# Tallinna Reaalkool

# Rubiku kuubiku manipulaatori ehitamine ja optimeerimisalgoritmide analüüsimine

Uurimistöö

Joosep Näks

11.a

Juhendaja: Jaagup Kippar

# Sisukord

1. Lahendusalgoritmid	5
1.1. Terminoloogia	5
1.2. Kasutatavate algoritmide ülevaade	5
1.2.1. Thistlewhite algoritm	5
1.2.2. Kociemba algoritm	6
1.2.3. Korfi algoritm	7
1.3. Realisatsioon ja tulemuste analüüs	7
2. Seadme arhitektuur ja disain	8
2.1. Mehaanika	8
2.2. Elektroonika	8
2.3 Tarkvara	8

# Sissejuhatus

Rubiku kuubik on oma leiutamisest aastal 1974 alates kuulsust kogunud ning levinud igasse vanuse- ja huvigruppi. Rubiku kuubik on pakkunud huvi ka matematikutele ja loogikutele, kuna sellel on kokku erinevaid võimalikke asendeid ligikaudu 43 kvintiljonit ehk kui toore jõuga lahendama hakata suvalisi pöördeid proovides, siis isegi arvutil läheks kõigi asendite läbi proovimiseks üle kümne tuhande aasta. Kuna proovimine võtab liiga kaua aega, on vaja luua algoritme kuubiku lahendamiseks. Peale niisama lahendamise on ka huvitav ülesanne leida vähimat arvu pöördeid, millega saab Rubiku kuubikut igast võimalikust asendist lahendatud asendisse viia. Seda numbrit nimetatakse God's Numberiks ning selle täpset väärtust on üritatud leida sellest ajast saati, kui Rubiku kuubik algselt loodi.

Teema valik tuleneb sellest, et autorit huvitab Rubiku kuubikute lahendamine ning tekkis huvi luua arvutis Rubiku kuubiku simulaator, et oleks lihtsam uusi valemeid katsetada ning et oleks lihtsam testida erinevate algoritmide efektiivsust. Autori huvi tõttu robootika vastu tekkis ka idee ehitada füüsiline Rubiku kuubiku manipulaator, millega visualiseerida programmi tööd.

Töö eesmärk on valmistada Rubiku kuubiku manipulaator, millele antakse ette füüsiline Rubiku kuubik ja sisestatakse Rubiku kuubiku asend ning mis lahendab seejärel Rubiku kuubiku ära. Selle saavutamiseks analüüsib autor erinevaid optimeerimisalgoritme, mille abil saab Rubiku kuubikut lahendada võimalikult väikese arvu käikudega ning realiseerida vähemalt üks algoritm, mida ta seejärel kasutab manipulaatoris.

Käesolev uurimustöö koosneb kahest osast. Esimeses osas kirjeldatakse erinevaid Rubiku kuubiku jaoks loodud optimeerimisalgoritme, ning lõpuks ka realiseeritakse vähemalt üks. Teises osas ehitatakse manipulaator, mis on võimeline füüsilist Rubiku kuubikut ära lahendama, kui talle on ette antud, mis Rubiku kuubiku algpositsioon on. Manipulaator koosneb kolmest etapist. Esimene etapp on mehaanika, mille käigus ehitatakse vineerist raam, mille külge saab kinnitada mootorid. Teine etapp on elektroonika, mille käigus ühendatakse mootorid mikrokontrolleri külge ja mikrokontroller omakorda väikese arvuti külge. Kolmas etapp on tarkvara, mille käigus kirjutatakse programm, mis koosneb virtuaalsest Rubiku kuubiku simulaatorist, uurimustöö esimeses osas realiseeritud algoritmist, kasutajaliidesest, kust saab algpositsiooni sisestada, ja mootoritele käske saatvast liidesest.

Põhilisteks allikateks on Jaap Scherphuisi koduleht ja Speedsolvers Wiki. Mõlemad neist sisaldavad paljude Rubiku kuubiku algoritmide kirjeldusi. Allikateks on põhiliselt

internetiallikad, kuna Rubiku kuubiku teemal on vähe kirjalikke allikaid. Autor leidis vaid üksikud populaarteaduslikud artiklid, kuid antud uurimustöös käsitletavad algoritmid on vähem tuntud, kuna neid ei kasutata käsitsi Rubiku kuubiku lahendamiseks, ei räägita nendest pindmisemates allikates. On ka mõned põhjalikumad raamatud, nagu David Singmasteri "Notes on Rubik's Magic Cube" aastast 1981, kuid selliseid raamatuid on raske leida.

