Videopptak av pc-skjermen ga også et detaljert bilde av hvordan digitale verktøy ble brukt til å løse til å løse oppgavene som elevene fikk utdelt. Ved å benytte denne innsamlingsmetoden fikk jeg en helhetlig oversikt over situasjonen og muligheten til å analysere og tolke dataene flere ganger i ettertid. Arkene som elevene noterte på, ble også samlet inn og brukt i analysen.

3.2.2 Observasjon

Observasjon er en datainnsamlingsmetode innen kvalitativ forskning. følge Bjørndal (2017, s.33) kan observasjon deles inn i observasjon av *første* og *andre* orden. Observasjon av første orden kjennetegnes av at forskerens primæroppgave er å observere og ikke delta i aktiviteten som observeres. På denne måten opprettholder forskeren en posisjon som observatør fra sidelinjen og har betydelig avstand til deltakerne i studien. Christoffersen & Johannessen (2012, s.69) omtaler en slik observasjonsrolle som *ikke-deltakende observatør.* I studier der det er grunn til å tro at forskerens deltakelse kan betydelig påvirke resultatene eller forstyrre situasjonen, kan det være mer hensiktsmessig å innta en passiv deltakerrolle. Thagaard (2013, s.79) understreker at det er viktig at forskeren tilstreber å gjøre seg lite bemerket slik at forskeres nærvær skal ha minst mulig påvirkning på undersøkelsessituasjonen. Forskeren deltar i liten grad i den ordinære samhandlingen mellom deltakerne, men kan engasjere seg i samtaler for å besvare eventuelle spørsmål. Ved observasjon av *andre orden* tar læreren en aktiv deltakerrolle i situasjonen som blir observert (Bjørndal, 2017, s.33). Observasjon av andre orden blir sett på som en sidestilt oppgave med undervisning eller veiledning. Forskeren omtales da som den *deltakende observatør* (Christoffersen & Johannessen, 2012, s.69). Det vil si at forskeren er en del av miljøet som studeres og engasjere seg i stor grad med deltakerne. En slik observatørrolle vil ha en større påvirkning på situasjonen og dermed også resultatene.

Jeg fikk ikke muligheten til å møte deltakerne før datainnsamlingen ble gjennomført, og hadde dermed ingen tidligere kjennskap eller relasjon til dem. For at situasjonen skulle oppleves mest naturlig, valgte jeg derfor å gjennomføre en observasjon av første orden ved å tre inn i rollen som ikke-deltakende observatør. Jeg var til stede under datainnsamlingen for å besvare praktiske spørsmål, men holdt ellers avstand til situasjonen. På denne måten håpte jeg at mitt nærvær i minst mulig grad skulle føre til usikkerhet eller ubehag blant deltakerne.

3.3 Beskrivelse av datainnsamlingsprosessen

3.3.1 Utvalg av deltakere

Et avgjørende aspekt i kvalitative metode er utvelgelse av deltakerne. I denne sammenhengen er *utvalgsstrategi* og *utvalgsstørrelse* to viktige faktorer å ta hensyn til.

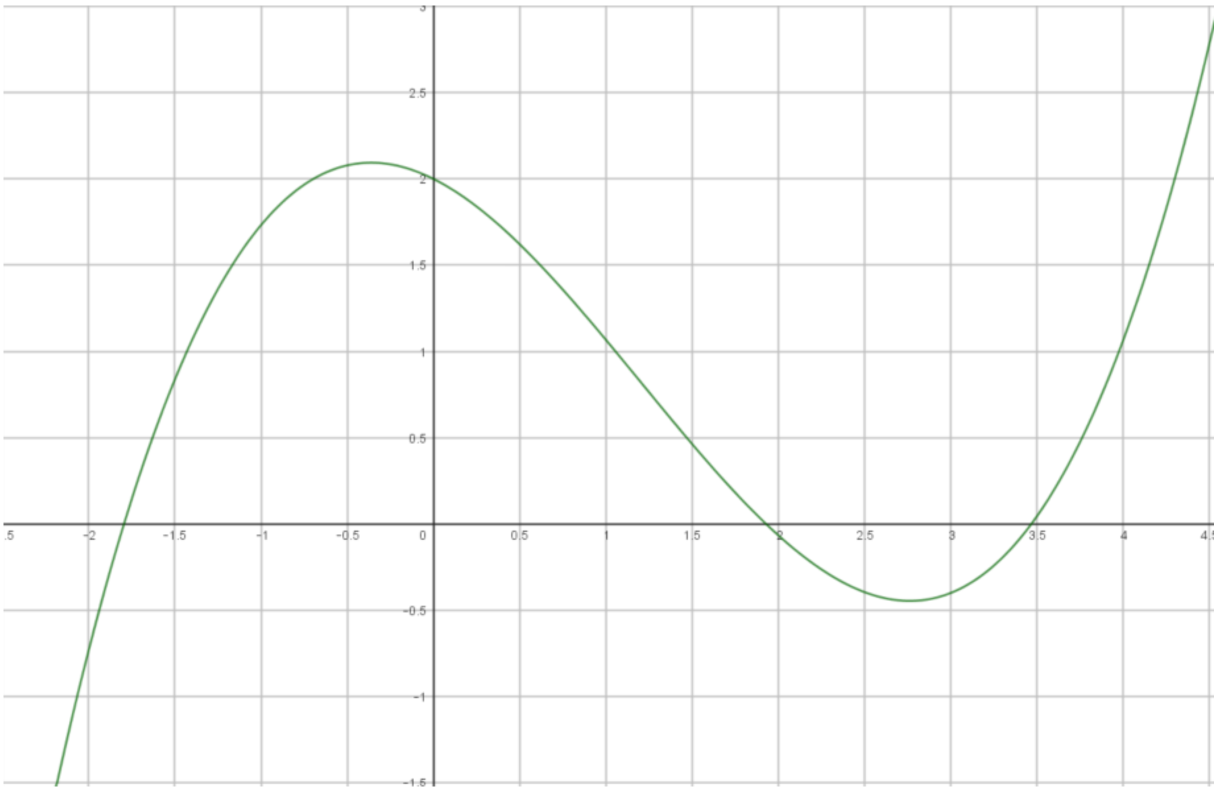
Hvilken utvalgsstrategi som brukes er avhengig av hva som er problemstillingen og hva som er hensiktsmessige å gjennomføre (Christoffersen & Johannessen, 2012, s.52). Ifølge Tjora (2020, s.41) er *kriterieutvelgelse* en passende metode når formålet er å undersøke deltakernes erfaringer og opplevelser. For å kunne besvare problemstillingen i denne studien måtte deltakerne oppfylle visse kriterier. Elevene måtte først og fremst ha matematikk som fag på skolen. Det var også ønskelig at elevene var muntlig aktive og ikke var redd for å tenke høyt, slik at det jeg kunne hente inn tilstrekkelig med data. Et annet kriterium var at elevene måtte ha matematikkunnskaper som tilsvarte 1T eller høyere nivå. Årsaken til dette var utforming av oppgaven som elevene fikk utdelt. Denne blir presentert i delkapittel 3.3.2. I denne studien ble derfor deltakerne valgt ut ved kriteriebasert utvelgelse.

Ifølge Christoffersen & Johannessen (2012, s.50) finnes det teoretisk sett ingen øvre eller nedre grense for utvalgsstørrelsen, men i praksis er det vanlig med et utvalg på rundt 10 deltakere i mindre studier. Dersom deltakerne er relativt lik hverandre på flere kriterier, noe som er tilfelle i denne studien, trenger forskeren færre deltakere. Dette skyldes at kvalitativ forskning innebærer en dypere undersøkelse av relativt få strategisk utvalgte enheter (Tjora, 2020, s.40). En annen retningslinje for å bestemme utvalgsstørrelsen som ble vurdert før datainnsamlingen, er at antall deltakere ikke bør være større enn det som er mulig å analysere grundig, da dette kan være tidskrevende og omfattende arbeid (Thagaard, 2013, s.65). Eksisterende sosialt nettverk ble benyttet for å komme i kontakt med en lærer på en videregående skole som var villig til å hjelpe med datainnsamlingsprosessen. Deltakerne i studien var ni videregående elever som hadde R1 matematikk som fag. Elevene ble delt inn i fire grupper, hvor tre av gruppene bestod av to elever, og den siste gruppen bestod av tre elever.

3.3.2 Utforming og begrunnelse av oppgaven

Oppgavene som elevene fikk utdelt (Figur 4) har fokus på algoritmisk tenkning i problemløsning og er inspirert av et undervisningsopplegg utviklet av Tom Jarle Christiansen og Rune Mathisen (Christiansen & Mathisen, 2020). NDLA (Nasjonal digital læringsarena) har også laget en oppgave med utgangspunkt i det samme undervisningsopplegget (Skurdal, 2020).

1) Gjett-tall aktivitet. Skriv ned en algoritme for hvordan dere går fram for å gjette tallet.



2) Hvordan kan vi finne nullpunktene til denne grafen? Formuler en algoritme, og skriv den ned med ord.

Side 1 av 2

3) Her er funksjonsuttrykket til grafen:

$$f(x) = \frac{1}{6}x^3 - \frac{3}{5}x^2 - \frac{1}{2}x + 2$$

Test ut algoritmen som dere formulerte i oppgave 2) for å finne f(x) = 0. Dere kan gjerne bruke GeoGebra/kalkulator/programmering som hjelpemidler.

Side 2 av 2

Figur 4: Oppgavene som elevene fikk utdelt

I oppgavene får ikke elevene oppgitt en klar algoritme for hvordan de skal gå fram. Elevene får først i oppgave å gjennomføre en aktivitet kjent som Gjett-tall leken. Dette er en aktivitet som innebærer å gjette på et tall som en annen person tenker på. Elevene blir oppfordret til å utforske og være kreative for å komme fram til en effektiv løsningsalgoritme. Dette innebærer at elevene må foreta egne valg, diskutere og reflektere over deres løsninger. Videre er det begrenset hvor mange funksjonstyper vi kan regne ut nullpunkter til ved å bruke formler. Det er vanlig at elevene lærer abc-formelen for å løse andregradslikninger og finne eventuelle nullpunkter (Kristensen & Kirfel, 2022, s.77). For polynomer med en grad større eller lik 5 er det ingen generell formel som kan brukes til å bestemme nullpunktene. Som et resultat må man ofte ty til kreative metoder. Én måte å gjøre dette på er ved hjelp av numeriske metoder. Numeriske metoder ble innført sammen med læreplanen LK20 i en rekke matematikkfag (Utdanningsdirektoratet, 2020b). Dette åpner opp nye muligheter for å arbeide med funksjoner på en utforskende måte. Halveringsmetoden er et eksempel på en numerisk metode som brukes for å finne en tilnærmet verdi for nullpunktene til en funksjon. Halveringsmetoden går ut på å løse en likning på formen $f(x) = 0$. Hvis f er en funksjon med sammenhengende graf i intervallet $[a, b]$, der $f(a)$ og $f(b)$ har motsatt fortegn, kan nullpunkt til f avgjøres med en viss nøyaktighet ved å anvende algoritmen nedenfor (Borgan et al., 2021, s.44):

1. Finn midtpunktet i intervallet $[a, b]$: $m = \frac{a+b}{2}$
2. Undersøk om $f(m) \approx 0$. Hvis det er tilfellet, avslutt. Hvis ikke, gå til trinn 3.
3. Undersøk om $f(a) * f(m) < 0$. Hvis det er tilfellet, ligger nullpunktet i intervallet $[a, m]$ og vi gjentar algoritmen for dette intervallet. Hvis ikke, gjentar vi algoritmen for intervallet $[m, b]$.
4. Gjenta algoritmen til svaret er nøyaktig nok.

En annen grunn til at oppgave 1) ble gitt, er at mange bruker prinsippet fra halveringsmetoden for å gjette riktig tall på færrest mulig forsøk, ved å halvere tallmengden i to. Tanken bak dette var å legge opp til at elevene forhåpentligvis skulle klare å overføre denne ideen for å løse oppgave 2). Elevene blir oppfordret til å se etter sammenhenger og gjenkjenne igjen fellestrekk mellom oppgavene. Det krever både evne til å generalisere og abstrahere for å dekomponere problemet og formulere en løsningsalgoritme. Oppgave 3) gir også elevene mulighet til å teste ut algoritmene de utvikler ved hjelp av digitale verktøy.

Oppgavene ble diskutert både med faglærer og veileder. De anses til å ha en moderat vanskelighetsgrad. Grunnen til at oppgavene ble utformet på denne måten var å legge til rette for at elevene fikk bryne seg over oppgavene over lengre tid. Som nevnt tidligere peker teorien på at algoritmisk tenkning og kreativitet kan komme til uttrykk når elevene må fikle, utforske problemet og bruke litt tid på det. På bakgrunn av overnevnte regnes derfor disse oppgavene som passende for å få fram elevenes algoritmiske tenkning og kreative framgangsmåter.

3.3.3 Gjennomføring av datainnsamlingen

I starten av datainnsamling ble elevene informert om hva studien gikk ut på og fikk deretter utdelt samtykkeskjema (Vedlegg 2).

For å samle inn datamaterialet ble videoopptak av pc-skjermen i hver gruppe benyttet, slik som det ble beskrevet i delkapittel 3.2.1. Elevene fikk utdelt oppgavene i papirform sammen med notatark for å gjøre beregninger, og skrive ned eventuelle løsninger på oppgavene. Disse notatene ble samlet inn på slutten av datainnsamlingen. I tillegg fikk elevene muligheten til å benytte seg av hjelpemidler som inkluderer skrivesaker, kalkulator, GeoGebra og programmeringsplattformer. Elevene fikk også beskjed om at bruk av internett ikke var tillatt.

Under datainnsamlingen ble elevene delt inn i grupper etter elevenes ønske. På denne måten bestod gruppene av elever som hadde en relasjon til hverandre. Det kan tenkes at elevene som har en god relasjon til hverandre kan bidra til å skape en trygg og åpen dialog der elevene føler seg komfortabel. For å unngå å forstyrre elevene, forsøkte jeg så godt det lot seg gjøre å posisjonere meg utenfor deres synsfelt. På denne måten kunne jeg være tilgjengelig for assistanse, uten å distrahere dem fra arbeidet med oppgavene. Ved å velge denne strategien, kunne jeg sørge for et mer uforstyrret arbeidsmiljø. Svakheten med denne strategien er at elevene kan ha lettere for å bli distrahert fra selve oppgaven og inngå i andre aktiviteter som ikke er relevant for studien. For å sikre at elevene jobbet effektivt, valgte jeg å følge den ordinære undervisningstiden, slik at elevene fikk en pause sammen med resten av klassen.

3.4 Analyse

Den kvalitative analysen har som mål å gjøre det mulig for en leser av forskningen å få økt kunnskap om saksområdet det forskes på, uten selv å måtte gå gjennom de data som er generert i løpet av studien (Tjora, 2020, s.195). Å analysere betyr å dele noe oppi i biter eller elementer. Det forskeren undersøker, betraktes som sammensatt av enkelte bestanddeler, og målet er å avdekke et budskap eller en mening, å finne et mønster i datamaterialet. For å sikre systematikk i analysearbeidet bestod analyseprosessen i denne studien av fire steg: 1) databearbeiding i form av transkribering, 2) databehandling ved hjelp av koding, 3) kondensering og 4) sammenfatning. Disse er basert på Malterud (2003) og Berg (2001) sine framstillinger (referert i Christoffersen & Johannessen, 2012).

3.4.1 Transkribering

Transkribering er prosessen med å bearbeide talespråk og visuell data til skriftlig tekst (Thagaard, 2013, s.186). På denne måten får man som forsker en mer strukturert og oversiktlig framstilling av datamaterialet som er bedre egnet for videre analyse. Samtidig er det viktig å være klar over at det ikke finnes en sann og objektiv oversettelse fra muntlig til skriftlig språk (Kvale og Brinkmann, 2009, s.194). Spørsmålet blir dermed ikke hvorvidt transkripsjonen er korrekt, men hva som vil være en nyttig transkripsjon i forhold til problemstillingen. I denne studien valgte jeg å transkribere ordrett det som ble uttrykt verbalt fordi jeg er interessert i å analysere elevenes tenkning. For mange detaljer kan gjøre transkripsjonen uoversiktlig og vanskelig å lese (Bjørndal, 2017, s.103). Derfor ble ytringer som ikke hadde en faglig relevans markert med []. Relevante kroppslige bevegelser slik som når elevene notere, gjorde beregninger og brukte digitale verktøy ble også inkludert i transkripsjonen. Skjermbilder fra skjermopptak er også en del av det transkriberte datamaterialet, og de er tatt med i transkripsjonen når det ble ansett som relevant. Elevens identitet ble anonymisert ved at elevene ble kalt for Elev x, som angir en ytring fra elev nummer x. Det ble også vurdert å bruke normalisert transkripsjon for å ytterlig anonymisere elevene. Dette innebærer at dialekter blir transkribert til bokmål (Tjora, 2020, s.174). Transkripsjonen bestod totalt av fire dokumenter for hver gruppe.

Tabell 1: En detaljert oversikt over transkripsjonskonvensjoner.

..	To punktum angir en pause med varighet på mindre enn tre sekunder.
…	Tre punktum angir en pause med varighet på mer enn tre sekunder.
()	Angir en beskrivelse av en kroppslig handling som er relevant for transkripsjonen.
(?)	Parentes med spørsmålstegn angir en ytring som er utydelig og ikke lar seg transkribere.
[]	Tomme klammer angir en irrelevant ytring.
\	Skråstrek angir at en ytring blir avbrutt.
ORD	Store bokstaver angir lyder som er spesielt sterke i forhold til talen rundt.

3.4.2 Koding av transkripsjonen

En kode er et ord eller en kort setning som danner en oppsummerende eller fremtredende merkelapp for en bestemt del av datamaterialet (Bjørndal, 2017, s.104). Kodingsprosessen handler dermed om å finne meningsbærende elementer i datamaterialet ved å foreta en systematisk gjennomgang av materialet og identifisere tekstelementer som gir kunnskap og informasjon om de hovedtemaene man undersøker (Christoffersen & Johannessen, 2012, s.101). På denne måten kan koding brukes for å trekke ut essensen i det empiriske datamaterialet og avdekke de meningsfulle utsnittene (Christoffersen & Johannessen, 2012, s.101). En deduktiv tilnærming vil innebære at forskeren bruker eksisterende koder (Christoffersen & Johannessen, 2012, s.106). I denne studien vil dette være kodene fra det sammensatte teoretiske rammeverket av Csizmadia et al. (2015) og Lithner (2008) som ble henholdsvis presentert i delkapittel 2.1.4 og delkapittel 2.2.4.

Tabell 2: koder for kjennetegn ved algoritmisk tenkning (Csizmadia et al., 2015, s.15).

