

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Alfred Saidlo

Paisktabeliprintsiibi õpetamine ülikoolikursustel
Bakalaureusetöö (9 EAP)

Juhendaja: Ahti Peder, PhD

Tartu 2019

Paisktabeliprintsiibi õpetamine ülikoolikursustel

Lühikokkuvõte:

Paisktabel on andmestruktuur, mida õpetatakse Tartu Ülikooli teise aasta informaatikastudentidele. Käesolev töö uurib, kuidas õpetatakse sama teemat teistes Eesti ja välismaa ülikoolides. Uurimuse käigus vaadatakse läbi mitmete ülikoolide loengumaterjalid ja esitatud ülesanded. Töö eesmärgiks on teada saada kas teised ülikoolid õpetavad antud teemat teistmoodi kui Tartu Ülikoolis ning kuidas nende lähenemine erinev on. Selle informatsiooni põhjal on võimalik parendada paisksalvestuse teema õpetamist Tartu Ülikoolis. Töö eesmärk saavutatakse kogutud materjali analüüsis.

Võtmesõnad:

Paisktabel, paisksalvestus, algoritm, andmestruktuur

CERCS: P175 Informaatika, süsteemiteooria

Teaching Hash Tables in University Courses

Abstract

Hash tables are one of the data structures that second year Computer Science students have to learn at the University of Tartu. This paper examines how other universities in Estonia as well as other universities around the world approach teaching the same subject. The paper details the information and exercises presented in different university courses. The aim of the paper is to see whether teaching hash tables is accomplished differently in other universities other than the University of Tartu as well finding out where the differences lie. The information obtained can be used to improve the teaching of hash tables at the University of Tartu. The purpose is achieved by analysing the materials collected during the research process.

Keywords

Hash table, hashing, data structure, algorithm

CERCS: P175 Informatics, systems theory

Sisukord

Sisukord.....	3
1. Sissejuhatus.....	5
1.1 Töö eesmärk ja hüpotees	5
1.2 Töö struktuur	6
2. Taustinformatsioon	7
2.1 Paisktabel ja seotud mõisted	7
2.2 Paisktabeli kasulikkus	8
2.3 Ülikoolide valim.....	10
2.4 Ülesannete tüübid.....	10
2.5 Töö metoodika	11
3. Paisktabeliprintsiibi õpetamine Tartu Ülikoolis	13
3.1 Loengutes käsitletavat teemad	13
3.2 Ülesanded	13
4. Paisktabeliprintsiibi õpetamine Eesti ülikoolides	16
4.1 Tallinna Ülikool.....	16
4.2 Tallinna Tehnikaülikool (TalTech)	17
4.3 Võrdlus Tartu Ülikooliga	18
5. Paisktabeli printsiibi õpetamine Tartu Ülikooliga sarnastes maailma ülikoolides	20
5.1 Houstoni Ülikool.....	20
5.2 Florida Riiklik Ülikool	21
5.3 Browni Ülikool.....	22
5.4 Chalmersi Tehnoloogiaülikool	23
5.5 Bergeni Ülikool	25
5.6 Helsingi Ülikool	26
5.7 Washingtoni Ülikool	28
6. Paisktabeliprintsiibi õpetamine tippülikoolides.....	31
6.1 Harvardi Ülikool	31
6.2 Massachusettsi Tehnoloogiainstituut	32
6.3 California Ülikool Berkeleys	33
6.4 Princetoni Ülikool.....	35
6.5 Võrdlus Tartu Ülikooliga	37
7. Analüüs	39

8. Kokkuvõte	41
Viidatud kirjandus.....	42
Lisad	47
Litsents	47

1. Sissejuhatus

Iga elukutseline programmeerija peaks tundma põhilisi algoritme ja struktuure andmetega manipuleerimiseks ning probleemide lahendamiseks [1]. Et seda eesmärki täita, peavad kõik Tartu Ülikooli informaatikatudengid omandama aine „Algoritmid ja andmestruktuurid“. Üks andmestruktuur, mis nii selles aines kui ka paljude teiste ülikoolide õppekavades esineb, on paisktabel.

Paisktabel on andmestruktuur, mis põhineb massiivil. Erinevalt tavalisest massiivist, kus konkreetne väärtus võib paikneda massiivi suvalisel positsioonil, on paisktabelis iga väärtuse asukoht määratud paiskfunktsiooni abil. Selline lähenemine annab paisktabelile mõned väga huvitavad omadused. Näiteks saab paisktabelist enamasti andmeid otsida konstantse ajaga, käimata läbi tervet andmestruktuuri.

Paisktabeli huvitavad omadused võimaldavad sellega lahendada palju probleeme efektiivsemalt kui naiivsete toore jõu (ingl k. *brute force*) algoritmidega. Samas on Tartu Ülikooli tudengite näitel näha, et paisktabeli realiseerimise ja kasutamise õpetamine on väga keeruline. Isegi kui tudeng teab, mis on paisktabel, ei pruugi talle selge olla, milliste probleemide lahendamiseks seda kasutada võiks. See informatsioon pärineb autori enda kogemusest antud teema õpetamisel.

1.1 Töö eesmärk ja hüpotees

Käesoleva lõputöö eesmärk on uurida, kuidas õpetatakse paisktabelit ja selle kasutamist teistes ülikoolides üle maailma. Täpsemalt pakub huvi see, milliseid näiteid tuuakse, milliseid materjale kasutatakse ja millised ülesanded tudengid lahendada peavad. Seda kõike soovime võrrelda selle materjaliga, mida kasutatakse Tartu Ülikooli aines „Algoritmid ja andmestruktuurid“, et paisktabelit ja selle kasutamist tudengitele õpetada. Saadud info pealt on loodetavasti võimalik parandada paisktabeli õpetamist Tartu Ülikoolis.

Töö kirjutamise alguses püstitas autor kaks hüpoteesi. Esimeseks hüpoteesiks on, et teistes Eesti ülikoolides ja ka välismaa ülikoolides õpetatakse paisksalvestuse teemat Tartu Ülikoolist erinevalt. Teiseks hüpoteesiks on see, et maailma tippülikoolides keskendutakse pigem

paisktabeli teooriale ning eeldatakse, et tudeng, kes teooriat oskab, suudab paisktabelit vajadusel ka probleemi lahendamiseks rakendada.

1.2 Töö struktuur

Antud töö koosneb viiest suuremast osast. Peale sissejuhatust järgneb lugejale tööst arusaamiseks vajalik taustinformatsioon. Kirjeldatakse paisktabel ja sellega seotud mõisted, selgitatakse ülikoolide kategoriseerimist ning töö metoodikat. Töö järgmises osas kirjeldatakse, kuidas õpetatakse paisksalvestuse teemat Tartu Ülikoolis. Pärast seda antakse kolmes osas ülevaade Eesti ja erineva taseme välismaa ülikoolide lähenemisest paisktabeli õpetamisele. Leitud informatsiooni võrreldakse Tartu Ülikoolis leiduva materjaliga. Töö lõpus analüüsitakse saadud tulemusi.

2. Taustinformatsioon

Selle peatüki eesmärk on anda lugejale taustinformatsioon, millel põhineb ülejäänud töö. Kõigepealt kirjeldatakse paisktabel ja sellega seotud mõisted. Seejärel tuuakse näide paisktabeli kasulikkusest programmeerimisülesannete lahendamisel. Peatüki lõpus kirjeldatakse käesoleva töö metoodika ehk kuidas uuritakse leitud ülikoolide kursusi.

2.1 Paisktabel ja seotud mõisted

Paisktabeli mõiste

Paisktabeli definitsiooni võib mitmeti sõnastada. Käesolevas töös lähtub autor üldiselt definitsioonist, mis on sõnastatud Jüri Kiho õpikus „Algoritmid ja andmestruktuurid“. Sellel õpikul [2] põhineb ka järgnev lõik.

Paisktabeliks nimetatakse m -realist massiivi, kus igal real on indeks. *Paiskfunktsiooniks* nimetame funktsiooni h , mis igale väärtusele võtmete hulgast U seab vastavusse mingi paisktabeli rea indeks. Seega kui paisktabelis on m rida, siis võib paiskfunktsiooni käsitleda kui funktsiooni

$$h: U \rightarrow \{p_0, p_1, \dots, p_{m-1}\}.$$

Sisuliselt tähendab see, et kui meil on soov salvestada mingi kirje, millel esineb võti e hulgast U , siis vastavad andmed paigutame nad paisktabelisse ritta indeksiga $h(e)$.

Eeldusel, et paiskfunktsioon on arvutatav konstantse ajaga, on paisktabelisse elementide lisamine ja eemaldamine ning tabelist elementide otsimine üldiselt teostatav konstantse ajaga. Elemendi e puhul on vaja vaid tegeleda paisktabeli reaga, mille indeks on $h(e)$.

Põrked

Paisktabelit kasutatakse enamasti juhtudel, kus kõikvõimalike väärtuste hulk U on palju suurem kui tegelikult salvestatavate väärtuste hulk [3]. Kuna mälu ei ole mõtet raisata, siis paisktabeli suuruse valime me üldiselt selle järgi, kui palju kirjeid me tahame salvestada. See aga tekitab ühe vältimatu probleemi. Kui $|U| > m$, kus m on paisktabeli suurus, siis pole võimalik, et paiskfunktsioon on injektiivne. Sellist olukorda, kus paiskfunktsioon seab kahele kirjele vastavaks sama paisktabeli rea indeks, nimetame *põrkeks* [3].

Kuna põrked on vältimatud, siis on vaja viisi, kuidas andmeid paisktabelisse salvestada ja neid sealt otsida, isegi siis kui kahe kirje võtmed annavad paiskfunktsiooni puhul sama väärtuse. Sellel probleemil leidub palju lahendusi. Kaks levinumat on lahtine adresseerimine ja välisahelate meetod [2].