# 1. Lahendusalgoritmid

#### 1.1. Terminoloogia

Tahk – Rubiku kuubiku üks külg. Kui Rubiku kuubikut enda ees hoida siis hoidja poole jääv tahk on eesmine tahk, sellest teisel pool on tagumine tahk, üleval on ülemine tahk, all on alumine tahk, paremale poole vaatav tahk on parem tahk ning vasakule poole vaatav tahk on vasak tahk.

Veerandpööre – tegevus, kus pööratakse ühte tahku 90° päripäeva või vastupäeva.

Poolpööre – tegevus, kus pööratakse ühte tahku 180° (pole vahet, kas päripäeva või vastupäeva, kuna tulemus on sama).

Valem – hulk pöördeid, mis viivad Rubiku kuubiku ühest asendist teise. Valem esitatakse üldjuhul tähtede jadana, kus 'R' tähistab parema tahu päripäeva veerandpöörde tegemist, 'L' vasaku tahu pööramist, 'U' ülemise tahu pööramist, 'D' alumise tahu pööramist, 'B' tagumise tahu pöörmaist ja 'F' eesmise tahu pööramist. Kui tähele järgneb ülekoma, tähendab et seda tahku tuleb vastupäeva pöörata. Kui tähele eelneb 2, tähistab see seda, et seda tahku tuleb 2 korda pöörata ehk tuleb teha poolpööre.

God's Number – number, mis näitab vähimat võimalikku pöörete arvu, et Rubiku kuubik igast võimalikust positsioonist lahendatud positsiooni viia. God's Numberil on ülempiir ja alampiir, mida on alates God's Numberi esimesest leidmisest 1979 aastast järjest üksteisele lähemale viidud, kuni aastal 2010 tõestati selle täpne väärtus, milleks on 20. (Rokicki, Kociemba, Davidson, Dethridge 2014)

#### 1.2. Kasutatavate algoritmide ülevaade

#### 1.2.1. Thistlewhite algoritm

Thistlethwaite'i algoritmi töötas välja sõlmeteoreetik ja Tennessee Ülikooli matemaatika professor Morwen B. Thistlethwaite aastal 1981. Thistlethwaite'i algoritm on algoritm, mille eesmärk on Rubiku kuubik lahendada võimalikult väikese pöörete arvuga ja kasutab üle tuhande erineva valemi, seega on see põhiliselt mõeldud arvuti abil lahendamiseks. (Scherphuis 2014b) Thistlethwaite'i algoritm oli üks esimisi algoritme, mis kasutas Rubiku kuubiku asendite alamgruppide läbimist Rubiku kuubiku lahendamiseks. Thistlethwaite'i

algoritm on tähtis ka sellepärast, et see algoritm viis God's Numberi ülempiiri 94 pöörde pealt 52 pöörde peale. (Grol 2010)

Thistlethwaite'i algoritm jagab kõik Rubiku kuubiku asendid viite erinevasse gruppi ning lahendamiseks tuleb Rubiku kuubik lahendada ühest grupist teise, kuni viimase grupini, mis sisaldab vaid ühte asendit, milleks on lahendatud Rubiku kuubik. Thistlethwaite'i algoritmile eelnevad algoritmid kõigepealt lahendasid mingi osa Rubiku kuubikust ära ning siis edasises lahenduses kasutasid valemeid, mis olemasolevat osa ära ei lõhuks, aga Thistlethwaite'i algoritmi puhul vaheetapi lõpetamine tähendab et mingit kindlat veerandpööret ei ole enam vaja teha. (Scherphuis 2014b)

Esimeses grupis on täiesti sassis asendid. Seal on ligikaudu 4.33E19 erinevat positsiooni ning sellest grupist järgmisesse jõudmiseks kulub maksimaalselt 7 pööret. Teises grupis on sellised asendid, mille lahendamiseks on vajalik ülemise ja alumise tahuga vaid poolpöördeid teha, selles grupis on 2.11E16 erinevat positsiooni ja sellest järgmisesse gruppi jõudmiseks kulub maksimaalselt 13 pööret. Kolmandas grupis on sellised asendid, mille lahendamiseks on vajalik lisaks ülemisele ja alumisele tahule ka eesmise ja tagumise tahuga vaid poolpöördeid teha, selles grupis on ligikaudu 1.95E10 asendit ja sellest järgmisesse gruppi jõudmiseks kulub maksimaalselt 15 pööret. Neljandas grupis on sellised asendid, mille lahendamiseks on kõigi tahkudega ainult poolpöördeid vaja teha, selles grupis on ligikaudu 6.63E5 erinevat asendit ja sellest järgmisesse gruppi jõudmiseks kulub algse algoritmiga maksimaalselt 17 pööret. See teeb kokku maksimaalselt 52 pööret kogu Rubiku kuubiku lahendamiseks. (Scherphuis 2014b)

