

**Matti Keskinie**

**Algoritmisen ajattelun opiskelu ITKP0003-kurssilla – vaikuttavat tekijät ja suoriutuminen**

Tietotekniikan pro gradu -tutkielma

23. kesäkuuta 2020

Jyväskylän yliopisto  
Informaatioteknologian tiedekunta

**Tekijä:** Matti Keskinie mi

**Yhteystiedot:** matti.j.keskinie mi@student.jyu.fi

**Ohjaajat:** Leena Hiltunen

**Työn nimi:** Algoritmisen ajattelun opiskelu ITKP0003-kurssilla – vaikuttavat tekijät ja suoriutuminen

**Title in English:** Studying algorithmic thinking in ITKP0003 – affecting factors and performance

**Työ:** Pro gradu -tutkielma

**Opintosuunta:** Koulutusteknologia

**Sivumäärä:** 47+0

**Tiivistelmä:** Tutkimuksessa pyrittiin selvittämään, mitkä tekijät vaikuttivat opiskelijoiden menestykseen ITKP0003-kurssin loppukokeessa. Tutkimuksessa analysoitiin opiskelijoiden loppukokeen vastauksia määrällisin menetelmin, sekä loppukoe ja kurssimateriaali sisällönanalyysia hyödyntäen. Loppukokeen analyysista ei löydetty selittäviä tekijöitä opiskelijoiden menestykselle. Kurssimateriaalin analyysista huomattiin, että kurssi ei käsittele kaikkia algoritmiseen ajatteluun kuuluvia ominaisuuksia.

**Avainsanat:** Algoritminen ajattelu, ohjelmoinnillinen ajattelu, algoritmi, ITKP0003

**Abstract:** This study tried to find out factors affecting performance of students during test of ITKP0003-course. Test results were analysed using quantitative methods. Course material and test were analysed with content analysis method. Analysis of test results did not lead into finding factors affecting performance. Analysis of course material showed that the course does not manage to introduce all features linked to algorithmic thinking.

**Keywords:** Algorithmic thinking, computational thinking, algorithm, ITKP0003

## **Esipuhe**

Tämä pro gradu -tutkielma ja sen tekoprosessi on omistettu Anteron muistolle. Lupauksista pidetään kiinni, kuten opetit. Tämän tutkielman valmistuminen on osoituksena siitä. Haluan lisäksi kiittää vaimoani Essiä tuesta ja tsemppaamisesta vaikeina hetkinä.

Jyväskylässä 11.06.2020

*Matti Keskinieni*

## **Termiluettelo**

SPSS	IBM:n numeraaliseen analysointiin ja tilastojen käsittelyyn erikoistunut ohjelmisto
JYU	Jyväskylän yliopisto, lyhenne sanoista Jyväskylä University
ICT	Information Communcation Technology - tietotekniikka

## **Kuviot**

Kuvio 1. Arvolauseiden jakautuminen aineistossa ja normaalijakaumakäyrä.....	25
Kuvio 2 Luokitellun suoritusajan jakautuminen aineistossa ja normaalijakaumakäyrä.....	26
Kuvio 3. Luokitellun aloitusajan jakautuminen aineistossa ja normaalijakauma.....	27

## **Taulukot**

Taulukko 1: Pisteiden varianssianalyysi suoritusajan ja aloitusajan suhteen .....	30
---	----

# Sisältö

1	JOHDANTO.....	1
2	ALGORITMISEN AJATTELUN TAIDOT .....	2
2.1	Osa-alueet algoritmiselle ajattelulle.....	2
2.2	Abstrakti ajattelu.....	3
2.3	Ongelman pilkkominen, dekompositio .....	4
2.4	Kyky ajatella algoritmisesti .....	4
2.5	Arvioiva ajattelu.....	5
3	ALGORITMINEN AJATTELU – MÄÄRITELMÄ JA KOGNITIIVISET PROSESSIT.....	7
3.1	Mitä on algoritminen ajattelu.....	7
3.2	Kognitiiviset prosessit (algoritmisen) ajattelun takana.....	8
3.2.1	Ajattelun kehittyminen .....	9
3.2.2	Looginen ajattelukyky .....	10
3.2.3	Ongelmanratkaisutaito ja algoritminen ajattelu.....	11
3.2.4	Kriittinen ajattelu.....	12
4	ALGORITMISEEN AJATTELUUN VAIKUTTAVAT TEKIJÄT .....	14
4.1	Oppimisvaikeudet .....	14
4.2	Oppimishäiriöt algoritmisen ajattelun kehittymättömyyden takana .....	15
4.3	Vireys.....	16
4.4	Harjoittelu .....	17
5	ITKP0003 – JOHDATUS ALGORITMISEEN AJATTELUUN .....	19
6	TUTKIMUSASETELMA JA AINEISTON KERUU .....	21
6.1	Tutkimusote ja -kysymykset .....	21
6.2	Aineiston keruu .....	23
7	LOPPUKOKEEN TULOKSET .....	25
8	AINEISTON JA LOPPUTENTIN ANALYSOINTI .....	31
8.1	Aineiston tulosten analyysi .....	31
8.2	Loppukokeen analysointi .....	33
8.3	Kurssin aineiston analyysi .....	36
8.3.1	Algoritmisen ajattelun osa-alueet kurssimateriaaleissa.....	36
8.3.2	Kognitiiviset prosessit kurssimateriaaleissa .....	38
9	POHDINTA JA LUOTETTAVUUDEN ARVIOINTI.....	39
	LÄHTEET .....	43

# 1 Johdanto

Algoritminen ajattelu on modernissa tietoyhteiskunnassa jo lähes lukemiseen verrattavissa oleva taito. Suomessa vuonna 2016 käyttöön otetussa opetussuunnitelmassakin mainitaan ohjelmoinnin opettelu (Opetushallitus, 2014). Ohjelmoinnin osaamisen oleellinen osa on algoritminen ajattelu ja sen taitaminen. Algoritminen ajattelu on kokoelma ajattelun ominaisuuksia, joihin kuuluu muun muassa looginen päättelykyky sekä ongelmanratkaisukyky useiden muiden ajattelun taitojen lisäksi (Lagame 2015; Knuth, 1985; Selby & Woollard, 2013; Wing 2006). Lisäksi algoritmiseen ajatteluun sisällytetään kyky pilkkoa isompi ongelma pieniin palasiin, löydettyjen ratkaisujen yleistäminen, sekä yksiselitteisyys (Wing, 2006). Knuth (1985) sekä Cooper, Dann ja Pausch (2000) puhuvat algoritmisesta ajattelusta tietojenkäsittelytieteen kontekstissa.

Algoritminen ajattelu ei ole uusi, mystinen ajattelun taito, vaan arkielämässäkkin hyödynnettävä taito. Jo uunilohen tekeminen vaatii algoritmista ajattelua! Ruoanvalmistuksessa hyödynnetäänkin algoritmisen ajattelun ominaisuuksia, kuten yleistämistä, kykyä ajatella algoritmisesti, sekä joissain tapauksissa myös kykyä ajatella arvioivasti (Selby & Woollard, 2013). Algoritminen ajattelu on jossain määrin opeteltavissa oleva taito, sillä taidot, joita algoritmiseen ajatteluun vaaditaan, on opeteltavissa. Esimerkiksi ongelmanratkaisutaito on opeteltavissa oleva taito (L. D. Miller et al. 2013, 1426-1432; Marcelino et al. 2018,470-477).

Tässä tutkielmassa tutkitaan Jyväskylän yliopiston Johdatus algoritmiseen ajatteluun -kurssin opiskelijoita ja heidän menestymistään kurssin loppukokeessa. Tavoitteena on selvittää tekijöitä, jotka vaikuttavat opiskelijan menestykseen loppukokeessa, sekä tutkia, arvioidaanko kurssin loppukokeessa oikeasti opiskelijan algoritmista ajattelua, vai jotakin muuta. Toisena tavoitteena on siis arvioida loppukokeen validiteettia, eli mittaako loppukoe oikeasti sitä, mitä sen pitäisi mitata. Tutkielmassa käytetty aineisto on kerätty kurssin ITKP0003 käyneiden opiskelijoiden loppukokeen vastauksista vuosien 2018-2019 aikana. Tänä aikana kurssi toteutui yhteensä 6 kertaa. Kurssi on suunnattu Jyväskylän yliopiston opiskelijoille, jotka tutustuvat tietotekniikkaan, esimerkiksi luokanopettajaopiskelijoille.

## 2 Algoritmisen ajattelun taidot

Tässä luvussa käsitellään algoritmisen ajattelun osa-alueita. Aluksi käydään lyhyesti osa-alueet, jotka kuuluvat algoritmiseen ajatteluun. Tämän jälkeen käydään tarkemmin läpi merkittäviä algoritmiseen ajatteluun vaikuttavia ajattelun taitoja, kuten loogista ajattelukykyä, ongelmanratkaisutaitoa, sekä kriittistä ajattelua.

### 2.1 Osa-alueet algoritmiselle ajattelulle

Meta-analyysissaan Selby ja Woollard (2013) ehdottavat jo olemassa olevan kirjallisuuden perusteella algoritmisen ajattelun taidon koostuvan viidestä osa-alueesta:

- Kyky ajatella abstraktisti
- kyky pilkkoa ongelma pienemmiksi paloiksi (decomposition)
- kyky ajatella algoritmisesti
- kyky ajatella arvioivasti
- kyky ajatella yleistäen.

Nykyään algoritmisen ajattelun käsitteen sisällöstä ollaan hyvin yksimielisiä (Esim. Selby & Woollard 2013; Weese 2017; Wing 2006; Wing 2008). Esimerkiksi Weese (2017) sekä termiä alun perin ehdottanut Wing (2006) ovat samaa mieltä siitä, että Selbyn ja Woollardin (2013) esittämät osa-alueet kuuluvat algoritmiseen ajatteluun. Knuth (1985) esittää jo 1980-luvulla, että edellä mainitut ominaisuudet kuuluvaksi algoritmiseen ajatteluun.

Sekä Wing (2006), Selby ja Woollard (2013) että Knuth (1985) ovat samaa mieltä siitä, että kyky hahmottaa abstrakteja kokonaisuuksia on erittäin olennainen osa algoritmista ajattelua. Abstraktien kokonaisuuksien hahmotusta käytetään esimerkiksi matematiikassa hankalien ongelmien ratkaisussa siinä, missä abstraktien kokonaisuuksien hahmotus on algoritmisessa ajattelussa esimerkiksi termien ja ratkaistavien ongelmien kautta tärkeä ominaisuus. Wing (2008) pitää tätä ominaisuutta tärkeimpänä algoritmisen ajattelun osa-alueena. Ongelman pieniksi osiksi pilkkominen nähdään myös matematiikassa tärkeänä



ominaisuutena, jota abstraktien kokonaisuuksien hahmottaminen auttaa (Weese 2017; Knuth 1985).

Edellä mainittujen algoritmisen ajattelun osien lisäksi on esitetty muitakin osia kuuluvaksi algoritmiseen ajatteluun. Esimerkiksi Weese (2017) on esittänyt, että edellä mainittujen osa-alueiden lisäksi algoritmiseen ajatteluun liitettäisiin samanaikainen pienten osasten ratkominen, parallelisaatio. Parallelisaatiolla tavoitellaan tilannetta, missä jakamalla tehtävästä pieniä osasia eri suorittajille, saataisiin kokonaisuus nopeammin valmiiksi. Parallelisaatiota voi verrata tehtaissa tapahtuvaan linjatyöskentelyyn, missä valmistettava tuote etenee linjastoa pitkin työvaiheesta toiseen. Linjaston toisessa kohdassa tuotteelle tehdään esivalmistelut, esimerkiksi maalataan tuote, kun taas toisessa kohtaa linjastoa siihen lisätään puuttuvia osia. Wing (2006) mainitsee myös saman ominaisuuden algoritmisesta ajattelusta kirjoittaessaan, mutta ei pidä sitä pääpiirteenä algoritmisessa ajattelussa. Knuth (1985) ei mainitse parallelisaatiota algoritmisen ajattelun osa-alueena omassa artikkelissaan.

## **2.2 Abstrakti ajattelu**

Abstraktin ajattelun tiivistää hyvin Piaget'n ajattelun kehityksen teoria. Abstraktissa ajattelussa ajateltavat kokonaisuudet eivät enää välttämättä ole konkreettisesti havaittavissa, vaan ne on kyettävä kuvittelemaan mielessä (Day 1981, 44-45; Kincheloe and Steinberg 1993, 296-321; Smith 1996). Piaget'n teorian mukaan abstraktin ajattelun kypsyminen vaatii formaalin ajattelun tason. Piaget'n teoria siis väittää, että lapsi ei voi oppia ajatteluun abstrakteja asioita kuin vasta 11 vuoden iässä, koska ajattelun kehittyminen on sidoksissa biologiseen kehitykseen. Yksinkertaisimmillaan abstrakti ajattelu on kykyä kuvitella ja pohtia asioita, joille ei välttämättä ole konkreettista vastinetta. Käsitteenä vapaus on hyvä esimerkki abstraktista asiasta. Vähintään formaalin ajattelun vaiheessa oleva ihminen kykenee käsittelemään ja ymmärtämään, mitä vapaus on.

Abstraktin ajattelun kehittymisen viivästymisellä ja lapsen kehittymisen häiriöillä on havaittu korrelaatio, erityisesti kielellisten ominaisuuksien osalta (Blank & Solomon, 1968). Heidän mukaan kehityksen viivästymä ilmenee siten, että abstraktien termien, kuten sanan

”päälle” ymmärtäminen osoittautuu lapselle kehitystasoaan hankalammaksi. Tutkimuksessaan Blank ja Solomon (1968) pyrkivät vahvistamaan lapsen abstraktia ajattelukykyä kehittämällä lapsen kielellisiä taitoja useiden opetustyylien kautta. Tutkimuksessa havaittiin, että kielellisten taitojen kehittäminen vahvisti myös abstraktin ajattelun taitoja.

### **2.3 Ongelman pilkkominen, dekompositio**

Dekompositio tarkoittaa ongelman pilkkomista pienempiin, ratkaistavissa oleviin palasiin, alaongelmiin (Ho, 2001). Dekompositio ei ole algoritmiseen ajatteluun rajoittuva ominaisuus, vaan sitä hyödynnetään yleisesti ongelmanratkaisussa, kuten Ho (2001) tutkimuksessaan kertoo. Selby ja Woollard (2013) sekä Wing (2006) pitävät dekompositiotaitoja merkittävänä osana algoritmista ajattelua. Dekompositio ei ole myöskään yksi strategia hajottaa ongelmaa pienemmiksi ongelmiksi, vaan ennemminkin kattoterminä eri strategioille, joita hyödyntämällä ongelman voi hajottaa pienemmiksi osiksi, esimerkiksi bottom-top -strategia tai top-bottom strategia (Ho, 2001). Dekompositiotaitoja, kuten ongelmanratkaisutaitoa yleensäkin, voi kehittää harjoittelemalla. Myöskin ongelman viitekehysten tunteminen helpottaa dekompositiota (Ho, 2001; Wing, 2006). Ho (2001) esittää tutkimuksessaan, että viitekehysten tunteva, kokenut työntekijän dekompositiostrategia on enemmän viitekehystietoinen verrattuna kokemattomampaan työntekijään. Tämä tarkoittaa sitä, että ongelmaa lähdetään pilkkomaan viitekehykselle tyypillisiä termejä hyödyntäen (Ho, 2001). Auton korjauksen kontekstissa dekompositiota hyödyntämällä voidaan miettiä, missä kohdalla autoa ongelma sijaitsee, kun auto ei liiku eteenpäin. Ongelman tutkimista voidaan lähteä systemaattisesti tutkimaan komponentti kerrallaan.

### **2.4 Kyky ajatella algoritmisesti**

Kykyä ajatella algoritmisesti ja algoritmista ajattelua ei tule sekoittaa toisiinsa. Siinä, missä algoritmisen ajattelu tarkoittaa ongelmanratkaisutapaa (Selby&Woollard, 2013) tarkoittaa kykyä ajatella algoritmisesti kykyä ajatella pieniä, yksiselitteisiä askelia ratkaisua kohti edessä (Lamagne, 2015). Matematiikka lienee koulussa yleisin algoritmista ajattelukykyä vaativa oppiaine.