Paiskfunktsioonid

Paikstabeli definitsioonis ei ole võtmete hulgale antud piiranguid. Seega ei pruugi erinevate andmete võtmed olla sama tüüpi. Näiteks võib inimese info salvestamisel paisktabelisse olla võtmeks tema eesnimi sõnena. Samas võiksime kasutada ka tema vanust täisarvuna. Et paisata erinevat tüüpi võtmeid, on vaja erinevaid paiskfunktsioone.

Üks lihtsamaid viise täisarvulisi võtmeid paisata on leida võtme jääk jagamisel paisktabeli pikkusega [3]. Sõnede paiskamiseks võime tõlgendada neid numbritena. Näiteks programmeerimiskeeles Java on kõik sümbolid kujutatud 16-bitiste täisarvudena. Sõne puhul võime selles sisalduvate sümbolite numbrilisi väärtusi kombineerida, et saada kätte indeks [4].

Paikstabeli põhioperatsioonide efektiivsus sõltub sellest, millist paiskfunktsiooni konkreetses olukorras kasutatakse. Erinevad paiskfunktsioonid põhjustavad erineva arvu põrkeid [5]. Suur põrgete arv aga suurendab paisktabeliga teostatavate operatsioonide ajakulu.

2.2 Paikstabeli kasulikkus

Olgu meil lahendada järgnev ülesanne.

Näidisülesanne 1

Olgu meile antud n -elemendilised täisarvude massiivid a ja b .
Väljastage ekraanile kõik massiivi b elemendid, mis ei sisaldu massiivis a .

Naiivne lähenemine antud probleemi lahendamiseks oleks iga b elemendi korral vaadata, kas see sisaldub massiivis a .

```
for arv in b:
    kontroll = True
    for x in a:
        if arv == x:
            kontroll = False
            break
```

```
if kontroll:
    print(arv)
```

Ülaloodud koodinäide on üks võimalik lahendus programmeerimiskeeles Python. Kontrollimaks, kas mingi arv x sisaldub massiivis a , on kaks võimalust. Kui see element massiivis a ei sisaldu, siis ei saa me seda teada enne kui oleme x -i kõigi a elementidega võrrelnud. Kui x sisaldub massiivis a , siis keskmisel juhul peame läbi vaatama vähemalt pooled a elemendid enne kui selles veenduda saame. Isegi kui me eeldame, et kõik b elemendid sisalduvad massiivis a , siis keskmiselt peame sooritama ikka $n \times \frac{n}{2}$ võrdusoperatsiooni. Järelikult on selline lahendusalgorithm ruutkeerukusega.

Samas saame antud ülesande lahendada palju efektiivsemalt paisktabeli abil. Lahendus koosneb kahest sammust. Esiteks lisame kõik a elemendid paisktabelisse. Seejärel käime läbi massiivi b elemendid ning kontrollime, kas see b element asub paisktabelis. Kui elementi paisktabelis pole, siis väljastame ta ekraanile. All on toodud näidislahendus programmeerimiskeeles Python. Näites on eeldatud, et pörkeid ei toimu ja et h on mingi paiskfunktsioon, mis on arvutatav konstantse ajaga.

```
tabel = [0]*(2*n)
for arv in a:
    tabel[h(arv)] = arv
for arv in b:
    if tabel[h(arv)] != arv:
        print(arv)
```

Ülaloodud algoritm koosneb kahest tsüklist, kusjuures mõlema tsükli iga samm on täidetav konstantse ajaga. Järelikult on mõlema tsükli keerukus lineaarne, sest tehakse n konstantse ajaga täidetavat sammu. Seega on terve algoritmi keerukus $\theta(n + n) = \theta(2n) = \theta(n)$. See aga tähendab, et suurte andmemahtude puhul on paisktabeli abil realiseeritud algoritm selle ülesande jaoks täidetav palju väiksema ajaga.

Kirjeldatud probleem ja selle lahendus on pealtnäha väga lihtsad. Kahjuks näitab autori kogemus antud teema õpetamisel, et tudengid ei oska tihti selliste probleemide lahendamiseks paisktabelit kasutada. Seega oleks huvitav teada, kas teised ülikoolid peale Tartu ülikooli ka selliseid ülesandeid käsitlevad.

2.3 Ülikoolide valim

Töö käigus uuriti ülikoole, mille puhul oli minimaalselt ligipääs loengumaterjalidele ja tudengitele antavatele ülesannetele. Lisaks uuriti nende olemasolul ka eksami- ja kontrolltööde näidiseid. Üldiselt lähtuti põhimõttest, et aine kodulehel olevad materjalid on ette nähtud tudengile aine läbimiseks.

Leitud ülikoolid jaotati seejärel kolme kategooriasse. Esiteks vaadeldakse töös Eesti suurimaid ülikoole. Seejärel käsitletakse välismaale ülikoole, mis on ülikoolide edetablis sarnase asetusega kui Tartu Ülikool. Kolmandaks uuritakse maailma tippülikoole.

Ülikoolide järjestuse aluseks võeti *Times Higher Education* lehekülg, mille edetabelis on üle tuhande ülikooli [6]. Tippülikoolideks loeti ülikoolid, mis jäid edetabelis esimese 20 hulka.

Ülikoolide kategoriseerimise eesmärk on veenduda, kas erineva taseme ülikoolid käsitlevad paisktabeli temaatikat teistmoodi.

2.4 Ülesannete tüübid

Et mugavalt võrrelda erinevates ülikoolides esitatud ülesandeid, on autor koos juhendajaga loonud järgmised ülesannete kategooriad:

- kirjalikud ülesanded;
- programmeerimisülesanded;
- probleemülesanded.

Kirjalikud ülesanded

Kirjalikud ülesanded on ülesanded, kus tudeng peab arvutit kasutamata mingile küsimusele vastuse leidma. Kirjalikud ülesanded võime selle töö raames jagada kahte kategooriasse:

1. lihtsad teooriaküsimused;
2. keerulised teooriaküsimused.

Esimesse rühma kuuluvad ülesanded nagu mõistete definitsioonid. Nendes ülesannetes ei ole tudengil vaja arutleda, vaid esitada selgeksõpitud vastus. Näiteks kuulub siia kategooriasse ka ülesanded, kus tudeng peab paisktabeli tööd paberil läbi tegema, kuna see on sisuliselt mõiste tundmise ülesanne.

Teise rühma kuuluvad ülesanded, mis puudutavad antud teema teooriat, kuid millele vastamiseks peab tudeng arutlema. Näiteks sobib siia kategooriasse ülesanne, kus nõutakse tudengil selgitada, miks ei sobi paisktabelile pikkusega 100 kompesammuks arv 20.

Programmeerimisülesanded

Programmeerimisülesanded seisnevad mingi nõutud funktsiooni või programmi koostamises. Need võima omakorda jagada kahte kategooriasse.

1. realiseerimisülesanded (ingl k *implementation*);
2. empiirilised ülesanded.

Esimese kategooria ülesanded seisnevad mingi konkreetse algoritmi tõlkimises programmeerimiskeelde. Kui ülesandeks on paisktabeli realiseerimine, siis see kuulub esimesse kategooriasse.

Teise rühma ülesanded seisnevad mingi algoritmi tööaja mõõtmises või tehtud sammude kokku lugemises. Paisktabeli teemas võib selliseks ülesandeks olla programmi koostamine, mis loendab paisktabelis toimunud pörgete arvu.

Probleemülesanded

Probleemülesanded seisnevad mingi probleemi lahendusalgoritmi disainimises. Hea näide on raamatust „Algoritmid ja andmestruktuurid. Ülesannete kogu“ ülesanne 8.49, kus on vaja sõnastada algoritm, mis suudab leida suvalisest lihtahelast seal leiduvate unikaalsete kirjete arvu [7]. Teiseks näiteks oleks siin töös esitatud näidisülesanne 1.

Sisuliselt võib probleemülesanne olla nii kirjalik kui ka programmeerimisülesanne.

2.5 Töö metoodika

Selle töö raames uuritakse erinevate ülikoolide lähenemist tudengitele paisktabeli õpetamisel. Info vastavate ainete kohta leitakse nende avalikelt kodulehtedelt. Leitud loengumaterjalidest antakse temaatilised ülevaated. Samuti tuuakse töös välja uuritavate ülikoolide põhiõpikud.

Viimase etapina uuritakse igas ülikoolis esitatud paisktabeli teemaga seotud ülesandeid. Leitud ülesanded jaotatakse kategooriatesse ning iga ülikooli kohta tuuakse näiteid erinevatest ülesannete tüüpidest. Autor sooviks juhtida tähelepanu sellele, et esitatud näiteülesanded ei ole alati täpselt sama sõnastusega kui allikas, kuna kõikide ebaoluliste detailide töösse mahutamine kulutaks mõttetult ruumi.

Pärast erinevate ülikoolide ainete kirjeldamist võrreldakse neid Tartu Ülikooli ainega „Algoritmid ja andmestruktuurid“. Seejärel analüüsitakse saadud tulemusi.

3. Paisktabeliprintsiibi õpetamine Tartu Ülikoolis

Antud peatükis anname ülevaate sellest, kuidas käsitletakse paisktabelitega seotud teemasid Tartu Ülikoolis (edaspidi TÜ). TÜ jagab maailma ülikoolide edetabelis 301.-350. kohta [6]. TÜ Informaatika bakalaureuseõppekava tudengid peavad läbima aine „Algoritmid ja andmestruktuurid“, mille üheks teemaks on paisksalvestus [8].

Aine põhineb nii eesti- kui ingliskeelsetel õppematerjalidel. Ainsaks kohustuslikuks õppematerjaliks on A. Pederi, J. Kiho ja H. Nestra raamat „Algoritmid ja andmestruktuurid. Ülesannete kogu“ [9]. Soovituslike õppematerjalidena on välja pakutud J. Kiho „Algoritmid ja andmestruktuurid“, K. Mehlhorni ja P. Sandersi „Algorithms and Data Structures“, T. Cormeni, C. Leisersoni, R. Rivesti ja C. Steini „Introduction to Algorithms“ ning A. Pederi, H. Heina ja L. Kimmeli „Objektorienteeritud programmeerimine“.