#### 1.2.2. Kociemba algoritm

Kociemba algoritmi töötas välja Saksa Rubiku kuubiku lahendaja Herbert Kociemba aastal 1992. Kociemba oli ka üks neljast inimesest, kes aastal 2010 tõestasid, et God's Number on 20. (Speedsolving Wiki 2012 s.v. Herbert Kociemba) Kociemba algoritm on Thistlethwaite'i algoritmi edasiarendus, millel on vähem samme. Kociemba sai põhiliselt oma algoritmi Thistlethwaite'i omast paremaks arvutite arvutusvõimsuse arengu tõttu. (Speedsolving Wiki 2017 s.v. Kociemba's Algorithm) Kociemba algoritm viis God's Numberi ülempiiri 37 pöörde pealt 28 pöördeni. (Scherphuis 2014a)

Sarnaselt Thistlethwaite'i algoritmiga jagab Kociemba algoritm asendid gruppidesse, aga tema algoritmis on vaid 3 gruppi asendeid. Ühest grupist teise jõudmiseks kasutab Kociemba

algoritm IDA\* otsimisalgoritmi. IDA\* iteratiivselt sügavnev süvitiotsing, mis arvutab erinevatele teedele kaalud välja, et valed teed kiiremini välistada. Kociemba algoritmi nimetatakse ka Kociemba kahe faasi algoritmiks, kuna lahenduse saab jagada kaheks faasiks: esimesest grupist teise jõudmine ja teisest grupist kolmandasse jõudmine.(Speedsolving Wiki 2017 s.v. Kociemba's Algorithm)

Esimene grupp sisaldab sassis asendeid. Teine grupp sisaldab asendeid, mille lahendamiseks on eesmise, tagumise, parema ja vasaku tahuga ainult poolpöördeid teha (ülemise ja alumise küljega on vaja veerandpöördeid teha). Kolmas grupp on lahendatud asendi grupp. Vähima pöörete arvuga lahenduse jaoks leitakse kõigepealt vähima pöörete arvuga esimese faasi lahendus, ütleme et esimese faasi lahendamiseks kulus n pööret. Seejärel leitakse saadud asendist edasi vähima pöörete arvuga teise faasi lahendus, ütleme et selleks kulus m pööret. Kokku kulus n+m käiku. Järgmisena hakatakse vaatama suurema pöörete arvuga esimese faasi lahendusi, kuna on võimalik, et mõne teise esimese faasi lahendusega saab teist faasi veel vähema pöörete arvuga lahendada. Järgmisi teise faasi lahendusi on kergem otsida, kuna kui esimene faas lahendati n+d pöördega, siis tuleb vaadata ainult neid teise faasi lahendusi, mis ei ületa m-d pööret, sest muidu tuleks lahendus algselt leitud lahendusest pikem. (Speedsolving Wiki 2017 s.v. Kociemba's Algorithm)

# 1.2.3. Korfi algoritm

### 1.3. Realisatsioon ja tulemuste analüüs

- 2. Seadme arhitektuur ja disain
- 2.1. Mehaanika
- 2.2. Elektroonika
- 2.3. Tarkvara

# Kokkuvõte

# Kasutatud kirjandus

Grol, van G. (2010) The Quest For God's Number. Loetud: <a href="http://digitaleditions.walsworthprintgroup.com/display-article.php?id=532775">http://digitaleditions.walsworthprintgroup.com/display-article.php?id=532775</a>, 20.01.2018

Rokicki, T., Kociemba, H., Davidson, M., Dethridge, J. (2014) God's Number is 20. Loetud: <a href="http://www.cube20.org/">http://www.cube20.org/</a>, 20.01.2018

Scherphuis, J. (2014) Computer Puzzling. Loetud: <a href="https://www.jaapsch.net/puzzles/compcube.htm">https://www.jaapsch.net/puzzles/compcube.htm</a>, 20.01.2018

Scherphuis, J. (2014) Thistlethwaite's 52-move algorithm. Loetud: <a href="https://www.jaapsch.net/puzzles/thistle.htm">https://www.jaapsch.net/puzzles/thistle.htm</a>, 20.01.2018

Speedsolving Wiki (2012) s.v. Herbert Kociemba. Loetud: <a href="https://www.speedsolving.com/wiki/index.php/Herbert Kociemba">https://www.speedsolving.com/wiki/index.php/Herbert Kociemba</a>, 20.01.2018

Speedsolving Wiki (2017) s.v. Kociemba's Algorithm. Loetud: <a href="https://www.speedsolving.com/wiki/index.php/Kociemba%27s\_Algorithm">https://www.speedsolving.com/wiki/index.php/Kociemba%27s\_Algorithm</a>, 21.01.2018