Knuth (1997) mainitsee algoritmiseen ajattelukyvyyn pohjautuvan algoritmeihin. Hän esittelee algoritmin koostuvan viidestä tärkeästä osasta, jotka yhdessä muodostavat algoritmin:

1. *Finiittisyys*. Algoritmin tulee päättyä äärellisen määrän jälkeen. Toisin sanoen, algoritmi ei saa jatkua lopetusehdon täytyttyä.
2. *Definiittisyys*. Jokainen algoritmin vaihe tulee olla yksiselitteinen, tarkasti määritelty. Varaa tulkitsemiselle ei saa olla. Esimerkiksi ohjelmointikielissä käsky tai komento on aina yksiselitteinen kontekstissaan.
3. *Syöte*. Algoritmilla on aina vähintään nolla syötettä, esimerkiksi parametreja, jotka annetaan algoritmille.
4. *Ulostulo*. Algoritmilla on aina yksi tai useampi ulostulo, joilla on suhde syöteen kanssa. Esimerkiksi lajittelualgoritmille annetaan kasa lajiteltavia alkioita ja ulostulona on lajiteltu alkiojoukko.
5. *Tehokkuus*. Algoritmin tulee olla verrattain tehokas.

Algoritmin määritelmään nojaten kyky ajatella algoritmisesti tarkoittaa siis kykyä ajatella ongelmaa siten, että ongelman ratkaisu täyttää edellä mainitut ominaisuudet. Esimerkiksi kahden luvun yhteenlaskutoimitus on algoritmista ajattelukykyä vaativa toimenpide, joka myös täyttää algoritmin piirteet. Uunilohen valmistusresepti täyttää myöskin algoritmin piirteet. Algoritmi päättyy, kun ruoka on valmista, ruoanlaittovaiheet on reseptiin kirjattu yksiselitteisesti, syöteenä toimii käytettävät raaka-ainemäärät, ulostuloja on yksi ja resepti on verrattain tehokas.

## 2.5 Arvioiva ajattelu

Kyky ajatella arvioivasti liitetään kriittiseen ajatteluun. Buckley, Archibald, Hargraves ja Trochim (2015) määrittelee artikkelissaan arvioivan ajattelun olevan kriittistä ajattelua arvioinnin kontekstissa. Samassa artikkelissa he määrittelevät arvioivan ajattelun tarkemmin. Heidän mukaansa arvioivaa ajattelua leimaa kriittisen ajattelun piirteiden lisäksi kyky reflektoida tehtyjä ratkaisuja, niiden hyviä ja huonoja puolia, sekä kyky pohtia ratkaisujen järkevyyttä. Earl ja Timperley (2015) määrittelevät arvioivaan ajatteluun kuuluvan lisäksi

prosessien, työskentelytapojen sekä ratkaisujen hionta ja hienosäätö. He pitävät arvioivaa ajattelua prosessinomaisen ajattelun taitona, jota soveltamalla voidaan jo olemassa olevia toimintatapoja, tai kehittyviä toimintatapoja, kehittää entistä paremmiksi. Buckley ja kumppanit (2015) toteavat kuitenkin, että arvioivan ajattelun määritelmästä ei olla vuosikymmenien jälkeenkään täysin yksimielisiä. Siitä kuitenkin näytetään olevan yksimielisiä, että arvioivaa ajattelua, kuten monia muitakin ajattelun taitoja, on mahdollista kehittää, kuten myös siitä, että arvioivan ajattelun taidot ovat yleistettävissä oleva taito, joka kuitenkin vaatii käyttäjiltään tietoa kontekstista, missä sitä käytetään (Buckley et al., 2015; Brookfield, 2020; Nelson & Eddy, 2008).

### **3 Algoritminen ajattelu – määritelmä ja kognitiiviset prosessit**

Tässä luvussa tiivistetään luvussa 2 käytyjen osa-alueiden pohjalta se, mitä algoritminen ajattelu (engl. computational thinking, programming thinking, algorithmic thinking) oikeastaan on. Ensimmäisessä alaluvussa määritellään algoritminen ajattelu historian kautta yleisellä tasolla. Toisessa alaluvussa käydään tarkemmin läpi algoritmiseen ajatteluun vaikuttavia kognitiivisia prosesseja, kuten ongelmanratkaisutaitoa.

#### **3.1 Mitä on algoritminen ajattelu**

Yksinkertaisuudessaan algoritminen ajattelu siis on joukko ajattelun taitoja, joita hyödyntämällä ongelma kyetään ratkaisemaan algoritmisesti (Lagame 2015; Knuth 1895; Wing 2008). Algoritmisen ajattelun synonyyminä käytetään usein ohjelmoinnillista ajattelua. Esimerkiksi Knuth (1985) sekä Cooper, Dann ja Pausch (2000) mainitsee tietojenkäsittelytieteeseen kuuluvan algoritmisen ajattelukyvyn, jonka Knuth (1985) esittelee tarkemmin. Hän mainitsee artikkelissaan ajattelukyvyn osa-alueita, joista puhutaan myöhemmissä tutkimuksissa ohjelmoinnillisena ajatteluna. (Wing 2008, 3717-3725; Eckerdal, Thuné, and Berglund, 2005, 135-142). Koska algoritminen ajattelu ja ohjelmoinnillinen ajattelu ovat synonyymejä tietojenkäsittelytieteen kontekstissa, käytetään tässä tutkielmassa terminä pelkästään algoritmista ajattelua tästä eteenpäin sekaannusten välttämiseksi.

Algoritminen ajattelu siis tarkoittaa ongelmanratkaisutapaa, missä ongelmaa pyritään ratkaisemaan hyödyntämällä algoritmista suoritustapaa (Wing 2008, 3717-3725; Eckerdal, Thuné, and Berglund, 2005, 135-142). Algoritminen ajattelu koostuu pienistä ajattelun osa-alueista, joita esitellään tarkemmin luvussa 2. Algoritmisen ajattelun ominaisuuksia määritellään tieteellisesti ensimmäisen kerran Knuthin (1985) teksteissä hänen puhuessaan tietojenkäsittelytieteessä tarvittaviin ajattelun ominaisuuksiin. Kuitenkin jo vuonna 1974 Woolley ja Miller mainitsevat algoritmisen ajattelun terminä puhuessaan ohjelmoinnin opiskelusta. Algoritminen ajattelu sai synonyymikseen 2000-luvulle tullessa ohjelmoinnillisen ajattelun. Wing (2006, 2008) sekä muun muassa Eckerdal ja kumppanit (2005) sekä Israel

ja kumppanit (2015) puhuvat ohjelmoinnillisesta ajattelusta viitaten täysin samoihin ajattelun taitoihin ja ominaisuuksiin, joita Knuth esitteli jo vuonna 1985 artikkelissaan.

Algoritmiseen ajatteluun kuuluu nimensä mukaisesti algoritmit. Mitä algoritmit sitten ovat? Tietojenkäsittelytieteessä algoritmilla tarkoitetaan yksiselitteistä, yksinkertaisista askeleista koostuvaa kokonaisuutta, jolla voidaan ratkaista ongelma. Algoritmin osat esitellään tarkemmin luvussa 2.4 algoritmisesta ajattelukyvyistä puhuttaessa.

Knuthin määritelmään pohjaten Blass ja Gurevich (2003) esittelevät tutkimuspaperissaan erilaisia algoritmeja niiden eroavaisuuksia. Kaikkia algoritmeja yhdistää se, että niissä on selkeät, määrätyt askeleet, joiden mukaan algoritmi toimii. Jokainen askel on yksiselitteinen, ja algoritmilla on lopputulema. Esimerkkinä algoritmista on Fibonaccin lukujono. Fibonaccin lukujonossa lasketaan kaksi edellistä lukua yhteen. Kaksi ensimmäistä lukua on klassisessa esimerkissä nolla ja yksi. Shah (1997) esittelee Fibonaccin luvun algoritmin todistuksen tutkimuksessaan.

Eckeraldin ja Berglundin (2005) tutkimuksessa opiskelijat vertaavat ohjelmointiin vaadittavaa ajattelutapaa tietokoneen ajattelutapaan. Tietokone tekee juuri sen, mitä käsketään, eikä mitään muuta. Samanlaisia loogisia ajattelun ominaisuuksia, yksiselitteisyys ja selkeys, on löydettävissä algoritmisesta ajattelustakin, kuten muun muassa Knuth (1985) ja Wing (2006, 2008) osoittavat. Fibonaccin lukujono on selkeä ja yksiselitteinen, hyvä esimerkki algoritmisesta ratkaisutavasta. Siinä toistetaan yksiselitteisillä ja selkeillä ohjeilla samoja askeleita uudestaan ja uudestaan, kunnes kohdataan loppuehto.

### **3.2 Kognitiiviset prosessit (algoritmisen) ajattelun takana**

Kuten luvussa 2 todettiin, liittyy algoritmiseen ajatteluun useita ajattelun taitoja. Tässä luvussa käydään aluksi läpi ajattelun kehittymistä, erityisesti Jean Piaget'n ajattelun kehittymisen teorian kautta. Tämän jälkeen käydään läpi yleisiä ajattelun taitoja, jotka liittyvät läheisesti algoritmiseen ajatteluun.

### 3.2.1 Ajattelun kehittyminen

Jean Piaget'n kuuluisan ajattelun kehittymisen teorian mukaan ihmisen ajattelun kehittyminen on nelivaiheinen portaikko (Day 1981, 44-45; Kincheloe and Steinberg 1993, 296-321; Smith 1996). Sensomotorisessa vaiheessa ajattelua leimaa aistien ylikorostettu rooli maailman havainnoimisessa. Jos ihminen ei tässä vaiheessa koske, näe tai kuule ilmiötä, ei sitä ole olemassa. Esioperationaalisessa vaiheessa puolestaan ympäristön havainnointi ja loogiset johtopäätökset johdetaan suoraa aistien välittämästä tiedosta. Klassisessa esimerkissä lapsen eteen kaadetaan kaksi tilavuudeltaan saman kokoista kuppia, joista toinen on laakeampi ja matalampi kuin toinen. Kuppeihin kaadetaan yhtä paljon vettä ja kysytään lapselta, kummassa kupissa on enemmän vettä. Esioperationaalisessa vaiheessa oleva lapsi vastaa, että korkeammassa kupissa on enemmän vettä. Formaalisessa ajattelun vaiheessa oleva ihminen ymmärtää, että vettä on yhtä paljon molemmissa kupeissa, koska tilavuus ei ole muuttunut mihinkään. Formaalin ajattelun vaihetta leimaa looginen, hypoteettisdeduktiivinen ajattelumalli. Ihminen tekee päätelmiä ympäröivästä maailmasta aistiensa antaman informaation, sekä sisäisten oppimalliensa, skeemojen, perusteella. Teorian mukaan ihminen siirtyy formaaliin ajatteluun 11-12 vuoden iässä, murrosiän kynnyksellä. Formaali ajattelu voidaan nähdä algoritmisen ja loogisen ajattelun perustana. Ilman päättelykykyä ja tulkintakykyä aistien ulkopuolisesta maailmasta, ihminen tuskin kykenee ajattelemaan monimutkaisia kokonaisuuksia, saati pilkkomaan niitä pienempiin palasiin. Postformaalisessa ajattelussa puolestaan ihminen kykenee käsittelemään abstraktejakin kokonaisuuksia symboleita hyväksi käyttäen. Suurin ero formaalin ja postformaalin ajattelun välillä onkin kyky käsitellä abstrakteja asioita, kuten esimerkiksi matemaattisia kaavoja.

Ajattelu itsessään on kenties tärkein osa algoritmista ajattelukykyä. Piaget'n teoriassa (Day 1981, 44-45; Smith 1996) ajattelu kehittyy vaiheittain, ikään kuin rakentaen edellisen vaiheen päälle. Piaget'n konstruktivinen ajattelun kehittymisen malli olettaa ajattelun kehittymisen olevan sidoksissa ikään ja biologiseen kehittymiseen, määrättyyn pisteeseen asti. Piaget'n teoria ei ota huomioon teoriassaan ollenkaan oppimisvaikeuksia, saati kehittymisen viivästyksiä. Tämä onkin Piaget'n teorian heikkous, mikä tulee ottaa huomioon teoriaa soveltaessa.

Ajattelun kehittymisen kilpailevana teoriana pidetään Lev Vygotskyn mallia ajattelun kehittymisestä. Vygotskyn sosiokulttuurinen ajattelun kehittymisen mallin mukaan kieli on ajattelun ”esiaste”, mitä lapsi hyödyntää sosialisoituakseen. Kielen kehittymisen myötä puhe kehittyy ajatteluksi, ns. sisäiseksi puheeksi. Ajattelu olisi siis vain riittävän kielellisen kehityksen jatke ja vastaavasti ilman kieltä ei olisi ajattelua. (Valsiner and Van der Veer 1988, 117-136) Yhdistettynä Vygotskyn lähikehityksen vyöhykkeen teoriaan päästään siis malliin, jossa lapsi oppii ajattelemaan kielellisesti lahjakkaamman avustuksella. Lähikehityksen vyöhykkeen teorian mukaan ihminen on kykenevä oppimaan uuden asian, jos häntä lahjakkaampi ihminen on häntä opastamassa ja ihmisen muu osaaminen tähän riittää. Esimerkiksi lapsi kykenee oppimaan yhteenlaskun, jos hän ymmärtää ja osaa numerot, sekä ymmärtää yksinkertaiset symbolit. Vastaavasti aikuinen voi oppia uusia taitoja, esimerkiksi ohjelmoimaan tietokoneita, jos hänelle tarjotaan avuksi henkilö, jonka ohjelmointitaidot ylittävät aikuisen oma taidot ja aikuisella on mielenkiintoa opiskella ohjelmointia.

### **3.2.2 Looginen ajattelukyky**

Algoritmisen ajattelun osa-alueiden kehittymiselle voidaan pitää tärkeänä ajattelun taitojen kehittymistä. Esimerkiksi loogista ajattelukykyä pidetään hyvin merkittävän osana algoritmisen ajattelun kehittymistä. Eckerdal ja Berglund (2005) mainitsevat ohjelmoinnissa käytetyn algoritmisen ajattelun kehittäneen matematiikassa tarpeellisia loogisia ajattelutaitoja. Looginen ajattelukyky taas on taito, jota voidaan harjoittaa. Se on ajattelun taito, joka tarkoittaa kykyä päästä johtopäätökseen annettujen esitietojen pohjalta. (Tobin K, William C. 1981). Loogisen ajattelukyvyn nähdään koostuvan viidestä osa-alueesta:

1. muuttujien säätely
2. suhteellinen päättely
3. yhdistelevä päättely
4. todennäköisyyspäättely
5. korrelaatiopäättely

Algoritmisen ajattelun taitojen näkökulmasta tarkasteltuna ainakin muuttujien säätelyn kautta tapahtuva päättely, sekä korrelaatiopäättely ovat oleellisia loogisen ajattelukyvyn



taitoja. Algoritmillä on usein joitain muuttuvia tekijöitä. Muuttuvien tekijöiden vaikutus algoritmisen suorituksessa ja lopputuloksen muuttumisen arviointi ovat oleellisia taitoja (Selby et al, 2013).

### **3.2.3 Ongelmanratkaisutaito ja algoritmisen ajattelu**

Ongelmanratkaisutaito sekä looginen ajattelukyky auttavat tehokkaasti toinen toistaan. Ongelmanratkaisutaito on, nimensä mukaisesti, kyky ratkaista ja hahmottaa ongelmia hyödyntäen eri ratkaisutapoja (Weese, 2017). Algoritmisen ajattelu näkökulmasta ongelmanratkaisutaito nivoutuukin oleellisesti lähes jokaisen algoritmisen ajattelun taitoon, mutta eniten se vaikuttaa kykyyn ajatella arvioivasti sekä algoritmisesti. Algoritmin muodostamista varten on kyettävä analysoida ongelma, sekä löytää ratkaisusta toistuvat rakenteet. Muun muassa Woods, Hrymak ja Bouchard (1997) määrittelevät ongelmanratkaisutaitoon kuuluvan useita osa-alueita. Algoritmisen ajattelun näkökulmasta merkittävimmät osa-alueet lienevät järjestelmällinen lähestymistapa, tiedon kerääminen sekä useiden lähestymistapojen kokeilu. Algoritmisessa ajattelussa järjestelmällinen lähestymistapa ilmenee jo ajattelutavan perusluonteessa. Algoritmi on yksiselitteinen, johdonmukainen tapa edetä ongelman parissa, kuten esimerkiksi Wing (2006,2008) sekä Knuth (1985) esittävät. Useiden lähestymistapojen kokeilu puolestaan ilmenee siinä, että samaan lopputulemaan voi päästä usealla erilaisella algoritmilla. Käytännön esimerkkinä on perunoiden keittäminen. Perunat voi pestä aluksi tai olla pesemättä. Voit laittaa veden keittymään ennen perunoiden pesua tai pesun jälkeen. Lopputulos on kutakuinkin sama järjestyksestä huolimatta.