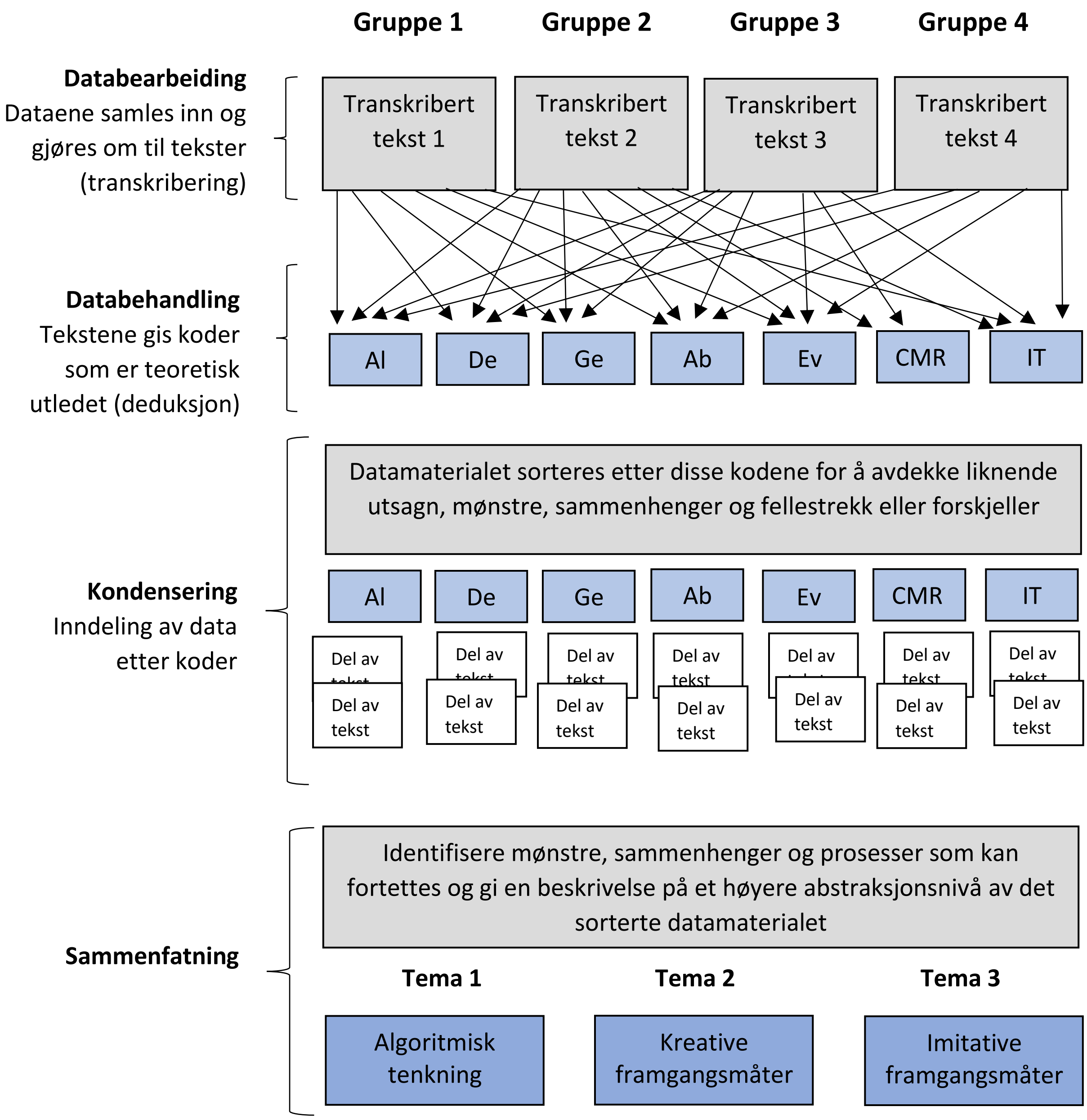
Csizmadia m. fl. (2015)	
Begrep	Kode
Algoritmebehandling	Al
Dekomponering	De
Generalisering	Ge
Abstraksjon	Ab
Evaluering	Ev

Tabell 3: Koder for imitativt og kreativt resonnement (Lithner, 2008).

Lithner (2008)	
Begrep	Kode
Kreativt matematisk resonnement	CMR
Imitativt resonnement	IR

3.4.3 Kondensering og sammenfatning

Det tredje steget tar utgangspunkt i kodingen og kalles for kondensering (Christoffersen & Johannessen, 2012, s. 104). Kondensering i denne sammenhengen betyr å redusere datamaterialet til å være mer kompakt og kortfattet. Det går ut på at forskeren først trekker ut de delene av teksten som er kodet. Deretter sorteres disse delene etter kodene for å få bedre oversikt over det reduserte datamaterialet. Å organisere kodene på denne måten kan gjøre det lettere å oppdage liknende utsagn, mønstre, sammenhenger og fellestrekk eller forskjeller (Christoffersen & Johannessen, 2012, s. 106). Den siste delen av analyseprosessen er sammenfatning. Formålet er å identifisere mønstre, sammenhenger og prosesser som kan fortettes og beskrives på et høyere abstraksjonsnivå. Gjennom sammenfatningen søker forskeren etter overordnede trekk og aspekter, som kan oppsummeres på en mer oversiktlig måte. Figur 5 viser de fire stegene i analyseprosessen.



Figur 5: Analyseprosessen fra start til slutt.

3.5 Studiens kvalitet

I dette delkapittelet diskuteres begrepene *reliabilitet* og *gyldighet* som kriterier for kvaliteten på studien.

3.5.1 Reliabilitet

Reliabilitet handler om forskningens pålitelighet og i hvor stor grad forskningen er utført på en tillitvekkende måte (Thagaard, 2013, s.193). Seale (1999) skiller mellom *ekstern* og *intern reliabilitet* (referert i Thagaard, 2013, s.202). Ekstern reliabilitet har i denne sammenheng referanse til *repliserbarhet*; det vil si i hvor stor grad en kan reprodusere studien og komme fram til samme resultat. Det er imidlertid et spørsmål om ekstern relabilitet er et relevant kriterium for denne studien, ettersom repliserbarhet er vanskelig å oppnå i kvalitativ forskning (Thagaard, 2013, s.202; Tjora, 2020, s.235). Repliserbarhet er knyttet til positivisme som er en vitenskapelig tilnærming som framhever nøytralitet og objektive fortolkninger (Tjora, 2020, s.264). Ettersom denne studien har tatt utgangspunkt i en kvalitativ tilnærming som baserer seg på fortolkninger av elevsamtaler og elevløsninger, kan ikke en fullstendig nøytralitet og objektivitet oppnås (Tjora, 2020, s.235).

En redegjøring av *intern reliabilitet* blir dermed viktig for å styrke studiens pålitelighet. Det innebærer å gi en detaljert beskrivelse av framgangsmåter ved innsamling og analyse av data, samt refleksjon over hvordan relasjonen til deltakere i prosjektet kan påvirke den informasjonen forskeren får (Thagaard, 2013, s.202). Deltakerne ble valgt ut ved et kriterieutvalg for å sikre at data som ble samlet inn var relevant i forhold til problemstillingen. For å få mest mulig pålitelig data, ble det vurdert som hensiktsmessig å inngå en ikke-deltakende observatørrolle. Ved å unngå å delta aktivt ønsket jeg å minimere min påvirkning av situasjonen, samt bevare den naturlige adferden til elevene. På denne måten kan elevenes tenkning komme til utrykk uten stor ytre påvirkning, og gi meg som forsker et mer nøyaktig bilde og forståelse av elevenes algoritmiske tenkning. Det er likevel viktig å ta i betraktning at min tilstedeværelse i rommet kan ha påvirket elevenes atferd, men ettersom jeg ikke hadde en tidligere relasjon til elevene kan det likevel tenkes at valget av en ikke-deltakende rolle ville forstyrre situasjonen minst mulig. Videopptak ble benyttet for å styrke studiens reliabilitet ettersom det gir en mer komplett og detaljert ikke-tolket gjengivelse av virkeligheten (Tjora, 2020, s.103). Dette gjør at jeg som forsker også kan se på opptak i etterkant og på denne måten kontrollere egne inntrykk og fortolkninger. Samtidig vil bruk av video i observasjonsstudier medføre store mengder data, noen som kan gjøre det krevende å trekke ut informasjon som er relevant og betydningsfull. I studien ble det derfor gjennomført en kvalitativ analyse for å sikre systematikk og framgang i analyseprosessen, slik som det ble beskrevet i kapittel 3.4. Det analyserte datamaterialet blir presentert i kapittel 4.

3.5.2 Validitet

Validitet er knyttet til tolkninger av data og handler om gyldigheten av de tolkninger forskeren kommer fram til. Det er mulig å vurdere validiteten av kvalitativ forskning med henblikk på spørsmålet om tolkningene man kommer fram til er gyldige i forhold til den virkeligheten man har studert (Thagaard, 2013, s.204).

Seale (1999) skiller mellom *intern* og *ekstern* validitet (referert i Thagaard, 2013, s.205). Intern validitet har referanse til vurderinger av fortolkninger og deres gyldighet innenfor en bestem studie. Tjora (2020, s.234) skriver at «den viktigste kilden til høy intern gyldighet er at forskningen pågår innenfor rammene av faglighet, forankret i relevant annen forskning». I denne studien tok jeg utgangspunkt i Csizmadia et al. (2015) sitt anerkjente rammeverk for å studere kjennetegn ved algoritmisk tenkning gjennom fem konkrete kjennetegn: *algoritmebehandling, generalisering, abstraksjon, dekomponering* og *evaluering*. Rammeverket til Lithner (2008) ble brukt for å videre kategorisere elevenes framgangsmåter under matematisk problemløsning som *kreative* eller *imitative.* Denne teoretiske forankringen hjelper dermed med å styrke studiens interne validitet. Imidlertid kan bruken av videopptak svekke studiens validitet, da ingen av deltakerne tidligere hadde deltatt i en liknende studie. Dette kan igjen ha påvirket elevenes atferd, noe som reiser spørsmålet om opptakene representerer en virkelighet som gir grunnlag for gyldige tolkninger (Thagaard, 2013, s.90). Det er vanskelig å fastslå i hvilken grad elevene atferd ble påvirket. Videoopptaket ble filmet direkte på elevenes datamaskin gjennom bruk av Zoom applikasjonen. I løpet av datainnsamlingen brukte elevene andre applikasjoner slik som GeoGebra og Visual Studio Code (VS Code). Når elevene benyttet seg av disse, ble Zoom applikasjonen automatisk skalert til et lite vindu i høyre hjørnet av pc-skjermen. Ettersom elevene ikke lenger var i hovedfokus, økte muligheten for at deres oppmerksomhet ikke lenger var rettet mot videoopptaket, og at elevenes naturlige atferd ble bevart til en viss grad under den matematiske problemløsningsprosessen

Ekstern validitet handler om den forståelsen som utvikles innenfor en studie, også kan være gyldig i andre sammenhenger. Som tidligere nevnt, kan bruk av kvantitative metoder gi større mulighet til å undersøke et fenomen mer generelt. Siden denne studien benytter seg av en kvalitativ metode begrenset til et lite utvalg, gir det ikke tilstrekkelig grunnlag for å generalisere studiens funn. Det man imidlertid kan diskutere er om studiens funn har en overføringsverdi til andre sammenhenger. For denne studien kan overførbarhet knyttes til gjenkjennelse. Det er mulig at leseren kan kjenne seg igjen i de tolkningene som formidles i teksten og dra paralleller mellom de erfaringene de har fra før (Thagaard, 2013, s.213). Gjenkjennelsen innebærer at tolkningene gir en dypere mening til tidligere kunnskaper og erfaringer, og samtidig utvider leserens forståelse. Leseren kan også finne noen interessante funn i studien, som kan være til nytte eller inspirasjon. På denne måten kan kunnskapen overføres og gi nytteverdi i andre sammenhenger (Postholm, 2010, s. 38).

3.6 Etiske retningslinjer

I all vitenskapelig forskning er det nødvendig at forskeren følger de etiske prinsippene som gjelder i forskningsmiljøet/forskningsdomene (Thagaard, 2013, s.24). Denne studien omfatter behandling av personopplysninger og er dermed meldepliktig. Det ble derfor søkt om tillatelse til å gjennomføre dette forskningsprosjektet gjennom RETTE, som er UiB sitt system for oversikt og kontroll med behandling av personopplysninger. Videre har etiske prinsipper blitt overholdt i henhold til De nasjonale forskningsetiske komiteers (NESH) retningslinjer for behandling av personopplysninger. Disse innebærer 1) *informert samtykke*, 2) *konfidensialitet* og 3) *konsekvenser av å delta i forskningsprosjekter* (Thagaard, 2013, s.24).

Prinsippet om *informert samtykke* går ut på at forskningsprosjekter som inkluderer personer, settes i gang bare etter deltakernes informerte og frie samtykke. De som deltar i studien skal orienteres om hva deltakelse innebærer, og at samtykke er avgitt uten ytre press. Deltakerne har også til enhver tid rett til å avbryte sin deltakelse uten at det får negative konsekvenser for dem (Thagaard, 2013, s.28). Hensikten med studien ble først formidlet av faglærer. Før selve datainnsamlingen fikk jeg selv muligheten til å presentere studiens formål for elevene, og hva deres deltakelse ville innebære. En detaljert beskrivelse av prosjektet er også uttrykt i informasjonsdokumentet som elevene fikk utdelt på papir sammen med samtykkeerklæringen.

I dokumentet går det tydelig fram hvilke behandlinger samtykke omfatter, og elevenes rettighet til å trekke seg uten å oppgi noe grunn.

Det andre prinsippet om *konfidensialitet* innebærer at alle deltakere har krav på at all informasjon de gir blir behandlet konfidensielt. Forskeren må håndtere informasjonen fra forskningsprosjektet på en måte som sikrer at deltakerens identitet forblir skjult. For å ta hensyn til dette ble alle elevene anonymisert slik det er beskrevet i delkapittel 3.4.1. I informasjonsskrivet går det også tydelig fram hvem som er behandlingsansvarlig og hvordan all datamaterialet blir behandlet. For å ytterlig ivareta elevenes konfidensialitet benyttet jeg meg av SAFE som er en løsning for sikker behandling av sensitive personopplysninger.

Det siste prinsippet er knyttet til *de konsekvensene forskningen kan ha* for deltakerne (Thagaard, 2013, s.30). Forskerens etiske ansvar i denne konteksten er å forsøke å unngå at forskningen fører til negative konsekvenser for deltakerne. En uheldig konsekvens som Thagaard (2013) skildrer er om deltakerne kjenner seg igjen i teksten når de leser resultatene av studien, så kan det å se egne utsagn skape negative reaksjoner. Elevene kan for eksempel bli provosert eller fornærmet om resultatene ikke er i overensstemmelse med deres egne forestillinger eller forventinger. For å ta hensyn til dette kommunisere jeg tydelig til deltakerne at resultatene som presenteres ville mest mulig være en generell beskrivelse av gruppen som helhet. Faglærer var heller ikke til stede under datainnsamlingen. På denne måten kunne elevene være helt sikre på at deres utsagn og arbeid ikke ville ha en negativ innvirkning på vurderingen i faget. Som følge av datainnsamlingen var elevene i tillegg borte fra den ordinære undervisning. Oppgavene som ble brukt i denne studien ble også gitt til resten av R1 elevene som ikke deltok i studien. Dette ble avtalt med faglærer for å sikre at deltakerne ikke gikk glipp av gjennomgangen av nytt fagstoff.

4 Resultat

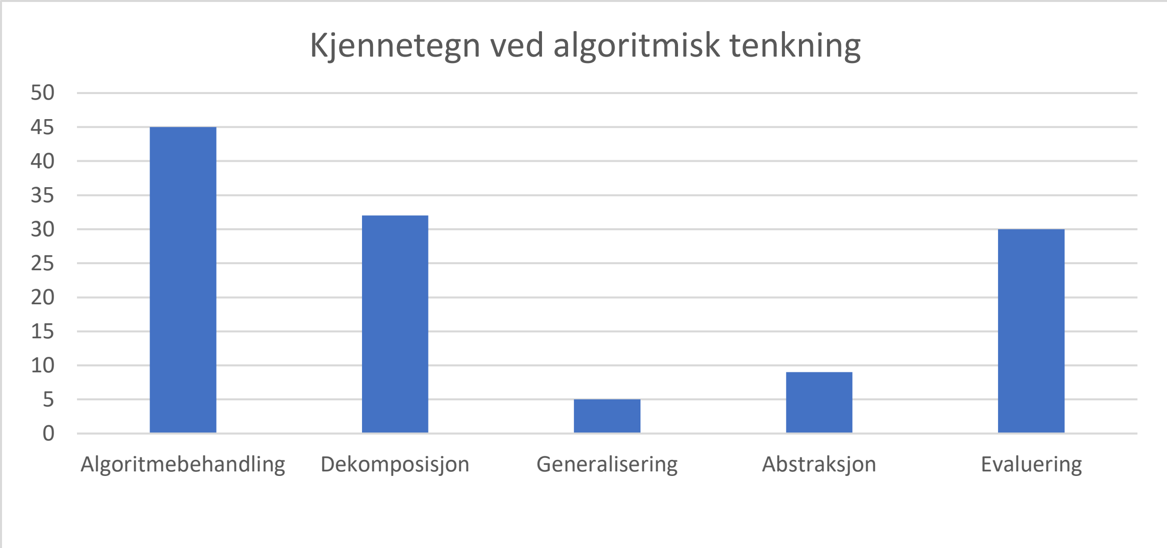
Formålet med denne oppgaven er å undersøke kjennetegn ved algoritmisk tenkning og videre vurdere om løsningene under den matematiske problemløsningsprosessen er av kreativ eller imitativ karakter. For å framlegge resultatene fra analysen på en oversiktlig måte er kapittelet delt inn i to. *Fase 1* referer til Gjett-tall aktiviteten, mens *Fase 2* refererer til oppgaven med å finne f(x) = 0 slik som det beskrevet i delkapittel 3.3.2.

I første del av kapittelet presenteres funn fra datamaterialet i henhold til Csizmadia et al. (2015) sine fem kjennetegn ved algoritmisk tenkning: *algoritmebehandling*, *dekomposisjon*, *generalisering*, *abstraksjon og evaluering*. Som følge av det omfattende analysearbeidet foreligger det et betydelig antall elevutsagn som viser til de ulike kjennetegnene ved algoritmisk tenkning. Noen resultater fra den kvalitative analysen blir derfor framstilt på en kvantitativ måte for å danne et bedre inntrykk av i hvilken grad de ulike kjennetegnene kom til uttrykk. Underveis blir utvalgte deler av elevsamtaler, enkelte elevutsagn og deres tilhørende arbeid presentert.