3.1 Loengutes käsitletavat teemad

TÜs käsitletakse paisksalvestuse loengutes järgnevaid teemasid [10]:

- konstantse ajaga otsing, kiiret otsingut vajavad ülesanded;
- paisktabeli mõiste, paiskfunktsioon;
- paiskamine jäägi meetodil, paiskamine korrutamise meetodil;
- lahtise adresseerimisega paisktabel, lineaarne kompimine, ruutkompimine, topelpaiskamine, kompesammude võrdlus;
- kimpudega paisktabel, kimbumeetodil järjendi sorteerimine, leksikograafiline sorteerimine.

Loengumaterjalides ei ole konkreetseid koodinäiteid. Loengu slaididel on välja pakutud lühiülesandeid kas loengus või iseseisvalt lahendamiseks.

3.2 Ülesanded

Enamus ülesanded selles aines pärinevad eespool mainitud ülesannete kogust [7]. Sellest ülesannete kogust pärinevad ka selles alapeatükis esitatud informatsioon ja näited. Paisktabeli teemal on ülesannete kogus 52 ning kuna tegemist on kohustusliku õppematerjaliga, siis nendest ülesannetest arusaamine on aine läbimiseks vajalik.

Ehk kõige lihtsam ülesanne on esitatud loengumaterjalides. See seisneb paisktabeli töö käsitsi läbitemises.

Näidisülesanne 2

Olgu $h = k \bmod 7$. Vaatame võtmeid 26, 53, 12, 65, 39, 6. Leiame neile koha paisktabelis.

Veidi keerulisem teooriaülesanne on ülesannete kogust ülesanne 8.3.

Näidisülesanne 3

Miks lahtise adresseerimisega lineaarse kompimise meetodis peab kompesamm olema tabeli pikkusega ühistegurita?

Samuti leidub ülesannete kogus programmeerimisülesandeid. Mõnikord valitakse ka ülesannete kogust mõni programmeerimisülesanne kodutööks. Näiteks ülesannete kogus ülesanne 8.26.

Näidisülesanne 4

Programmeerida järgmised funktsioonid paisktabeli koostamiseks:

- a) lahtine adresseerimine lineaarse kompimisega;
- b) kimbumeetod (Kimbud võivad olla realiseeritud järjendina, mille uued elemendid lisatakse lõppu).

Tulemuseks tagastada loodud järjend. Paiskfunktsiooniks võite võtta korrumtamismeetodil põhineva paiskfunktsiooni.

Ülesandeks on võrrelda elemendi otsimise efektiivsust kummalgi meetodil realiseeritud paisktabelis. Selleks:

- fikseerida tabeli suurus;
- genereerida täisarvujärjendid pikkustega 10%, 20%, ..., 90%, 99% tabeli pikkusest, mille elementideks on juhuarvud;
- paigutada järjendite elemendid paisktabelisse mõlemal meetodil;
- genereerida teatud arv juhutäisarve samades piirides;
- iga täisarvu puhul loendada, mitu võrdlemist tehakse selle arvu otsimisel paisktabelist;
- tulemused kujutage graafikuna.

Ülaloodud ülesanne hõlmab endas nii paisktabeli realiseerimist kui ka selle efektiivsuse empiirilist mõõtmist. Tihti kasutatakse samasugust ülesannet ka kodutööna. Teine levinud kodutöö ülesanne on erinevate paiskfunktsioonide efektiivsuse võrdlemine.

Probleemilahendusülesandeid on ülesannete kogus mitu. Üks näide, mida töö tausta osas mainisime, oli ülesannete kogu ülesanne 8.49.

Üldiselt võime täheldada, et kõik erinevat tüüpi ülesanded on TÜ aines „Algoritmid ja andmestruktuurid“ esindatud. Tartu Ülikoolis käsitletakse nii paisktabeli mõistet ja selle teooriat kui ka paisktabeli rakendusi.

4. Paisktabeliprintsiibi õpetamine Eesti ülikoolides

Järgnevas peatükis uurime kuidas õpetatakse paisktabelit ja selle kasutamist teistes Eesti ülikoolides. Vaatluse alla on võetud kaks suuremat ülikooli: Tallinna Ülikool (edaspidi TLÜ) ja Tallinna Tehnikaülikool (edaspidi TalTech). Töö autor uuris ka IT kolledži andmestruktuuride aineid, kuid avastas, et nende andmestruktuuride ainekst on paisktabeli teema välja jäetud [11]. Lisame, et TalTechi asetus maailma ülikoolide edetabelis on 601. – 800. koht [6]. Samal leheküljel aga TLÜ asetuse kohta infot ei leidu.

4.1 Tallinna Ülikool

Tallinna Ülikoolis õpetatakse ainet „Algoritmid ja andmestruktuurid“ ning aine raames käsitletakse ka paisktabelite teemat [12].

Järgnevas lõigus olev info pärineb TLÜ õppeinfosüsteemist [12]. Aine kohustuslikeks taustmaterjalideks on V. Leppiksoni „Programmeerimine C Keeles“ ja J. Kiho „Algoritmid ja andmestruktuurid“. Loengute asenduskirjandusena on soovitatud lugeda R. Sedgewicki raamatut „Algorithms in C“ ja A. Isotamme teost „Programmeerimine C keeles“. Kohustuslikest õppematerjalidest on ilmselt olulisem Jüri Kiho õpik, ning Leppiksoni õpik on pakutud tudengitele, kes vajavad C keeles juurde teadmisi.

TLÜ loengutes käsitletakse järgnevaid teemasid [13][14]:

- paisktabeli mõiste, pörked;
- paiskamine jäägimeetodil, paiskamine korrutamise meetodil;
- lõplik paisksalvestus, universaalne paisksalvestus;
- pörgete lahendamine välisahelatega, pörgete lahendamine lahtise adresseerimisega;
- lineaarne kompimine, ruutkompimine, topelt paisksalvestus;
- paisktabeli kiiruse võrdlus kahendotsimispuuga.

Enamus teemasid on illustreeritud konkreetsete näidetega, kuid lõplik paisksalvestus ja universaalne paisksalvestus on mainitud väga lühidalt [13]. Need esinevad loengumaterjalides, kuid on slaididelt välja jäetud [14].

Alltoodud näited pärinevad Inga Petuhhovi kodulehe ülesannete alajaotuse juurest [15]. Paisktabeli teemal on tudengitel kohustuslik lahendada ka mõned ülesanded. Algul nõutakse paisktabeli andmestruktuuri käsitsi läbimängimist.

Näidisülesanne 5

Olgu meil antud võtmed: 10 22 31 4 15 28 17 88 59
Loo tabeli pikkusega 11. Loetletud võtmete jaoks leidke
1. paiskväärtused jäägimeetodil;
2. paiskväärtused korrutamise meetodil.
Lisaks tehke paisktabelid, kus kasutate kollisioonide lahendamist
3. ahelatega;
4. avatud adresseerimisega:
 4.1 lineaarne kompimine;
 4.2 ruutkompimine;
 4.3 topeltpaiskamine.

Pärast kirjaliku ülesande teostamist peavad tudengid lahendama ka programmeerimisülesande.

Näidisülesanne 6

Koosta programm, mis realiseerib eelmises ülesandes paberile kirjutatud paisktabelid.

Rohkem ülesandeid kursuse kodulehel tudengitele välja pakutud ei olnud [15]. Eksami- materjalides on toodud välja paisktabeliga seotud mõisted, mida tudeng peab tundma. On selgitatud, et tudeng peab oskama erinevaid paisktabeleid koostada ja sealt kirjeid otsida [16].

4.2 Tallinna Tehnikaülikool (TalTech)

Järgnev info pärineb Tallinna Tehnikaülikooli õppeinfosüsteemist [17]. TalTechis leidub aine „Algoritmid ja andmestruktuurid“. Aine materjalide all on ära toodud väga palju erinevaid õpikuid: M. Weissi „Data Structures and Algorithm Analysis in C++“, A. Drozdeki „Data Structures and algorithms in C++“ ning R. Sedgewicki „Algorithms in C. Fundamentals, data structures sorting, searching“. On ka ära mainitud lisalugemiseks õpikud Thomas Cormenilt ja Jüri Kiholt.

TalTechis käsitletakse paisksalvestuse loengutes järgmisi teemasid [18]:

- paisktabeli mõiste, põrked;
- paiskamine jäägimeetodil, paiskamine voltimise meetodil;

- välisahelatega paisktabel, lahtise adresseerimisega paisktabel, põrgete lahendamine keldrimeetodil;
- lineaarne kompimine, topeltpaiskamine;
- laiendatav paisksalvestus.

Üldiselt koosnevad loengumaterjalid vaid konkreetsete paisktabelite skeemidest. Teksti juures on vähe. Tudengid peavad seega loengus ise konspekteerima või õpikust juurde lugema.

Aine kodulehel ei ole esitatud ühtegi harjutusülesannet paisktabeli teemal [19]. Lisaks peavad TalTechi tudengid osalema aine läbimiseks laborites. Üllataval kombel ei ole üheski laboris ülesannet paisktabeli kasutamise kohta [20].

4.3 Võrdlus Tartu Ülikooliga

TLÜ, TÜ ja TalTechi taustmaterjalid on osaliselt kattuvad. J. Kiho õpik esineb nendel kõigil kirjanduse loetelus. Siiski on õpikud üldiselt koolide vahel erinevad.

Loengutes käsitletavat teemad on suures osas kattuvad. Tartu Ülikoolis jõutakse peale paisktabeli mõiste rääkida ka selle kasulikkusest sorteerimisel. On ka teemasid, mis TÜ loengulaididel ei esine. Näiteks keldrimeetodil põrgete lahendamist tutvustatakse vaid TalTechis. Teema mis esineb ainult TLÜ loengutes on universaalne paisksalvestus. Lisaks käsitlevad nii TalTech kui ka TLÜ topeltpaiskamist.