Ongelmanratkaisutaitoa prosessina esittelee puolestaan Zsakó ja Szlavi (2012) artikkelissaan, joka käsittelee ICT -taitoja. He mainitsevat ongelmanratkaisutaidon koostuvan viidestä kohdasta:

1. Ongelman tunnistaminen
2. Ongelman ymmärtäminen
3. Ongelman esittäminen
4. Ongelman ratkaisu
5. Tulosten kommunikointi

Algoritminen ajattelu on tapa ratkaista ongelmia. Täten algoritminen ajattelu keskittyy erityisesti ongelmanratkaisussa kohtaan 4, ongelman ratkaisuun. Ongelmanratkaisutaito onkin enemmän ajattelun taito, jonka osa algoritminen ajattelu on, eikä toisin päin. Toisaalta ongelmanratkaisukin voidaan mieltää algoritmiseksi prosessiksi, jos siihen kyetään luomaan yksiselitteiset ohjeet, sekä toistettava rakenne.

Ongelmanratkaisutaito on harjoiteltavissa oleva taito (L. D. Miller et al. 2013, 1426-1432; Marcelino et al. 2018, 470-477). Esimerkiksi hankalien matemaattisten tehtävien ratkaisuun tarvitaan kehittyneitä ongelmanratkaisutaitoa loogisen ajattelukyvyn lisäksi. Ongelmanratkaisutaidon kehittyminen vaatii ongelman ratkaisemiseen vaadittavien tietojen ja taitojen keräämistä, kuten matematiikan tehtävässä. Kyetäkseen laskemaan, on henkilön ymmärrettävä symbolien ja numeroiden merkitys. Tämän lisäksi on ymmärrettävä yleiset laskusäännöt, eli ehdot, joiden puitteissa ongelma on mahdollista ratkaista. Algoritmiseen ajatteluun kuuluukin ongelmanratkaisutaito ja siihen liitettävät taidot olennaisesti. Kyetäkseen pohtimaan algoritmista ajattelua, tulee ymmärtää ratkaisua koskevat lainalaisuudet. Tarkastelemalla algoritmisen ajattelun osa-alueita, jotka esitellään luvussa 2, huomataan kyvyn hahmottaa ongelmia liittyvän selkeästi vähintään kykyyn pilkkoa ongelmaa pienemmiksi paloiksi.

### **3.2.4 Kriittinen ajattelu**

Puhuttaessa algoritmisesta ajattelusta tai ongelmanratkaisutaidoista, mainitaan kriittisen ajattelun taidot erittäin usein samassa yhteydessä (Dolek, Bazalais, Lemay, Saxena, Basnet. 2017; Tsalapatras, Heidmann, Alimisi, Houstis. 2012). Dolek ja kumppanit (2017) esittävät kriittisen ajattelun määritelmäksi ajattelun taitoa, jossa ongelmaa kyetään arvioimaan ja tutkimaan siten, että annettujen tietojen pohjalta päädytään rationaaliseen lopputulokseen. Snyder ja Snyder (2008) puolestaan määrittelevät kriittisen ajattelun tarkoittamaan järjestelmällistä prosessia, missä aktiivisesti käsitteellistetään, analysoidaan, syntetisoidaan ja/tai arvioidaan kerättyä tai havaittua tietoa. Heidän mukaansa kriittiseen ajatteluun on myös viitattu metakognitiivisena taitona, taitona ajatella ajattelua, ja sitä kautta kehittää yksilön ajattelun taitoa.

Selby ja Woollard (2013) mainitsevat meta-analyysissään algoritmiseen ajatteluun kuuluvan yhtenä osa-alueena kyvyn ajatella arvioivasti. Kuten huomataan, kriittiseen ajatteluun kuuluu kyky arvioida kerättyä tai havaittua tietoa. Voidaankin siis todeta, että kriittisen ajattelun taito on merkittävä osa algoritmista ajattelua. Selby ja Woollard (2013) puhuvat kyvystä ajatella arvioivasti erityisesti algoritmin lopputulosten kanssa. Kriittisen ajattelun taidot nousevat siis esille erityisesti, kun arvioidaan algoritmin lopputuloksia, suorittiko algoritmi halutun toimenpiteen. Toisaalta kriittistä ajattelua tarvitaan myös algoritmia suunnitellessa tai jo tehtyä algoritmia arvioidessa. Algoritmia tulee kyetä tarkastelemaan ja arvioimaan, tekeekö se sen, mihin se on tarkoitettu. Onnistutko keittämään ohjeella perunoita, vai poltatko ne pohjaan. Kriittistä arviointia tarvitaan myös paljon ennen algoritmin suoritusta, kun arvioidaan algoritmin ja parametrien soveltuvuutta keskenään. Voitko keittää makaronia perunoiden keitto-ohjeella? Voitko perunoiden keitto-ohjeella keittää kinkkiä? Esille nousee kriittisen ajattelun tarve päätyä rationaaliseen lopputulokseen, sekä arvioidaan jo kerättyä tietoa, sekä pohditaan niiden järjellisyttä.

## 4 Algoritmiseen ajatteluun vaikuttavat tekijät

Tässä luvussa käydään läpi tekijöitä, jotka vaikuttavat algoritmiseen ajatteluun. Ensimmäisessä alaluvussa käydään läpi hieman yleisesti oppimisvaikeuksia. Toisessa alaluvussa käydään läpi oppimisvaikeuksien vaikutusta algoritmiseen ajatteluun. Kolmannessa alaluvussa käydään läpi muita ajatteluun vaikuttavia tekijöitä, kuten esimerkiksi vireyden ja valppauden vaikutus algoritmiseen ajatteluun. Viimeisessä alaluvussa käydään läpi harjoittelun vaikutusta algoritmiseen ajatteluun.

### 4.1 Oppimisvaikeudet

Oppimisvaikeudet ovat neurologisia häiriöitä, jotka vaikeuttavat ihmisen kykyä opiskella uusia asioita. Suomessa käytettävässä ICD-10 -tautiluokituksessa oppimisvaikeuksilla ja -häiriöillä on oma ryhmänsä, F81. (THL, 2018). Selvyiden vuoksi tästä eteenpäin puhutaan vain oppimishäiriöistä. Oppimishäiriöt jaetaan yleisesti kolmeen alalajiin, häiriön kohteen mukaan: Lukemisen häiriöt, kirjoittamiskyvyn häiriöt, sekä laskemiskyvyn häiriöt. (THL,2018). Näiden lisäksi on vielä olemassa laaja-alaiset oppimisen häiriöt (eng. non-specific learning disorder, non-specific learning difficulty), jotka liittyvät usein muihin kehityksen häiriöihin, esimerkiksi Downin syndroomaan. Laaja-alaisissa oppimisen häiriöissä henkilö on kehittynyt neurologisesti epätyypillisesti, mikä johtaa oppimisen häiriöihin.

Korkeakouluissa oppimisvaikeuksista kärsivien opiskelijoiden määrä ei ole aivan yhtä suuri, kuin muun väestön keskuudessa. Noin 7.5% Yhdysvalloissa yliopistossa opiskelevista kärsii jostain oppimisvaikeudesta (Williams, 2015). Lähes saman tuloksen oli Kunttu, Pesonen ja Saari (2016) saaneet tekemästään korkeakouluopiskelijoiden terveystutkimuksesta. Suomen korkeakouluopiskelijoista noin 8% kärsii jostain oppimisvaikeudesta, lukemisen vaikeuksien ollessa selkeästi yleisin vaikeus. Lukemisen vaikeus esiintyi noin 5% vastaajista. Kuntun ja kumppanien (2016) tutkimuksessa on hyvä huomioda, että tutkimuksessa on myös ammattikorkeakouluopiskelijat, joten se ei ole täysin suoraa vertailtavissa Williamsin (2015) tutkimukseen. Tulosten samankaltaisuus viittaisi kuitenkin siihen, että kan-

sainvälisesti korkeakouluissa opiskelee suunnilleen saman verran erilaisista oppimisvaikeuksista kärsiviä henkilöitä, kuin Suomessakin.

## **4.2 Oppimishäiriöt algoritmisen ajattelun kehittymättömyyden takana**

Todennäköisesti suurimmat syyt algoritmisen ajattelun ongelmien takana liittyvät harjoittelun puutteeseen ja laskemiskyvyn häiriöihin. Niilo Mäki -instituutin (2019) mukaan laskemiskyvyn oppimishäiriöille on tyypillistä laskujärjestyksen ja -vaiheiden oppiminen. Laskujärjestyksen oppiminen vaatii kykyä loogiseen, johdonmukaiseen ja laskusääntöjen mukaiseen etenemiseen. Samanlaista taitoa vaaditaan algoritmisen ajattelun kehittymiseen. Kappaleissa 3.1 ja 2 todettiin, että algoritmit ovat yksiselitteisiä, johdonmukaisia ja verrattain tehokkaita ongelmanratkaisumenetelmiä. Algoritmisessa ajattelussa tarvitaan matemaattikassakin tuttua kykyä ajatella abstraktejakin kokonaisuuksia (Knuth, 1985).

Matemaattisiin taitoihin liittyvillä oppimishäiriöillä voi olla vaikutusta algoritmisen ajattelun kehittymiseen. Tätä tukee entisestään tieto siitä, että matemaattisten taitojen kehittäminen on mahdollista, kuten myös algoritmisen ajattelun taitojen (Israel et al. 2015, 45-53; Marcelino et al. 2018, 470-477; Swanson, Olide, and Kong 2018, 931-951). Esimerkiksi Marcelino ja kumppanit (2018,4770-477) kehittivät tutkimuksessaan algoritmisia taitoja vahvistamalla opiskelijoiden loogista ajattelua ja ongelmanratkaisutaitoa Scratch -ohjelmointiympäristöllä. Kontrolliryhmään verratessa ero algoritmisessa ajattelussa oli tilastollisesti merkitsevä. Calao, Moreno León, Correa ja Robles (2015) olivat päätyneet tutkimuksessaan tulokseen, jonka mukaan algoritmisen ajattelun taitojen kehittäminen johti matemaattisten taitojen kehittymiseen. Tutkimuksessa algoritmisen ajattelun taitojen kehittämiseen käytettiin Scratch-ohjelmointiympäristössä suoritettavia tehtäviä. Knuth (1985) esitti, että matematiikassa ja algoritmisessa ajattelussa vaadittavat ajattelun taidot olisivat vähintään osin päällekkäisiä.

Grigorenko ja kumppanit (2019) osoittivat, että matemaattisia taitoja on mahdollista opettaa oppimishäiriöstä kärsivälle lapselle, kun taitoja opetellaan niille ominaisessa kontekstissa. Tämä tarkoittaa matemaattisten taitojen kohdalla sitä, että niitä opiskellaan laskemalla tehtäviä, mieluummin konkreettisia sellaisia. Koska laskemiskyvyn häiriöstä kärsivän opis-

kelijan on mahdollista harjoittaa matemaattisia taitoja, on mahdollista harjoittaa myöskin algoritmisen ajattelun vaatimia taitoja jonkin verran, kuten esimerkiksi Marcelino ja kumppanit (2018) sekä Calao ja kumppanit (2015) osoittivat. Lisäksi Calao ja kumppaneiden tutkimuksessa huomattiin, että algoritmisen ajattelun taitojen kehittäminen vahvisti matemaattisten taitojen osaamista. Algoritmisen ajattelun osaaminen vahvistaa siis matemaattisten osaamista. Luvussa 2 todettiin myös matemaattisen ajattelun osa-alueiden käyvän osittain limittäin algoritmisen ajattelun osa-alueiden kanssa.

Matemaattisten kykyjen lisäksi algoritmisen ajattelu vaatii ajattelun kypsymistä. Piaget'n teoriaan (Day 1981, 44-45; Kincheloe and Steinberg 1993, 296-321; Smith 1996) peilaten algoritmisen ajattelun kehittyminen vaatii riittävän ajattelun tason, toisin sanoen oikean kehittymisen vaiheen. Ajattelun kehittymistä hidastavat tekijät voivat siis vaikuttaa Piaget'n teoriaan pohjaten algoritmisen ajattelun kehittymiseen. Tällaisia, ajattelun kehittymiseen vaikuttavia oppimishäiriöitä on esimerkiksi laaja-alaiset oppimisen häiriöt. Tutkimuksia siitä, voiko laaja-alaisesta oppimishäiriöstä kärsivän ihmisen loogista päättelykykyä kehittää, ei löytynyt. Muihin tutkimuksiin perustuen voisi olettaa sen olevan mahdollista jossain määrin, koska muita ajattelun ominaisuuksia on mahdollista kehittää (Grigorenko et al. 2019; Israel et al. 2015, 45-53; Swanson, Oline, and Kong 2018, 931-951).

### **4.3 Vireys**

Valppaus tarkoittaa sitä, kuinka toimintavalmis ja ärsykeisiin reagoiva ihminen on muutokseen (Paget, Lambert, ja Sridhar 1981, 359-365). Toisin sanoen valppaudella tarkoitetaan sitä, kuinka terävästi ihminen reagoi ympäristön aistiärsykkeiden muutokseen (Paget, et al. 1981, 359-365). Yleensä ihmisen valppaustila on korkeimmillaan heräämisen jälkeen ja matalimmillaan juuri ennen nukahtamista. Pitkäaikaisissa, rutiininomaisissa tehtävissä valppaudella on tapana hetkellisesti tippua, mikä puolestaan vähentää ihmisen herkkyyttä reagoida ärsykeisiin (Paget, et al. 1981, 359-365; Warm, Parasuraman, ja Matthews 2008, 433-441).

Aikaisemmissa tutkimuksissa on havaittu, että erityisesti unen puute ja ulkoisten ärsykkeiden suuri määrä vaikuttavat merkittävästi ihmisen valppauteen (Paget et al. 1981, 359-

365). Erityisesti äänillä tuntuu olevan vaikutusta ihmisen valppauteen. Valppautta laskee esimerkiksi pitkäaikainen, samanlainen ääni, katujuvä esimerkiksi. Toisaalta taas satunnaiset, kovat äänet saattavat hetkellisesti myöskin häiritä valppautta, koska ihminen kohdistaa aistinsa äänen aiheuttajaan ja pyrkii tunnistamaan sen. Unen puute vaikuttaa erityisesti hidastamalla ärsykkeisiin reagointiaikaa merkittävästi. Myös huomiota vaativat tehtävät, kuten esimerkiksi tentin tekeminen, kärsivät unen vähydestä. Tarkkuutta ja huomiota vaativien tehtävien tekemiseen kului myöskin merkittävästi enemmän aikaa. Esimerkiksi Pagetin ja kumppaneiden (1981) tutkimuksessa anestesiaalääkäreiden univajeet ja kolmivuorotyö vaikuttivat merkittävästi lääkäreiden valppaustilaan.

Algoritminen ajattelu sekä sitä vaativat tehtävät ovat kognitiivisesti kuormittavia ja huomiota vaativia prosesseja. Unen puute, kuten edellä todettiin, hidastaa ihmisen reaktiota ja pidentää huomiota vaativien tehtävien suoritusaikaa. Algoritmista ajattelu vaativat tehtävät vaativat myös aktiivista huomion kohdentamista. Algoritmista ajattelua vaativien tehtävien tekeminen pitäisi hidastua ja virheiden määrän kasvaa, jos tehtäviä joutuu tekemään huonosti levänneenä tai matalalla vireystilalla. Taustamelullakin on Pagetin ja kumppaneiden (1981) tutkimukseen peilaten mahdollisuus heikentää algoritmista ajattelua vaativista tehtävistä suoriutumista heikentämällä keskittymiskykyä.