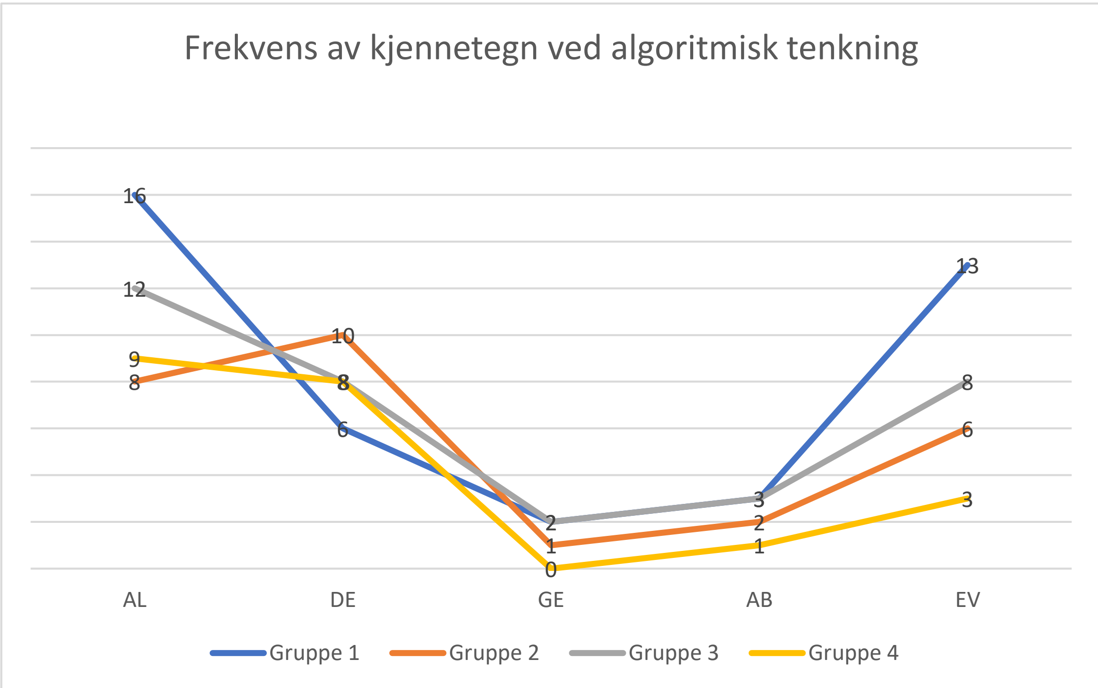
Andre del av kapittelet presenterer funn fra elevenes framgangsmåter ut fra Lithner (2008) sitt rammeverk om *kreative* og *imitative* resonnement.

4.1 Algoritmisk tenkning

Alle kjennetegn ved algoritmisk tenkning beskrevet i rammeverket til Csizmadia et al. (2015) ble observert i denne studien (Figur 6). Studiens funn viser at kjennetegnene *algoritmebehandling*, *dekomposisjon* og *evaluering* kom til uttrykk flest ganger. Kjennetegnene *abstraksjon* og *generalisering* ble observert i mindre grad. Hvordan forekomsten av de ulike kjennetegnene ved algoritmisk tenkning varierer mellom gruppene er vist i Figur 7. I Gruppe 1 og Gruppe 3 ble algoritmebehandling og evaluering observert flest ganger. Gruppe 1 og Gruppe 4 skiller seg ut når det gjelder evnen til å evaluere. Antall observasjoner av dette kjennetegnet er betydelig høyere i Gruppe 1 i forhold til resten, mens i Gruppe 4 er dette tallet betydelig lavere.



Figur 6: Totalt antall observerte kjennetegn ved algoritmisk tenkning.



Figur 7:Linjediagram av hyppigheten av kjennetegn ved algoritmisk tenkning i gruppe.

4.1.1 Algoritmebehandling

Kjennetegnet algoritmebehandling kom til utrykk gjentatte ganger i alle gruppene. Totalt ble det observert fire ulike løsningsalgoritmer i Fase 1 som innebar bruk av flere ferdigheter som går under algoritmebehandling. Disse er oppsummert i tabellen under:

Tabell 4: Algoritmene observert i Fase 1 som innebar bruk av algoritmebehandling.

1)	Gjenkalle prinsippet fra halveringsmetode umiddelbart
2)	Lineær søkealgoritme (Gjette på a, a+1, a+2,.., b i et intervall [a,b] helt til tallet er funnet)
3)	Systematisk eliminasjon ved kategorisering av tall
4)	Komme fram til prinsippene fra halveringsmetoden underveis/uten tidligere kjennskap til den

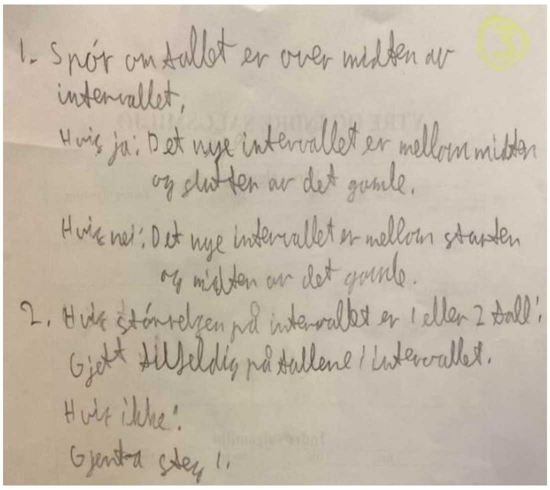
Algoritmebehandling handler blant annet om å følge og bruke kjente algoritmer, når en handling skal utføres gjentatte ganger i lignende problemer (Csizmadia et al., 2015). Dette ble observert i Gruppe 1. Elevene benyttet seg umiddelbart av en kjent metode, nemlig *halveringsmetoden*, for å løse problemet i Fase 1. Elevene fulgte prinsippene fra halveringsmetoden for å stille spørsmål om tallet hadde en høyere eller lavere verdi enn midten av et tallintervall. Her er samtalen som viser til dette:

Utdrag 1

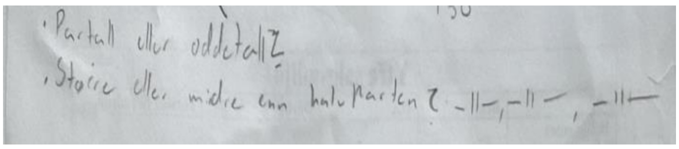
Elev 2	Vil du tenke på et tall?
Elev 1	Okey, jeg tenker på et tall
Elev 2	Fra 1 til 20? (Elev 1 nikker) Tenker du på tallet 10?
Elev 1	Nei
Elev 2	Okey, jeg prøver. Er det høyere eller lavere enn 10?
Elev 1	Høyere enn 10
Elev 2	Er det 15?
Elev 1	Nei
Elev 2	Er det lavere eller høyere enn 15?
Elev 1	Høyere
Elev 2	Okey men da kan jeg si 18 fordi det er cirka på midten av 15 og 20. Okey, er det da .. 17?
Elev 1	KORREKT!...Jeg tror du brukte halveringsmetoden
Elev 2	Ja, eller i hvert fall prinsippet fra halveringsmetoden med å sjekke om det er høyere eller lavere enn ett eller annet, og så får du prøve på nytt igjen..Og så sjekker du for et nytt punkt, så kan du se om det er høyere eller lavere enn det. Da går det fortere å finne fram til en verdi

Algoritmebehandling handler også om å finne en løsning på et problem gjennom å definere et sett med instruksjoner (Csizmadia et al., 2015). I starten av Fase 1 gjettet Gruppe 2, Gruppe 3 og Gruppe 4 usystematisk på tilfeldige tall i et begrenset tallintervall. Denne framgangsmåte brukte elevene nokså lang tid på. Etter en del forsøk med prøving og feiling gikk elevene over til å formulere konkrete instruksjoner som hadde flere kjennetegn av algoritmebehandling. Dette ble blant annet observert når en av elevene i Gruppe 3 brukte en *lineær søkealgoritme.* Eleven fulgte tydelige instruksjoner (spør om tallet er a i et tallintervall [a,b]), lagret denne informasjonen og formulerte neste instruks (hvis ikke, spør om tallet er a+1). Denne handlingen ble gjentatt i en bestemt rekkefølge for å komme fram til riktig tall. Andre elever i både Gruppe 2, Gruppe 3 og Gruppe 4 eliminerte tall ved dele tallene inn i kategorier. Elevene stilte spørsmål på en systematisk og strukturert måte for å få informasjon om tallets matematiske egenskaper, for eksempel om tallet var et partall eller oddetall. I neste steg formulerte elevene spørsmål basert på den nye informasjonen om tallet, og gjentok prosessen med å eliminere tall helt til de kom fram til riktig svar. Det siste tilfelle hvor algoritmebehandling ble observert var når elever i Gruppe 2, Gruppe 3 og Gruppe 4 utledet prinsippene for halveringsmetoden på egenhånd. Elevene formulerte instruksjoner som brukte logiske operasjoner for å avgjøre om et tall var større, lik eller mindre enn midten av et tallintervall. Deretter repeterte elevene denne handlingen, inntil ønsket resultat var oppnådd.

Til tross for ulike framgangsmåter som ble observert, endte alle gruppene opp med en løsningsalgoritme som hadde likhetsstrekk med halveringsmetoden som sluttresultat. Det var imidlertid en stor variasjon i hvor detaljert og oversiktlig algoritmen ble beskrevet. Figur 8 viser for eksempel algoritmen som Gruppe 3 og Gruppe 4 noterte ned:



1. Spør om tallet er over midten av intervallet,
Hvis ja: Det nye intervallet er mellom midten og slutten av det gamle.
Hvis nei: Det nye intervallet er mellom starten og midten av det gamle.
2. Hvis størrelsen på intervallet er 1 eller 2 tall: Gjett tilfeldig på tallene i intervallet.
Hvis ikke: Gjenta steg 1.



· Partall eller oddetall?
· Større eller mindre enn halvparten? -‖-, -‖-, -‖-

Figur 8: Algoritmen til Gruppe 3 og Gruppe 4 henholdsvis til venstre og høyre.

I Fase 2 skulle elevene i første omgang formulere en algoritme for å finne f(x) = 0 til en tredjegradsfunksjon. Elevene skulle kun ta utgangspunkt i bildet til grafen, uten å ha fått oppgitt det tilhørende funksjonsuttrykket (Kapittel 3.3.2, Figur 4, «oppgave 2)»). Deretter skulle elevene anvende algoritmen på et funksjonsuttrykk som tilhørte grafen på bildet (Kapittel 3.3.2, Figur 4, «oppgave 3)»). Til sammen ble det observert 4 løsningsalgoritmer hvor kjennetegnet algoritmebehandling kom til uttrykk:

Tabell 5: Algoritmene observert i Fase 2 som innebar bruk av algoritmebehandling.

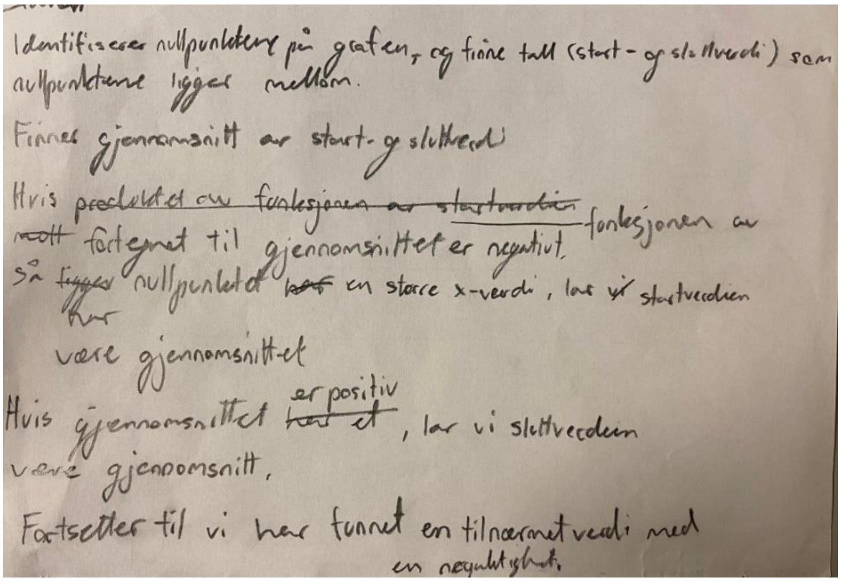
1)	Bruke halveringsmetoden
2)	Se på fortegnet til funksjonsverdier rundt nullpunktet for å oppgi en tilnærmet verdi for nullpunktet
3)	Bruke polynomdivisjon
4)	Tilpasse og overføre algoritmen fra Fase 1 for å finne en tilnærmet verdi for nullpunktet

Gruppe 1 var nokså rask med å formulere en løsningsalgoritme. Samtalen under tyder på at Elev 2 gjenkaller halveringsmetoden direkte for å finne nullpunktene til funksjonen:

Utdrag 2

Elev 2	Så det vi gjør (starter å skrive ned på arket). Vi identifiserer nullpunktene.. hvor de ligger.. hvilke tall de ligger imellom
Elev 1	Finne intervallet nullpunktet ligger mellom
Elev 2	Altså start og sluttverdi. Og da, når vi har funnet start og sluttverdi så kan vi se på gjennomsnitt fordi vi har lyst til å krympe intervallet for å komme nærmere nullpunktet (skriver mer på arket)…Og etter at vi har gjort det så sjekker vi med fortegnet..hvis vi tar den verdien ehm…hvis vi tar funksjonen av den startverdien og ganger med funksjonen av gjennomsnittet. Hvis de har forskjellige fortegn, altså at det blir negativt så vet vi jo at…nullpunktet ligger over gjennomsnittet fordi hvis vi hadde ganget..funksjonen med gjennomsnittet og fått 0..da har vi funnet nullpunktet, men siden det blir negativt så vet vi at det ligger under grafen.. Eller hvis det blir positivt da vet vi at nullpunktet ligger over gjennomsnittet og hvis det blir negativt så vet vi at den ligger under.. Hvis produktet av funksjonen av startverdien\

Her kommer algoritmebehandling til uttrykk gjennom at Elev 2 forsøker å formulere instruksjoner for å identifisere et intervall, og betingelser som avgjør hvor nullpunktet ligger ved å se på fortegnet til produktet av f(startverdi) og f(midtpunkt). Eleven ser ut til å ha utfordringer med å formulere riktige betingelser. Først antyder eleven at nullpunktet befinner seg *over* (til høyre for) midtpunktet hvis produktet er negativt, men senere sier eleven at nullpunktet ligger *under* (til venstre for) midtpunktet dersom produktet har negativ verdi. Det riktige her er at hvis produktet av f(startverdi) og f(midtpunkt) er negativt så vil nullpunktet befinne seg i intervallet [startverdi, midtpunkt] og dermed vil nullpunktet ligge til venstre for midtpunktet. Videre i samtalen foreslår den andre eleven i gruppen å *kun* se på funksjonsverdien til midtpunktet for å avgjøre i hvilken del av intervallet nullpunktet ligger. Elevene tar utgangspunkt i dette forslaget og ender opp med algoritmen vist i Figur 9. Algoritmen inneholder repetisjon av stegene i algoritmen, og oppdatering av start- og sluttverdiene. Denne algoritmen er imidlertid ikke fullstendig korrekt, da det ikke er nok med å bare se på fortegnet til midtpunktet. Algoritmen elevene kom fram til gjelder kun i det tilfelle hvor grafen er stigende i intervallet rundt nullpunktet, og når midtpunktet har en x-verdi som er mindre enn x-verdien til nullpunktet. Denne algoritmen vil dermed kun gi svar på 2 av de 3 nullpunktene til grafen $f(x) = \frac{1}{6}x^3 - \frac{3}{5}x^2 - \frac{1}{2}x + 2$.


Identifiserer nullpunktene på grafen, og finne tall (start- og sluttverdi) som nullpunktene ligger mellom.
Finner gjennomsnitt av start- og sluttverdi
Hvis fortegnet til funksjonen av gjennomsnittet er negativt,
så har nullpunktet en større x-verdi, lar vi startverdien være gjennomsnittet
Hvis gjennomsnittet er positiv, lar vi sluttverdien være gjennomsnitt,
Fortsetter til vi har funnet en tilnærmet verdi med en nøyaktighet.

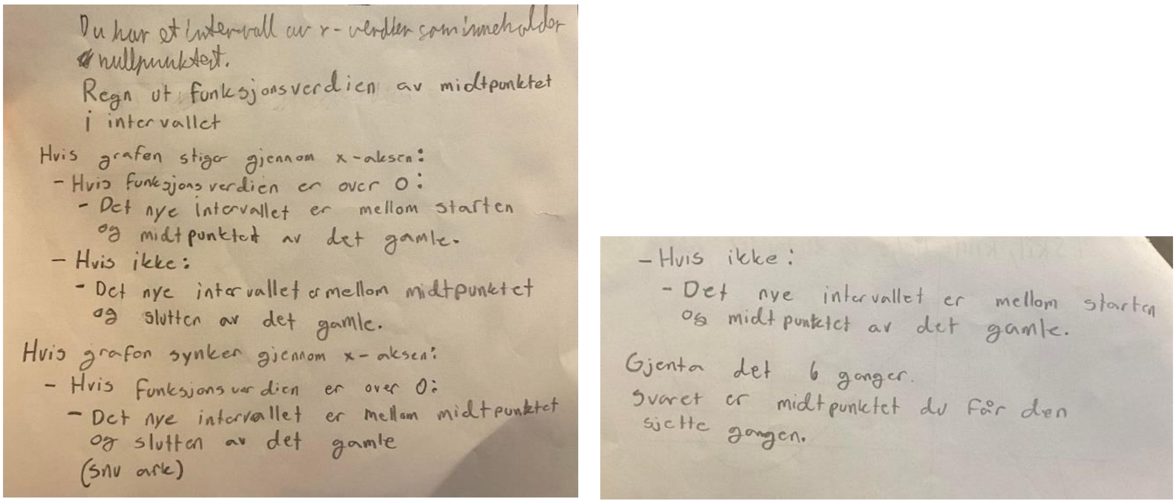
Figur 9: Gruppe 1 Algoritmen for å finne en tilnærmet verdi av f(x)=0.

I Gruppe 2 ble algoritmebehandling observert når elevene kom fram til at de kunne studere fortegnet til funksjonsverdien rundt et nullpunkt. Elevene oppdaget at funksjonsverdien på hver side av et vilkårlig nullpunkt hadde motsatt fortegn, og brukte denne informasjonen videre til å finne en tilnærmet verdi for nullpunktet. Her er utdraget fra samtalen som viser at Elev 1 tydelig definerer en rekke klare instruksjoner for å løse oppgaven:

Utdrag 3

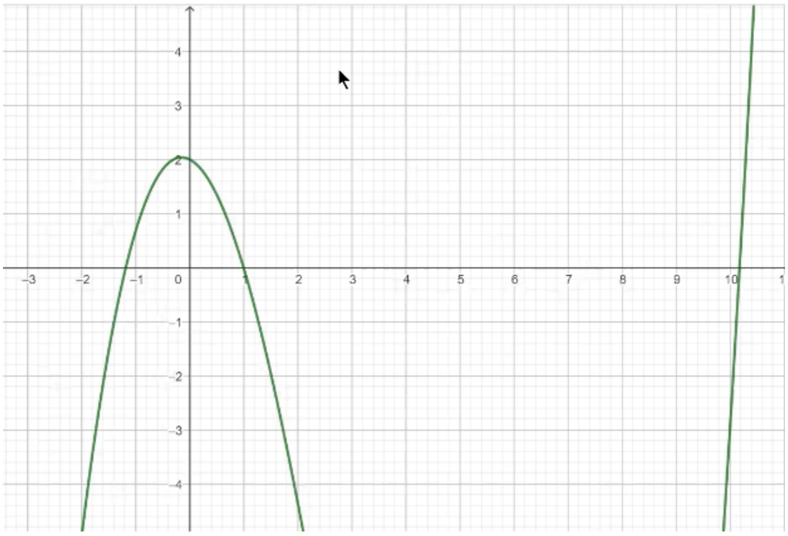
Elev 1	Så hvis vi klarer å korte ned intervallet så kan vi se om f(x) er positiv og negativ for de tallene, for hvis de ikke er det..for eksempel hvis vi finner at f(x) er negativ på begge sider av nullpunktet så ligger ikke nullpunktet der, men i..den andre delen av intervallet liksom. Og slik kan vi fortsette til vi har kommet nærme nok. (skriver ned noe på arket)

Gruppe 3 klarte å formulere klare og veldefinerte instruksjoner for å løse oppgaven vist i Figur 10, gjennom å hente inspirasjon fra algoritmen som de formulerte i Fase 1. Gruppen formulerte instruksjoner som blant annet gjentok stegene i algoritmen i form av seks iterasjoner. Elevene formulerte betingelser som tok hensyn til monotoniegenskapene til grafen rundt et nullpunkt. De brukte matematiske og logiske operasjoner for å avgjøre om f(midtpunkt) hadde en positiv eller negativ verdi. Videre brukte elevene denne informasjonen for å skrive instruksjoner som avgjør i hvilken del av intervallet nullpunktet ligger i. Elevene formulerte også instruksjoner som i hver iterasjon oppdaterte start- og sluttverdien.