Nagu näha on mõned teemad teistes Eesti ülikoolides, mida TÜs ei käsitleta. Samas ei ole need teemad kuskil väljaspool loenguid materjalides nähtaval. Seega ei paista nad väga olulised olevat.

Kõige suurem erinevus Eesti ülikoolide vahel on aga ülesannete osas. Tartu Ülikoolil on kasutusel terve ülesannete kogumik, kus leidub palju materjali tudengitele harjutamiseks. TLÜ-s on aga aine materjalide all välja toodud paar väikest ülesannet, mis on sealjuures väga lihtsat tüüpi ülesanded. TalTech ei ole oma aine materjalide hulgas ühtegi harjutus- ega kontrollülesannet esitanud, mis sisaldaks paisktabeli kasutamist.

Lähtudes ülaltoodust võime väita, et paisktabeli temaatika on Tartu Ülikoolis käsitletud palju põhjalikumalt kui teistes Eesti ülikoolides. Osalt võib selle põhjuseks teemale antud aeg.

Näiteks TÜs on paisktabeli teema kohta kaks loengut ja praktikume samuti kaks. Samas TLÜ käsitleb sama teema ära ühe loengu ja praktikumiga.

5. Paisktabeli printsiibi õpetamine Tartu Ülikooliga sarnastes maailma ülikoolides

Järgmises peatükis kirjeldame paisktabeli temaatika õpetamist maailma ülikoolides, mis on asetusest Tartu Ülikooliga sarnased. Vaatluse all olevad ülikoolid ja nende paigutus rahvusvahelises ülikoolide edetabelis on [6]:

- Houstoni Ülikool (301.-350. koht);
- Florida Riiklik Ülikool (koht 251.-300. koht);
- Browni Ülikool (53. koht);
- Chalmersi Tehnoloogiaülikool (201.-250. koht);
- Bergeni Ülikool (197. koht);
- Helsingi Ülikool (97. koht);
- Washingtoni Ülikool (28. koht).

5.1 Houstoni Ülikool

Selles alapeatükis olev info põhineb uuritava aine kodulehel ning sealt allalaaditavatel loengumaterjalidel ja eksami näidisülesannetel [21].

Houstoni ülikoolis käsitletakse paisktabeleid aines „Data Structures“. Põhiliseks taustmaterjaliks on M. Goodrichi „Data Structures and Algorithms in Java“ .

Houstoni ülikooli loengutes õpetatakse tudengitele järgnevat:

- sõnastik kui abstraktne andmetüüp, sõnastik massiiv, mittetäisarvulised võtmed;
- paiskfunktsioon, paisktabel, paiskamine jäägimeetodil;
- pörked, välisahelatega paisktabel, lahtise adresseerimisega paisktabel, lineaarkompimine;
- topeltpaiskamine, paisktabelioperatsioonide keerukus.

Loengus on väga palju aega kulutatud ka paisktabeli konkreetsetele realiseerimisele programmeerimiskeeles Java. Paiskfunktsioonide kirjeldamisel tuuakse välja, et paiskfunktsioon koosneb tavaliselt kahest osast: nn. *hash code* osa, mis loob andmetest „suvalise“ numbriga ja kokkusurumise osa (ingl k *compression*), mis surub saadud väärtuse soovitud väärtusvahemikku.

Kuigi paisktabel on üks kursuse teemadest, ei anta Houstonis selle põhjal tudengitele ühtegi kodust ülesannet. Paisktabelit käsitletakse vaid loengus ja eksamil, kus tudengid peavad oskama vastata lihtsatele küsimustele paisktabeli teemal. All on toodud paar küsimust näidis-eksamilt.

Näidisülesanne 7

1. Millal tekivad paisktabelis pörked?
2. Joonistage 11 reaga paisktabel, mislle saate kirjete 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 paiskamisel, kui paiskfunktsiooniks on $h(i) = 4i+7 \pmod{11}$. Põrgete lahendamiseks kasutage välisahelaid.

5.2 Florida Riiklik Ülikool

Florida Riiklikus Ülikoolis õpetatakse ainet „Data Structures, Algorithms, and Generic Programming“. Aine raames käsitletakse ka paisksalvestuse teemat [22]. Aine põhineb M. Weissi raamatul „Data Structures and Algorithm Analysis in C++“ [23]. Tegemist on sama õpikuga, mida kasutatakse TalTechis.

Selle aine loengute all käsitletakse järgnevaid ideid [24]:

- paisktabeli mõiste, paisktabeli rakendused;
- pörked, paiskamine jäägimeetodil algarvulise ja kordarvulise suurusega paisktabeli puhul, sõnede paiskamine;
- välisahelatega paisktabel, lahtise adresseerimisega paisktabel;
- lineaarkompimine, ruutkompimine, topeltpaiskamine;
- paisktabeli suuruse muutmise.

Sarnaselt Houstoni ülikoolile on loengutes palju koodinäiteid paisktabeli realiseerimisest programmeerimiskeeles Java [25].

Florida Riiklikus Ülikoolis on paisktabeli teema peale üks kodutöö, kus nõutakse paisktabeli realiseerimist ja selle kasutamist [24]. Kuna tegelik sõnastus on väga pikk ja detailne, siis toome siin näitena välja vaid lühikokkuvõtte ülesande püstitusest.

Näidisülesanne 8

1. Realiseerige paisktabeli andmestruktuur. Sisemiselt peab paisktabel olema vektor listidest.
Paisktabeli klass peab omama meetodeid paari {võti, väärtus} lisamiseks, sisalduvuse kontrolliks, tabeli tühjendamiseks ja andmete faili kirjutamiseks või failist lugemiseks. Lisaks peab klass loomulikult omama konstruktorit ja destruktorit.
2. Looge lihtne parooliserveri klass. Klass peab kasutama sisemise andmestruktuurina paisktabelit, kus hoitakse kasutajanimed ja krüpteeritud paroole.
Klass peab omama meetodeid kasutajate otsimiseks, lisamiseks, kustutamiseks ja kasutajate paroolide muutmiseks. Lisaks peab klass võimaldama andmeid tekstifaili salvestada ja sealt lugeda.

5.3 Browni Ülikool

Põhilisi andmestruktuure õpetatakse Browni Ülikoolis tudengitele aines „Introduction to Algorithms and Data Structures“ [26]. Aine põhiõpikuteks S. Dasgupta „Algorithms“ ja T. Roughgardeni „Algorithms Illuminated“, kuid need raamatud on märgitud soovituslikuks lugemiseks ehk nende lugemine pole kohustuslik [27].

Alltoodud loetelu ja sellele järgnev lõik põhinevad aine „Introduction to Algorithms and Data Structures“ loengumaterjalidel [28].

Browni Ülikooli ühes loengus käsitletakse paisktabeli teemat järgnevalt:

- otsingumootorid, hulk kui abstraktne andmestruktuur, sõnastik kui abstraktne andmestruktuur;
- massiivil põhinev sõnastik, paisktabeli mõiste, pörked;
- välisahelatega paisktabel, välisahelatega paisktabeli operatsioonide keerukus;
- universaalne paiskamine;
- universaalset paiskfunktsiooni kasutava paisktabelist otsingu ajalise keerukuse tõestamine;
- paisktabeli rakendus otsingumootoris.

Browni Ülikoolis juhatatakse paisktabeli teema sisse otsingumootorite näitega, mis on teistest ülikoolidest erinev. Veel üks erinev detail on see, et loengus ei mainita lahtist adresseerimist. Loengus peavad tudengid lahendama ka ülesande, kus neil palutakse ennast matriklinumbrite

järgi paisata. Ülesande idee on näidata, et kui andmed on kitsast vahemikust, siis võib kehvasti valitud paiskfunksiooni tekitada palju põrkeid.

Erinevalt Florida ülikoolist on Browni Ülikoolis teoreetilisema suunitlusega ülesanded. Näiteks alltoodud ülesanded nõuavad paisktabeli täituvuse uurimist [29].

Näidisülesanne 9

1. Kirjutage programm, mis loob massiivi B pikkusega n . Seejärel genereerib programm juhuarve vahemikust $0, \dots, n-1$. Nendest numbritest võib mõelda kui paiskfunksiooni väärtustest. Iga genereeritud numbril i korral suurendage väärtust $B[i]$ ühe võrra. Töö lõppedes trükige välja n ja suurim element massiivis B. Korrake seda protsessi erinevatel juhtudel, kus $n = 2, 4, \dots, 2^{15}$.

Pakkuge välja funktsioon, mis kirjeldab $B[i]$ suurimat väärtust sõltuvalt n -ist. Vajadusel tehke andmete kohta graafik. Vastusele matemaatilist tõestust ei nõuta.

2. Tehke sama, mis esimeses ülesandes, kuid genereerige igal sammul kaks arvu i ja j ning suurendage lahtrit $B[i]$ või $B[j]$ sõltuvalt sellest, kummas on väiksem number.

Võrrelge tulemusi esimese ülesandega. Kas B suurim väärtus kasvab nüüd aeglasemini? Miks?

Soovitatakse kindlasti testida suuremate n -i väärtustega, et erinevused välja paistaksid.

Kursusel on välja toodud ka vaheeksami kordamisteemad. Nende seas on ka paisktabel, kuid on rõhutatud, et loengus toodud tõestused ei ole eksamiteema [30]. Samas failis on ka välja pakutud mõned näidisülesanded.

Näidisülesanne 10

Milline on hea paiskfunksioon? Halb paiskfunksioon? Mis asi on paisktabel ja kuidas see töötab?

Näidiseksami ülesannete hulgas aga paisktabeli teema ei kajastu [31].