#### **4.4 Harjoittelu**

Algoritminen ajattelu on harjoiteltavissa oleva taito. Esimerkiksi Liao ja Lian (2017) osoittivat tutkimuksessaan, että yhdistetyllä oppimismallilla voi oppia tehokkaasti algoritmista ajattelua. Grigorenkon ja kumppaneiden (2019) havainnot matemaattisten taitojen opettamisesta tukevat Liaon ja Lian (2017) havaintoja. Lisäksi Tsalapatás ja kumppanit (2012) osoittivat, että algoritmiseen ajatteluun tarvittavia taitoja, kuten ongelmanratkaisutaitoja, sekä järjestelmällistä, yksiselitteistä suoritustapaa on mahdollista harjoitella tekemällä algoritmista ajattelua vaativia tehtäviä jo peruskouluikäisenä. Liaon ja Lian (2017) tutkimuksessa tutkimusryhmässä käytettiin sulautetun oppimisen menetelmiä vahvistamaan algoritmisen ajattelun oppimista. Tutkimuksen perusteella algoritmisen ajattelun oppimista voi vahvistaa sulautetulla oppimisella verrattuna normaaliin luokkaopetusmenetelmään. Tutkimuksen perusteella algoritmista ajattelua on mahdollista harjoitella. Myös Miller ja

kumppanit, sekä Marcelino ja kumppanit päätyivät siihen tulokseen, että algoritmista ajattelua on mahdollista harjoitella ( Miller et al. 2013, 1426-1432; Marcelino et al. 2018, 470-477).

Tutkimuksia siitä, kuinka paljon harjoittelu auttaa kehittämään algoritmista ajattelua verrattuna harjoittelemattomuuteen, löytyi niukasti. Kuitenkin tutkimuksissa, joissa algoritmista ajattelua harjoitettiin, havaittiin harjoittelulla olevan tehostava vaikutus algoritmisen ajattelun taitoja kohtaan. Algoritmisen ajattelun taitojen harjoittamisella havaittiin olevan vaikutusta myös samankaltaisiin ajattelun taitoihin (Calao, et al.,2015; Marcelino et al. 2018, 4770-477), Algoritmisen ajattelu ja matemaattinen ajattelu pohjaavat osittain samoihin kognitiivisiin prosesseihin, kuten luvuissa 2 ja 3.2 todettiin. Lisäksi Grigorenko ja kumppanit (2019) osoittivat, että oppimishäiriöstä kärsivän opiskelijan on mahdollista opetella matemaattista ajattelua sekä matemaattisia taitoja. Tämä tieto tukee entisestään oletusta siitä, että algoritmista ajattelua on mahdollista opiskella ja kehittää sekä sitä, että algoritmisen ajattelun harjoittelusta on hyötyä taidon kehittämisessä (Calao, et al.,2015; Miller et al. 2013, 1426-1432; Marcelino et al. 2018, 470-477).



## 5 ITKP0003 – Johdatus algoritmiseen ajatteluun

Johdatus algoritmiseen ajatteluun, ITKP0003, on Jyväskylän yliopiston kurssi, jota tarjotaan osana informaatioteknologian tiedekunnan erillisiä opintokokonaisuuksia. ITKP0003 kuuluu pakollisena osana digitalisaation opintokokonaisuuteen. Kurssin osaamistavoitteina mainitaan muun muassa algoritmisen ajattelun peruskäsitteiden ymmärtäminen, sekä algoritmisen ajattelun perusteet, kuten ongelman purkaminen pieniksi palasiksi ja säännöllisesti toistuvien toimintamallien löytäminen. ITKP0003 on kahden opintopisteen laajuinen kurssi, laskennallisesti noin 53 tunnin työskentelyn vaativa kurssi. (JYU)

ITKP0003 muodostuu seitsemästä opiskelijan itsenäisesti opiskeltavissa olevasta osiosta, sekä loppuentistä. Seitsemäs osio on yhteenveto kuudesta edellisestä osiosta. Kurssi on jaettu seuraaviin osioihin:

- Mitä on algoritmisen ajattelu
- Arjen algoritmit
- Ongelmanratkaisu
- Algoritmisia rakenteita
- Sanastoa
- Käytännön esimerkkejä algoritmeista
- Yhteenveto

Kurssi on loppukoe myöden saatavilla Moodlessa verkkokurssina. Kontaktiopetusta ei järjestetä. Kurssin aineisto yhdistelee videoita, itsenäisesti suoritettavia tehtäviä sekä kirjoitettua aineistoa. Opiskelija voi edetä kurssilla omaan tahtiin, kunhan suorittaa loppukokeen annetun aikarajan sisällä.

Loppukoe koostuu 23 kysymyksestä. Kysymyksiä ei voida tutkielmassa paljastaa, sillä Johdatus algoritmiseen ajatteluun-kurssi on toteutuksessa oleva kurssi. Tämän takia turvaututaan kuvailemaan loppukokeen kysymykset paljastamatta kysymysten sisällöstä liikaa. Kysymykset ovat soveltavia valintatehtäviä, monivalintatehtäviä, tulkintatehtäviä sekä järjestelytehtäviä. Avoimia kysymyksiä on myös muutamia, mutta niiden määrä on selkeästi vähäisempi verrattuna yksiselitteisiin väittämätehtäviin sekä valintatehtäviin. Jokaisel-

le saman periodin opiskelijalle arvotaan tehtäväpankin kahdeksan kategorian pohjalta yksilöllinen loppukoe jokaisella suorituskerralla. Kysymyspankissa on useampia kysymyksiä samasta aihealueesta, mikä lisää loppukokeen vaihtelevuutta eri kurssitoteutusten välillä. Tehtävistä saa yhdestä neljään pistettä riippuen tehtävän vaativuudesta. Osapisteitä ei jaeta. Yhteensä tentistä voi saada 56 pistettä, keskimäärin siis kaksi pistettä kysymystä kohti. Loppukoetta ei ole rajoitettu paikkaan, vaan opiskelija voi tehdä tentin, annettujen päivien aikana. Loppukokeen tekemahdollisuus avautuu yleensä viikko ennen periodin loppua ja sulkeutui periodin viimeisenä päivänä kuitenkin siten, että tentin tekemiseen tarjoutuu 7 päivää. Tentin aloittamisen jälkeen opiskelijalla on 30 minuuttia aikaa suorittaa loppukoe Moodle-ympäristössä.

## 6 Tutkimusasetelma ja aineiston keruu

Tässä luvussa avataan tutkimusotetta, sekä esitellään tutkimuskysymykset sekä aineiston keruumenetelmä.

### 6.1 Tutkimusote ja -kysymykset

Tutkimus keskittyy kurssin ITKP0003 loppukokeen suorittaneiden opiskelijoiden vastausaineistoon, sekä kurssin materiaaliin, joten tutkimusta voi pitää selittävänä tutkimuksena, missä pohditaan opiskelijoiden kurssimenestyksen syy-seuraussuhteita teorian pohjalta. Toisaalta tutkimusta voisi pitää tapaustutkimuksena. Tapaustutkimuksen piirteet piirteet täytyvät, sillä tutkittavana kohteena on yksi kurssi yhdessä Suomen yliopistossa. Tapaustutkimuksen muunnelma, monitapaustutkimus, olisi myös ollut mahdollinen vertailemalla eri toteutusten menestymistä keskenään. Tapaustutkimukseen kuuluu kuitenkin myös sosiokulttuurinen näkökulma, mihin tämä tutkielma ei ota kantaa. Tätä tutkielmaa ei toisaalta ole järkevää pitää kehittämistutkimuksenakaan, missä hyödynnetään määrällisiä menetelmiä, kuten myös teorialähtöistä sisällönanalyysiä kehityskohteita tutkittaessa. Näiden pohdintojen perusteella tutkimusotteeksi määriteltiin selittävä tutkimus, missä tarkasteltiin Johdatus algoritmiseen ajatteluun-kurssin käyneiden opiskelijoiden menestystä sekä kurssin loppukoetta ja kurssimateriaalia.

Selittävä tutkimus määrällisten tutkimusten laji, missä pyritään etsimään syy-seuraussuhteita ilmiöiden välillä. (Vilka, 2007). Selittävässä tutkimuksessa etsitään syy-seuraussuhteita esimerkiksi korrelaatiotutkimuksilla, varianssianalyysillä, sekä vastaavilla tilastollisten riippuvuuksien tutkintamentelmillä. Määrällisen luonteensa takia selittävä tutkimus pyrkii mitatun aineiston pohjalta etsimään aineistossa esiintyviä syy-seuraussuhteita. Määrällistä tutkimusta arvioidaan validiteetin ja reliabiliteetin näkökulmasta. Validiteetilla tarkoitetaan sitä, että tutkimus mittaa sitä, mitä sen tuleekin mitata. Reliabiliteetilla taas tarkoitetaan sitä, että samalla aineistolla toteutettuna eri tutkijat saavat saman lopputuloksen tutkimuksesta. Tutkimuksen validiteetti on hyvä, jos siihen ei sisälly systemaattista virhettä. (Vilka, 2007). Systemaattinen virhe voi muodostua, jos vastaajat valehtelevat systemaattisesti. Reliabiliteetti puolestaan on hyvä, jos toinen tutkija kykenee

toteuttamaan saman tutkimuksen samalla aineistolla ja menetelmillä saaden siitä saman tuloksen.

Selittävässä tutkimuksessakin voidaan käyttää laadullisia menetelmiä syy-seuraussuhteiden selvittämiseksi, esimerkiksi sisällönanalyysiä (Tuomi & Sarajärvi, 2018). Sisällönanalyysillä pyritään nostamaan aineistosta merkityksellisiä asioita, yhtäläisyyksiä ja eroavaisuuksia. Teorialähtöisessä sisällönanalyysissä valitaan teoriasta kumpuavia määritteitä ja yhtäläisyyksiä ja tutkitaan, pitääkö teoriasta kumpuavat väitteet aineistossa paikkansa. Tämän tutkielman kontekstissa tämä tarkoittaa, että mittaako loppukoe aineistossa opiskeltuja asioita ja sisältääkö kurssiaineisto algoritmisen ajattelun taitoja ja osa-alueita. Aineistolähtöisessä sisällönanalyysissä puolestaan etsitään aineistosta näitä yhtäläisyyksiä riippumatta olemassa olevasta teoriasta. Sisällönanalyysiä hyödynnetään kurssin aineistoa analysoidessa.

Tutkielman aiheen keskittyttyä algoritmiseen ajatteluun, tuntui luontevalta suunnata tutkimuskysymyksiä algoritmiseen ajatteluun vaikuttaviin tekijöihin. Ennen aineiston näkemistä minua kiinnosti suunnattomasti, vaikuttaako esimerkiksi tentin suoritusaike tentistä saatavaan arvosanaan, saati tehtäväkohtaisiin pisteisiin. Hypoteesi on, että tenttiä pidempään tehneet saavat keskimäärin parempia arvosanoja sekä pisteitä. Mielenkiintoa myöden tutkimuskysymyksiksi valikoituikin, että mitkä tekijät aineiston pohjalta vaikuttavat tentistä kerättäviin pisteisiin ja sitä kautta lopulliseen arvosanaan.

Algoritmisen ajattelun kehittymistä ohjaavalla kurssilla on usealta eri osa-alueelta olemassa tehtäviä. Kurssi on jaettu seitsemään osioon, joista jokaisesta tulee kysymyksiä loppukokeeseen. Tätä myöden mieleeni kohosi toinen kysymys, jonka päädyin ottamaan toiseksi tutkimuskysymykseksi:

1. Mitkä tekijät vaikuttivat opiskelijan menestykseen loppukokeessa?
2. Mittaako loppukoe lopulta ollenkaan algoritmista ajattelua, vai onko kyseessä jostain muuta ominaisuutta mittaava loppukoe?
3. Onko kurssin aineiston perusteella mahdollista tehdä loppukoe? Opetetaanko kurssin aineistossa kaikki tarpeellinen algoritmiseen ajatteluun liittyvä?

## 6.2 Aineiston keruu

Vastausaineiston koon ollessa lähes 300 opiskelijaa (N=269) valikoitui määrälliset menetelmät aineiston analysointimenetelmäksi. Koska aineistosta tekijöitä, jotka aineiston perusteella vaikuttivat menestykseen, päädyttiin aineistoa analysoimaan monipuolisesti määrällisiä menetelmiä hyödyntäen. Tutkittaessa lopputentin algoritmisen ajattelu mittausta, päädyttiin vertaamaan tenttiä ja tentin kysymyksiä algoritmisen ajattelun piirteisiin sekä määrittelemään, mittaako lopputentti oikeasti algoritmista ajattelua.

Tutkimukseen käytettävä aineisto kerättiin kurssin ITKP003 loppukokeesta. Aineisto kerättiin vuoden 2018-2019 toteutusten vastauksista. Opiskelijoille annettiin mahdollisuus kieltäytyä aineiston käytöstä. Tutkimusta varten aineisto anonymisoitiin siten, että yksittäisiä loppukokeen tuloksia on mahdoton kohdistaa yksiselitteisesti loppukokeen tekijään. Aineiston analysoinnissa käytettiin apuna SPSS-ohjelmistoa.

Aineistoon kerättiin opiskelijan tulokset yksilöivä, juokseva tunnusluku, tentin aloitus- ja lopetusaika, käytetty aika sekunnin tarkkuudella, kokonaispisteet, pisteet tehtäväkohtaisesti sekä vastaukset tehtäviin. Henkilötunnuksia, saati nimiä ei aineistossa ollut. Näiden tietojen pohjalta luotiin aineiston analysointia varten uusia muuttujia. Kokonaan uusia muuttujia luotiin 3 kappaletta, joiden lisäksi olemassa olevaa aineistoa muokkaamalla saatiin 2 lisämuuttujaa.

Aineistoa muokkaamalla luodut muuttujat olivat tentin aloitus- ja lopetusaika. Alkuperäisessä aineistossa aloitusaika oli muodossa DD-MM-YYY HH:MM. SPSS-ohjelmiston Date And Time Wizard -ominaisuuden avulla eroteltiin aloituskellonaika sekä -lopetusaika omiksi muuttujiksi. Nämä muuttujat saivat nimekseen aloitettu\_klo ja suoritettu\_klo. Aloitettu\_klo -muuttujaa hyödynnettiin myöhemmin uusien muuttujien luontiin.

Kokonaispisteiden ja tentin arvosteluperusteiden avulla laskettiin opiskelijalle uusi muuttuja, arvolause asteikolla 0-5, missä nolla tarkoitti hylättyä ja viisi tarkoitti erinomaista suoritusta. Tentin kokonaispistemäärä oli 56 pistettä ja arvonsanaan yksi tarvittiin tästä piste-määrästä 50%, eli 28 pistettä. Arvosanaan kaksi puolestaan 60%, arvonsanaan kolme 70%

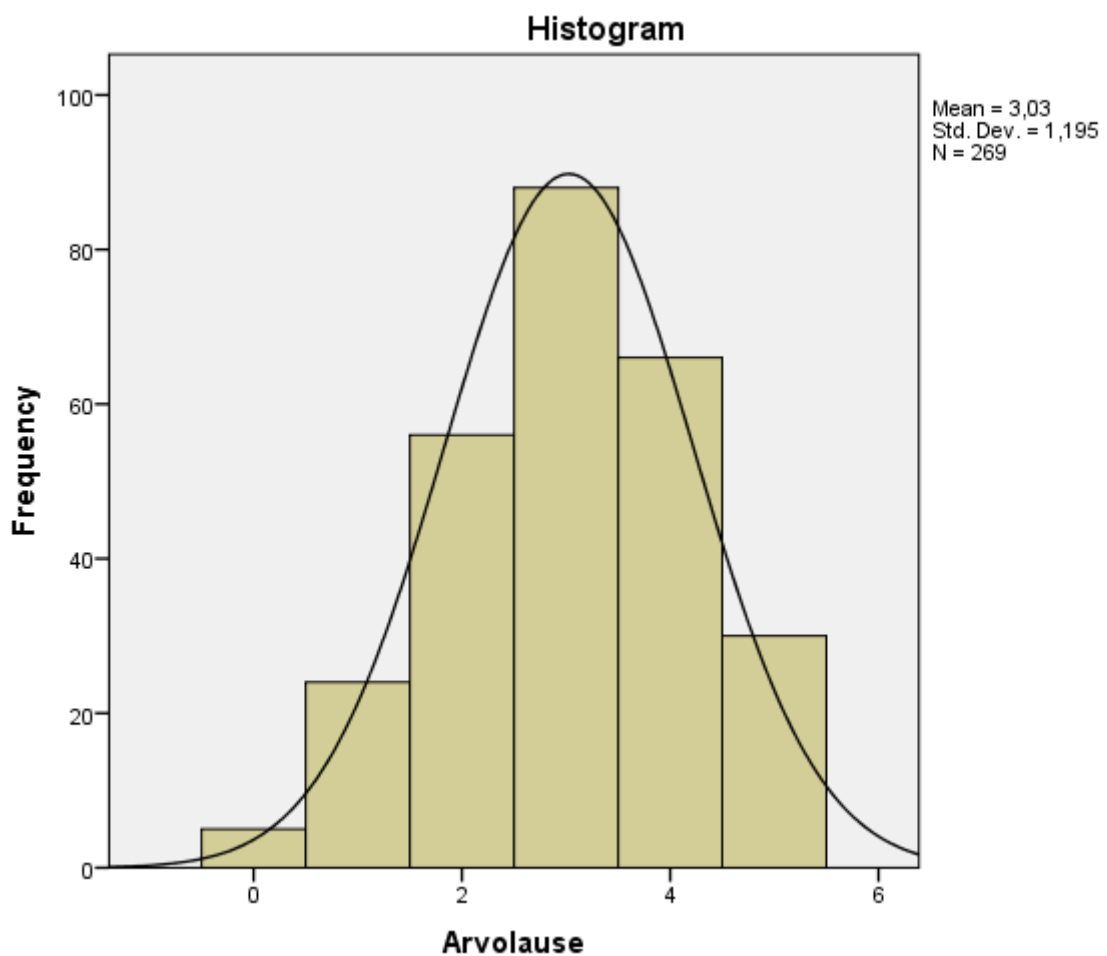
pisteistä, arvosanaan neljä 80% ja arvosanaan viisi 90% tai enemmän pisteitä. Arvosanojen pisterajat saatiin kurssin materiaalin tiedoista.