Figur 10: Gruppe 3 Algoritmen for å finne en tilnærmet verdi av f(x)=0.

Hos Gruppe 4 kom algoritmebehandling i form av å følge og bruke en kjent metode. En av elevene i gruppen tastet inn feil funksjonsuttrykk i GeoGebra (Figur 11), og fikk dermed en graf som var annerledes enn grafen i oppgaven. Elevene benyttet seg videre av polynomdivisjon, etterfulgt av abc-formelen for finne nullpunktene til grafen. Jeg kommer til å si noe mer om dette i neste delkapittel, 4.1.2.



Figur 11: Den nye grafen som Gruppe 4 fikk etter å ha tastet inn feil funksjonsuttrykk i GeoGebra.

Ifølge Csizmadia et al. (2015) er algoritmebehandling en kjerneferdighet som elevene også utvikler i programmeringskontekst. Både Gruppe 1 og Gruppe 3 implementerte koden i utviklingsmiljøet VS Code ved hjelp av programmeringsspråket Python. Ettersom elevene formulerte en rekke konkrete instruksjoner (i form av input-funksjoner, variabler, if-else og while-løkker) på et programmeringsspråk for å løse problemet, ble dette kategorisert som algoritmebehandling.

4.1.2 Dekomposisjon

Dekomposisjon er evnen til å bryte ned et komplekst problem i mindre komponenter, slik at problemet blir mer overkommelig (Csizmadia et al., 2015).

I Fase 1 delte alle gruppene problemet i mindre delproblemer ved å formulere ulike spørsmål som gjorde det enklere å gjette på riktig tall. På denne måten eliminerte elevene en viss mengde med tall slik at problemet ble mer håndterbart. Dette er tydelige tegn på dekomposisjon. Utdrag 4 nedenfor viser noen av spørsmålene som elevene stilte. Disse er hentet fra ulike grupper og skal ikke leses som en sammenhengende samtale.

Utdrag 4

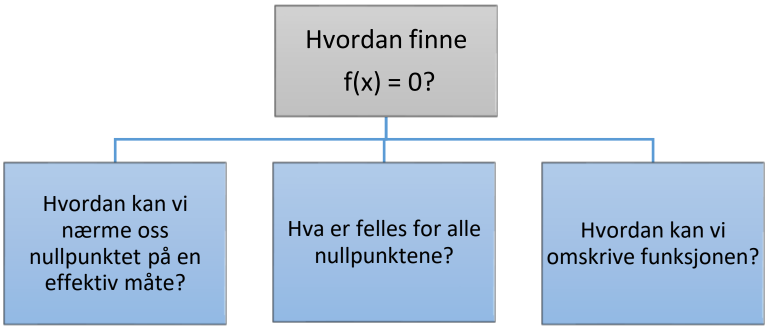
Så hvis det ikke kan deles på 3, så kan det kanskje deles på 2..Da kan jeg spørre om det er et partall?
Er det over eller under 10?
Er det et oddetall?
Ehm…er det nærmere 100 eller 50?
Er det 1- eller 2-sifret?
Okey…ehm, kan jeg finne tallet i 5-gangen?
Har tallet et primtall i seg? Eller jeg mener, er tallet et primtall?
Har det et 3-tall i seg?
Inneholder tallet to av de samme tallene?

Til slutt kombinerte elevene svarene på hvert delproblem for å løse det originale problemet. Utdrag 5 nedenfor viser hvordan en av elevene i Gruppe 2 bruker svarene fra flere delproblem, for å så kombinere disse sammen. Underveis formulerer eleven også flere mindre delproblemer og løser dem direkte. I forkant av utdraget har eleven allerede stilt et par spørsmål og vet følgende om tallets matematiske egenskaper: tallet er et oddetall, er delelig på 3 og ligger i intervallet mellom 125 og 250. Eleven har også eliminert muligheten for at tverrsummen kan være 3, og nå forsøker eleven å kombinere informasjonen som er tilgjengelig for å foreslå et gjett.

Utdrag 5

Elev 1	(Begynner å skrive) Hvis tverrsummen skal bli 6 eller 9..Okey, jeg vet at tallet er et oddetall, og at det kan deles på 3, og nå vet jeg at tverrsummen må være 6 eller 9..Så siden det er et oddetall så må det slutte på 1,3,5,7 eller 9.. (skriver ned) Ikke sant? Og så vet jeg at det er 3 siffer i det tallet, siden det er fra 125 til 250, så første siffer må enten være 1 eller 2..Hm, okey (skriver ned). Så hvis første tallet er 1, da kan det slutte på 1 og da må andre siffer være..4 for at tverrsummen skal bli 6. Er det 141?

I Fase 2 spilte dekomposisjon en viktig rolle for å finne nullpunktene som ikke lot seg lese av direkte. Det første delproblemet som alle gruppene formulerte var å avgjøre hvor grafen skjærer x-aksen mellom to heltallsverdier. Deretter dekomponerte elevene problemet i hovedsak ved hjelp av tre ulike tilnærminger (Figur 12).



Figur 12: Tre ulike måter gruppene dekomponerte problemet med å finne f(x)=0.

Den første tilnærmingen gikk ut på at noen elever forsøkte å identifisere et passende intervall rundt nullpunktet. Istedenfor å oppgi en konkret verdi for nullpunktet, forsøkte elevene dermed å svare på spørsmålet: I hvilket intervall ligger f(x)=0? I dette lå også en vurdering av et nytt delproblem: Hvilken informasjon trenger vi?

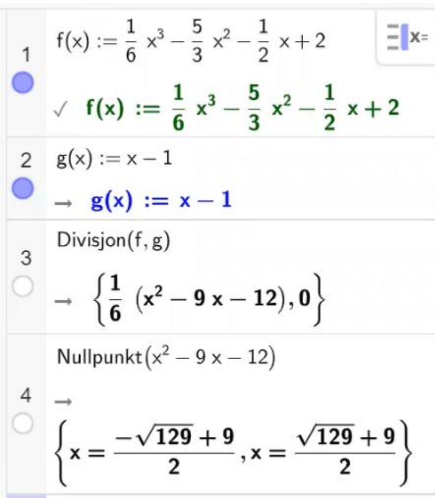
Gruppe 1 dekomponerte problemet ved å anvende halveringsmetoden. Elevene halverte intervallet og brukte informasjonen om funksjonsverdien til midtpunktet for å videre avgjøre hvor nullpunktet lå. På denne måten dekomponerte de problemet nok en gang med å avgjøre: I hvilken del av intervallet ligger f(x)=0? Elevene dekomponerte problemet ytterligere med å forsøke å bestemme en viss nøyaktighet for den tilnærmede verdien av nullpunktet. Gruppe 2 halverte også intervallet i stadig mindre deler. Det at nye delproblemer ble formulert påvirker frekvensen av dekomposisjon. Disse omhandlet blant annet spørsmålet: Hva har det å si om grafen synker eller stiger i intervallet? Nedenfor er et utdrag som viser til en del av samtalen hvor denne dekomposisjonen kom til uttrykk:

Utdrag 6

Elev 3	Okey så vi regner ut funksjonsverdien av midtpunktet i intervallet, og så ser vi på hvilken y verdi det tilsvarer
Elev 2	Ja, men problemet er hva vi skal gjøre om grafen stiger gjennom x-aksen eller synker gjennom x-aksen…ehm…..Så da kan vi kanskje ha sånn enda ett spørsmål, hvis grafen stiger så gjør det og det, og hvis grafen synker så gjør det og det..men det blir veldig langt

Dekomposisjon kom til uttrykk i Gruppe 2 ved at de omformulerte problemet til: Hva er det som er likt for alle nullpunktene? Som nevnt i delkapittel 4.1.1 oppdaget elevene som et resultat av dette spørsmålet at funksjonsverdien på hver side et nullpunkt hadde motsatt fortegn. Denne informasjonen brukte de for å minske intervallet gjentatte ganger, til de selv var fornøyd med svaret. På denne måten ble oppgaven med å finne f(x)=0 delt i mer håndterbare deler.

Den tredje tilnærmingen ble observert i Gruppe 4. I starten av Fase 2 diskuterte elevene spørsmålet: Hvordan kan vi omskrive funksjonen slik at vi kan bruke polynomdivisjon? Ettersom elevene ikke fikk oppgitt et konkret funksjonsuttrykk i oppgave 2), hadde elevene utfordringer med å gjøre dette. Elevene lyktes ikke med å dekomponere problemet på en annen måte og gikk etter hvert videre til oppgave 3). Som det ble nevnt i delkapittel 4.1.1. tastet en av elevene inn feil funksjonsuttrykk i GeoGebra. Dette gjorde at elevene nå kunne løse delproblemet som de formulerte i starten av Fase 2. Elevene benyttet seg av CAS i GeoGebra for å omskrive funksjonen ved hjelp av divisjon-kommandoen, og deretter nullpunkt-kommandoen for å finne de nullpunktene som ikke lot seg lese av grafisk (Figur 13). På denne måten viste elevene evne til å dekomponere tredjegradsfunksjonen i to mindre polynomuttrykk av første og andre grad.



Figur 13: Gruppe 4 dekomposisjon ved hjelp av polynomdivisjon og abc-formelen.

Til slutt ble dekomposisjon observert i programmeringssammenheng når Gruppe 1 og Gruppe 3 implementerte koden i VS Code. Dekomposisjon kom til uttrykk når elevene måtte bryte ned den algoritmen som de formulerte med ord i mindre deler, slik at det ble enklere å oversette den til Python-kode. Elevene definerte funksjoner og variabler hver for seg, formulerte løkker og tilhørende betingelser, samt andre kodeblokk. Utdrag 7 viser hvordan Elev 2 i Gruppe 1 foreslår en måte å definere en vilkårlig tredjegradsfunksjon på Pythonspråket ved å dekomponere funksjonen i 4 koeffisienter a, b, c og d, slik at koden kan ta inn en vilkårlig funksjon:

Utdrag 7

Elev 1	Men går det an å ta return input sånn at det blir hvilken som helst funksjon? Eller er det umulig?
Elev 2	Hvilken som helst..Altså det går jo kanskje an å gjøre det ehm..
Elev 1	For da hadde det vært et brukbart program
Elev 2	Det man kan gjøre er å kanskje definere grafen og ta input av a-koeffisienten, b-koeffisienten og c-koeffisienten og d hver for seg, siden du trenger alle de 4 tallene sånn at vi kan regne ut for hvilken som helst 3.gradsfunksjon?

4.1.3 Generalisering

Generalisering var det kjennetegnet som ble observert i minst grad. Ifølge Csizmadia et al. (2015) handler generalisering om å bruke tidligere kunnskap og erfaring til å løse nye problemer. I denne studien kom generalisering til uttrykk gjennom tre konkrete tilfeller.

I det første tilfelle overførte Gruppe 1 prinsippene fra halveringsmetoden for å løse Gjett-tall problemet i Fase 1. Gjett-tall er en aktivitet der en person tenker på et tall, mens en annen prøver å gjette hvilket tall det er ved å komme med ulike tallforslag. På den andre siden er halveringsmetoden en numerisk metode som brukes for å finne nullpunktene til matematiske ligninger ved å systematisk halvere et gitt intervall. Elevene anvendte denne ideen i Gjett-tall aktiviteten ved å identifisere midtpunktet i tallintervallet. Deretter bestemte de hvilken halvdel av intervallet som inneholdt tallet ved å stille spørsmål om tallet var høyere eller lavere enn midtpunktet. På samme måte som med halveringsmetoden, ble denne prosessen gjentatt til tallet som den andre tenkte på ble funnet. Elevene tilpasset deler av halveringsmetoden slik at det gjaldt i et nytt problemområde, noe som kjennetegner generaliseringskomponenten.

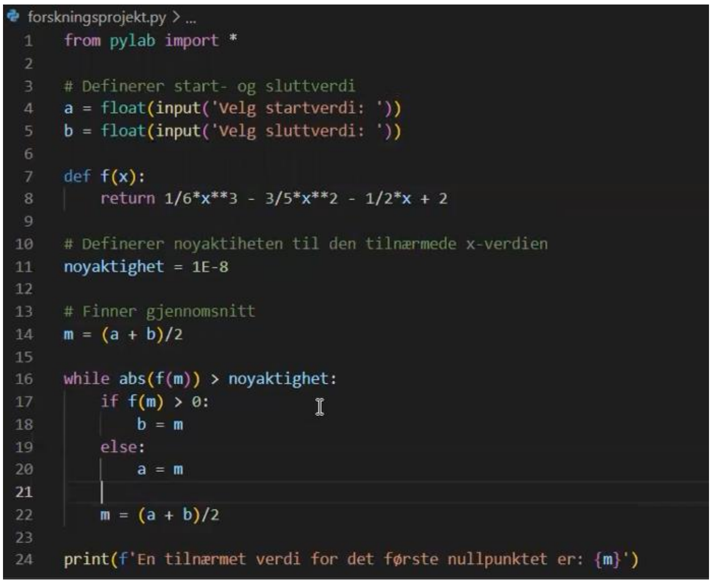
Det andre tilfellet hvor generalisering ble observert var når elevene overførte og tilpasset ideene fra algoritmen som de formulerte i Fase 1 til Fase 2. Dette kom til utrykk i Gruppe 3. Elevene gjentok prosessen med å dele intervallet i to, for å deretter sjekke hvilken del av intervallet nullpunktet lå i basert på visse betingelser. Utdrag 9 viser til når elevene først ser fellestrekk med Gjett-tall aktiviteten og forsøker å overføre ideen med å halvere intervallet til et nytt problemområde. Elev 2 ser også hvordan problemet i Fase 2 er annerledes fra Fase 1 med å påpeke at dette problemet inkluderer et funksjonsuttrykk i tillegg. Spørsmålet *Hvordan er dette annerledes fra tidligere problem?* kjennetegner generaliseringskomponenten i algoritmisk tenkning.

Utrdag 9

Elev 3	Bare ta halvparten?
Elev 1	Hva er det vi skal ta halvparten av?
Elev 3	Vi vet at det ene nullpunktet er mellom -2 og -1.5
Elev 2	Ja
Elev 3	Og så kan vi kanskje bruke algoritmen som vi lagde på det intervallet?
Elev 1	Ja for å sikkert finne en mer nøyaktig verdi for nullpunktet som mulig
Elev 2	Ja, men det er ikke..Vi kan ikke bruke akkurat samme algoritmen, fordi her har vi en funksjon i tillegg

En lignende generaliseringsprosess ble også observert i Gruppe 2, men elevene i denne gruppen halverte ikke intervallet konsekvent. Noen ganger halverte elevene intervallet for å identifisere funksjonsverdier som hadde motsatt fortegn, mens andre ganger sjekket de funksjonsverdien for x-verdier som var i nærheten av nullpunktet. Likevel klarte de å se et mønster mellom Gjett-tall aktiviteten og oppgaven med å finne f(x)=0. Dermed ble dette tilfelle også kategorisert som generalisering.

Det kan også argumenteres for at generalisering kom til uttrykk når Gruppe 1 og Gruppe 3 implementerte algoritmen som elevene formulerte med ord i VS Code. For å skrive Python-kode måtte elevene avgjøre hvordan det å skrive kode er annerledes fra den algoritmen som de formulerte. Dette er et viktig trekk i generaliseringsprosessen i Csizmadia et al. (2015). De måtte tilpasse og endre algoritmen slik at den kunne gjelde i programmeringssammenheng. Det innebærer at elevene identifiserer fellestrekk i algoritmen for å oversette disse til Python-språk for hvordan dataene skal håndteres og manipuleres. Elevene måtte også ta hensyn til spesifikke detaljer som er annerledes fra den håndskrevne algoritmen. For eksempel tok Gruppe 1 hensyn til at datatypen til x-verdien kunne være en *float* og at man skulle sjekke når absoluttverdien til f(midtpunkt) var større enn nøyaktigheten (Figur 14). Det er tydelig at elevene tenkt over spørsmålet: Hvordan er dette annerledes?



Figur 14:Gruppe 1 Python-kode som implementerer halveringsmetoden

Kjennetegnet generalisering ble ikke observert i Gruppe 4.

4.1.4 Abstraksjon

Abstraksjon går ut på å identifisere og fokusere på de viktigste detaljene i en situasjon for å løse et problem (Csizmadia et al., 2015). Abstraksjonskomponenten ble observert i alle gruppene. Her vil jeg trekke fram 3 abstraksjonsprosesser som ble observert i denne studien.

Den første abstraksjonsprosessen er knyttet til Fase 1. For å komme fram til en algoritme som har likhetstrekk med halveringsmetoden, krever det abstraksjonsferdigheter. Abstraksjon kom til uttrykk når elevene ignorerte informasjon om tallets matematiske egenskaper (partall,

primtall osv.). Elevene abstraherte bort disse detaljene og endte opp med å følge prinsippene fra halveringsmetoden. På denne måten viste elevene evne til å velge ut relevante detaljer slik at problemet ble forenklet. Det andre tilfelle hvor abstraksjon ble observert var i Fase 2. Det at elevene i Gruppe 2 oppdaget at funksjonsverdien rundt et nullpunkt hadde motsatt fortegn, betyr at elevene klarte å fokusere på viktige detaljer for å redusere kompleksiteten rundt definisjonen av et nullpunkt. Elevene i Gruppe 1 og Gruppe 3 hadde også forstått essensen av problemet i Fase 2, når de valgte ut nok informasjon og relevante detaljer for å finne en tilnærmet verdi for f(x) = 0. På denne måten fant elevene et tilnærmet svar på likningen f(x) = 0 uten å måtte vurdere alle mulige verdier i det opprinnelige intervallet. Algoritmen som gruppene kom fram til ga en god representasjon av en løsning på en forenklet måte. Konvertering av en algoritme fra en håndskrevet notasjon til en programmeringskode ble også kategorisert som abstraksjon. Dette ble observert både i Gruppe 1 og Gruppe 3. Elevene identifisere og fokuserte på de viktigste elementene i algoritmen for å representere dem på en måte som kan forstås av datamaskinen. Dette innebærer å forenkle og å skape en representasjon i form av Python-språk som skjuler den fullstendige kompleksiteten til algoritmen.
Abstraksjonskomponenten ble ikke observert hos Gruppe 4 i Fase 2.