5.4 Chalmersi Tehnoloogiaülikool

Chalmersi Tehnoloogiaülikoolis käsitletakse põhilisi andmestruktuure aines „Data Structures“ [32]. Ainel endal põhiõpikut ei ole ning on rõhutatud, et aine saab läbida ilma õpikut lugemata.

Soovijatele on lugemiseks välja pakutud R. Sedgewicki raamat „Algorithms, 4th Edition“ [33].

Chalmersi ülikooli loengus tutvustatakse paisksalvestuse teemat järgnevalt [34]:

- hulk, mille operatsioonid on täidetavad logaritmilise ajaga, hulga operatsioonide täitmine konstantse ajaga;
- otsingumootori probleem, paisktabeli mõiste, paiskamine jäägimeetodil;
- põrked, välisahelatega paisktabel, selle operatsioonide ajaline keerukus;
- paisktabeli suuruse muutmine, koormustegur;
- paiskfunktsioonid, hea paiskfunktsiooni defineerimine, sõnade paiskamine paiskfunktsioonid programmeerimisekeeles Java;
- paisktabel lahtise adresseerimise ja lineaarse kompimisega, selle efektiivsus;
- lineaarse kompimise ja välisahelatega paisktabelite võrdlus, ruutkompimine.

Paisktabeli teema juures kasutatakse sama näidet, mis Browni Ülikoolis - kuidas luua otsingumootorit, kui läbi vaadatavate andmete hulk on sadades miljonites [35]. Sarnaselt mõndadele mainitud ülikoolidele on Chalmersi Tehnoloogiaülikooli aine Java keele keskne.

Chalmersi Tehnoloogiaülikoolis ei ole paisktabeli teemal ühtegi kohustuslikku ülesannet [34]. Siiski on soovitatav lahendada kursuse kodulehel toodud harjutusülesandeid, et end eksamiksi ette valmistada [35]. Ülesanded seisnevad paisktabelite teemal lühikeste teooriaküsimustele vastamises [35]. Mõned neist ülesannetest on kirjeldatud all olevas näites.

Näidisülesanne 11

1. Mis on paisktabeli eelised kahendotsimispuude ees?
2. Esitage sobivad invariandid paisktabelite jaoks, mis kasutavad
 - a) välisahelaid;
 - b) lineaarset kompimist.
3. Konstrueerige näide, mis tõestab, et lineaarse kompimisega paisktabelist ei saa lihtsalt kirje eemaldamiseks paisktabeli vastav lahter tühjaks teha.

Väga sarnased ülesanded on esitatud ka eksamil. Näiteks alltoodud eksamiülesandes nõutakse valikvastustega ülesande lahendamist. [36]

Näidisülesanne 12

Olgu meil paiskfunksiooniks $h(x) = x \bmod 10$ ja selle abil täidetud paisktabel.

0	1	2	3	4	5	6	7	8	9
10	XX	2	33		5	16			19

1. Teame, et tabeli 1. real olev kirje on kustutatud. Milline järgnevatest kirjetest võis paikneda tabelis enne selle kirje eemaldamist? Õigeid vastuseid võib olla mitu.

- a) 26
- b) 9
- c) 20
- d) 38
- e) 41
- f) 13

Eksamil on paisktabeli teema kohta ka teistsuguseid ülesandeid. Alltoodud ülesanded näitavad, et kõik teooriaülesanded pole Chalmersi Tehnoloogiaülikoolis päris nii lihtsad kui näidisülesanne 12 [36][37].

Näidisülesanne 13

1. Paisktabelid on üldiselt kiiremad andmestruktuurid kui tasakaalustatud otsingupuud. Siiski kasutatakse mõlemat andmestruktuuri laialdaselt. Üks põhjus on see, et paisktabel ei toeta kõiki operatsioone, mis on tehtavad tasakaalustatud otsingupuul peal.

Tooge näide operatsioonist, mis on efektiivselt realiseeritav kahendotsimispuul, kuid mitte paisktabelil.

2. Kui me tahame paisktabelis salvestada sõnesid, siis üheks võimalikuks paiskfunksiooniks oleks $h(x) = x.length()$. Kas see on hea paiskfunksioon? Põhjendage oma vastust.

5.5 Bergeni Ülikool

Bergeni ülikooli aine „Algoritmar, datastrukturar og programmering“ sisaldab enda paisktabelite teemat [38]. Aine põhiõpikuks on R. Sedgewicki ja K. Wayne'i „Algorithms“ [39].

Bergeni ülikooli loengutes kasutatakse :

- paisktabeli mõiste, paiskfunksioon, pörked;
- ühtlane paiskamine;
- Java meetod hashCode, selle realiseerimine enda klasside jaoks;

- välisahelatega paisktabel, selle realisatsioon Javas, operatsioonid ja nende efektiivsus;
- lahtise adresseerimisega paisktabel, lineaarkompimine, selle Java realisatsioon, operatsioonide efektiivsuse analüüs;
- topeltpaiskamine, *cuckoo hashing*, paisktabeli võrdlus kahendotsimispuudega.

Princetoni ülikooli loengumaterjalid on väga detailsed ja käsitletakse väga palju teemasid. Suur fookus on programmeerimiskeelel Java.

Selles lõigus sisalduv info ja näidisülesanne pärinevad Bergen Ülikooli kodutöö kirjeldusest [40]. Üks Bergen Ülikooli kodutöödest hõlmab ka ülesandeid paisktabeli kohta. Peab lahendada nii teooriaküsimusi kui ka oskama programmeerida. Teooriaküsimused on väga sarnased ülesandele, mis on kirjeldatud näidisülesanne 7.2 juures. Seega neid me siin üle ei korda.

Näidisülesanne 14

Looge klass *PerfectHash*, mis lahendab järgneva ülesande.

Andmete paiskamisel on piisavalt väikese andmete hulga puhul võimalik luua täiuslik paiskfunktsioon (ingl k *perfect hash function*). Selline paiskfunktsioon annab iga võtme korral unikaalse väärtuse. Üks paiskfunktsioon sõnade paiskamiseks oleks järgmine: $h(s) = \sum_{i=0}^l s_i a^{l-i} \bmod M$, kus $l = s.length() - 1$ ja s_i on sõne i -s sümbol.

Teie ülesanne on kirjutada programm, mis leiaks konstantide a ja M väärtused nii, et M oleks minimaalne ja et tegemist oleks täiusliku paiskfunktsiooniga.

Sisendfaili esimesel real on täisarv $2 \leq n \leq 80$ ning järgnevalt n real on unikaalsed sõned. Väljundfaili kirjutada ühele reale leitud mittenegatiivsed täisarvulised a ja M -i väärtused.

5.6 Helsingi Ülikool

Helsingi Ülikoolis õpetatakse tudengitele paisktabeli andmestruktuuri aines „Tietorakenteet ja algoritmit“ [41]. Aines kohustuslikku õpikut pole, kuid lisamaterjali on soovitatud otsida T. Cormeni raamatust „Introduction to Algorithms“ [42].

Paisktabelid on aine loengutes käsitletud. Kirjeldatud teemad on järgmised [41]:

- paisktabel, paiskfunktsioon, põrked;
- välisahelatega paisktabel, selle operatsioonid ja ajalise keerukuse leidmine;
- paiskamine jäägimeetodil, paiskamine korrutamise meetodil;

- universaalne paiskamine, ümberpaiskamine;
- lahtise adresseerimisega paisktabel, lineaarne kompimine, topeltpaiskamine ruutkompimine, erinevate kompesammude efektiivsuse võrdlus;
- paisktabeli realisatsioon Javas.

Paisktabeli teemat tutvustatakse tudengitele näitega telefoniraamatust [41]. Ainukesed operatsioonid, mida telefoniraamat vajab on kirjade lisamine, eemaldamine ja kirje otsamine. See on aga ideaalne koht, kus paisktabelit rakendada, sest kasutatakse paisktabeli põhioperatsioone.

Järgnevad lõigud ja näited selles alapeatükis põhinevad Helsingi Ülikooli paisktabeli teemalise kodutöö materjalidel [43].

Helsingi ülikoolis on paisktabeli teemal uuritud ülikoolidest kõige rohkem kohustuslikke ülesandeid. Koduses töös peavad tudengid kõigepealt vastama kümnele kirjalikule küsimusele. Kaks neist on sõnastatud järgnevalt.

Näidisülesanne 15

1. Olgu meil paiskfunktsioon $f:O \rightarrow \{0, \dots, N-1\}$. Eeldame, et f töötab nii hästi kui võimalik ehk funktsiooni väärtused on jaotatud ühtlaselt.
 - a) Olgu meil suvaline $o \in O$. Valigem $o_1, o_2, \dots, o_n \in O$. Mis on tõenäosus, et vähemalt ühe valitud elemendi paiskväärtus on sama, mis o puhul.
2. Leidke suvalise $n \in \mathbb{N}$ jaoks 2^n sõne pikkusega $2n$, nii et kõik need sõned annavad `java.lang.String.hashCode()` meetodi puhul sama väärtuse. Esitage protsess, kuidas te vastuseni jõudsite.

Leidub ka ülesandeid, mis sarnanevad TLÜ-s esitatud kirjalikule ülesandele (vt näidisülesanne 5).

Lisaks kirjalikele ülesannetele peavad tudengid lahendama ära ka vähemalt neli programmeerimisülesannet seitsmest. Ülesannetes on lubatud kasutada Java sisseehitatud `HashMap` ja `HashSet` klasse. Seega seisnevad ülesanded paisktabeli kasutamise oskamises, mitte selle realiseerimisel. Kõik programmid peavad lõpetama oma töö ülimalt 1 sekundiga.

Näidisülesanne 16

1. Olgu meil mees nimega Olev, kes kõnnib lõpmatul kahemõõtmelisel laual alustades koordinaatidst $(0,0)$.

Kirjeldagem Olevi teekonda tähtedega „Ü“ - üles, „A“ - alla, „P“ - paremale ja „V“ - vasakule. Looge meetod mis võtab sisendiks sõne, mis kirjeldab Olevi teekonda.