Arvosanojen lisäksi luotiin suoritusajaluokka -muuttuja. Muuttuja laskettiin annetun suoritusajan perusteella siten, että 15 minuuttia tai vähemmän tentin suorittamiseen aikaa käyttäneet opiskelijat saivat arvon yksi, 15-20 minuuttia kuluttaneet arvon kaksi, 20-25 minuuttia käyttäneet arvon kolme ja yli 25 minuuttia kuluttaneet taas arvon neljä. Raja-arvot luokittelevalle muuttujalle valittiin, kun aineistosta oli saatu selville pisin ja lyhyin tenttiin käytetty aika.

Kolmas laskettu muuttuja oli aloitusaikaluokka-muuttuja. Aloitusaikaluokka-muuttuja määriteltiin loppukokeen aloituskellonajan perusteella, mikä puolestaan irrotettiin jo olemassa olevasta muuttujasta SPSS-ohjelmistolla. Jos tentin aloitusaika oli välillä 08.00 – 12.00, oli muuttujan arvo 1. Klo 12.01-16.00 välillä aloitettu loppukoe puolestaan sai arvon 2, 16.01-20.00 arvon 3, klo 20.01-23.59 sai arvon 4 ja muuhun aikaan aloitettu loppukoe sai arvon 5.

## 7 Loppukokeen tulokset

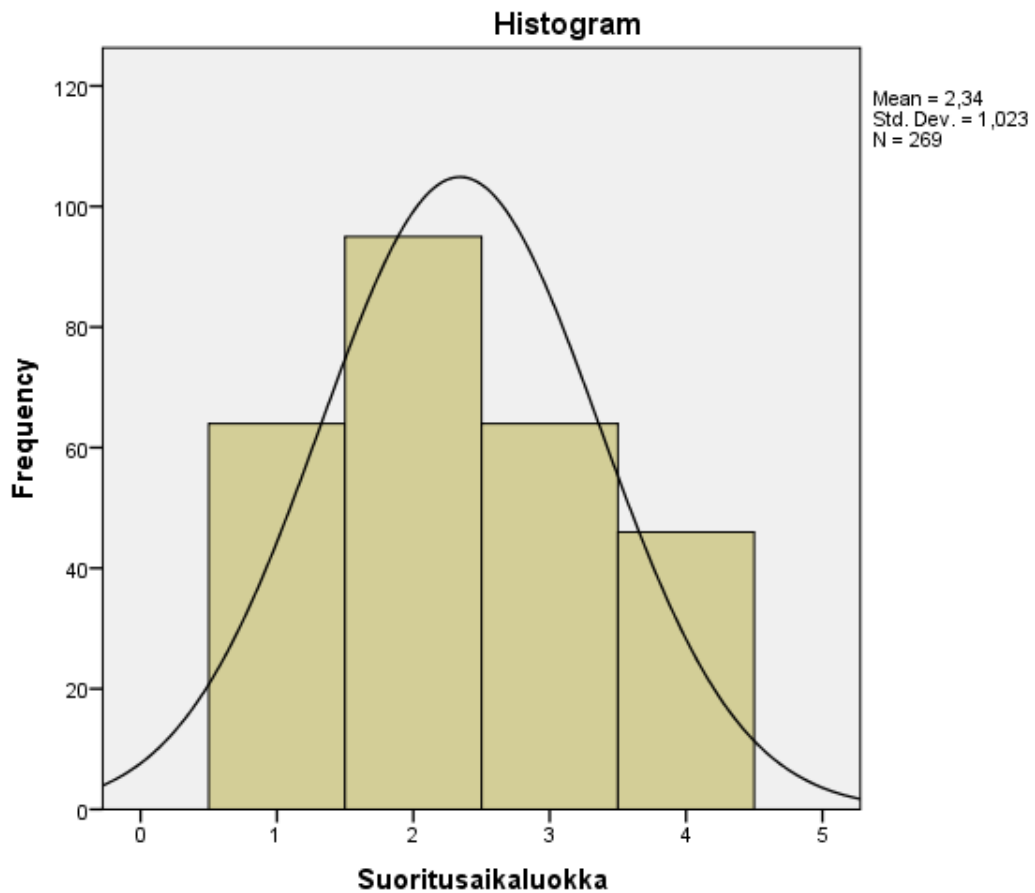
Aineiston analysointi aloitettiin luomalla luvussa 6 esiteltyt muuttujat aineistoon analysointia helpottamaan. Aineiston koon ollessa suuri (N=269) katsoin järkeväksi aloittaa aineiston analysoinnin tutkimalla arvosanjakaumaa, sekä ajankäytön jakautumista tentin tekemiseen. Yleisin arvosana tentin suorittaneelle opiskelijalle oli kolme, jonka sai 32,6% opiskelijoista. Harvinaisin arvosana puolestaan oli hylätty, nolla, jonka sai 1,9% opiskelijoista. Puuttuvia arvoja, eli tentin kesken jättäneitä opiskelijoita oli yksi kappale.



Kuvio 1. Arvolauseiden jakautuminen aineistossa ja normaalijakaumakäyrä

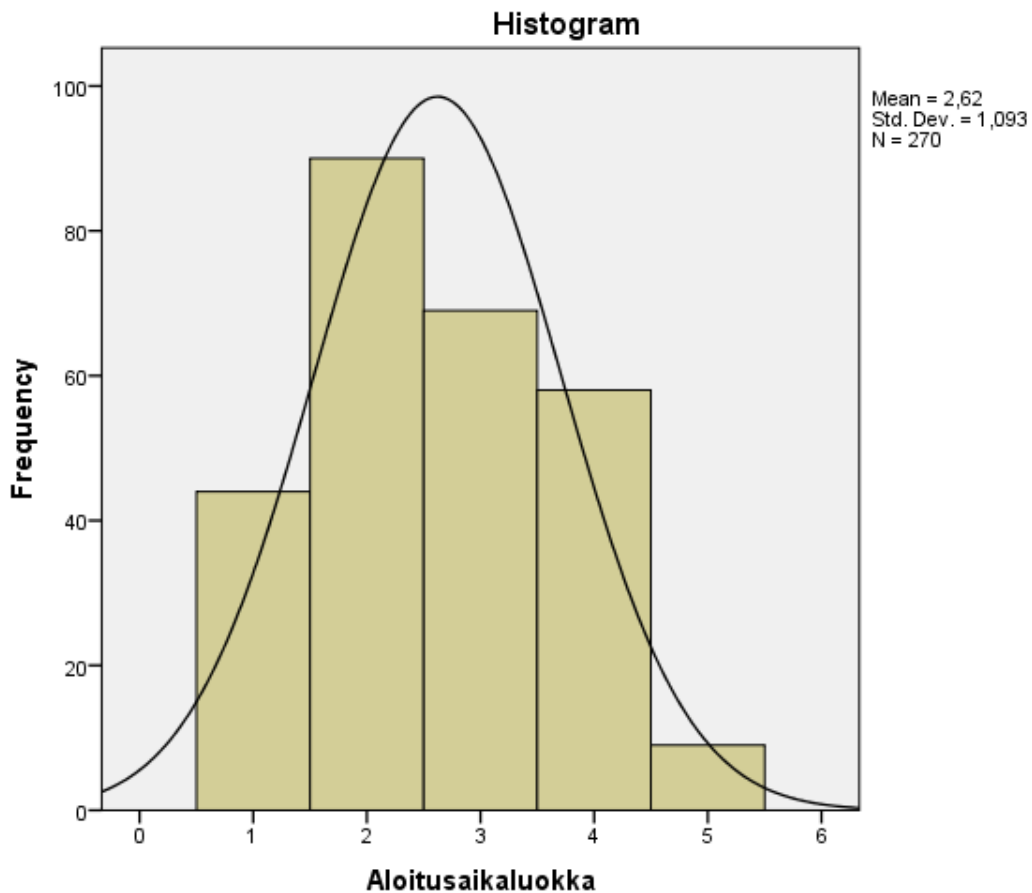
Suoritusaikojen yleisyyden arviointiin käytettiin aikaisemmin luotua suoritusajaluokka - muuttujaa, sillä ilman luokittelua eri arvoja olisi tullut noin 266 kappaletta. Yleisimmillään opiskelija käytti tenttiin 15-20 minuuttia; 35,2% eli 95 opiskelijaa käytti tämän verran ai-

kaa. Samassa havaittiin, että yhtä monta opiskelijaa käytti aikaa alle 15 minuuttia, kuin 20-25 minuuttia. Yli 25 minuuttia tentin tekemiseen käytti puolestaan pienin määrä opiskelijoita. 46 opiskelijaa, eli 17% aineiston opiskelijoista kulutti tenttiin yli 25 minuuttia.



Kuvio 2: Luokitellun suoritusajan jakautuminen aineistossa ja normaalijakaumakäyrä





Kuvio 3. Luokitellun aloitusajan jakautuminen aineistossa ja normaalijakauma

Ensimmäinen tilastollisen analyysin menetelmä, jota aineistoon käytettiin, oli korrelaatiotutkimus. Korrelaatiotutkimuksella haluttiin saada tietoon, onko kahdella kasvavalla muuttujalla korrelaatiota kasvussaan. Muuttujissa oli pari paria, jolle suora korrelaatiovertailu oli mielekästä tehdä. Nämä muuttujat olivat saadut pisteet ja tentin tekemiseen käytetty aika. Korrelaatiotesti osoitti, että näiden kahden muuttujan välillä ei ollut tilastollisesti merkittävää yhteyttä, ( $N=269$   $p = 0.36$ ). Toinen pari, jolle korrelaatiovertailu oli mielekästä tehdä, oli loppukokeen aloitusaika (aloitusaikaluokka) sekä saatu arvolause. Testi osoitti, että aloitusajalla ja saadulla arvolauseella ei ollut tilastollisesti merkitsevää korrelaatiota ( $n=269$ ,  $p=-.018$ ).

Korrelaatiotutkimusten lisäksi aineistolle suoritettiin yksisuuntaisia varianssianalyyseja. Varianssianalyysin tavoitteena on mitata, eroavatko kahden tai useamman ryhmän keskiar-

vot tilastollisesti toisistaan. (Vilkka 2007). Soveltuvien muuttujien varianssianalyysillä tavoitellaan tässä tutkimuksessa muuttujien riippuvuuksien selvittämistä soveltavin osin.

Aineiston tarkastelun jälkeen oli selvää, että varianssianalyysiä pystyisi soveltamaan luokitellun suoritusajan ja saatujen pisteiden välisiin riippuvuuksiin. Varianssianalyysi osoitti, että luokitellun suoritusajan ollessa selittävä tekijä, ei suoritusajan ja pisteiden välillä ollut tilastollisesti merkitseviä eroja ryhmien kesken. (N=269, p=.225).

**ANOVA**

Pisteet /56

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	178,160	3	59,387	1,461	,225
Within Groups	10768,450	265	40,636		
Total	10946,610	268			

Kuva 1: Varianssianalyysi pisteiden välillä ryhmitellyn suoritusajan mukaan

Toinen varianssianalyysi suoritettiin aloitusajalle ja saaduille pisteille aloitusajan ollessa selittävä muuttuja ja saadut pisteet riippuva muuttuja. Varianssianalyysi osoitti, että luokiteltuna aloitusajalla ja saaduilla pisteitä ei ollut tilastollisesti merkitsevää riippuvuutta toisistaan (N=269, p=.554).

Kolmas varianssianalyysi suoritettiin luokitellulla aloitusajalle ja loppukokeen yksittäisistä kysymyksistä saaduille pistemäärille kysymyskohtaisesti. Käytännössä loppukokeen kysymyspankista otettiin kysymys ja opiskelijoista valittiin ne opiskelijat, jotka olivat kysymykseen vastanneet. Tavoitteena oli tutkia, onko tentin aloitusajalla vaikutusta yksittäisiin kysymyksistä saatuihin pisteisiin. Tämän analyysin suorittaminen vaati aineiston erottele- mistä soveltavin osin, sillä loppukokeen rakenteen takia sama kysymys saattoi olla opiskelijoilla eri kysymysnumerolla. Aineistoa muokattiin siten, että kysymykset, joissa kysymys-vastausparin saattoi erottaa, erotettiin toisistaan ja kyseisen kysymyksen kohdalla

opiskelijat analysoitiin omina ryppäinään. Esimerkiksi kysymyspankin määritelmä - kategorian kysymykset kyettiin erottamaan toisistaan, pois lukien ensimmäinen ja viimeinen kysymys. Koska nämä kysymykset olivat kuitenkin vastausvaihtoehdoiltaan identtiset ja kysymyksenasettelultaan hyvin samanlaiset, yhdistettiin näiden kysymysten tulokset keskenään.

Tilastollisesti merkitsevät eroavaisuudet aloitusajan suhteen havaittiin kysymyksien 12 ja 21 kohdalla. Kysymys 12 käsitteli arjen algoritmeja. Bonferronin post hoc-testin tarkastelu osoitti, että aloitusaikaluokan 1,4 ja 5 ryhmien keskiarvot erosivat tilastollisesti merkitsevästi toisistaan, kuten myös aloitusaikaluokan 2 ja 5, sekä 3 ja 5. Kysymys 21 puolestaan käsitteli algoritmisia rakenteita. Siinä merkitsevät erot ilmenivät aloitusaikaryhmien 1 ja 4 sekä 3 ja 4 välillä. Suoritusajan suhteen tilastollisesti merkitseviä eroavaisuuksia havaittiin vain kysymyksen 13 kohdalla. Post hoc-testin tarkastelu osoitti merkitsevien eroavaisuuksien olevan suoritusajaryhmien 1 ja 4, 1 ja 5 sekä 2 ja 5 välillä

Kysymykset 7-11 olivat määritelmäväittämiä, joihin vastausvaihtoehtoina oli tosi ja epätosi. Kysymykset 22-39 olivat sanastoon liittyviä tosi - epätosiväittämiä. Näille ei suoritettu analyysia, sillä aineistosta ei yksiselitteisesti pystytty erottamaan kysymystä, johon opiskelija oli vastannut. Myös kysymykset 40 ja 41 olivat kysymyksiä, joita ei voinut ottaa analyysiin mukaan. Kysymykset olivat hyvin erilaisia keskenään, mutta vastausvaihtoehdot olivat samankaltaisia. Aineistosta ei pystynyt päättämään, kuka opiskelija oli vastannut kysymykseen 40 ja kuka kysymykseen 41.