4.1.5 Evaluering

Evaluering er et av de kjennetegnene ved algoritmisk tenkning som ble observert i stor grad. Ifølge Csizmadia et al. (2015) handler evaluering om å kontrollere om en løsning er god, enten det er en algoritme, et system eller en prosess.

I denne studien kom evaluering blant annet til uttrykk gjennom at elevene vurderte effektiviteten til de ulike løsningsalgoritmene. Som nevnt tidligere i kapittel 4.1.1 startet Gruppe 2, Gruppe 3 og Gruppe 4 med å gjette tilfeldig på tall i Fase 1. Alle gruppene påpekte etter hvert at denne løsningen var lite effektiv og hadde et ønske om å forbedre den. Utdrag 10 nedenfor viser til en del av samtalen mellom elevene i Gruppe 3. I forkant av utdraget har to elever i gruppen forsøkt å gjette på tallet som den tredje eleven tenkte på, i intervallet [1,100]. Det tok lang tid før elevene lyktes med å gjette på riktig tall, og de startet nå å reflektere over hvor effektiv denne framgangsmåten var:

Utdrag 10

Elev 1	Dette tok altfor langt tid ass
Elev 2	Vi må finne på en bedre måte å gjøre det på, fordi når det er få tall å velge mellom så går det relativt fort, men sånn..når det er 1 til 100, så er det altfor mange tall å velge mellom hvis vi bare skal gjette sånn helt vilt liksom
Elev 3	Kan vi komme opp med en bedre algoritme?

På slutten av Fase 1, da alle gruppene utviklet en algoritme som hadde fellestrekk med halveringsmetoden, kom også evalueringskomponenten til uttrykk. Elevene vurderte igjen hvor god og effektiv denne løsningsalgoritmen var. Eksempler på dette er vist i utdrag 11. Disse ytringene tilhører elever fra ulike grupper.

Utdrag 11

Ja sikkert, men for at noe skulle vært raskere så måtte det ha eliminert mer enn 50%
Og det er den raskeste, eller mest stabile måten å komme fram til et svar på..
Så du blir alltid kvitt halvparten

Et annet tilfelle hvor evaluering kom til uttrykk var da elevene vurderte om algoritmen passet dets formål. Dette ble observert i Gruppe 1, Gruppe 2 og Gruppe 3. Utdrag 12 viser til når elevene i Gruppe 1 konkluderer med at algoritmen som de formulerte i Fase 1 alltid gir ønsket resultat. Elevene kommer fram til denne konklusjonen etter å ha vurdert noen fordeler og ulemper med algoritmen. De sammenligner også deres algoritme med strategien som innebærer å gjette på tilfeldige tall. Dette er elementer som kjennetegner evalueringsprosessen.

Utdrag 12

Elev 2	Haha ja det er dypt…Så vi repeterer denne prosessen…Men hvis vi skal tenke på fordeler og ulemper med denne metoden..Hva tenker du?
Elev 1	Altså...ulempen er jo at man kan være raskere hvis man gjetter i noen tilfeller
Elev 2	Du kan det, men\
Elev 1	Det er lite sannsynlig
Elev 2	Hvis det blir veldig store..\
Elev 1	Tall\
Elev 2	Veldig mange tall du kan velge mellom så vil jo denne metoden være raskere, for den vil være mer konsekvent, så den vil konsekvent være effektiv..men du kan jo selvfølgelig gjette på ett forsøk, det er mulig. Men det er lite sannsynlig statistisk sett
Elev 1	Ja, og da er jo fordelen med det at det funker alltid

I Fase 2 gikk elevene i Gruppe 1, Gruppe 2 og Gruppe 3 gjennom algoritmen som de formulerte steg for steg. De benyttet seg av GeoGebra for å teste ut algoritmene sine og gjøre ulike beregninger. Elevene tolket resultatene og vurderte nøyaktigheten av verdien for nullpunktet som de kom fram til. Her er et utdrag som viser til samtalen mellom elevene i Gruppe 3, hvor elevene evaluerer svaret som de fikk etter fire iterasjoner:

Utdrag 13

Elev 3	Det er ikke nøyaktig det vi fikk da
Elev 1	Vi har jo bare gjort det 4 ganger, så hvis vi fortsetter så får vi sikkert det
Elev 2	Nei, vi får mest sannsynlig aldri det nøyaktig, vi kommer bare nærmere og nærmere siden vi kan ha uendelig mange desimaler. Siden vi bruker sånn numerisk løsning, så blir det litt unøyaktig

Gruppe 1 og Gruppe 2 benyttet seg i tillegg av nullpunkt-kommandoen i GeoGebra for å kontrollere om svaret var tilfredsstillende. Utdrag 14 viser til en del av samtalen mellom elevene i Gruppe 1. I forkant av dette utdraget hadde elevene utført 6 iterasjoner i GeoGebra for å regne ut f(x) = 0 i intervallet [-2, -1.5] ved hjelp av algoritmen som de selv formulerte i Fase 2. Da kom elevene fram til en x-verdi som tilsvarte -1.7891. Videre brukte elevene nullpunkt-kommandoen i GeoGebra som ga en x-verdi lik -1.7951. Elev 2 sammenlignet disse x-verdiene for å vurdere nøyaktigheten av deres eget svar, og hva de eventuelt kunne ha gjort for å få en mer nøyaktig x-verdi. Til slutt sammenlignet Elev 2 også hvor effektiv deres algoritme var i forhold til et eventuelt program. Alle disse elementene kjennetegner evalueringsprosessen.

Utdrag 14

Elev 1	-1.7891?
Elev 2	Ja, så det vil være x-verdien til nullpunktet
Elev 1	Okey
Elev 2	Så da har vi funnet den ved å se på fortegnet til gjennomsnittet og bestemme om det er negativt eller positivt..Det går an å se på nullpunktet (Regner ut x-verdi for nullpunkt ved hjelp av nullpunkt-kommandoen i GeoGebra). Vi ser jo at nullpunktet som GeoGebra har funnet er en tilnærmet verdi på -1.7951 som er litt forskjellig fra det vi har funnet…og da i neste iterasjon så ville jo vi har skrevet om dette..vi ville ha endret på sluttverdien..og gjort den om til gjennomsnittet, og sånn kan man fortsette . Men det er gunstig å gjøre det i et program, da dette går mye mer effektivt og da kan man definere hvor nøyaktig man vil ha svaret

I Gruppe 1 og Gruppe 3 kom evalueringen også til uttrykk i programmeringssammenheng når elevene kjørte Python-kodene for å se om den ga forventet verdi for nullpunktet. De sammenlignet svarene som de fikk i GeoGebra, gjorde justeringer for å øke nøyaktigheten, og diskuterte hvor effektivt det var å bruke som programmering som verktøy framfor GeoGebra.

4.2 Kreativ eller imitativ vei til svaret

Det ble totalt observert 6 ulike framgangsmåter både i Fase 1 og Fase 2. Disse er henholdsvis oppsummert i Tabell 6 og Tabell 7. Resonnementene som ligger til grunn for framgangsmåtene, blir her referert til som en sti, i samsvar med begrepet i Lithner (2008) sitt rammeverk. Tallinndeling fra 1 til 6 sier imidlertid ingenting om at Sti 6 er mer kreativt enn Sti 5 for eksempel. Denne inndelingen er bare ment for å legge fram resultatene på en oversiktlig og strukturert måte.

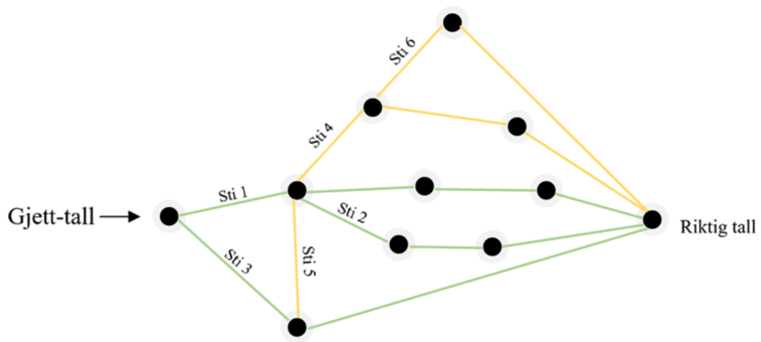
Tabell 6: Alle framgangsmåtene som ble observert i Fase 1: Gjett-tall aktiviteten.

	Beskrivelse	*Gruppe 1*	*Gruppe 2*	*Gruppe 3*	*Gruppe 4*	
Sti 1	Usystematisk gjetting (gjetter på tilfeldige tall)		x	x	x	Imitativ resonnement
Sti 2	Lineær søkealgoritme (Gjetter på a, a+1, a+2,.. og oppover i et intervall [a,b] helt til tallet er funnet)			x		
Sti 3	Gjenkaller prinsippet fra halveringsmetode umiddelbart	x				
Sti 4	Systematisk eliminasjon ved kategorisering av tall (stiller spørsmål som tar stilling til tallets matematiske egenskaper)		x	x	x	Kreativt resonnement
Sti 5	Er kjent med halveringsmetoden, men klarer ikke å gjengi algoritmen umiddelbart. Må selv utforske for å komme fram til prinsippene bak metoden		x		x	
Sti 6	Kommer selv fram til prinsippene fra halveringsmetoden uten tidligere kjennskap til den			x		

Sti 1 og Sti 2 blir betraktet om standardmetoder for å gjette tall. Disse stiene preges ikke av originalitet og oppfyller ingen andre kriterier for CMR. Dermed ble stiene kategorisert som imitative. Sti 3 ble også kategorisert som imitativ. Elevene som fulgte denne stien gjenkalte umiddelbart en memorert og kjent metode (halveringsmetoden), og anvendte prinsippene fra denne metoden for å finne en løsningsalgoritme. Elevene låste seg til denne framgangsmåten, og forsøkte ikke å finne andre tilnærminger. Dermed viste elevene liten evne til å være fleksibel.

Sti 4 og Sti 6 ble kategorisert som kreative ettersom disse oppfylte flere kriterier for CMR. Stiene var preget av originalitet for elevene som resonnerte, løsningene var plausible og hadde matematisk forankring. I Sti 5 ble en glemt metode gjenkalt. Dette skjedde imidlertid ikke umiddelbart. Elevene måtte gjennom litt prøving og feiling for å komme fram til prinsippene bak halveringsmetoden. Dermed oppfyller stien også kravet for CMR. Kreative stier kom også til uttrykk blant de gruppene som viste evne til å nærme seg problemet fra ulike vinkler og komme opp med flere løsninger. Resultatene fra Tabell 6 viser at Gruppe 2, Gruppe 3 og Gruppe 4 fulgte flere stier for å gjette på riktig tall i Fase 1. Dermed viste elevene evne til fleksibilitet som kjennetegner kreative resonnement.

En visualisering av stiene og hvordan disse henger sammen er vist i grafen under (Figur 15). Hjørnene i grafen gir imidlertid ikke en representasjon av det faktiske antallet deloppgaver. Alle gruppene fulgte i starten av Fase 1 en sti av imitativ karakter. Etter hvert som elevene jobbet med oppgaven og flere kreative løsninger kom til uttrykk, utspringet nye stier fram. Uansett hvilken sti elevene fulgte, førte det til riktig gjett.



Figur 15:Stiene i grafen representerer resonneringssekvensene som ble observert i Fase 1: Gjett-tall aktivitet. En grønnfarget sti betegner imitativt resonnement, og gulfarget sti betegner kreativt resonnement.

Tabell 7: Alle framgangsmåtene som ble observert i Fase 2: finne f(x) = 0.

	Beskrivelse	*Gruppe 1*	*Gruppe 2*	*Gruppe 3*	*Gruppe 4*	
Sti 1	Gjette tilfeldig på x-verdier i nærheten av nullpunktet/ lese av nullpunktet på grafen ved å minske skalaen på x-aksen				x	Imitativ resonnement
Sti 2	Bruke polynomdivisjon				x**	
Sti 3	Bruke ChatGPT til å skrive en ferdig Python-kode som implementerer halveringsmetoden				x	
Sti 4	Bruke halveringsmetoden	x*				
Sti 5	Oppdage at funksjonsverdien rundt et nullpunkt har motsatt fortegn. Se på funksjonsverdier rundt nullpunktet for å oppgi en tilnærmet verdi		x			Kreativt resonnement
Sti 6	Tilpasse og overføre algoritmen fra Gjett-tall aktiviteten			x*		

* Algoritmen ble også implementert i utviklingsmiljøet VS Code
**Elev i gruppen skrev inn feil funksjonsuttrykk, som gjorde det mulig å benytte seg av polynomdivisjon for å løse oppgaven

Fire stier ble kategorisert som imitative i Fase 2. Sti 1 som ble observert i Gruppe 4 gikk ut på at elevene gjettet på tilfeldige x-verdier i nærheten av nullpunktet og regnet ut tilhørende funksjonsverdier i GeoGebra. Elevene minsket stadig skalaen på x-aksen, helt til de tastet inn en x-verdi som ga en funksjonsverdi lik 0. Denne framgangsmåten ble ikke ansett som original og nytenkende. I lys av dette ble Sti 1 kategorisert som IR. En annen sti som ble observert i samme gruppe er Sti 2. Denne innebærer bruk av en kjent prosedyre, nemlig polynomdivisjon. Elevene fulgte denne framgangsmåten for å finne et analytisk svar av nullpunktene. Bruk av memorerte og kjente metoder er også karakteristisk for IR. En annen tilnærming som ble observert i Gruppe 4 involverte bruk av ChatGPT, som er en kunstig intelligens-basert tekstgenereringsmodell lansert i 2022 (OpenAI, 2022). Gruppen ba ChatGPT om å generere en ferdig Python-kode som implementerer halveringsmetoden. ChatGPT foreslo en Python-kode, etterfulgt av en forklarende tekst til hvordan koden skal brukes. Elevene hadde utfordringer med å anvende denne informasjonen videre og kom derfor ikke fram til noe svar. Denne framgangsmåten anses som lite original og plausibel.

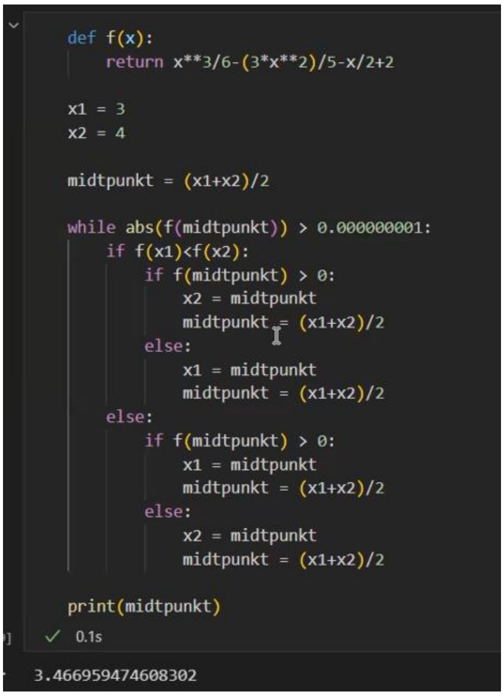
Sti 4 som ble observert i Gruppe 1 gikk ut på at elevene gjenkalte halveringsmetoden umiddelbart. Strategiimplementering baserte seg utelukkende på at elevene fulgte den gjenkalte metoden for å finne en tilnærmet verdi for nullpunktet. Ettersom elevenes framgangsmåte består av å bruke en memorert løsningsmetode ble dette kategorisert som IR. Elevene i Gruppe 1 implementerte også halveringsmetoden i VS Code. Implementeringen ble ikke ansett som ny for elevene, og elevenes framgangsmåte var preget av å gjenkalle flere deler av en memorert kode. Python-koden som gruppen utviklet (Figur 14, delkapittel 4.1.3), har flere likhetstrekk med Python-koden som blir presentert i R1 læreboken (Figur 15). I begge kodene er alle variabelnavnene like. Det kan tenkes at dette er en tilfeldighet, men variabelnavnet "noyaktighet" som blir brukt i begge kodene, tyder på at elevene har vært borti denne koden før. Elevene importerte også hele pylab-biblioteket i koden sin, til tross for at dette ikke var nødvendig for deres spesifikke kode. Funksjonsuttrykket i koden i læreboken inneholder derimot en log-funksjon som gjør det nødvendig å importere verktøy fra pylab-biblioteket. Dette er noe som understreker den imitative karakteren av elevens Pyton-kode. På bakgrunn av dette ble denne løsningen også vurdert som IR.

```
from pylab import *

a = 0                    # startverdi intervall
b = 1                    # sluttverdi intervall
noyaktighet = 0.0001     # angir hvor nøyaktig svaret skal være

def f(x):  # definerer funksjonen
  return 5*log(x**3 + 2) + x - 6

# Halveringsmetoden
# finner midtpunkt i opprinnelig intervall
m = (a + b)/2

while abs(f(m)) >= noyaktighet:
  if f(a)*f(m) < 0:
    b = m
  else:
    a = m

    m = (a + b)/2

print("Løsningen på likningen er tilnærmet lik ", round(m, 4))
```

Figur 16: Python-kode fra R1-læreboken som implementerer halveringsmetoden (Borgan et al., 2021, s.45).

Sti 5, som går ut på å oppdage at funksjonsverdien rundt et nullpunkt har motsatt fortegn og videre se på funksjonsverdier rundt nullpunktet for å oppgi en tilnærmet verdi, ble observert i Gruppe 2. Resonneringssekvensen var ikke preget av memorerte regler eller løsningsalgoritmer. Elevene i gruppen kom fram til funksjonens egenskap om fortegnsskiftet rundt et nullpunkt på egenhånd, og brukte denne informasjonen videre for å finne en tilnærmet verdi for nullpunktet. Dermed ble denne stien ansett som original og plausibel. Elevene viste evne til å tilnærme seg problemet på en måte som var ny for dem. Argumentene var også forankret i matematiske egenskaper og dermed oppfylte stien det tredje kriteriet i Lithner (2008) sitt rammeverk. Av den grunn ble stien kategorisert som CMR.