Meetod peab tagastama täisarvu, mis väljendab, millal Olev külastab esimest korda ruutu, kus ta juba varem olnud on. Kui Olev ei külasta sama ruutu kunagi kaks korda, siis tagastage 0.

Sisendsõne pikkus võib olla 1 kuni 10^5 sümbolit.

2. Olgu teil seljakott, kus tahate kiiresti kontrollida esemete sisalduvust. Looge klass, mis seda teha võimaldab. Seljakoti klassil peavad olema järgnevad meetodid.

- a) Konstruktor;
- b) `lisa(int x)`;
- c) `eemalda(int x)`;
- d) `sisaldab(int x)`.

Meetodeid võidakse töö käigus välja kutsuda kuni 500 000 korda. Kõik seljakoti esemed on tähistatud numbritega vahemikust $0 \dots 10^9$. Pidage meeles, et seljakott võib sisaldada ühte eset mitu korda ja eemalda meetod peab eemaldama vaid ühe koopia esemest.

3. Teil on antud DNA lõik, mis koosneb tähtedest A, C, G ja T. Teie ülesanne on leida lühima pikkusega DNA jada, mis ei sisaldu ette antud DNA lõigus.

DNA lõigu pikkus on vahemikust $0 \dots 10^5$.

Ülaltoodud ülesanded on samuti pärit sellest kodutööst [43]. Nagu näha, tegemist on probleemülesannetega.

5.7 Washingtoni Ülikool

Washingtoni Ülikoolis õpetatakse paisktabeleid aines „Data Structures and Algorithms“ [44]. Kursuse põhiõpikuks on M. Weissi „Data Structures and Algorithm Analysis in Java (3rd edition)“, kuid selle lugemine pole kohustuslik [44].

Washingtoni ülikoolis käsitletakse paisksalvestuse loengutes järgnevaid teemasid [45][46]:

- sõnastik kui abstraktne andmestruktuur, sõnastiku realisatsioonid ilma paisktabelita;
- paisktabel, paiskamine jäägimeetodil, põrked, välisahelatega paisktabel;
- sõnede paiskamine, lahtise adresseerimisega paisktabel;
- lineaarkompimine, ruutkompimine, topeltpaiskamine;

- paisktabelioperatsioonide ajaline keerukus.

Sarnaselt paljude teiste ülikoolidega on loengus lisaks muudele näidetele ka koodinäited programmeerimiskeeles Java [45].

Loengus antakse tudengitele harjutusülesanne paisktabelisse kirjete lisamise kohta. Ülesanne on väga sarnane sellele, mis esitati TÜ loengus (vt näidisülesanne 2) [45]. Samasugust näidet kasutatakse mitu korda. Näiteks hiljem tutvustatakse sellise ülesande abil tudengitele kirjete kuhjumist [46].

Uuritavas aines ei ole paisktabeli teemal harjutusülesandeid, kuid selle kohta on üks kodune ülesanne [47]. Anname siin selle lühikirjelduse [48].

Näidisülesanne 17

1. Realiseerige sõnastik kasutades sisemise andmestruktuurina lahtise adresseerimisega paisktabelit.
2. Realiseerige sõnastik kasutades sisemise andmestruktuurina välisahelatega paisktabelit.
3. Realiseerige hulk kasutades sisemise andmestruktuurina välisahelatega paisktabelit.

Tegemist on lihtsalt andmestruktuuri realiseerimisülesandega. Samas teeb ülesande raske-
maks asjaolu, et tudengid peavad muretsema andmestruktuuri efektiivsuse pärast [48]. Seega peavad nad õigel hetkel oma paisktabeli suurust suurendama või vähendama.

Peale kodutöö on aines vaheeksam ja eksam [47]. Selleks õppimiseks on aine kodulehel saadaval varasemate aastate ülesanded [49]. Enamustes variantides on need ülesanded lihtsalt paisktabelisse kirjete sisestamise läbimängimine (vt näiteülesanne 4) [49]. Mõnes variandis leidub ka teistsuguseid ülesandeid nagu järgmises näites [50].

Näidisülesanne 18

Selgitage, miks on kuupäevade paiskamine aasta esimese kahe numbriga järgi hea/halb idee.

Mõnedes eksamivariantides esineb ka probleemülesandeid [51]. Üks näide oleks andmestruktuuri valiku ülesanne. Tudengile antakse ette 5-6 andmestruktuuri ning nad peavad valima, millist andmestruktuuri millise probleemi lahendamiseks kasutada [51].

5.8 Võrdlus Tartu Ülikooliga

Ka välismaa ülikoolide loengutes käsitletud teemad langevad suures osas kokku TÜ aine „Algoritmid ja andmestruktuurid“ teemadega. Üks erinevustest on see, et mõnedes ülikoolides tuuakse loengumaterjalides välja paisktabelite või paiskfunktsioonide realiseerimise programmeerimisekeeles Java. TÜ kursusel aga loengutes koodinäidetega ei tegeleta. Samas ei ole üheski välisülikoolis mainitud ära, et paisktabeleid saab kasutada sorteerimiseks.

Teine erinevus TÜ ja välisülikoolide on see, kuidas paisktabeli teemat sisse juhatatakse. Tartu Ülikoolis tutvustatakse paisktabelit kui kiire otsinguga andmestruktuuri. Paaris ülikoolis aga tutvustatakse paisktabelit kui viisi sõnastiku abstraktse andmestruktuuri realiseerimiseks. Browni ja Chalmersi ülikoolid aga toovad paisktabeli tutvustamiseks otsingumootori näite.

Nagu enne veendusime, on Tartu Ülikoolis kõik meie poolt esitatud ülesannete tüübid olemas. Kui aga välismaa ülikooli vaadata, siis on olukord teistsugune. Uuritud seitsmest ülikoolist kuues esinevad lihtsad teooriaülesanded. Ainult kolmes esinevad ka veidi keerulisemad teooriaülesanded.

Programmeerimisülesanded esinevad seitsmest ülikoolist viies, enamustes koolides on vaid ühte tüüpi programmeerimisülesanne. Realiseerimisülesanded on Florida ja Washingtoni ülikoolides, empiirilised ülesanded Browni ja Bergen ülikoolides. Probleemülesandeid õnnestus leida vaid Helsingi ja Washingtoni ülikoolide materjalidest.

6. Paisktabeliprintsiibi õpetamine tippülikoolides

Järgmises peatükis uurime kuidas tutvustatakse paisktabeli temaatikat tudengitele maailma tippülikoolides. Valitud ülikoolid ja nende asetus on järgmine [6]:

- Harvardi Ülikool (6. koht) ;
- Massachusettsi Tehnoloogiainstituut (4. koht) ;
- California Ülikool Berkeleys (15. koht) ;
- Princetoni Ülikool (7.koht) .

6.1 Harvardi Ülikool

Harvardi ülikoolis õpetatakse paisktabeleid tudengitele aines „Data Structures and Algorithms“ [52]. Aine põhiõpikuks on Thomas Cormeni raamat „Introduction to Algorithms“ [52]. Lisalugemiseks pakutud teos on Jon Kleinbergi „Algorithm Design“ [52].

Järgnev loetelu ja lõik põhinevad Harvardi Ülikooli loenguslaididel [53]. Aine loengues käsitletakse järgnevaid teemasid:

- paisktabeli mõiste, pörked;
- välisahelatega paisktabel;
- universaalne paiskamine;
- päringu kiirus eeldusel, et on tegemist universaalse paiskamisega.

Harvardi loengus on suur rõhk pandud universaalsele paiskamisele ja selle matemaatilisele definitsioonile. Loengus on antud näide kuidas konstrueerida universaalset paiskfunktsiooni ja tõestatud teoreem universaalset paiskfunktsiooni kasutava paisktabeli kiiruse kohta. Samas paisktabeli mõiste enda kohta pole toodud ühtegi näidet ja pigem eeldatakse, et tudeng otsib selle info ise või paneb loengus hästi tähele.

Edasine info siin peatükis pärineb uuritava aine harjutusülesannete failist [54]. Harvardi Ülikooli praktikumides lähenetakse paisktabeli teemale tõenäosusteooria alusel ning tudengid peavad oskama rakendada teadmisi matemaatikast omandatud teadmisi, et edukalt ülesandeid lahendada. Praktikume on aines 13 ning üks neist keskendub paisktabeli teemale. Näiteks palutakse praktikumis lahendada järgnev ülesanne.

Näidisülesanne 19

Olgu meil paisatud n kirjet välisahelatega paisktabelisse suurusega m suvalise paiskfunktsiooniga h (s.t et suvalise x jaoks võib $h(x)$ olla võrdse tõenäosusega ükskõik millise rea indeks). Vastake küsimustele.

- 1) Milline on eeldatav tabeli ridade arv, milles asub täpselt üks kirje?
- 2) Milline on eeldatav tabeli ridade arv, milles asub täpselt kaks kirjet?
- 3) Milline on eeldatav tabeli ridade arv, milles asub rohkem kui kaks kirjet?

Lisaks tõestatakse praktikumis veel teoreeme universaalsete paiskfunktsioonide kohta. Universaalse paiskamise kohta peavad tudengid lahendama alltoodud keerulise ülesande.

Näidisülesanne 20

Ütleme, et paiskfunktsioonide $P: \{1, \dots, U\} \rightarrow \{1, \dots, m\}$ pere on universaalne, kui iga $1 \leq x \leq y \leq U$ korral $P(h(x) = h(y)) \leq \frac{1}{m}$, kus h on suvaline paiskfunktsioon vaadeldavast perest. Teisisõnu, pörke tõenäosus on maksimaalselt $\frac{1}{m}$.