Taulukko 1: Pisteiden varianssianalyysi suoritusajan ja aloitusajan suhteen

Kysymysnumero	Suoritusajan suhteen p-arvo	Aloitusajan suhteen p-arvo
1	.42	.75
2	.74	.76
3	.27	.44
4	.42	.75
12	.45	.012
13	.015	.67
14	.23	.45
15	.10	.44
16	.19	.23
17	.87	.93
18	.54	.99
19	.54	.99
20	.67	.21
21	.44	.022
42	.95	.63

## **8 Aineiston ja lopputentin analysointi**

Tässä luvussa analysoidaan aineistosta saadut tulokset, sekä loppukokeen kyky arvioida algoritmista ajattelua. Ensimmäisessä alaluvussa käydään läpi loppukokeen tulokset ja toisessa alaluvussa puolestaan analysoidaan itse loppukoe. Loppukokeen analysoinnissa hyödynnetään luvuissa 2-4 esiteltyä teoriataustaa.

### **8.1 Aineiston tulosten analyysi**

Kurssin loppukokeesta saadun aineiston perusteella näyttää siltä, että opiskelijan loppukokeeseen käyttämällä ajalla ei ole tilastollisesti merkitsevää vaikutusta kokeesta saatuun arvolauseeseen. Kun opiskelijat ryhmiteltiin kokeeseen käytetyn ajan mukaan (suoritusajaluokka), oli jokaisessa ryhmitellyssä luokassa havaittavissa jokaista arvosanaa, eikä keskihajonnassa ollut tilastollisesti merkitseviä eroavaisuuksia. Tämän perusteella voidaan väittää, että loppukokeen suorittamisaika ei vaikuttanut tilastollisesti merkitsevästi opiskelijan saamaan arvolauseeseen. Sama ilmiö toistuu aineiston perusteella myös saatujen pisteiden ja suoritusajan välillä. Loppukokeeseen käytetty aika ei vaikuttanut loppukokeesta saatavaan arvolauseeseen tai pisteisiin tilastollisesti merkitsevästi. Toisin sanoen, vastaus tutkimuskysymykseen yksi on, että aineiston analyysin perusteella ei kyetä yksiselitteisesti päättämään tekijöitä, jotka vaikuttavat opiskelijan menestykseen loppukokeessa, mutta voidaan pyrkiä päättämään tekijöitä, jotka vaikuttavat yksittäisiin tehtäviin.

Vaikka menestystä selittäviä tekijöitä ei löytynyt loppukokeen analysoinnissa, havaittiin siellä kuitenkin tilastollisesti merkitseviä eroja muissa tekijöissä. Aineiston perusteella loppukokeen suorituksen aloitusajankohdalla oli muutaman kysymyksen kohdalla tilastollisesti merkitseviä vaikutus kysymyksestä saataviin pisteisiin. Eräs tällainen oli kysymyspankin kysymys 12. Aloitusajaluokkien 1 ja 4, 1 ja 5, 2 ja 5, sekä 3 ja 5 välillä oli tilastollisesti merkitseviä eroavaisuuksia pisteissä. Opiskelijat, jotka olivat aloittaneet loppukokeen tekemisen ennen puolta yötä olivat analyysin mukaan saaneet merkitsevästi parempia pisteitä, kuin puolen yön jälkeen aloittaneet opiskelijat. Tehtävän menestystä voi selittää paljolti luvussa 4.3 esitelty vireystilan vaikutus ihmisen kykyyn tehdä kognitiivisesti vaativia tehtäviä. Aloitusaikaryhmän 5 huono suoriutuminen tehtävästä vahvistaa entisestään

vireystilan heikentymisestä johtuvan menestymisen mahdollisuutta, sillä muiden aloitusaikaryhmien tuloksissa ei ollut kyseisen tehtävän kohdalla tilastollisesti merkitseviä eroavaisuuksia. Kysymyksen 13 kohdalla suoritusajalla näytti olevan tilastollisesti merkitsevä vaikutus suoritusaikaryhmien 1 ja 4, 1 ja 5, sekä 2 ja 5 välillä. Tässäkin tapauksessa suoritus aika 5, yli 25 minuuttia loppukokeeseen käyttäneet opiskelijat, näyttivät menestyneen huonoiten. Kyseessä voi olla tilastollinen harha, joka johtuu pienestä määrästä opiskelijoita, joilla meni yli 25 minuuttia suorittaa loppukoe. Kysymyksessä 13 pohdittiin algoritmisen ajattelun soveltamista arkielämään, joten abstraktin ajattelun taidon puute voi myös selittää tehtävässä menestymistä. Jos opiskelija ei kykene ajattelemaan algoritmisen ajattelun taitoja yleisenä ajattelun taitona, jota voi soveltaa, voi tehtävä tuottaa hankaluuksia. Toisaalta se, että nopeammin tehtävän suorittaneet opiskelijat saivat keskimäärin parempia pisteitä, ei tue tätä teoriaa. Selvää kuitenkin on, että yli 25 minuuttia loppukoetta tehnyt opiskelija suoriutui keskimääräisesti heikommin kysymyksestä 13, kuin loppukokeen nopeammin tehneet opiskelijat. Myös alle 15 minuuttia (suoritusaikaluokka 1) loppukoetta tehneet opiskelijat suoriutuivat verrattain huonosti, mutta silti paremmin, kuin yli 25 minuuttia loppukoetta tehneet opiskelijat. Nopeasti loppukokeen tehneiden ja merkittävän hitaasti loppukokeen tehneiden kehnompi menestyminen tehtävässä voi viitata myös harjoittelun puutteeseen. Opiskelija voi ohittaa tehtävän hutiloiden, koska ei osaa vastata siihen kunnolla, tai yrittää opetella asiaa loppukoetta tehdessään, jolloin suoritus aika venyy.

Algoritmisia rakenteita käsittelevissä kysymyksissä oli havaittavissa yllättävän vähän tilastollisia merkitsevyyksiä kysymyksissä. Kysymyksen 20 kohdalla oli aloitusaikaluokkien 1 ja 5, 2 ja 5, sekä 3 ja 5 välillä tilastollisesti merkitseviä eroavaisuuksia. Kyseisessä tehtävässä tulkittiin Scratch-ohjelmointiympäristöstä napattua algoritmia. Kyseinen algoritmi muodostui neljästä ehtolauseesta, sekä kolmesta silmukasta. Kysymyksessä oli muihin kysymyksiin verrattuna selkeästi eniten ohjelmoinnillisia rakenteita ja monimutkaisuutta. Onkin luontevaa epäillä, että aloitusaikaluokan 5 opiskelijoiden vireystilan laskusta johtuva looginen ajattelukyvyyn lasku on syyllinen huonoon menestykseen. Harjoittelun vähyyttä ei voi pois sulkea, sillä kysymyksessä 21 oli aikaisempien aloitusaikaluokkien kohdalla tilastollisesti merkitseviä eroavaisuuksia opiskelijoiden menestyksessä. Kysymyksessä 21 on yksi silmukkarakenne, sekä kaksi ehtolauseetta. Molempia kysymyksiä yhdistää

tarve kyetä pilkkomaan ongelmaa olemassa olevien vihjeiden perusteella pienemmiksi palasiksi, sekä tarve kyetä ajattelemaan algoritmisesti. Molemmat ovat harjoiteltavissa olevia taitoja, kuten luvuissa 2 ja 4 esitetään, kunhan ajattelu on kehittynyt riittävästi, kuten luvussa 3.2.1 esitetään.

Suuri osa loppukokeen kysymyksistä jäi analysoimatta, koska Moodle-ympäristö ei tarjonnut mahdollisuutta yhdistää yksiselitteisesti kysymystä ja opiskelijan antamaa vastausta esimerkiksi tosi – epätosi -kysymyksissä. Näiden kysymysten analysoinnissa en nähnyt järkeä, sillä esimerkiksi Sanastoa – kategoriassa on 18 kysymystä, joista opiskelijalle arvotaan 8. Ei ole mitään takuuta siis siitä, että opiskelijoilla olisi sama kysymys samassa kohdassa. Pelkästään kahdessa ensimmäisessä kysymyspaikassa nämä 18 kysymystä voivat järjestäytyä 306 eri tavalla. Määritelmät -kategorian ensimmäistä ja neljättä, sekä käytännön algoritmit – kategorian kahta ensimmäistä kysymystä oli vastausten perusteella mahdotonta erottaa toisistaan. Jälkimmäiset ovat loppukokeen analyysin kannalta merkittävä haitta, sillä se heikentää mahdollisuutta pohtia kurssimateriaalin luvun kuusi sisältöä merkittävästi.

## **8.2 Loppukokeen analysointi**

Loppukokeen analysointi pohjautuu koko kysymyspankkiin, eikä yksittäiseen kysymykseen. Kysymykset on kuvailtu siten, että niistä ei ilmene loppukokeen suorittamiseen helpottavaa tietoa, mutta kuitenkin riittävästi tietoa analysointia varten.

Loppukokeen ensimmäiset kysymykset käsittelevät algoritmisen ajattelun määritelmiä. Määritelmässä käsitellään yleisellä tasolla algoritmisen ajattelun vaiheita, sekä siihen kuuluvia osia. Verratessa kappaleessa 2 esiteltyjä algoritmisen ajattelun osia, huomataan kysymyksissä esiintyvän algoritmisen ajattelun osa-alueet arvioivaa ajattelua lukuunottamatta. On kuitenkin huomioitava, että määritelmäkysymyksissä eriytetään ohjelmoinnillinen ajattelu ja algoritmisen ajattelu toisistaan, vaikka ne ovatkin synonyymejä keskenään, kuten todettiin Knuthin (1985), Wingin (2008), Selbyn & Woollardin (2013) sekä Cooperin (2000) tutkimuksissa sekä artikkeleissa.

Seuraavat kysymykset lähestyvät algoritmisen ajattelun määritelmää väittämien kautta. Kyseessä on oikein-väärin -väittämiä, jotka keskittyvät algoritmin määritelmään. Knuthin (1997) algoritmin osa-alueisiin peilaten väittämäkysymykset eivät keskity algoritmin määrittelyyn väittämien kautta. Sen sijaan kysymyksissä pohditaan yleisemmin, mitä algoritmit ovat, mitkä ovat niiden rajoitteita, sekä mitä ominaisuuksia algoritmille kuuluu. Osa Knuthin (1997) algoritmin piirteistä tulee esille, kuten esimerkiksi algoritmin definiittisyys eli yksiselitteisyys. Väittämät onnistuvat hyvin yleisellä tasolla pohtimaan, mitä algoritmit ovat, mutta ne eivät pureudu algoritmin määreeseen yhtä syvällisesti, kuin Knuthin (1997) määritelmä. Väittämissä ei myöskään pureuduta ajattelun taitoihin, joita algoritmiin ajatteluun tarvitaan.

Määritelmien jälkeen algoritmeja lähestytään arjen esimerkeillä. Kysymyksissä esitetään arkisia tilanteita ja vastaajan tulee arvioida, mitkä algoritmit tilanteeseen sopivat. Mukana on myös yksi kysymys, joka sisältönsä puolesta soveltuu paremmin algoritmin määrittelyyn, kuin arjen esimerkkeihin algoritmeista. Tässä kysymyksessä pohditaan toimivan algoritmin osa-alueita. Kokonaisuutena kysymysten anti muistuttaa enemmän yhdistelmää alun määritelmäkysymyksistä ja väittämäkysymyksistä sen sijaan, että näissä kysymyksissä tutkittaisiin uusia algoritmisen ajattelun osa-alueita. Kysymykset eivät suoranaisesti mittaa algoritmista ajattelua, vaan ennemminkin pyrkivät soveltamaan algoritmisen ajattelun taitoja jokapäiväisten esimerkkien kautta sitouttamalla.

Arjen esimerkkien jälkeen päästään seuraavaan kysymyskategoriaan: Algoritmiset rakenteet. Kysymyksissä esitetään algoritmi, joka vastaajan tulee tulkita oikein. Algoritmit koostuvat silmukoista, ehtolauseista, sekä ulostuloista. Kysymykset testaavat vastaajalta kykyä ajatella algoritmisesti. Vastaajan pitää kyetä kuvittelemaan päässään algoritmi, joka hänelle näytetään tekstimuodossa tai kuvana kyetäkseen vastaamaan kysymykseen. Esitettävät algoritmit ovat yksiselitteisiä, eli niissä ei jää tulkinnan varaa, kuten algoritmista ei kuuluukaan jäädä. Kyetäkseen vastaamaan kysymykseen, tulee opiskelijan kyetä lisäksi seuraamaan algoritmia ja sen suoritusta, mikä vaatii kykyä ajatella algoritmisesti ja jopa hie-man dekompositiotaitoja. Opiskelijan tulee osata pilkkoa algoritmi pieniin palasiin ja ajatella algoritmisesti kyetäkseen seurata suoritusta. Algoritmisissa rakenteissa siis testataan opiskelijan kykyä ajatella algoritmisesti ja jossain määrin dekompositiotaitoja.



Algoritmisten rakenteiden jälkeen kysymyspankista löytyy yleistä sanastoa algoritmiseen ajatteluun ja algoritmeihin liittyen. Kysymykset liittyvät moniin erinäisiin algoritmisen ajattelun termeihin, sekä algoritmin määrittelyyn. Tämän osion tehtävät testaavat muun muassa vastaajan ymmärrystä algoritmin määritelmästä. Tässä osiossa on myös yleisiä, ohjelmoinnissakin paljon käytettyjä termejä, joiden merkitystä kysellään. Lisäksi Sanasto-osioista löytyy yleisiä kysymyksiä, jotka eivät liity suoraan algoritmiseen ajatteluun, vaan enemmänkin algoritmisen ajattelun historiaan. Historiaa käsitteleviä kysymyksiä on selkeästi vähemmän verrattuna sanastokysymyksiin. Tämän kysymysoSION kysymyksissä käsitellään hyvin Knuthin (1997) määritelmä algoritmista. Jokaista Knuthin algoritmin osaa varten on vähintään yksi kysymys pankissa. Sanasto-osion kysymykset käsittelevät hyvin kattavasti algoritmin määrittelyn, sekä joitakin ohjelmoinnillisia käsitteitä, kuten muuttuja tai pseudokoodi.

Kysymyspankin viimeisessä ryhmässä pohditaan käytännön esimerkkejä algoritmeista ja algoritmisesta ajattelusta. Esimerkkiosion kysymyksissä ei suoraan pohdita algoritmisen ajattelun tai algoritmien hyötyä ongelmanratkaisumenetelmänä tai ajattelumallina. Kysymykset pohtivat lähinnä yritysten käyttämiä algoritmeja. Algoritmisen ajattelun ja algoritmien käytännön esimerkeissä voi mielestäni pohtia myös tilanteita, missä algoritmisen ajattelu voi tuottaa nopeampia ratkaisuja tai lisäarvoa ongelman ratkaisuun yleistettävien ratkaisumenetelmien muodossa. Käytännön esimerkissä voisi olla tehtävä, jossa kuvaillaan ongelma, joka ratkaistaan tehtävänannossa. Vastaajan tulisi kertoa ratkaisun hyvät tai huonot puolet. Vaihtoehtoisesti, samalla tehtävänannolla voisi esittää kysymyksen, missä kaikkialla tehtävänannossa esitetty ratkaisutapa olisi käytettävissä muuten, vai onko ollenkaan.

Loppukokeen kysymysten perusteella voi todeta loppukokeen testaavan melko kokonaisvaltaisesti, ymmärtääkö vastaaja, mitä on algoritmi, sekä ymmärtääkö vastaaja algoritmisen ajattelun tunnusomaiset piirteet. Tutkimuskysymykseen kaksi voidaan vastata loppukokeen mittaavan algoritmista ajattelua, muttei täydellisesti. Kysymyspankissa on myös kysymyksiä, jotka eivät liity suoraan algoritmisen ajattelun ymmärtämiseen, saati algoritmeihin. Nämä kysymykset eivät mielestäni tuo lisäarvoa loppukokeelle, sillä ne testaavat lähinnä sitä, onko loppukokeen vastaaja tutustunut kaikkeen opiskeltavaan materiaaliin

kaikilta osin. Ne eivät testaa sitä, osaako opiskelija ajatella algoritmisesti, saati ymmärtääkö opiskelija algoritmisen ajattelun osa-alueita. Käytännön esimerkit – kategoriassa jokainen kysymys on tällainen. Sanastoa-osiossa on myös algoritmisen ajattelun historiaan liittyviä kysymyksiä, jotka eivät mittaa opiskelijan ymmärrystä algoritmisesta ajattelusta. Ne eivät myöskään testaa, ymmärtääkö opiskelija, mikä on algoritmi. Kurssin nimi on Johdatus algoritmiseen ajatteluun. Nimeen peilaten kysymykset, jotka eivät suoraa mittaa ymmärrystä algoritmeista ja algoritmisesta ajattelusta, ovat perusteltuja. Aihealueen johdatte- luun esimerkiksi historian tutkiminen on mielestäni hyvä tapa.