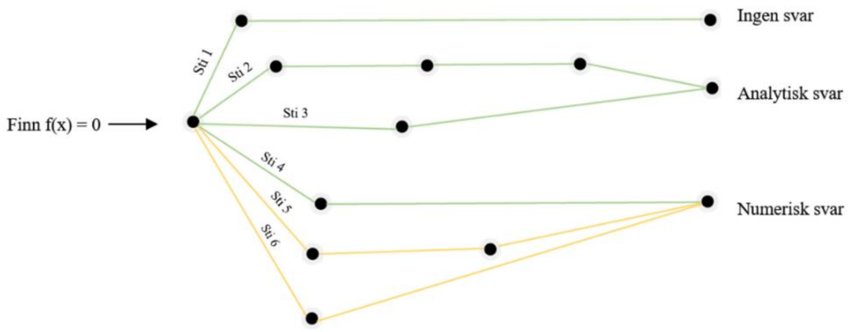
I Gruppe 3 ble Sti 6 som går ut på å tilpasse og overføre algoritmen fra Gjett-tall aktiviteten observert. Elevenes tilnærming til problemet med å finne f(x)=0 gikk ut på å modifisere algoritmen som de utviklet i Fase 1, noe som ble anses som plausibelt. Til tross for at algoritmen som elevene utviklet hadde likhetstrekk med halveringsmetoden, tydet samtalene underveis på at elevene ikke gjenkalte en memorert løsningsmetode. Elevene diskuterte ulike betingelser og kom fram til en løsning på egenhånd. Dermed ble denne stien vurdert som CMR. Videre implementerte elevene også algoritmen i VS Code (Figur 17). Med tanke på at algoritmen som elevene formulerte for hånd ble ansett som kreativ, ble implementeringen av algoritmen i VS Code også ansett som ny for elevene. Implementeringen var ikke preget av å gjenkalle en memorert kode. Python-koden som elevene utviklet, er original i den forstand at den kun tar utgangspunkt i den algoritmen som elevene selv formulerte og inneholder dermed flere unike elementer.


```
def f(x):
    return x**3/6-(3*x**2)/5-x/2+2

x1 = 3
x2 = 4

midtpunkt = (x1+x2)/2

while abs(f(midtpunkt)) > 0.000000001:
    if f(x1)<f(x2):
        if f(midtpunkt) > 0:
            x2 = midtpunkt
            midtpunkt = (x1+x2)/2
        else:
            x1 = midtpunkt
            midtpunkt = (x1+x2)/2
    else:
        if f(midtpunkt) > 0:
            x1 = midtpunkt
            midtpunkt = (x1+x2)/2
        else:
            x2 = midtpunkt
            midtpunkt = (x1+x2)/2

print(midtpunkt)
```
0.1s

3.466959474608302

Figur 17: Gruppe 3 Python-kode som implementerer halveringsmetoden.

Figur 18 gir en visualisering av de ulike resonneringssekvensene som stier i en graf. Det var flere resonneringssekvenser som ble ansett som imitative. Disse førte til forskjellige svar, blant annet ingen svar, analytisk svar og numerisk svar. De stiene som ble kategorisert som kreative resulterte i numeriske svar.



Figur 18: Stiene i grafen representerer resonneringssekvensene som ble observert i Fase 2: finne f(x)=0. En grønnfarget sti betegner imitativt resonnement, og gulfarget sti betegner kreativt resonnement.

4.3 Oppsummering

Det ble observert flest tilfeller av algoritmebehandling. Elevene utviklet og formulerte ulike algoritmer, både på egenhånd og ved å bruke kjente algoritmer og metoder. I Fase 1 ble det observert 4 løsningsalgoritmer som involverte bruk av algoritmebehandling. Alle gruppene endte opp med en algoritme som hadde noen likheter med halveringsmetoden. I Fase 2 ble det også observert totalt 4 algoritmer. Disse inkluderte bruk av kjente metoder som halveringsmetoden og polynomdivisjon, samt mer originale løsningsalgoritmer, som tilpasning av algoritmen fra Fase 1 og bruk av informasjon om fortegnskiftet rundt et nullpunkt.

I Fase 1 delte alle gruppene ned problemet i mindre delproblemer ved å formulere spørsmål for å gjette på riktig tall. Deretter kombinerte de svarene på hvert delproblem for å løse det originale problemet. I Fase 2 benyttet elevene seg primært av tre forskjellige tilnærminger for å dekomponere problemet. Noen av disse ble betraktet som nye for elevene, mens andre ble ansett som kjent dekomposisjon.

Det var få observerte tilfeller av kjennetegnet generalisering. Studien identifiserte tre konkrete tilfeller hvor generalisering kom til uttrykk: når elevene overførte ideen fra halveringsmetoden til Fase 1, når elevene overførte ideen fra algoritmen i Fase 1 til Fase 2, og når elevene tilpasset en algoritme til Python-språk. Generalisering ble ikke observert hos Gruppe 4, verken i Fase 1 eller Fase 2.

Abstraksjon ble også observert i tre tilfeller. Det første tilfellet involverte elevenes evne til å abstrahere bort detaljer om tallets egenskaper i Fase 1. Elevene som forstod essensen av problemet i Fase 2 viste også evne til abstraksjon. Den tredje abstraksjonsprosessen involverte konvertering av en algoritme fra håndskrevet notasjon til programmeringskode. Abstraksjonskomponenten ble ikke observert hos Gruppe 4 i Fase 2.

Det ble gjort mange observasjoner av kjennetegnet evaluering. Alle elevene vurderte effektiviteten av løsningsalgoritmene og reflekterte over forbedringsmuligheter i Fase 1. Med unntak av Gruppe 4, evaluerte elevene også om algoritmene ga ønsket resultat og vurderte nøyaktigheten av resultatene de fikk i Fase 2. Evaluering kom også til uttrykk når elever sammenlignet og vurderte bruk av GeoGebra og programmering som verktøy, med hensyn til nøyaktighet og effektivitet.

Framgangsmåtene som ble observert var av både imitativ og kreativ karakter. I Fase 1 baserte alle løsningene seg først på imitative resonnement. Deretter ble det observert flere kreative framgangsmåter, hvor elevene viste evne til fleksibilitet i tilnærmingen til problemet. I Fase 2 ble det observert et fåtall kreative framgangsmåter. Elevene i disse gruppene klarte å løsrive seg fra kjente metoder, og komme opp med originale løsninger som hadde matematisk forankring. Mange imitative framgangsmåter ble observert i Fase 2. Det ble også observert brukt av ChatGPT, og resonnementene tilknyttet denne framgangsmåten ble ansett som imitative.

5 Diskusjon

I dette kapittelet diskuteres sentrale funn fra analysen med mål om å belyse problemstillingen: «*Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk hos R1 elever i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?*». Funn fra analysen drøftes i lys av tidligere presentert teori.

5.1 Algoritmisk tenkning: en dynamisk prosess

Som det kom fram av funn fra analysen ble alle kjennetegn ved algoritmisk tenkning observert i henhold til Csizmadia et al. (2015) sitt rammeverk. Noen kjennetegn forekom imidlertid i større grad enn andre. Det var spesielt algoritmebehandling, dekomposisjon og evaluering som var de mest fremtredende kjennetegnene.

Det kan være flere mulige grunner til hvorfor nettopp disse ble observert flest ganger. En mulig årsak kan være at elevene hadde mer erfaring med algoritmebehandling, dekomposisjon og evaluering enn de andre kjennetegnene ved algoritmisk tenkning. Det er mulig at tidligere matematikkundervisning har fokusert og lagt til rette for at elevene fikk utviklet disse ferdighetene i større grad. En annen årsak kan være at oppgavene som elevene fikk utdelt, oppfordret spesielt til bruk av disse. For at problemet i Fase 1 og Fase 2 skulle bli mer overkommelig, krevde det at elevene dekomponerte problemet, noe som ble observert i alle gruppene. I Fase 1 var det spesielt i de gruppene hvor elevene ikke benyttet seg av prinsippene fra halveringsmetoden umiddelbart, at dekomposisjon kom mest til uttrykk. Disse gruppene hadde utfordringer med å dekomponere problemet på en effektiv måte i starten, noe som førte til at flere delproblemer ble formulert. For eksempel når elevene stilte spørsmålet om tallet var en- eller tosifret i intervallet [1,20], så eliminerte elevene en god del tall. I neste omgang kunne elevene imidlertid ikke stille det samme spørsmålet om igjen, fordi alle tallene som gjenstod å velge mellom oppfylte det samme kravet om å enten være en- eller tosifret. Elevene måtte dermed formulere nye delproblemer for å eliminere enda flere tall, basert på andre matematiske egenskaper ved tallet. Dette står i motsetning til de elevene som brukte prinsippene fra halveringsmetoden umiddelbart. Elevene dekomponerte problemet i kun ett delproblem, ved å spørre om tallet var høyere, lik eller lavere enn midten.

Videre i oppgaven blir det presisert at elevene skal formulere en algoritme med egne ord og skrive den ned. Som følge av dette måtte elevene utarbeide et sett med instruksjoner for å kunne løse oppgaven både i Fase 1 og Fase 2. Dette kan kanskje også bidra til å forklare at komponenten *algoritmebehandling* ble observert flest ganger totalt. Samtidig må det nevnes at algoritmebehandling mest frekvent forekom i form av å formulere instruksjoner på programmeringsspråket Python. Elevene utviklet og implementerte en sekvens av instruksjoner som behandlet data enten i form av å lagre den, manipulere eller flytte den. Dette kan dermed forklare årsaken til at komponenten algoritmebehandling kom spesielt til uttrykk i Gruppe 1 og Gruppe 3.

Det at oppgavene spesifikt ba elevene om å formulere algoritmer for å løse oppgavene, kan videre knyttes til hvorfor evalueringskomponenten ofte kom til uttrykk. Når elevene selv formulerte og utviklet algoritmer, kan det tenkes at evalueringsprosessen ble mer omfattende og grundig. Det krever at elevene viser en dypere forståelse for instruksjonene i algoritmen og konseptene bak disse. Elevene må evaluere alle stegene i algoritmen for å teste om den gir ønsket resultat. Dette kom til syne i Fase 1 når elevene vurderte hvor effektive og brukervennlige deres algoritmer og strategier var. Det faktumet at alle gruppene endte til slutt opp med en algoritme som brukte prinsippene fra halveringsmetoden, indikerer nettopp at elevene lykkes med dette. Evnen til å evaluere kom også til uttrykk når elevene som fant tilnærmede løsninger til f(x)=0 vurderte x-verdiene som de fikk som svar i Fase 2. Ved å stadig nærme seg nullpunktet vurderte elevene nøyaktigheten til svaret, om løsningsalgoritmen de brukte var effektiv nok, og om det i det hele tatt var mulig å finne en eksakt verdi for nullpunktet. På denne måten diskuterte resultatene sine, noe som igjen førte til at de reflekterte over måter de kunne forbedre algoritmen sin på. Dette førte igjen til at elevene forsøkte å endre på sine eksisterende algoritmer og formulerte nye instruksjoner. Slik vekslet elevene fram og tilbake med å vise evne til evaluering, dekomposisjon og algoritmebehandling. Elevene i de gruppene som brukte programmering for å implementere algoritmene sine, viste ytterligere evne til å evaluere bruk av ulike digitale verktøy. Elevene i disse gruppene diskuterte begrensninger og muligheter knyttet til bruk av GeoGebra og programmering. På denne måten sammenlignet og tolket elevene resultatene, noe som førte til at de fant feil i koden når svarene ikke stemte overens, eller når elevene vurderte nøyaktigheten til svaret, for å deretter formulere nye instruksjoner i form av Python-kode for å oppnå en bedre nøyaktighet. Dette kan være med å forklare hvorfor evalueringskomponenten kom spesielt ofte til utrykk i Gruppe 1 og Gruppe 3. Gruppe 2 forsøkte også å bruke programmering som verktøy, men de møtte på noen utfordringer. Basert på samtalene mellom elevene virket det som om de hadde noe kunnskap om programmering, men at de strevde med å formulere en konkret kode for å implementere algoritmen deres. De hadde blant annet problemer med syntaksen[1], og klarte heller ikke å tolke feilmeldingene. Å overføre løsningsalgoritmen til et programmeringsspråk kan dermed by på utfordringer dersom eleven ikke er kjent med det tilhørende programmeringsspråket. Dette kan ha hindret elevene i å vise flere tegn til evalueringsevne.

En annen trend fra resultatene som er verdt å diskutere i sammenheng med evaluering er det store spriket innad i gruppene. Resultatene viser at evalueringskomponenten ble observert i veldig liten grad hos elevene i Gruppe 4. Det er derfor interessant å undersøke hvorfor dette var tilfelle. Elevene i gruppen hadde en annen tilnærming for å løse problemet med å finne f(x) = 0 i Fase 2. De hadde utfordringer med å komme opp med en algoritme for å finne en tilnærmet verdi av nullpunktet i oppgave 2), hvor funksjonsuttrykket ikke var oppgitt. Elevene ga uttrykk for at de ikke forstod hva oppgaven spurte om og forsøkte å overhøre hva de andre gruppene diskuterte. Kanskje elevene ikke var vant til å møte matematiske problemer på denne måten, eller at de hadde manglende kunnskaper i matematikk? Dette kan ha begrenset deres evne til å løse oppgaven og videre formulere en løsningsalgoritme. Når elevene gikk videre til oppgave 3) hvor et konkret funksjonsuttrykk ble gitt, skrev de inn feil funksjonsuttrykk i GeoGebra. Det nye funksjonsuttrykket hadde nå et skjæringspunkt i x=1, som gjorde at elevene benyttet seg av polynomdivisjon for å komme fram til et analytisk svar. Det kan tenkes at fordi elevene brukte en kjent prosedyre og fikk et analytisk svar, at de ikke anså det som relevant å vurdere i hvor stor grad svaret var tilfredsstillende. På den andre siden, så viste elevene også liten evne til å evaluere algoritmen sin i Fase 1, ved at de for eksempel ikke tok hensyn til om tallet som den andre tenkte på kunne være i midten av tallintervallet.

Det at elevene skulle løse to spesifikke oppgaver kan ha begrenset i hvor stor grad komponentene *generalisering* og *abstraksjon* kunne komme til uttrykk. En av hensiktene med oppgavene var å legge opp til at elevene kunne overføre noen av ideene fra Fase 1 til Fase 2, og på denne måten tilpasse løsninger slik at de kan gjelde i tilsvarende problemer. Det var dermed forventet å observere generaliseringskomponenten i liten grad, ettersom elevene løste to konkrete oppgaver innenfor begrensede matematiske områder. Ikke alle gruppene lyktes imidlertid med å identifisere fellestrekk mellom Fase 1 og Fase 2. Elevene i Gruppe 4 hadde utfordringer med å se hvordan problemet med å finne f(x) = 0 kan ha likhetstrekk med Gjett-

[1]Syntaks er de reglene som styrer hvordan setninger og fraser kan bygges opp i et språk. Syntaksen bestemmer for eksempel hvilken ordstilling vi har i setninger og hvilken form et ord kan ha når det opptrer i relasjon til andre ord (Hagemann, 2021).

tall aktiviteten. Det er vanskelig å peke nøyaktig på hva som er årsaken til det. På den ene siden kan det tenkes at problemet stilte for høye kognitive krav til elevene. Elevenes tidligere erfaringer med og ferdigheter i matematikk kan igjen ha spilt en rolle i denne sammenhengen. Dette samsvarer i så fall med det Mason (1996) sier, om at mønstergjenkjenning er en forløper til generalisering, som igjen er grunnlaget for matematisk tenkning. På den andre siden så er en av arbeidsmåtene i den algoritmiske tenkeren «å holde ut». I denne sammenhengen vil det si å vise evne til å vedvare og fortsette å arbeide med oppgaven, selv når det blir vanskelig og komplekst. Elevene i gruppen ga nokså raskt opp med å forsøke å komme opp med en egen løsningsalgoritme og brukte mer tid på å diskutere temaer som ikke var faglig relevante enn de andre gruppene. Det kan dermed tenkes at hvis elevene ikke hadde gitt opp umiddelbart, men heller brukt litt mer tid på oppgaven, at de etter hvert hadde begynt å se mønstre og fellestrekk mellom Fase 1 og Fase 2. Elevene i Gruppe 2 hadde for eksempel utfordringer med å komme opp med en løsningsalgoritme i Fase 2 i starten, men de viste evne til å holde ut ved å stadig vende tilbake til problemet. Det førte til at elevene til slutt klarte å overføre noen av ideene fra Fase 1 til Fase 2 ved å systematisk sjekke hvilken del av intervallet nullpunktet lå i, ved å studere fortegnet til funksjonsverdier på hver side av nullpunktet. Dette kan underbygge påstanden om at det kreves en viss tålmodighet og utholdenhet for at kjennetegn ved algoritmisk tenkning skal komme til uttrykk. Utdanningsdirektoratet omtaler dette som «kognitiv-kondis» (Utdanningsdirektoratet, 2019, s.2). På samme måte som med generalisering, var det videre forventet at antall observasjoner av abstraksjonskomponenten ville være lavt, gitt oppgaveformuleringen. Resultatene antyder imidlertid at det eksisterer en sammenheng mellom abstraksjonskomponenten og generaliseringskomponenten. I alle gruppene hvor abstraksjon kom til uttrykk, ble også generaliseringskomponenten observert. Dette er i samsvar med at abstraksjon legger grunnlaget for generalisering (Bocconi et al., 2016). Abstraksjon handler om å fokusere på de viktigste elementene ved å fjerne unødvendig informasjon. På denne måten kan man lettere indentifiserer mønstre og fellestrekk, slik at generalisering kan komme til uttrykk.

Basert på det som har blitt diskutert til nå ser det ut til at det er mulig til en viss grad å styre hvilke kjennetegn som kommer til uttrykk gjennom utformingen av oppgavene. I undervisningssammenheng kan det bety at læreren kan legge til rette for progresjon og utvikling av algoritmisk tenkning, ved å bevisst velge oppgaver som legger vekt på å utvikle spesifikke kjennetegn. Dette kan bidra til å gjøre arbeidet med å integrere algoritmisk tenkning i undervisningen mer klart og tydelig. Samtidig kan det være utfordrende å velge oppgaver som er innenfor alle elevene sitt kompetansenivå. Lærerens evne til å tilpasse undervisningen kan dermed spille en avgjørende rolle i å sikre at alle elever får muligheten til å uttrykke og utvikle sine ferdigheter innen algoritmisk tenkning. I tillegg ser det ut til at de forskjellige aspektene ved algoritmisk tenkning er sammenkoblet. Å kunne dekomponere et problem effektivt krever for eksempel abstraksjonsevne for å identifisere de viktigste aspektene av et problem. Abstraksjon ser igjen ut til å kunne påvirke elevenes evne til generalisering. Algoritmebehandling og evaluering i konteksten av denne studien ser også ut til å påvirke hverandre. Videre indikerer resultatene at programmering kan være et godt verktøy for å få fram kjennetegn ved algoritmisk tenkning, til tross for at denne studien har betraktet algoritmisk tekning som en prosess som ikke nødvendigvis trenger å skje innenfor en programmeringskontekst. Dette støtter ideen om at algoritmisk tenkning kan komme til uttrykk og utvikles i flere sammenhenger (Bocconi et al., 2016). Samtidig ble det observert en stor forskjell i elevenes programmeringsferdigheter. Som lærer bør man derfor reflektere over hvordan arbeidet med programmering kan tilpasses forskjellige nivå og behov, på en slik måte som bidrar til å utvikle elevenes algoritmiske tenkning. I LK20 brukes programmering innenfor en rekke andre fag slik som fysikk, kjemi, kunst og håndverk og musikk, og kan dermed tenkes å bidra til tverrfaglighet og en helhetlig forståelse av hvordan teknologi kan brukes i ulike fagfelt. Disse resultatene tyder dermed på at algoritmisk tenkning er en dynamisk og fleksibel prosess. Dette kan underbygge oppfatning om at algoritmisk tenkning er en kompleks prosess som har vært vanskelig å definere.