Olgu meil antud algarv p . Paiskame n -elemendilisi sõnesid, mille iga element on arv hulgast $\{0, \dots, p-1\}$. Paisktabeli reaaindeksid on $\{0, \dots, p-1\}$. Paiskfunktsioonid saame järgnevalt: loome n -elemendilise enniku (c_1, \dots, c_n) , kus iga element on valitud suvaliselt hulgast $\{0, \dots, p-1\}$. Suvalise sõne $x = (x_1, x_2, \dots, x_n)$ jaoks leiame $h(x) = \sum_{i=1}^n c_i x_i \pmod{p}$.

Tõestage, et selline protsess loob universaalse paiskfunktsioonide pere.

Sarnast tüüpi ülesanded esinevad ka kodutöös [54]. Kuigi teooriaülesandeid on palju, ei nõua Harvardi Ülikool tudengitelt ühegi programmeerimisülesande lahendamist.

6.2 Massachusettsi Tehnoloogiainstituut

Massachusettsi Tehnoloogiainstituudi (edaspidi MIT) tudengid puutuvad paisktabelitega kokku aines „Introduction to Algorithms“ [56]. Kursuse põhiõpik on sama mis Harvardis - Cormeni „Introduction to Algorithms“ [57].

MIT loengute ülesehitus on järgmine [58][59][60]:

- sõnastik kui abstraktne andmestruktuur, sõnastikud keeles Python;
- otsene paisksalvestus, paiskfunktsioonid, pörked;
- välisahelatega paisktabel, paiskamise keerukus ühtlasel paiskamisel;

- paiskamine jäägi meetodil, paiskamine korrutamise meetodil;
- universaalne paiskamine, paisktabeli suuruse muutmine;
- alamsõnede otsimise ülesanne, Karp-Rabini algoritm;
- lahtise adresseerimisega paisktabel, lineaarne kompimine, topeltpaiskamine;
- otsimise keerukus lahtise adresseerimisega paisktabelis ühtlase paiskamise korral.

Kokku on MITis paisktabeli ja paiskfunktsioonide teema kohta tervelt kolm loengut [56]. Seega jõutakse loengutes käsitleda väga palju erinevaid teemasid. Lisaks räägitakse kolmandas loengus veel krüptograafilistest räsifunktsioonidest [60].

Erinevalt Harvardist antakse MITis tudengitele nii teoreetilisi kui ka praktilisi ülesandeid [61]. Esiteks nõutakse tudengitel valikvastustega küsimustele vastamist [61]. Näiteks on esitatud ülesanne paisktabeli suuruse muutmise kohta.

Näidisülesanne 21

Juku tahab realiseerida oma paisktabeli jaoks pörgete lahendamise viisi ja ka paisktabeli suuruse dünaamilise muutmise. Samas ei soovi ta teha üleliigset tööd ja seega tekib tal küsimus, kas mõlemat on vaja, et säilitada paisktabeli töökindlus ja efektiivsus. Kui Jukul on dünaamiline paisktabeli suuruse muutmine, siis pörgete vältimiseks võib ta ju alati paisktabelit suuremaks teha. Sarnaselt kui tal on algoritm pörgete vältimiseks, siis ei põhjusta pörked paisktabeli töös mingeid häireid. Milline alltoodud väidetest on õige?

- 1) Dünaamiline paisktabeli suuruse muutmine säilitab nii efektiivsuse kui töökindluse.
- 2) Dünaamiline paisktabeli suuruse muutmine säilitab ainult töökindluse.
- 3) Ainult pörgete lahendamine ilma paisktabeli suurust dünaamiliselt muutmata tagab efektiivsuse, kuid mitte töökindluse.

Sarnased küsimused ootavad tudengeid ka eksamil [62]. Praktilise ülesandena on vaja realiseerida DNA lõikude võrdlemisalgoritm, kasutades paiskfunktsioone [61]. Kuna ülesande sõnastus on väga pikk, siis jätame selle siin kirjeldamata.

6.3 California Ülikool Berkeleys

California Ülikool Berkeleys käsitleb paisktabelite teemat aines „Data Structures and algorithms“ [63]. Ainel ei ole kohustuslikku õpikut, kuid soovitatakse lugeda R. Sedgewicki teost „Algorithms“ [64].

California ülikool Berkeleys käsitleb paiksaldvestuse loengute raames järgnevaid teemasid [65]:

- lineaarne otsing, paiskfunktsioonid;
- välisahelatega paisktabel, pörked, lahtise adresseerimisega paisktabel;
- linearkompimine, ruutkompimine, topeltpaiskamine;
- paisktabeli suuruse muutmine;
- sõnede paiskamine, järjendite paiskamine, puude paiskamine;
- Java klassid, mis kasutavad paisktabelit;
- monotoonne paiskamine, täielik paiskamine;
- paisktabeli operatsioonide keerukus, paisktabeli efektiivsus võrreldes teiste andmestruktuuridega.

California ülikool Berkeleys nõuab laborites tudengitelt mõnede lühiküsimuste vastamist, mis on seotud paiskfunktsioonidega [66]. Näiteks järgnev ülesanne nõuab tudengilt sobiva paiskfunktsiooni loomist [66].

Näidisülesanne 22

Kujutagem `trips-traps-trulli` lauda kahemõõtmelise massiivina. Arve 0,1,2 kasutagem, et kujutada vastavalt tühja ruutu, risti ja ringi.

Sõnastage paiskfunktsioon, mis ei anna erinevate lauaseisude korral sama väärtust.

Mõned California ülikooli ülesanded vajavad tudengitel Java keelega lähemat tutvumist [66].

Näidisülesanne 23

Kas Java `HashSet` andmestruktuuri on võimalik lisada piiramatu hulk sõnesid, ilma pörkeid tekitamata?

Kui ei, siis milline on minimaalne arv erinevaid sõnesid, mis garanteerib pörke tekkimise `HashSeti` lisamisel?

Lisaks on tudengitele antud kodutöök paisktabeli teemal programmeerimisülesanne [67].

Näidisülesanne 24

Looge sõnede hulga klass *ECHashStringSet*. Klass peab kasutama andmete säilitamiseks välisahelatega paisktabelit.

Et tagada andmestruktuuri kiiruse peaksite tabelit suurendama, kui koormustegur on kõrgem kui 5. Samuti peaksite tabeli suurust vähendama, kui koormustegur on väiksem kui 20%, et vältida mälu raiskamist.

Kodutöö käigus ei ole lubatud kasutada teisi Java klasse, mis sisaldavad nime „hash“. On aga lubatud kasutada *String* klassi meetodit *hashCode*.

6.4 Princetoni Ülikool

Princetoni Ülikoolis õpetatakse paisktabeleid koos muude andmestruktuuridega aines „Algorithms and Data Structures“ [68]. Aine põhineb R. Sedgewicki raamatul „Algorithms“ [69].

Princetoni Ülikooli loengumaterjalid on täpselt samad, mis Bergen Ülikoolis, seega ei hakka me neid siin uuesti kirjeldama [68].

Princetoni ülikoolis ei ole esitatud ühtegi hindelist paisksalvestuse teemalist ülesannet [68]. Esitatud kodutööd paisktabeleid ei kasuta [70]. Loengute kohta on kodulehel väiksed küsitlused, kuid nende sisu pole avalik [71]. Siiski on lisatud loengumaterjalide hulka valim harjutusülesandeid, et teemat omandada [72]. Ülesanded on jaotatud hinnete järgi erinevatesse raskusastmetesse [72]. Lihtsaim ülesanne on paisktabeli ja paiskfunktsiooni rakendamise peale [72]:

Näidisülesanne 25

Olgu meil funktsioon `hashCode()`, mis liidab arvus sisalduvad numbrid kokku. Näiteks numbri „342“ puhul on paiskfunktsiooni väärtus $3+4+2 = 9$. Olgu meil järgmised andmed:

Väärtus	Võti	hashCode	Indeks
A	13	4	4
B	15	6	1
C	2	2	
D	34	7	
E	16	7	
F	100	1	

1. Kui on tegemist paisktabeliga, mille suurus on 5, siis leidke puuduvad reaindeksid kasutades jäägi leidmise meetodit.
2. Joonistage paisktabel, mis tekib kui need kuus kirjet lisatakse paisktabelisse ja pörgete lahendamiseks kasutatakse välisahelaid.

Veidi raskema ülesandena on soovitatud lahendada ülesanne 3.4.15 kursuse õpikust [72].

Näidisülesanne 26

Kui palju võrdlusi peame tegema halvimal juhul, et sisestada N kirjet tühja paisktabelisse, mis kasutab välisahelaid?

Kõige raskema taseme ülesandena pakutakse järgneva andmestruktuuri loomist [72].

Näidisülesanne 27

Looge LRU andmestruktuuri klass, mis omab järgmisi operatsioone:

1. `LRU(int N)` - loob LRU isendi, mille mahutavus on N ,
2. `void cache(Key key)` - salvestab võtme andmestruktuuri. Kui andmestruktuur on täis, siis eemaldatakse kõige varasemalt lisatud kirja ning uus kirje sisestatakse,
3. `boolean inCache(Key key)` - kontrollib kas antud võti asub andmestruktuuris.

Meetodid `cache` ja `inCache` peavad olema keskmiselt konstantse keerukusega operatsioonid universaalse paiskamise eeldusel.

Kirjeldage loodud andmestruktuuri ja tooge näide selle tööst.

Princetoni ülikoolis on avalikult saadavad ka eelmiste aastate eksamid ja vaheksamid [73]. Seal leiduvad ülesanded on üldiselt sama tüüpi nagu üleval kirjeldatud näidetes [73]. Mõnedel eksamitel on ka andmestruktuuri disaini ülesanne [74]. Toome siin näite ühest sellisest ülesandest, mille lahendamiseks saab kasutada paisktabelit [74].

Näidisülesanne 28

Looge FIFO järjekord, mis sisaldab sõnesid, kusjuures järjekorras saab ühte sõne sisaldada ainult ühe koopia.