### **8.3 Kurssin aineiston analyysi**

Kurssin aineisto on sekoitus tekstiä ja videoita, jotka käsittelevät algoritmista ajattelua eri lailla. Aineiston laatu vaikutti päätökseen analysoida aineisto teorialähtöisillä sisällönanalyysin metodeilla aineistolähtöisyyden sijaan. (Tuomi & Sarajärvi, 2018) Kurssiaineiston analyysissä keskitytään kerrallaan yhteen teoriaosuudessa käsiteltyyn algoritmisen ajattelun osa-alueeseen ja analysoidaan sen käsittelyä kurssiaineistossa. Lisäksi analysoidaan kurssiaineistoa tutkien, löytyykö kurssiaineistosta ajattelun taitoja, joita tarvitaan algoritmiseen ajatteluun. Analyysin tavoitteena on kyetä vastaamaan tutkimuskysymykseen kolme kattavasti. Analyysia tutkiessa on syytä huomata, että sisällönanalyysin ollessa laadullinen menetelmä, on tutkijan omilla näkemyksillä ja ennakkoasenteilla taipumus vaikuttaa lopputulokseen. Kurssiaineiston analyysin perusteella voi sanoa, että loppukoe on mahdollista tehdä aineiston pohjalta, mutta aineistossa ei opeteta kaikkea algoritmiseen ajatteluun liittyviä ominaisuuksia tai ajattelun taitoja, mitä tutkimuskysymyksessä kolme pohdittiin.

#### **8.3.1 Algoritmisen ajattelun osa-alueet kurssimateriaaleissa**

Luvuissa 2 sekä 3 käsiteltiin algoritmiseen ajatteluun liittyviä ajattelun taitoja, sekä kognitiivisia prosesseja, joita tarvitaan algoritmisessa ajattelussa. Aineistossa käsitelläänkin algoritmista ajattelua ja siihen kuuluvia prosesseja, kuitenkin vajavaisesti.

Algoritmisen ajattelun määritelmä esiteltiin alaluvussa 2.1. Aineiston analysoinnissa havaitaan, että missään ei selkeästi mainita kaikkia algoritmiseen ajatteluun kuuluvia osa-

alueita. Johdantokappaleessa on kaksi videota, joissa esitellään ohjelmoinnillinen ajattelu sekä algoritminen ajattelu. Videoiden perusteella voisi päätellä, että algoritminen ajattelu ja ohjelmoinnillinen ajattelu ovat eri asioita, tarkkaan ottaen algoritminen ajattelu olisi ohjelmoinnillisen ajattelun osa-alue. Toisaalta, koska algoritmiseen ajatteluun kuuluu kyky ajatella algoritmisesti, on mahdollista, että termit sekoittuvat keskenään, sillä molempiin, algoritmiseen ajatteluun ja kykyyn ajatella algoritmisesti, viitataan samalla termillä englannin kielessä: Algorithmic thinking. Vasta tutustuttuaan Selbyn ja Woollardin meta-analyysiin (2013) selviää, että algoritmista ajattelua käsittelevällä videolla keskitytään nimenomaan kykyyn ajatella algoritmisesti itse algoritmisen ajattelun sijaan. Algoritmisen ajattelun määrittelykappaleessa ei myöskään missään vaiheessa mainita Selbyn ja Woollardin (2013) meta-analyysissä löydettyä arvioivaa ajattelua osana algoritmista ajattelua. Kurssin aineistosta puuttuu tutkittuun kirjallisuuteen peilaten yksi viidestä algoritmisen ajattelun ominaisuudesta. Arvioivan ajattelun lisääminen algoritmisen ajattelun määritelmäosioon onkin tärkeää, jos halutaan saada algoritmiselle ajattelulle täydellisempi määritelmä.

Algoritmista ajattelua ja sen taitoja pyritään yleistämään kurssimateriaalissa arkipäiväisiin asioihin. Koska algoritminen ajattelu on ajattelun taito, on arkipäiväiset esimerkit yksi tapa osoittaa, kuinka algoritminen ajattelu näkyy arkisessa elämässä. Kurssin tavoitteena on johdatella algoritmiseen ajatteluun, kurssin nimen mukaisesti. Johdatteluun reaali maailman esimerkit ovat toimiva ratkaisu.

Algoritmiseen ajatteluun kuuluu lisäksi nimensä mukaisesti algoritmit. Algoritmin määritelmää ja algoritmisia rakenteita käydään materiaalissa useamman osion verran läpi. Esimerkiksi Sanastoa-osiossa käydään algoritmin määritelmää läpi videolla. Videolla käydään läpi algoritmin määritelmä matematiikan näkökulmasta, mutta se sisältää myös Knuthin (1997) algoritmin määritelmän mukaiset osat. Lisäksi samalla videolla käydään läpi algoritmisia rakenteita, kuten ehtolauseita ja toistorakenteita. Nämä liittyvät algoritmin yksiselitteisyyteen ja luettavuuteen. Samoja rakenteita käydään läpi myös Algoritmisia rakenteita – osiossa.

### **8.3.2 Kognitiiviset prosessit kurssimateriaaleissa**

Muita kognitiivisia prosesseja, kuten ongelmanratkaisutaidon tai kriittisen ajattelun taitojen merkitystä käsitellään kurssin materiaaleissa jonkin verran. Ongelmanratkaisulle on omistettu kokonaan oma osionsa. Osiossa käsitellään ja pohditaan, millaisiin ongelmiin algoritmista ajattelua voidaan hyödyntää sen sijaan, että keskityttäisiin ongelmanratkaisun taitoihin, joita algoritmisessa ajattelussa vaaditaan. Sisältö on siis yhtenevä otsikon kanssa. Ainoa kerta, kun ongelmanratkaisutaito mainitaan algoritmisen ajattelun yhteydessä, on johdanto-osion Computational Thinking -videossa. Samassa yhteydessä mainitaan loogisen ajattelun ja kriittisen ajattelun tarve. Kurssin materiaaleissa ei muualla mainita näitä ajattelun taitoja, jotka oleellisesti liittyvät algoritmiseen ajatteluun. Luvussa 3 mainittiin algoritmiseen ajatteluun liittyviä kognitiivisia prosesseja, jotka liittyvät merkittävästi algoritmiseen ajatteluun. Näiden kognitiivisten prosessien esittely puuttuu käytännössä kokonaan. Lisäämällä tietoa kognitiivisista prosesseista, jotka vaikuttavat algoritmisen ajattelun taustalla, monipuolistetaan kurssimateriaalia ja laajennetaan kurssin puitteissa algoritmisen ajattelun määritelmää täydellisemmäksi.

## 9 Pohdinta ja luotettavuuden arviointi

Loppukokeen tuloksia analysoimalla saatiin varsin yllättäviä havaintoja. Mielestäni merkittävin tulos oli, että loppukokeeseen käytetty aika ei vaikuttanut tilastollisesti merkittävästi saatuun arvosanaan, vain muutamiin kysymyksiin. Syitä tulokseen on monia, mutta uskon tärkeimpänä olevan opiskeltavana olleen aineiston sisäistämisen palvelevan merkittävimmin tätä tulosta, sekä vireystilan olevan merkittävä osa loppukokeessa selviytymistä. Tulos viittaa myös siihen, että jos opiskelija osaa opiskeltavan asian tai on lahjakas testattavassa aiheessa, ei hänelle tuota vaikeuksia hyödyntää opittua tietoa. Toisaalta loppukoe on verkossa suoritettava, joten ei voida poissulkea lunttauksen mahdollisuutta. On myös mahdollista, että loppukokeen on tehnyt joku muu, kuin Moodle-ympäristöön kirjautunut käyttäjä. Tuloksia tarkastellessa onkin syytä ottaa huomioon se, että verkossa suoritettavassa loppukokeessa, jossa ei näennäistä valvontaa ole, on lunttaus varsin houkutteleva vaihtoehto. Moodle-ympäristön tuomat rajoitteet sotkevat muutenkin tutkimusta, sillä suuri osa loppukokeen kysymyksistä jäi kunnolla analysoimatta ympäristön rajoitteista johtuen.

Loppukokeen tulosten validiteetin lisäämiseksi olisi mielestäni hyvä tehdä kurssin alussa tasotestin tapainen koe, millä nähdään opiskelijan senhetkinen osaaminen algoritmisessa ajattelussa. Tällöin pystyttäisiin tutkimaan entistä tarkemmin, mittaako loppukoe oikeasti algoritmisen ajattelun osa-alueiden kehittymistä. Tässä tutkimuksessa aloitustasotestin tekeminen ei ollut mahdollista, sillä tutkimusaineisto oli kerätty hieman jälkijunassa. Tulevaisuudessa, kun tutkitaan algoritmisen ajattelun kehittymistä tämän kurssin puitteissa tai jonkin muun opintokokonaisuuden puitteissa, lähtötasotestin tekeminen ja tulosten vertailu olisikin merkittävä edistysaskel todentamaan tutkimustuloksia. Lisäksi loppukokeen tulosten validiteetin lisäämiseksi olisi tarvittu mahdollisuus analysoida jokainen kysymysvastauspari. Tätä Moodle ja loppukokeen rakenne eivät mahdollistaneet, mikä syökin loppukokeen tulosten validiteettia.

Loppukokeen ja kurssiaineiston teorialähtöistä sisällönanalyysin perusteella hyvällä validiteetilla varustettuja. Loppukoe mittaa analyysin perusteella sitä, mitä kurssiaineistossa opiskellaan, mutta aineistossa ei käydä läpi kaikkia algoritmisen ajattelun ominaisuuksia. Luvuissa 2-4 on käyty hyvin kattavasti läpi algoritmiseen ajatteluun liittyviä osa-alueita ja

näitä puolestaan peilattu kurssimateriaaliin. Tämän perusteella kurssimateriaalin sisällyönanalyysin validiteetti on melko hyvä. Validiteettia voi väittää hyväksi, sillä aineiston analyysi on perustunut monipuoliseen ja laajaan teoriataustaan, jota esitellään luvuissa 2-4.

Loppukokeessa oli muutamia kysymyksiä, joissa osalla opiskelijoista oli selkeästi vaikeuksia. Suurimmat vaikeudet vaikuttivat olevan algoritmien tulkinnessa niillä opiskelijoilla, jotka tekivät loppukokeen myöhään illalla tai yöllä. Tämä viittaa siihen, että vireystilalla on suuri merkitys siinä, miten algoritmien tulkitseminen, kyky ajatella algoritmisesti ja pilkkoa algoritmia pieniin palasiin, sujuu. Tulos ei yllättänyt.

Loppukokeesta saatuihin pistemääriin vaikuttaa myös algoritmisen ajattelun harjoittelu. Aineistossa ei Moodlen teknisten rajoitteiden takia ollut mahdollisuutta tutkia, kauanko opiskelija vietti aikaa opiskeltavan materiaalin parissa. Harjoittelun vaikutuksesta algoritmiseen ajatteluun on tehty positiivisia havaintoja, kuten luvussa 4.4 todetaan. Loppukokeen validiteettia silmällä pitäen tasotestin lisäksi olisi mielenkiintoista tutkia, kauanko opiskelija kulutti aikaa materiaalin parissa, sekä kuinka paljon tämä vaikutti loppukokeen ja tasotestin väliseen eroon. Toisaalta jo pelkästään tieto opiskelijan materiaalissa viettämästä ajasta toisi lisäarvoa tutkimukselle. Materiaalissa vietettyä aikaa voisi verrata loppukokeessa menestymiseen ja sitä kautta tutkia, vaikuttiko harjoittelu loppukokeessa menestymiseen.

Loppukokeen analysointi puolestaan osoitti, että loppukokeessa testataan algoritmisen ajattelun osaamista monipuolisesti, mutta algoritmipainotteisesti. Kysymyksissä on lisäksi termin historiaan liittyviä kysymyksiä, joilla ei suoranaisesti mitata algoritmista ajattelua mitenkään. Luvussa 2 käytiin läpi algoritmiseen ajatteluun liitettäviä ominaisuuksia, algoritmin käsite sekä ajattelun taitoja, jotka vaikuttavat algoritmiseen ajatteluun.

Loppukokeen kysymyspankista löytyy vähintään yksi kysymys jokaista algoritmin määritelmään liitettyä termiä kohden. Lisäksi kysymyksissä pohditaan algoritmin määritelmää yleisesti. Luvussa 2 esiteltujen algoritmin määritelmien perusteella loppukokeessa pyritään mittaamaan, ja mitataankin, vastaajan ymmärrystä siitä, mikä on algoritmi. Ymmärrys siitä, mikä on algoritmi, onkin olennainen osa algoritmista ajattelua, kuten nimestäkin voi päätellä. Lisäksi loppukokeessa testattiin vastaajalta algoritmisen ajattelun ominaisuuksia,

kuten abstraktien asioiden käsittelyä, hahmotuskykyä, sekä kykyä pilkkoa ongelmaa pienempiin paloihin. Yleistämistä testaavia kysymyksiä ei harmillisesti ollut. Parantaakseen algoritmisen ajattelun testaamista, olisi loppukokeeseen hyvä lisätä kysymyksiä, jotka käsittelevät yleistämistä. Esimerkiksi tehtäväpari, joista ensimmäisessä luodaan algoritmi ja toisessa pohditaan, mihin kaikkeen algoritmia voisi hyödyntää, voisi palvella yleistämisen testaamista. Lisäämällä yleistämistä testaavia kysymyksiä, loppukoe muuttuisi paremmin algoritmista ajattelua testaavaksi. Lisäksi loppukokeessa voisi mielestäni yhdistää algoritmisen ja ohjelmoinnillisen ajattelun kysymykset keskenään. Kuten luvussa 3.1 todettiin, on ohjelmoinnillinen ajattelu ja algoritmisen ajattelu pohjimmiltaan yksi ja sama asia. Samoista ajattelun taidoista kahden eri termin käyttäminen näinkin lyhyen opintojakson sisällä voi aiheuttaa hämmennystä.

Kurssimateriaalin analysointi osoitti, että materiaalista puuttuu algoritmisen ajattelun taitoja, jotka kuuluvat algoritmiseen ajatteluun Selbyn ja Woollardin (2013) meta-analyysin, sekä Weesen (2017) tutkimuksen mukaan. Kurssin nimi – Johdatus algoritmiseen ajatteluun – viittaa siihen, että algoritmisen ajattelun taustoja esitellään, kuten myös taitoja, joita algoritmiseen ajatteluun vaaditaan. Kurssimateriaalin jatkokehityksen kannalta näen tärkeänä lisätä materiaalin vähintäänkin täydentävää materiaalia siten, että algoritmisen ajattelu määritellään, kuten Selbyn ja Woollardin (2013) tutkimuksessa se on määritelty. Lisäämällä algoritmisen ajattelun määrittelyyn puuttuvat osat, parannetaan kurssin sisältöä. Näkisin myös algoritmiseen ajatteluun vaadittavien kognitiivisten prosessien, kuten kriittisen ajattelun, sekä ongelmanratkaisutaidon ja loogisen ajattelukyvyyn lisäämisen materiaaliin hyvänä lisänä. Lisäämällä näistä prosesseista tietoa, saadaan aikaan entistä kattavampi johdatus algoritmiseen ajatteluun. Vaarana on kurssin paisuminen paljon nykyistä suuremmaksi. Kurssin osioita tarkastelemalla näkisin, että osioita yhdistelemällä saataisiin tilaa uudelle tiedolle. Esimerkiksi Arjen Algoritmit -osion voisi yhdistää Käytännön esimerkkejä algoritmeista – osion kanssa. Tämän voisi toteuttaa esimerkiksi siten, että erilaisia algoritmeja esiteltäisiin, kuten nyt Arjen Algoritmit – osiossa, käytännön esimerkkien kautta. Vastaus tutkimuskysymykseen kolme onkin, että loppukoe on mahdollista tehdä materiaalien perusteella, mutta kurssilla ei opeteta kaikkea algoritmiseen ajatteluun liittyvää.