5.2 Å løsrive seg fra kjente algoritmer og metoder

Framgangsmåtene som ble observert i denne studien var både av imitativ og kreativ karakter, og forekomsten av disse var ulik innad i gruppene. Det ble ikke observert kreative framgangsmåter hos Gruppe 1 verken i Fase 1 eller Fase 2. Dette kan forklares med at deres framgangsmåter baserte seg på imitative resonnement. Disse gikk ut på at elevene umiddelbart gjenkalte en kjent metode, og at de videre benyttet seg av denne både i Fase 1 og Fase 2. Problemet med å finne en tilnærmet verdi for nullpunktene til f(x)=0 i oppgave 2) var dermed ikke nytt for elevene. På den ene siden kan man derfor ikke lenger referere til oppgaven som et problem, men heller en rutineoppgave. På den andre siden klarte ikke elevene å løsrive seg fra den kjente metoden, noe som kan tyde på at elevenes resonnement var lite fleksibel. Det kan tenkes at det er utfordrende å komme opp med ny løsning når man allerede kjenner en fra før. Det er flere mulige grunner til hvorfor dette kan være tilfelle. Ved å følge en allerede etablert og kjent tilnærming, vil elevene være trygge på at den vil fungere og løse problemet. Dermed trenger ikke elevene å tenke utenfor boksen for å finne alternative løsninger. På denne måten er elevene mindre tilbøyelig for å utforske andre måter å løse problemet på. Ved at elevene i Gruppe 1 ble fiksert på den ene framgangsmåten, kan man si at de ble værende i samme problemrom (Kaufmann, 2006). Dette kan tyde på at dersom man gjenkaller en kjent løsningsmetode, kan man blokkere framdriften av kreative ideer. I tillegg kan sammensettingen av gruppen kan ha spilt en rolle. Kreativitet kan ifølge litteraturen bli påvirket av omgivelsene. En av elevene i gruppen forsøkte å komme opp med nye ideer for å finne f(x)=0, men ble oversett av den andre eleven. Hvordan samarbeidet foregikk kan dermed ha påvirket utviklingen av kreative ideer. Samtidig er det få alternative måter å løse oppgaven på med tanke på det konkrete funksjonsuttrykket som elevene fikk. En av disse er Newtons metode som også nå er pensum i R1 (Borgan et al., 2021). Metoden er likevel ganske komplisert, og elevene hadde trolig nok ikke klart å komme fram til denne metoden på egen hånd i løpet av den korte tiden de fikk til rådighet. Disse erfaringene er noe jeg kommer til å ta med meg videre.

I denne studien utfoldet derimot mange av de grunnleggende elementene i den kreative prosessen seg, i de gruppene som ikke gjenkalte en kjent løsningsmetode umiddelbart. Elevene som kom fram til kreative framgangsmåter i Fase 1 viste evne til å komme opp med originale løsninger. Deres framgangsmåter baserte seg først på imitative resonnement som innebar usystematisk gjetting og bruk av lineær søkealgoritme. Etter en stund med prøving og feiling utarbeidet elevene nye framgangsmåter for å løse problemet som var av kreativ karakter. Disse framgangsmåtene var plausible og hadde matematisk forankring. At elevene nærmet seg problemet på ulike måter, kan være et tegn på divergent tenkning. I tillegg kan dette funnet tyde på at veien til kreative løsninger innebærer en rekke imitative resonnement. Dette er i tråd med det som omtales som den forberedende fase i Gestalt-modellen (Sriraman, 2009), og at tidligere erfaringer som skal danne et passende miljø for at kreative ideer skal kunne komme til uttrykk (Ervynck, 1991). Videre i Fase 2 ble kreative framgangsmåter kun observert i Gruppe 2 og Gruppe 3. Deres framgangsmåter baserte seg heller ikke på bruk av kjent metoder eller løsningsalgoritmer. Gjennom å oppdage fellestrekk mellom løsningen i Fase 1 klarte elevene å komme opp med en algoritme som bare preg av originalitet for å finne f(x)=0.

Framgangsmåten som ble observert hos elevene i Gruppe 4 var på den andre siden av imitativ karakter. Elevene forsøkte å høre etter hva de andre gruppene diskuterte. De forsøkte å gjette seg fram til hva jeg som forsker forventet at de skulle gjøre, og gjentok selv flere ganger at de «ikke klarte å tenke utenfor boksen». Videre så elevene etter hint i oppgavehefte, noe som førte til at de fant et konkret funksjonsuttrykk i oppgave 3). Når elevene tastet inn feil funksjonsuttrykk i GeoGebra, benyttet deg seg av en kjent metode (polynomdivisjon) for å finne svar på ligningen f(x)=0. Ettersom elevene lyktes med å komme opp med kreative framgangsmåter i Fase 1, kan det se ut til at elevene manglet noen fagspesifikke kunnskaper i Fase 2 for å kunne frigjøre potensialet for kreativitet. I denne konteksten kan det også være mulig at elevenes intuisjon spilte en rolle i prosessen med kreative framgangsmåter. Både Gruppe 2 og Gruppe 3 handlet på en intuitiv basis for å overføre noen av prinsippene fra Fase 1 til Fase 2. Elevene bestemte seg for å trå til med noe som hadde nokså uforutsigbare konsekvenser. Dette kom til uttrykk når elevene foreslo at de kanskje kunne bruke algoritmen fra Fase 1 i Fase 2, men uten å ha noen logiske begrunnelser eller argumenter som støttet det. Intuisjon var i dette tilfelle nyttig når elevene skulle finne tilnærminger til et komplekst problem. Elevenes intuisjon alene og evnen til å foreslå nye ideer var likevel ikke tilstrekkelig for å løse problemet. Elevene videreførte sine intuitive tanker og formulerte en plausibel løsningsalgoritme som sluttprodukt. Elevene i Gruppe 2 omformulerte for eksempel problemet med å finne f(x)=0 til «Hva er felles for alle nullpunktene?» i et forsøk på å tilpasse algoritmen fra Fase 1 til å gjelde i Fase 2. Som tidligere nevnt er reformulering av et problem et viktig element i kreativ problemløsning (Kaufmann, 2006). Evnen til å omformulere et problem kjennetegner også dekomposisjonskomponenten i algoritmisk tenkning. På denne måten kom også flere av kjennetegn ved algoritmisk tenkning til uttrykk, blant annet gjennom at elevene dekomponerte problemet og videre formulerte instruksjoner for å omgjøre de kreative ideene til en konkret algoritme. Dersom man ønsker å gjøre store og kreative sprang kan det derfor se ut til at det kan lønne seg å handle intuitivt og ikke nødvendigvis rasjonelt. En annen mulig forklaring til hvordan elevene kom fram til kreativ løsning kan være knyttet til inkubasjon. Det gikk nokså lang tid fra elevene i Gruppe 2 møtte på problemet i Fase 2, til de kom opp med en løsningsalgoritme. I denne problempausen kan det tenkes at elevene underbevist bearbeidet problemet. Dette kan minne om den kreative problemløsningsprosessen hvor Ervynvk (1991) og Sriraman (2009) foreslår at intuisjon og inkubasjon spiller en viktig rolle for kreativ opplysning. Til tross for at det i Gruppe 1 ikke ble observert kreative framgangsmåter for å løse oppgavene som elevene fikk tildelt, er det likevel verdt å nevne at en av elevene viste evne til å dekomponere et mindre delproblem på en kreativ måte. Utdrag 7 som ble presentert i delkapittel 4.1.2 viser hvordan en av elevene forsøker å komme opp med en måte å definere en vilkårlig tredjegradsfunksjon på Python-språk. Elevkommentaren «Hvilken som helst..Altså det går jo kanskje an å gjøre det ehm..» kan tolkes som en veldig kort inkubasjonstid. Det resulterte i at eleven foreslo å definere koeffisientene hver for seg, noe som var en original og plausible måte å definere en tredjegradsfunksjon ved hjelp av programmering.

Hvorvidt det er en direkte sammenheng mellom algoritmisk tenkning og kreativitet er det vanskelig å fastslå. Det foreligger imidlertid noen funn som antyder en potensiell kobling mellom disse to. En av framgangsmåtene som involverer kategorisering av tall for å gjette riktig tall i Fase 1, ble identifisert som kreativ. Dette krevde at elevene demonstrerte evnen til dekomposisjon, et kjennetegn ved algoritmisk tenkning. Dekomposisjon dannet også grunnlaget for en kreativ tilnærming i Fase 2, der elevene stilte spørsmålet «Hva er likt for alle nullpunktene?». Dette førte til oppdagelsen av at funksjonsverdien rundt hvert nullpunkt hadde motsatt fortegn, noe som videre resulterte i en original løsning. Kjennetegnet abstraksjon var også til stede i dette tilfelle. Framgangsmåten hvor elevene selv kom fram til prinsippene bak halveringsmetoden for å løse oppgaven i Fase 1, ble også ansett som kreativ. I denne sammenhengen var kjennetegnet evaluering spesielt fremtredende, ettersom elevene vurderte sine tidligere framgangsmåter for å utvikle en mer effektiv løsning. Videre forsøkte de samme elevene å overføre og tilpasse algoritmen fra Fase 1 til Fase 2. Kjennetegnet generalisering og algoritmebehandling var særlig til stede i realiseringen av denne kreative ideen. Resultatene peker altså på at algoritmisk tenkning kan være nyttig innenfor kreativ problemløsning for å legge grunnlaget for eller videreutvikle innovative ideer.

5.3 ChatGPT: forbud eller verktøy?

I Fase 2 ble det observert at elevene i Gruppe 4 benyttet seg av ChatGPT for å generere en ferdig Python-kode som implementerte halveringsmetoden. Elevene hadde utfordringer med å forstå hvordan de kunne bruke den ferdige koden videre, og kom dermed ikke fram til et svar. I konteksten av denne studien ble denne framgangsmåten vurdert til å være av imitativ karakter, ettersom den ikke oppfylte noe av kriteriene til kreativ resonnering. Dette funnet reiser flere spørsmål som er verdt å diskutere i undervisningssammenheng. Kan ChatGPT hemme kreativitet i matematikkfaget? Bør kunstig intelligens være en del av undervisningen? Hvordan kan det i så fall integreres på en måte som gir produktiv læring? Kunstig intelligens er en teknologi som har utviklet seg i et høyt tempo, og kunstig intelligens som ChatGPT har vært et stort samtaleemne det siste halvåret. Meningene rundt ChatGPT i skolesammenheng ser imidlertid ut til å være todelt. Noen mener at ChatGPT bør forbys, mens andre mener at det kan være et nyttig verktøy. NTNU har blant annet inkludert bruk av ChatGPT i regelverket som juks, med mindre eksamensoppgavene åpner for bruk av dette (Elverum, 2023). Bill Gates er en av dem som mener at kunstig intelligens kan være en stor ressurs når det gjelder hvordan undervisning og læring foregår. I sin artikkel *The Age of AI has begun* skrev han følgende: «..I think in the next five to 10 years, AI-driven software will finally deliver on the promise of revolutionizing the way people teach and learn» (Gates, 2023). Dette kan minne om Papert (1980) sin idé om hvordan bruk av teknologi og datamaskinen vil endre måten mennesker lærer og tenker på.

På den ene siden kan det argumenteres for at bruk av ChatGPT kan hemme elevenes evne til å være kreative og tenke nyskapende. Potensielt kan bruk av ChatGPT føre til at elevene blir avhengig av å bruke denne teknologien til å få ferdige løsninger og svar. Dette kan redusere elevenes evne til å være i stand til å evaluere løsningsforslagene på egenhånd. Veien til svaret og diskusjonene underveis har en stor verdi i læringsprosessen. Hvis elevene hele tiden blir gitt svar uten å resonnere selv, kan det tenkes at elevenes kreative tenkning og problemløsningsevne blir svekket. På denne måten kan det også tenkes at elevenes mulighet til å utforske matematiske konsepter og sammenhenger blir begrenset. Dermed vil ikke elevene nødvendigvis få den samme erfaringen med matematikkfaget og utvikle den samme graden av intuisjon og forståelse. Dette kan potensielt føre til at elevene går glipp av muligheten til å forstå matematikken på et dypere nivå og ikke være i stand til å anvende kunnskapen sin i nye og komplekse problemer.

På den andre siden er kunstig intelligens allerede en integrert del i samfunnet. Man blir møtt av kunstig intelligens baserte chatbotter på en rekke nettsteder i form av kundeservicetjenester, og på sosiale plattformer slik som Snapchat. Dermed er det naturlig å anta at kunstig intelligens vil fortsette å være en viktig del av hverdagen til mange, og ha en innvirkning på arbeidslivet framover. Om undervisningen skal bidra til å gjøre elevene bedre rustet til å møte framtidenes utfordringer, kan det dermed tenkes at ferdigheter og kunnskap om kunstig intelligens kanskje vil være det neste kompetansebehovet i det 21.århundre. Det som vil da være avgjørende er hvordan denne teknologien kan integreres på en trygg måte og hensiktsmessig måte. Dette vil stille mange krav knyttet til skolevesenet og ikke minst til lærerrollen. Om ChatGPT skal integreres i skolen, bør vurderingsformen vurderes også, slik at den kan legge til rette for bruk av ChatGPT, på samme måte som den legger til rette for programmering og GeoGebra. Lærerne bør også være åpen for å utforske nye måter å videreutvikle sin undervisningspraksis på. Det krever at lærerne vurderer og evaluerer hensikten med teknologiske hjelpemidler i relasjon til konteksten de blir brukt i. ChatGPT kan for eksempel bli brukt til å generere ulike funksjoner, som elevene kan sammenligne og på denne måten identifisere forskjeller og likheter på egenhånd. Et annet eksempel er å bruke ChatGPT til å generere ulike datasett, som elevene kan analysere og framstille på ulike måter. I programmeringssammenheng kan ChatGPT bli brukt til å genere en kode som inneholder feil. Elevene kan evaluere og feilsøke koden, og videre formulere instruksjoner på det tilhørende programmeringsspråket for å rette koden. En slik interaktiv arbeidsform kan legge til rette for utforskende og kreative tilnærminger i matematikkfaget. Samtidig får elevene muligheten til å sammenligne og diskutere sine framgangsmåter og svar med hverandre. Dette kan oppfordre elevene til å tenke kritisk og reflektere rundt sine egne og andres standpunkter. Disse diskusjonene kan bli ledet av læreren for å sikre fruktbare og produktive diskusjoner.

6 Konklusjon

Problemstillingen for denne studien er: «*Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk hos R1 elever i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?*».

Alle kjennetegn ved algoritmisk tenkning beskrevet i Csizmadia et al. (2015) sitt rammeverk ble observert i denne studien. Forekomsten av de ulike kjennetegnene varierte innad i de forskjellige gruppene som deltok i studien. *Algoritmebehandling*, *dekomposisjon* og *evaluering* var de mest frekvente kjennetegnene som kom til uttrykk. Funnene antyder at elevenes tidligere erfaringer og matematiske ferdigheter kan være en viktig faktor, når det gjelder i hvor stor grad kjennetegnene ved algoritmisk tenkning kommer til uttrykk. Studiens funn indikerer at det også er mulig i stor grad å styre hvilke kjennetegn som kommer til uttrykk gjennom utformingen og formulering av oppgavene. Det er viktig at elevene får muligheten til å beskjeftige seg med matematiske ideer og konsepter gjennom aktive arbeidsformer som lar elevene utforske og diskutere. Samtidig bør oppgavene utformes på en slik måte som tilpasses elevenes kompetansenivå og deres «kognitive kondis». Dermed kan det tenkes at læreren vil spille en avgjørende rolle når det gjelder å legge til rette for at elevene får arbeide med matematiske problemer på en måte som kan utvikle deres algoritmiske tenkning.

Denne studien betraktet algoritmisk tekning som en prosess som ikke nødvendigvis trenger å skje innenfor en programmeringskontekst. Likevel er det viktig å påpeke at resultatene indikerer at programmering kan være et godt verktøy for å få fram kjennetegn ved algoritmisk tenkning. Alle kjennetegn ved algoritmisk tenkning kom til uttrykk i de gruppene som benyttet seg av programmering som verktøy. I tillegg foreslår resultatene at kjennetegn ved algoritmisk tenkning ikke forekommer kronologisk i en lineær prosess. Det ser ut til at de ulike kjennetegnene kan påvirke hverandre, og i denne studien ble dette spesielt observert mellom kjennetegnene evaluering, dekomposisjon og algoritmebehandling. Elevene vekslet fram og tilbake mellom disse kjennetegnene når de kontinuerlig evaluerte løsningene sine i ønske om å forbedre disse. Dette førte til oppdagelsen av nye delproblem, som igjen oppfordret elevene til å formulere konkrete instruksjoner for å løse disse. Resultatene peker også på at algoritmisk tenkning kan være nyttig innenfor kreativ problemløsning, på den måten at det kan legge grunnlaget for, og videreutvikle de kreative ideene. Med utgangspunkt i disse antakelsene kan det argumenteres for at algoritmisk tenkning er dynamisk og fleksibelt, noe som kan utnyttes tverrfaglig.

Studiens funn peker på at kreative framgangsmåter kjennetegnes ved at elevene klarte å frigjøre seg fra kjente løsningsalgoritmer og metoder. Elevene viste evne til å komme opp med originale løsninger som var plausible og hadde matematisk forankring. Gjennom en del prøving og feiling klarte elevene å nærme seg matematiske problemer på ulike måter, noe som tyder på at evnen til fleksibilitet og divergent tenkning er viktig i den kreative utviklingsfasen. Resultatene indikerer også at veien til kreative løsninger kan innebære en rekke imitative resonnementer, ved at tidligere erfaringer kan danne et passende miljø for at kreative ideer skal kunne utfolde seg. Elevenes framgangsmåter viste også spor av at faktorer som intuisjon og inkubasjon kan spille en rolle i den kreative problemløsningsprosessen. De tilfellene hvor elevene handlet på intuitiv basis, framfor å følge logiske og rasjonelle slutninger, resulterte i kreative løsninger. I gruppene hvor det ikke ble observert kreative framgangsmåter, baserte elevenes framgangsmåter seg på imitative resonnementer og bruk av kjente metoder. Manglende fleksibilitet og begrenset evne til å utforske alternative løsninger, ser dermed ut til å hemme framdriften av kreative framgangsmåter. Vanskelighetsgraden på oppgavene kan også se ut til å påvirke elevenes evne til å være kreative. En oppgave hvor elevene umiddelbart kjenner igjen en løsningsmetode kan begrense elevene til å utforske andre tilnærminger. En altfor utfordrende oppgave kan på den andre siden hemme potensiale for kreativitet, da elevene kan mangle noen fagspesifikke kunnskaper for å løse oppgaven. Utformingen og formulering av oppgavene ser altså ut til å kunne påvirke graden av kreativitet.

Bruk av kunstig intelligens kan potensielt hemme kreativitet i matematikkfaget. Funn fra studien viser at ChatGPT ble brukt i framgangsmåten til noen av elevene. Dette førte til en imitativ løsning av det matematiske problemet. Om kunstig intelligens skal integreres i skolen, bør man vurdere muligheter og begrensninger som det kan medføre undervisningen. Det vil være viktig å finne en balanse mellom bruken av denne type teknologi, samtidig som man sikrer at elevene utvikler de nødvendige ferdighetene og forståelsen i matematikkfaget på en selvstendig og kreativ måte. Dette krever refleksjon, tilpasning og kontinuerlig evaluering av undervisningsmetoder og anvendelsen av slike verktøy.