Tutkimuksen arvo tieteellisenä tutkimuksena ei ole nykyisellään kovin merkittävä, sillä kyseessä on enemmän tarkastelu yliopiston kurssista, kuin tiedettä tekevä tutkimus. Aineistona oli yhden suomalaisen yliopiston kurssin loppukokeen aineisto, mikä ei anna kattavaa kuvaa edes Suomen yliopisto-opiskelijoiden algoritmisen ajattelun osaamisesta. Aineistosta puuttui myös monia mielenkiintoisia tietoja, joita analysoimalla tutkimukseen olisi saatu paljon tutkimusarvoa, sekä validiteettia. Esimerkiksi opiskelijan tiedekunnalla, materiaaliin käytetyllä ajalla, sekä lähtötasolla olisi saatu tutkimukseen merkittävästi lisää tieteellistä arvoa. Ymmärryksen mukaan Moodle-ympäristö ei näitä tietoja tarjoa saataville. Oppimisympäristöä lienee siis tarpeellista kehittää tai vaihtaa näiden tietojen saamista aj. Jatkotutkimusideana samaa kurssia voisi analysoida paremmalla aineistolla, mistä ilmeneisi tiedekunnat. Tällöin voisi tutkia eroja tiedekuntien välillä ja etsiä sitä kautta selittäviä eroja algoritmisen ajattelun tasossa. Toisaalta, ennen tiedekuntien välisten erojen tutkimusta olisi hyvä kurssimateriaalia päivittää siten, että siellä käsitellään kaikki algoritmisen ajattelun osa-alueet. Nykyisessä ratkaisussa arvioiva ajattelu jää kokonaan pois. Lisäksi, on nykyisellä Moodlesta saatavalla aineistolla mahdoton selvittää yksiselitteisesti yhteen vastaukseen yksiselitteisesti kysymystä, jos loppukoe sisältää vähintään kaksi tosi-epätosi - kysymystä. Loppukokeen analysointiin pohjaavalle tutkimukselle Moodle ei nykyisellään ole paras ympäristö, sillä se vaatii kokeen tekijältäkkin paljon pohdintaa kokeen rakenteesta ja vastauksista, että kysymykset olisivat myöhemmin yksiselitteisesti jäljitettävissä. Myös pitkittäinen kehittämistutkimus olisi mahdollista toteuttaa, sillä kurssi toteutuu useamman kerran lukuvuoden aikana. Tällöin kurssitoteutusten välinen syklinen kehitys olisi mahdollista. Kurssin kehittäminen kehittämistutkimuksena vaatisi kuitenkin lisää tietoja muun muassa aineiston käyttöön kulutetusta ajasta.



# Lähteet

## Bibliography

- Abele, Andrea. 1985. "Thinking about Thinking: Causal, Evaluative and Finalistic Cognitions about Social Situations." *European Journal of Social Psychology* 15 (3): 315-332.
- Barbarese, William J., Slavica K. Katusic, Robert C. Colligan, Amy L. Weaver, and Steven J. Jacobsen. 2005. "Math Learning Disorder: Incidence in a Population-Based Birth Cohort, 1976–82, Rochester, Minn." *Ambulatory Pediatrics* 5 (5): 281-289. doi:10.1367/A04-209R.1. <https://www.sciencedirect.com/science/article/pii/S1530156705603632>.
- Blank, Marion and Frances Solomon. 1968. "A Tutorial Language Program to Develop Abstract Thinking in Socially Disadvantaged Preschool Children." *Child Development*: 379-389.
- Blass, Andreas and Yuri Gurevich. 2004. "Algorithms: A Quest for Absolute Definitions." In *Current Trends in Theoretical Computer Science: The Challenge of the New Century Vol 1: Algorithms and Complexity Vol 2: Formal Models and Semantics*, 283-311: World Scientific.
- Brookfield, Stephen. 2020. "Teaching for Critical Thinking." In *Handbook of Research on Ethical Challenges in Higher Education Leadership and Administration*, 229-245: IGI Global.
- Calao, Luis Alberto, Jesús Moreno-León, Heidy Ester Correa, and Gregorio Robles. 2015. "Developing Mathematical Thinking with Scratch." In *Design for Teaching and Learning in a Networked World*, 17-27: Springer.
- Cracolice, Mark S., John C. Deming, and Brian Ehlert. 2008. "Concept Learning Versus Problem Solving: A Cognitive Difference." *Journal of Chemical Education* 85 (6): 873. doi:10.1021/ed085p873. <https://search.proquest.com/docview/212032513>.
- Day, Mary Carol. 1981. "Thinking at Piaget's Stage of Formal Operations." *Educational Leadership* 39 (1): 44-45.
- Dinges, D. F., F. Pack, K. Williams, K. A. Gillen, J. W. Powell, G. E. Ott, C. Aptowicz, and A. I. Pack. 1997. "Cumulative Sleepiness, Mood Disturbance, and Psychomotor Vigilance Performance Decrements during a Week of Sleep Restricted to 4–5 Hours Per Night." *Sleep* 20 (4): 267. doi:10.1093/sleep/20.4.267. <https://www.ncbi.nlm.nih.gov/pubmed/9231952>.

- Doleck, Tenzin, Paul Bazelaïs, David John Lemay, Anoop Saxena, and Ram B. Basnet. 2017. "Algorithmic Thinking, Cooperativity, Creativity, Critical Thinking, and Problem Solving: Exploring the Relationship between Computational Thinking Skills and Academic Performance." *Journal of Computers in Education* 4 (4): 355-369.
- Earl, Lorna and Helen Timperley. 2015. "Evaluative Thinking for Successful Educational Innovation." .
- Eckerdal, Anna, Michael Thuné, and Anders Berglund. Oct 1, 2005. "What does it Take to Learn 'Programming Thinking'?" *ACM*, . doi:10.1145/1089786.1089799. <http://dl.acm.org/citation.cfm?id=1089799>.
- Grigorenko, Elena L., Donald L. Compton, Lynn S. Fuchs, Richard K. Wagner, Erik G. Willcutt, and Jack M. Fletcher. 2019. "Understanding, Educating, and Supporting Children with Specific Learning Disabilities: 50 Years of Science and Practice." *American Psychologist*. doi:10.1037/amp0000452. <http://search.ebscohost.com/login.aspx?direct=true&db=psych&AN=2019-25332-001&site=ehost-live>.
- Harris, Anette, Holger Ursin, Robert Murison, and Hege R. Eriksen. 2007. "Coffee, Stress and Cortisol in Nursing Staff." *Psychoneuroendocrinology* 32 (4): 322-330. doi:10.1016/j.psyneuen.2007.01.003. <https://www.clinicalkey.es/playcontent/1-s2.0-S0306453007000121>.
- Ho, Chun-Heng. 2001. *Some Phenomena of Problem Decomposition Strategy for Design Thinking: Differences between Novices and Experts*. Vol. 22. doi:[https://doi.org/10.1016/S0142-694X\(99\)00030-7](https://doi.org/10.1016/S0142-694X(99)00030-7). <http://www.sciencedirect.com/science/article/pii/S0142694X99000307>.
- Israel, Maya, Quentin M. Wherfel, Jamie Pearson, Saadeddine Shehab, and Tanya Tapia. 2015. "Empowering K–12 Students with Disabilities to Learn Computational Thinking and Computer Programming." *TEACHING Exceptional Children* 48 (1): 45-53. doi:10.1177/0040059915594790. <https://journals.sagepub.com/doi/full/10.1177/0040059915594790>.
- JYU. "ITKP0003 Johdatus Algoritmiseen Ajatteluun .", accessed 6.12., 2020, <https://www.jyu.fi/ops/fi/it/informaatioteknologian-tiedekunnan-erilliset-opintokokonaisuuudet/unit/20711>.
- Kincheloe, Joe L. and Shirley R. Steinberg. 1993. "A Tentative Description of Post-Formal Thinking: The Critical Confrontation with Cognitive Theory." *Harvard Educational Review* 63 (3): 296-321. doi:10.17763/haer.63.3.h423221226v18648. <https://search.proquest.com/docview/212279037>.
- Knuth, Donald E. 1985. "Algorithmic Thinking and Mathematical Thinking." *The American Mathematical Monthly* 92 (3): 170-181.

- Knuth, Donald Ervin. 1997. *The Art of Computer Programming*. Vol. 3 Pearson Education.
- KORKEAKOULUOPISKELIJOIDEN and TERVEYSTUTKIMUS 2016. *Ylioppilaiden Terveystutkimuksen Tutkimuksia* 48.
- Kruskal, William H. and W. A. Wallis. 1952. "Use of Ranks in One-Criterion Variance Analysis." *Journal of the American Statistical Association* 47 (260): 583-621. doi:10.1080/01621459.1952.10483441. <https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441>.
- L. D. Miller, L. Soh, V. Chiriacescu, E. Ingraham, D. F. Shell, S. Ramsay, and M. P. Hazley. 2013. *Improving Learning of Computational Thinking using Creative Thinking Exercises in CS-1 Computer Science Courses*. doi:10.1109/FIE.2013.6685067.
- L. Liao and J. Liang. 2017. *An Empirical Study on Blended Learning to Promote the Development of Computational Thinking Ability of College Students*. doi:10.1109/ISET.2017.64.
- Lamagna, Edmund A. 2015. "Algorithmic Thinking Unplugged." *Journal of Computing Sciences in Colleges* 30 (6): 45-52.
- Licht, William. 2016. "Understanding College Students with a Learning Disability -- A Phenomenological Study." ProQuest Information & Learning. <http://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2016-26512-254&site=ehost-live>.
- Lihua Liao and Jin Liang. 2017. "An Empirical Study on Blended Learning to Promote the Development of Computational Thinking Ability of College Students." IEEE, .
- Marcelino, Maria José, Teresa Pessoa, Celeste Vieira, Tatiana Salvador, and António José Mendes. 2018. "Learning Computational Thinking and Scratch at Distance." *Computers in Human Behavior* 80: 470-477. doi:10.1016/j.chb.2017.09.025. <https://www.sciencedirect.com/science/article/pii/S0747563217305526>.
- Mikko Ketokivi. 2015. *Tilastollinen Päätely Ja Tieteellinen Argumentointi* Gaudeamus Oy.
- Nelson, Meta and Rebecca M. Eddy. 2008. "Evaluative Thinking and Action in the Classroom." *New Directions for Evaluation* 2008 (117): 37-46.
- Niilo Mäki-instituutti. a. *Lasten Neurokognitiiviset Häiriöt Ja Oppimisvaikeudet*.  
 ———. "Matemaattisten Oppimisvaikeuksien Määrittely"., accessed Sept. 10., 2019, <http://www.lukimat.fi/matematiikka/tietopalvelu/oppimisvaikeudet/matemaattisten-oppimisvaikeuksien-maarittely>.

Opetushallitus. 2014. *Perusopetuksen Opetussuunnitelman Perusteet 2014*

Paget, N. S., T. F. Lambert, and K. Sridhar. 1981. "Factors Affecting an Anaesthetist's Work: Some Findings on Vigilance and Performance." *Anaesthesia and Intensive Care* 9 (4): 359-365. doi:10.1177/0310057X8100900407. <https://www.ncbi.nlm.nih.gov/pubmed/7032351>.

Selby, Cynthia and John Woollard. 2013. "Computational Thinking: The Developing Definition." .

SHAH, JAYANT. 1991. "A History of Piṅgala's Combinatorics." *Journal of History of Science* 26: 1.

Smith, Jeffrey A. 2016. "Viewpoint." *The Canadian Journal of Economics* 49 (3): 871-905. <http://www.econis.eu/PPNSET?PPN=886123259>.

Smith, Leslie. 1996. *Critical Readings on Piaget*, edited by Leslie Smith. Florence: Routledge.

Süleyman, YAMAN. 2005. "Exploring Effectiveness of Problem Based Learning on Developing Logical Thinking Skills in Science Teaching." *Journal of Turkish Science Education* 2 (1): 31. <https://search.proquest.com/docview/1658721797>.

Swanson, H. L., Andres F. Olide, and Jennifer E. Kong. 2018. "Latent Class Analysis of Children with Math Difficulties and/Or Math Learning Disabilities: Are there Cognitive Differences?" *Journal of Educational Psychology* 110 (7): 931-951. doi:10.1037/edu0000252. <http://search.ebscohost.com/login.aspx?direct=true&db=psych&AN=2017-57183-001&login.asp&site=ehost-live>.

Terveyden ja hyvinvoinnin laitos. *Tautiluokitus ICD-10*.

Tobin, Kenneth G. and William Capie. 1981. "The Development and Validation of a Group Test of Logical Thinking." *Educational and Psychological Measurement* 41 (2): 413-423.

Tsalapatas, Hariklia, Olivier Heidmann, Rene Alimisi, and Elias Houstis. 2012. "Game-Based Programming Towards Developing Algorithmic Thinking Skills in Primary Education." *Scientific Bulletin of the Petru Maior University of Targu Mures* 9 (1).

Tuomi, Jouni and Anneli Sarajärvi. 2018. *Laadullinen Tutkimus Ja Sisällönanalyysi*. Uudistettu laitos ed. Helsinki: Kustannusosakeyhtiö Tammi. <https://www.ellibslibrary.com/jyu/9789520400118>.

- Valsiner, Jaan and Rene Van der Veer. 1988. "On the Social Nature of Human Cognition: An Analysis of the Shared Intellectual Roots of George Herbert Mead and Lev Vygotsky." *Journal for the Theory of Social Behaviour* 18 (1): 117-136.
- Vilkka, Hanna. 2007a. *Tutki Ja Mittaa : Määrällisen Tutkimuksen Perusteet*.  
[https://www.openaire.eu/search/publication?articleId=od\\_\\_\\_\\_\\_1073::3abce3585b8934fc15e10f7e7e4c5dfa](https://www.openaire.eu/search/publication?articleId=od_____1073::3abce3585b8934fc15e10f7e7e4c5dfa).
- . 2007b. *Tutki Ja Mittaa: Määrällisen Tutkimuksen Perusteet*.
- Warm, Joel S., Raja Parasuraman, and Gerald Matthews. 2008. "Vigilance Requires Hard Mental Work and is Stressful." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 50 (3): 433-441. doi:10.1518/001872008X312152.  
<http://www.ingentaconnect.com/content/hfes/hf/2008/00000050/00000003/art00016>.
- Weese, Joshua Levi. 2017. "Bringing Computational Thinking to K-12 and Higher Education." ProQuest Dissertations & Theses.  
<http://www.pqdtcn.com/thesisDetails/845C76B57138A5BA2BDA50F461D2F53A>.
- Wing, Jeannette M. 2006. "Computational Thinking." *Communications of the ACM* 49 (3): 33-35.
- . 2008. "Computational Thinking and Thinking about Computing."  
*Phil.Trans.R.Soc.A* 366: 3717-3725.
- Woods, Donald R., Andrew N. Hrymak, Robert R. Marshall, Philip E. Wood, Cameron M. Crowe, Terrence W. Hoffman, Joseph D. Wright, Paul A. Taylor, Kimberly A. Woodhouse, and C. G. Kyle Bouchard. 1997. "Developing Problem Solving Skills: The McMaster Problem Solving Program." *Journal of Engineering Education* 86 (2): 75-91. doi:10.1002/j.2168-9830.1997.tb00270.x.  
<https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2168-9830.1997.tb00270.x>.
- Woolley, John D. and Leland R. Miller. 1974. "LINUS: A Structured Language for Instructional Use."
- Zhao, Fengqing, Sen Li, Tianran Li, and Guoliang Yu. 2019. "Does Stereotype Threat Deteriorate Academic Performance of High School Students with Learning Disabilities? the Buffering Role of Psychological Disengagement." *Journal of Learning Disabilities* 52 (4): 312-323. doi:10.1177/0022219419849107.  
<http://search.ebscohost.com/login.aspx?direct=true&db=psych&AN=2019-34451-003&site=ehost-live>.
- Zsakó, László and Péter Szlávi. 2012. "ICT Competences: Algorithmic Thinking." *Acta Didactica Napocensia* 5 (2): 49-58.