6.1 Kritisk refleksjon og veien videre

I denne studien har jeg betraktet algoritmisk tenkning i henhold til Csizmadia et al. (2015) sitt rammeverk. Det kan derfor tenkes at resultatene hadde blitt annerledes om jeg hadde benyttet meg av et annet rammeverk. En annen faktor som potensielt kunne ha påvirket resultatene, er om elevene i pausen diskuterte oppgavene med elever fra andre grupper. Jeg fant foreløpig ingen indikasjoner på dette, men det er likevel vanskelig å utelukke dette med sikkerhet. Det at elevene fikk utdelt alle oppgavene samtidig, kan ha hatt innvirkning på resultatene for elevene i Gruppe 4. Dersom elevene ikke hadde bladd videre i oppgaveheftet, er det mulig at de ville ha kommet fram til alternative løsningsforslag etter hvert. Gruppesammensetningen kan også ha hatt innvirkning på dynamikken og interaksjonen mellom deltakerne i studien. I noen grupper ble det observert at visse elever var mer aktive enn andre. Dette førte til at forslagene fra de passive elevene av og til ble oversett, og i enkelte tilfeller dominerte de mer aktive elevene fullstendig diskusjonen. Dersom gruppesammensetningen hadde vært annerledes, kunne diskusjonen ha tatt en annen retning, og dermed påvirket studiens resultater. Samtidig er det ikke sikkert at gruppesammensetningen ville ha hatt en betydelig innvirkning på studien, dersom begrensningen lå i elevenes matematiske kunnskaper.

Et større utvalg kunne ha styrket oppgavens validitet, men studiens omfang og tidsramme har vært begrensende faktorer. Denne studien har imidlertid vært en begynnelse på å undersøke kjennetegn ved algoritmisk tenkning og kreative framgangsmåter. For veien videre ville det vært spennende å gjennomføre en lignende studie over en lengre periode, for å undersøke om flere kjennetegn ved algoritmisk tenkning kom til uttrykk med tanke på inkubasjonstiden i den kreative problemløsningsprosessen. Det hadde også vært interessant å utforske i mer detalj hva intuisjon i matematikk innebærer, og om det skiller seg fra intuisjon i andre sammenhenger. En annen retning for videre studier kan være å utforske grundigere forholdet mellom algoritmisk tenkning og matematisk tenkning. Andre ideer til videre studier kan være å undersøke hvordan algoritmisk tenkning kan bidra til å danne en bro mellom matematikkfaget og programmering. Med tanke på den raske utviklingen av kunstig intelligens, kunne det også være spennende å undersøke om algoritmisk tenkning og kreativitet kan utvikles ved hjelp av kunstig intelligens som ChatGPT.

Referanser

Andresen, M. & Brun, T. (2015). Matematisk kreativitet i problemløsning. Presentasjon av forskningsprosjektet “Elevstrategier”. Universitetet i Bergen, ISBN 978829216050-3.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework- Implications for Teacher Knowledge. *Educational Technology & Society*, 19(3), 47–57.

Balanskat, A. & Engelhardt, K. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. *European Schoolnet.* Technical report, Brussels.

Barr, V. & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48–54.

Bergqvist, T. & Lithner, J. (2005). Simulating creative reasoning in mathematics teaching. Department of Mathematics and Mathematical statistics, Umeå University, *Research Reports in Mathematics Education*, no. 2.

Bicer, A., Chamberlin, S. & Perihan, C. (2020). A Meta-Analysis of the Relationship between Mathematics Achievement and Creativity. *Journal of Creative Behavior*, *2021* (55)3:569–590. https://doi.org/10.1002/jocb.474

Bjørndal, C. R. P. (2017). *Det vurderende øyet. Observasjon, vurdering og utvikling i pedagogisk praksis.* Gyldendal akademisk.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A. & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*; EUR 28295 EN; https://doi.org/10.2791/792158

Bocconi, S., Chioccariello, A. & Earp, J. (2018). *The Nordic approach to introducing Computational Thinking and programming in compulsory education*. Report prepared for the Nordic@BETT2018 Steering Group. https://doi.org/10.17471/54007

Borgan, Ø., Borge, I. C, Engeseth, J., Heir, O., Moe, H., Norderhaug, T. T. & Vie, S. M. (2021). *Matematikk R1* (4.utg.). Aschehoug Co.

Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at the 2012 annual meeting of the American Educational Research Association, Vancouver.

Bråting, K., Kilhamn, C., & Rolandsson, L. (2020). Integrating programming in Swedish school mathematics: description of a research project. Paper presented at the twelfth research seminar of the Swedish Society for Research in Mathematics Education, MADIF12, Linnaeus University, Växjö.

Bråting, K. & Kilhamn, C. (2021) Exploring the intersection of algebraic and computational thinking, Mathematical Thinking and Learning, 23:2, 170-185. https://doi.org/10.1080/10986065.2020.1779012

Bueie, H. (2019). *Programmering for matematikklærere.* Oslo: Universitetsforlaget.

Cansu, S. K. (2019) International Journal of Computer Science Education in Schools, April 2019, Vol. 3, No. 1 ISSN 2513-8359: https://files.eric.ed.gov/fulltext/EJ1214682.pdf

Chamberlin, S. A., & Moon, S. M. (2005). Model-eliciting activities as tool to develop and identify creativity gifted mathematicians. *Journal of Secondary Gifted Education*, 17(1), 37–47.

Christiansen, T. J & Mathisen, R. (2020). *Å finne nullpunkt med halveringsmetoden.* Github. https://github.com/fagstoff/Skolekoden/blob/master/docs/undervisningsopplegg/Nullpunkt_halveringsmetoden/Nullpunkt_halveringsmetoden.md.

Christoffersen, L. & Johannessen, A. (2012). *Forskningsmetode for lærerutdanningene*. Oslo: Abstrakt forlag.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). *Computational thinking-A guide for teachers*. Report, University of Southampton. https://eprints.soton.ac.uk/424545/

Denning, P. J. (2009). The profession of IT Beyond computational thinking. Communications of the ACM, 52(6), 28-30.

Elverum, H. (2023, 20.februar). *Nå kommer NTNU med «KI-sikre» eksamener*. Underdusken.no. https://underdusken.no/chatgpt-kunstig-intelligens-ntnu/na-kommer-ntnu-med-ki-sikre-eksamener/311394

Ervynck, G. (1991). Mathematical creativity. I D. Tall, *Advanced mathematical thinking,* Kluwer Academic Publishers New York, 42-52.

Evans, B. T. (2017). *Thinking and Reasoning: A Very Short Introduction*. Oxford University Press.

Gates, B. (2023, 21.mars). *The Age of AI has begun.* Gatesnotes. https://www.gatesnotes.com/The-Age-of-AI-Has-Begun

Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten–tidsskrift for matematikkundervisning, 30*(3), 31-37.

Grégoire, J. (2016). Understanding Creativity in Mathematics for Improving Mathematical Education. *Journal of Cognitive Education and Psychology, 15*(1), 24-36. https://doi.org/10.1891/1945-8959.15.1.24

Grover, S. & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, *42* (21), 38-43. https://doi.org/10.3102/0013189X12463051

Hagemann, K. (2021, 29.november). *Syntaks*. Store Norske Leksikon. https://snl.no/syntaks

Hannemyr, G. (2022, 9.februar). *Datamaskin*. Store norske leksikon. http://snl.no/datamaskin

Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a new 21st century skill-computational thinking: Concepts and trends. *International Education Studies,* 11(4), 29–42. https://doi.org/10.5539/ies.v11n4p29

Haylock, D. W. (1987). A Framework for Assessing Mathematical Creativity in Schoolchildren. *Educational Studies in Mathematics,* 18(1), 59-74.

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021a). The Associations Between Computational Thinking and Creativity: The Role of Personal Characteristics. *Journal of Educational Computing Research*, *58*(8), 1415–1447. https://doi.org/10.1177/0735633120940954

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021b). A Log-Based Analysis of the Associations Between Creativity and Computational Thinking. *Journal of Educational Computing Research*, *59*(5), 926–959. https://doi.org/10.1177/0735633120973429

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, 23(2), 159–187. https://doi.org/10.1080/14794802.2020.1852104

Kaufmann, G. (2006). *Hva er kreativitet.* Universitetsforlaget.

Kravik, R., Berg, T. K. & Siddiq, F. (2022). Teachers' understanding of programming and computational thinking in primary education – A critical need for professional development. *Acta Didactica Norden, 16*(4). https://doi.org/10.5617/adno.9194

Kristensen, T. E., & Kirfel, C. (2022). *Programmering i matematikkfaget.* Cappelen Damm Akademisk.

Kunnskapsdepartementet. (2018). Retningslinjer for utforming av nasjonale og samiske læreplaner for fag i LK20 og LK20S. https://www.regjeringen.no/contentassets/3d659278ae55449f9d8373fff5de4f65/retningslinjer-for-utforming-av-nasjonale-og-samiske-lareplaner-for-fag-i-lk20-og-lk20s-fastsatt-av-kd.pdf

Kvale, S. & Brinkmann, S. (2009). *Det kvalitative forskningsintervju* (2.utg). Gyldendal. Oslo.

Liljedahl, P., & Sriraman, B. (2006). Musings on mathematical creativity. *For The Learning of Mathematics*, 26(1), 17-19.

Lithner, J. (2008). A Research Framework for Creative and Imitative Reasoning. *Educational Studies in Mathematics*, *67*(3), 255–276. http://www.jstor.org/stable/40284656

Lodi, M. & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Sci & Educ* 30, 883–908. https://doi.org/10.1007/s11191-021-00202-5

Mason, J. (1996). Expressing Generality and Roots of Algebra. I N. Bernarz, C. Kiera & L. Lee (Red.), *Approaches to Algebra: Perspectives for research and teaching*. Dordrecht: Springer, 65-86. https://doi.org/10.1007/978-94-009-1732-3_5

Nadjafikhah, M., Yaftian, N., & Bakhshalizadeh, S. (2011). Mathematical creativity: Some definitions and characteristics. *Procedia-Social and Behavioral Sciences*, 31, 285–291.

Nordbakke, M. (2018). Utvikling av kjerneelementer. *Tangenten – tidsskrift for matematikkundervisning*, 29(4), 35-40.

Nordby, S.K., Bjerke, A. H. & Mifsud, L. (2022). Primary Mathematics Teacher's Understanding of Computational Thinking. *KI - Künstliche Intelligenz* (2022) 36:35–46. https://doi.org/10.1007/s13218-021-00750-6

NOU 2020: 2. (2020). *Fremtidige kompetansebehov III - Læring og kompetanse i alle ledd.* Utgiversted: Kunnskapsdepartementet. Hentet fra: https://www.regjeringen.no/contentassets/053481d65fb845be9a2b1674c35d6d14/no/pdfs/nou202020200002000dddpdfs.pdf

Nouri, J., Zhang, L., Mannila, L. & Norén. E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17.

OECD. (2019). *OECD Skills Outlook 2019: Thriving in a Digital World*, OECD Publishing, Paris, *https://doi.org/10.1787/df80bc12-en*.

OpenAI. (2022, 30.november). *Introducing ChatGPT.* https://help.snapchat.com/hc/en-gb/articles/13266788358932-What-is-My-AI-on-Snapchat-and-how-do-I-use-it-

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.

Postholm, M. B. (2010). *Kvalitativ metode: En innføring med fokus på fenomenologi, etnografi og kasusstudier* (2. utg.). Universitetsforlaget.

Román-González, M., Pérez-González, J.C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441-459.

Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E. & Voll, L. O. (2016). *Teknologi og programmering for alle - En faggjennomgang med forslag til endringer i grunnopplæringen*. Oslo: Utdanningsdirektoratet.

Selby, C. C., & Woollard, J. (2013). Computational thinking: The developing definition. *Proceedings of the 45th ACM technical symposium on computer science education*. Canterbury: ACM.

Selby, C. C, og Woollard, J. (2014). Refining an Understanding of computational thinking. Hentet fra: https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf

Selvik, K et al. (2016). *Programmering i skolen*. Notat fra Senter for IKT i utdanningen, Hentet fra: https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Skurdal, B. (2020, 28. juli). *Halveringsmetoden – gjett på et tall*. NDLA. https://ndla.no/article/24211

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms. *The Science Teacher*, 81(5), 53.

Solvang, R. (1991). *Matematikkdidaktikk.* NKI Forlaget.

Sriraman, B. (2009). The characteristics of mathematical creativity. *ZDM Mathematics Education*, 41(1-2), 13-27.

Stein, M.K., Grover, B.W. & Henningsen, M.A. (1996). Building Student Capacity for Mathematical Thinking and Reasoning: An Analysis of Mathematical Tasks Used in Reform Classrooms. *American Educational Research Journal*, 33, 455 - 488.

Store norske leksikon. u.å. *Kreativ*. Hentet 10.februar fra: https://snl.no/kreativ

Svartdal, F. (2018, 7.mai). *Tenkning*. Store norske leksikon. https://snl.no/tenkning

Tallis, R. (2004). *Why the mind is not a computer. A pocket lexicon of neuromythology.* Imprint Academic.

Tekdal, M. (2021). Trends and development in research on computational thinking. *Educ Inf Technol* 26, 6499–6529. https://doi.org/10.1007/s10639-021-10617-w

Thagaard, T. (2013). *Systematikk og innlevelse: En innføring i kvalitativ metode.* Fagbokforlaget Vigmostad & Bjerke.

Tjora, A. (2020). *Kvalitative forskningsmetoder: i praksis.* Gyldendal Akademisk.

Utdanningsdirektoratet. (2017). *Overordnet del – verdier og prinsipper for grunnopplæringen.* file:///C:/Users/Eier/Downloads/Overordnet%20del%20p%C3%A5%20bokm%C3%A5l%20(6).pdf

Utdanningsdirektoratet. (2019). *Algoritmisk tenkning*. https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/

Utdanningsdirektoratet. (2020a). *Hva er nytt i matematikk?* https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/

Utdanningsdirektoratet. (2020b). *Læreplan i matematikk*. https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/

Vinnervik, P. & Bungum, B. (2022). Computational thinking as part of compulsory education: How is it represented in Swedish and Norwegian curricula? Nordina 18(3), 384-400. https://doi.org/10.5617/nordina.9296

Voogt, J., Fisser, P., Good, J., Mishra, P. & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. https://doi.org/10.1007/s10639-015-9412-6

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.

Wing, J. M. (2011). Research notebook: Computational thinking-What and why. *The Link Magazine*,20–23. Carnegie Mellon University's School of Computer Science.

Wu, W. R & Yang, K. L. (2022). The relationships between computational and mathematical thinking: A review study on tasks. *Cogent Education*, 9(1). https://doi.org/10.1080/2331186X.2022.2098929

Vedlegg

Vedlegg 1: personlig korrespondanse med Utdanningsdirektoratet

Vår referanse: 2023/470

Hei Yasmina Madi

Takk for henvendelsen. Så spennende at du skriver masteroppgave om algoritmisk tenkning i matematikkfaget.

Comptational thinking ble oversatt til "algoritmisk tenkning" da læreplanen til forsøksvalgfaget programmering ble laget, i 2016. Oversettelsen ble diskutert i læreplangruppa, blant annet på grunn av assosiasjonen til matematikkfaget, som du nevner, men vi fant ikke noe bedre oversettelse som dekket Computational thinking godt nok. Vi så på andre lands oversettelse, som det svenske "datalogisk tänkande", men endte på algoritmisk tenkning.

Da programmering og algoritmisk tenkning skulle inn i nye læreplaner gjennom fagfornyelsen, ble det lagt til matematikkfaget (men skal også brukes i naturfag, kunst og håndverk og musikk), og algoritmisk tenkning ble lagt inn som del av kjerneelementet "Utforskning og problemløsning". Begrepet var allerede etablert gjennom blant annet valgfaget, så det ble ikke vurdert å se på begrepet på nytt.

Utkast til læreplanene i LK20 ble lagt ut på høring, og det kom mange høringssvar, blant annet på plasseringen av algoritmisk tenkning og programmering som del av matematikkfaget: https://hoering-publisering.udir.no/343/uttalelser

Lykke til med masteroppgaven din,

Vennlig hilsen

Vibeke Guttormsgaard
seniorrådgiver
Utdanningsdirektoratet

Vedlegg 2 samtykkeskjema

Vil du delta i forskningsprosjektet

«Algoritmisk tenkning og kreativitet»?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke algoritmisk tenkning og kreative framgangsmåter i detalj. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål
Prosjektet er en del av en masteroppgave. Formålet med prosjektet er å få en bedre forståelse for hva algoritmisk tenkning innebærer, og hvordan kreative framgangsmåter kan observeres i klasserommet. På bakgrunn av dette skal følgende problemstilling belyses «*Hvilke kjennetegn ved algoritmisk tenkning kommer til uttrykk hos R1 elever i matematisk problemløsning, og hvordan kan man avgjøre om framgangsmåten er av kreativ karakter?*».

Hvem er ansvarlig for forskningsprosjektet?
UiB Matematisk Institutt er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?
Opplegget som skal gjennomføres og de oppgavene som vil bli gitt er beregnet til elever på den videregående skolen. Jeg kom i kontakt med din faglærer og fikk positiv tilbakemelding om å gjøre datainnsamlingen i din matematikklasse. Det er ønskelig at du kan delta i forskningsprosjektet, slik at det kan samles inn tilstrekkelig med data til prosjektet.

Hva innebærer det for deg å delta?
Deltagelse i prosjektet innebærer ikke noe ekstra arbeid for din del. Elevene som deltar, blir delt inn i grupper. Alle gruppene skal gjennomføre det samme opplegget. Økten vil bli filmet, elevenes samtaler i økten vil bli tatt opp, samt skjermopptak av datamaskinen elevene jobber på. Dette for å analysere hvordan elevene arbeider med oppgavene, hvilke problemløsningsstrategier de velger og hvordan de arbeider i møte med programmeringen. Noen elever kan bli kalt inn til intervju for en kort samtale i etterkant av undervisningsøkten. Foresatte/verge har rett til å se intervjuguide på forhånd dersom deltakerne er umyndige.

Det er frivillig å delta
Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Vi bruker SAFE (Sikker Adgang til Forskningsdata og E-infrastruktur) som er en løsning for sikker behandling av sensitive personopplysninger i forskning, som bygger på Norm for informasjonssikkerhet i helse- og omsorgstjenestene (Normen) og sikrer at informasjonssikkerheten med hensyn til konfidensialitet, integritet og tilgjengelighet blir ivaretatt ved behandling av sensitive personopplysninger. Det er kun meg, Yasmina Madi, og min veileder, Jonas Oskarsson, som vil ha tilgang til video, lydopptakene og skjermopptakene som blir tatt opp i forbindelse med prosjektet.

Dersom informasjon om data skal publiseres, skal ingen data kunne knyttes direkte hverken til skole, lærer eller elev. Alle opplysningene vil bli transkribert, hvor personopplysninger blir anonymisert i transkripsjonen.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?
Opplysningene anonymiseres når prosjektet avsluttes, noe som etter planen er 1.juni.2022.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.
På oppdrag fra UiB Matematisk Institutt har NSD- Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter
Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvor kan jeg finne ut mer?
Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

Høyskolen i Innlandet ved Jonas Oskarsson, jonas.oskarsson@inn.no

Yasmina Madi, yma005@uib.no

Universitetet i Bergen sitt personvernombud kan kontaktes på e-post: personvernombud@uib.no

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

NSD- Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «*Algoritmisk tenkning og kreativitet*», og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i video- og lydopptak i en undervisningsøkt
- at mine opplysninger behandles fram til prosjektet er avsluttet

----

(Signert av prosjektdeltaker, dato)