Klassifail peab toetama järgmisi operatsioone:

1. konstruktor;
2. lisamismeetod – lisab järjekorda sõne, kui see seal juba ei sisaldu;
3. eemaldamise meetod – eemaldab kõige varasemalt lisatud sõne.

6.5 Võrdlus Tartu Ülikooliga

Sarnaselt ülejäänud välismaa ülikoolidega on ka maailma tippülikoolide loenguteemad väga sarnased Tartu Ülikooli omadega. Samas on maailma tippülikoolide hulgas paar erandit.

Harvardi Ülikoolis käsitletakse enam-vähem loengumaterjalides paisktabeliga seotud olulised mõisted ära. Seda aga tehakse väga lühidalt ja lakooniliselt. Rohkem materjali mahust on keskendatud universaalse paiskamise selgitamisele, mida TÜ loengutes ei mainita.

MIT on aga eriline selle poolest, et paisksalvestuse loengutes käsitletakse teemasid, mis on seotud krüptograafiliste räsifunktsioonidega. Loomulikult on need paisktabeli teemaga natukene seotud, kuid Tartu Ülikoolis on need teemad üldiselt eraldatud.

Ülesannete tüüpide poolest eelistavad tippülikoolid pigem teooriaülesandeid. Lihtsamaid teooriaülesandeid sisaldavad nii MIT kui ka Princetoni Ülikool. Veidi keerulisemaid teooriaülesandeid esineb aga kõikide vaadeldud tippülikoolide ülesannete hulgas. Programmeerimisülesandeid esineb MIT-s ja California Ülikoolis. Üks neist on realisatsiooni ülesanne, teine aga probleemülesanne. Lisaks leidub probleemülesandeid ka Princetoni Ülikoolis.

Jällegi paneme tähele, et ühelgi tippülikoolil ei esine kõik ülesannete tüübid, erinevalt Tartu Ülikoolist. Sarnaselt madalama asetusega ülikoolidega on kõige tavalisemad ülesanded kirjalikud teooriaülesanded. Veel üks sarnasus teiste välismaa ülikoolidega on see, et probleemülesandeid esineb väga vähe.

Mõned tudengitele antud ülesanded erinevad Tartu Ülikooli omadest, sest nad nõuavad tudengilt Java programmeerimiskeele väga head tundmist. Näiteks näidisülesande 23 lahendamiseks pead sa teadma, kuidas täpselt on Java klass *HashSet* realiseeritud.

7. Analüüs

Järgnevas peatükis analüüsitakse erinevate ülikoolide kohta leitud informatsiooni.

Enne, kui analüüsime ülesandeid ja loengumaterjale, vaatleme korra uuritud ainete kasutatud kirjandust. Töö käigus uuritud kursused omavad kõik väga erinevaid õpikuid. Kõige populaarsemad raamatud olid R. Sedgewicki ja K. Wayne'i „Algorithms“ ning autorite T. Cormen, C. Leirserson, R. Rivest, C. Stein teos „Introduction to Algorithms“. Kuigi need raamatud olid kõige populaarsemad, esinesid nad vaid kolme erineva ülikooli materjalide hulgas. Seega on näha, et paisktabeli õpetamisel ei leidu ühtset taustmaterjali, mida kõik ülikoolid kasutavad.

Nagu eelpool nägime, on suures osas loengumaterjalid kõikidel uuritud ülikoolidel sama sisuga. Samas on ikkagi väiksemaid erinevusi. Näiteks paisktabeli kasutamist sorteerimiseks mainitakse ainult Tartu Ülikoolis. Lisaks on ka teemasid nagu universaalne paiskamine, mis esineb mitme ülikooli loengutes, kuid Tartu Ülikoolise sellest ei räägita. Kolmas erinevus on see, et mitme ülikooli loengud on Java keele kesksed, sisaldades koodinäited ja kirjeldades Java keeles leiduvaid paisksalvestusega seotud klasse. See on jällegi asi, mis Tartu Ülikooli loengutes käsitlemata jääb.

Seega loengumaterjalide vaatamine osaliselt toetab meie esimest hüpoteesi, kuid ainult selle põhjal me väita ei saa, et paisktabeli teema õpetamine teistes ülikoolides väga erinev on. Lõppkokkuvõttes ei ole alati kõik teemad loengus sama tähtsusega. Näiteks Browni Ülikoolis esitatakse loengus pikk ja keeruline universaalse paiskfunktsiooni tõestus, kuid seejärel mainitakse, et seda tegelikult tudengilt eksamil ei küsita.

Hoopis teistsugune olukord on aga uuritud ainetes esitatud ülesannetega. Esiteks on Tartu Ülikoolis tudengitele harjutamiseks välja pakutud palju rohkem materjali ülesannete kogu kujul. Seega on TÜ tudengitel palju rohkem võimalusi harjutada. Teiseks on Tartu Ülikool uuritud ülikoolidest ainuke, kus esinevad igat tüüpi ülesanded – nii eri raskus astmega teooriaülesanded, realiseerimisülesanded kui ka probleemülesanded. Kui vaadata teisi ülikoole, siis näeme, et probleemülesandeid on välja pakutud väga vähestes ülikoolides. Kõige levinuimad ülesannete tüübid on kirjalikud teooriaülesanded.

Seega ülesannete valik teistes ülikoolides toetab väga tugevasti esimest hüpoteesi. Paisktabeli teemat käsitletakse tõesti teistes ülikoolides Tartu Ülikoolist erinevalt. Ühtlasi tõestab

ülesannete erinevus ka teise hüpoteesi, sest ka tippülikoolides esineb probleemülesannete vähesus ning suurem fookus on teooriaülesannetel.

Samas saame ka tähele panna, et teine hüpotees kehtib ka madalama asetusega ülikoolide kohta, kuna seal on ülesande tüüpide jaotus sarnane. Seega selgub, et kuigi me jaotasime ülikoolid edetabeli järgi erinevatesse kategooriatesse, pole nende ülikoolide lähenemine paisktabeli temaatika õpetamisele üldse kuigi erinevad.

Kuna loengumaterjalid on enamustes ülikoolides sarnased Tartu Ülikooli omadega ja ülesannete valik kitsam, siis võime järeldada, et Tartu Ülikoolis käsitletakse paisktabeli teemat võrreldes teiste ülikoolidega suhteliselt hästi ja suuri muudatusi ilmselt tarvis ei ole.

8. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli teada saada, kuidas õpetatakse paisktabelit ja selle andmestruktuuri kasutamist teistes Eesti ja välismaa ülikoolides. Töös kirjeldati ära 14 erineva ülikooli ained, mis sisaldasid paisktabeli teemat. Uuritud ülikoolid olid pärit viiest erinevast riigist.

Töö alguses püstitati kaks hüpoteesi:

- paisktabeli õpetamine teistes ülikoolides on teistsugune kui Tartu Ülikoolis;
- paisktabeli õpetamine tippülikoolides keskendub rohkem teooriale kui paisktabeli rakendustele.

Loengumaterjalide analüüsist selgus, et enamustes ülikoolides käsitletakse paisktabelite teemat loengutes väga sarnaselt. Esinesid vaid väiksed erinevused mõnede nagu koodinäidete olemasolu ning see kuidas paisktabeli teemat sisse juhatati.

Töö käigus tõlgiti ja sõnastati ümber 28 näidisülesannet, mis illustreerivad erinevates ülikoolides leiduvaid ülesannete tüüpe. Tegelikult jõudis töö kaante vahele neid ülesandeid isegi rohkem, sest mõne näite all kirjeldati mitu ülesannet.

Ülesannete analüüsist saadi teada, et valdav enamik ülikooli ei esita tudengitele ülesandeid, kus eesmärgiks on mingi probleem lahendada. Selle asemel leiti, et levinuim ülesannete liik on erineva raskusastmega teooriaülesanded. Eranditeks olid ülikoolid nagu Helsingi ja Princetoni Ülikool, kus esitati lahendamiseks ka probleemülesandeid.

Lähtudes ülesannete analüüsist, said kinnitust mõlemad hüpoteesid. Lisaks saadi teada, et paisktabeli õpetamine ei ole tippülikoolides väga palju erinev kui teistes kõrgkoolides. Taustmaterjalide uurimisest sai autor teada, et paisktabeli õpetamiseks kasutatakse väga palju erinevaid õpikuid ning ei paista olevat ühte kindlat populaarset õpikut.

Viidatud kirjandus

- [1] Mehlhorn, K. ja Sanders, P. (2008) *Algorithms and Data Structures*. Berliin: Springer.
- [2] Kiho, J. (2003) *Algoritmid ja andmestruktuurid*. Tartu: Tartu Ülikooli Kirjastus.
- [3] Cormen, T., Leiserson, C., Rivest, R., Stein, C. (2001) *Introduction to Algorithms*. Cambridge, Massachusetts: The MIT press.
- [4] Sedgewick, R., Wayne, K. (2011) *Algorithms*. New Jersey: Pearson Education.
- [5] Kleinberg, J. ning Tardos, É. (2005) *Algorithm Design*. Boston: Addison-Wesley Longman Publishing Co.
- [6] Times Higher Education World University Rankings.
<https://www.timeshighereducation.com/world-university-rankings/2019/world-ranking> (09.05.2019).
- [7] Peder, A., Kiho J., Nestra, H. (2017) *Algoritmid ja andmestruktuurid. Ülesannete kogu*. Tartu: Tartu Ülikooli Kirjastus.
- [8] Tartu Ülikooli Informaatika eriala bakalaureusetaseme õppekava.
<https://ois2.ut.ee/#/curricula/2476/version/2019/details> (10.05.2019).
- [9] Tartu Ülikooli aine „Algoritmid ja andmestruktuurid“ kirjeldus õppeinfosüsteemis.
<https://ois2.ut.ee/#/courses/LTAT.03.005/details> (10.05.2019).
- [10] Peder, A. (2018). Aine „Algoritmid ja andmestruktuurid“ loenguslaidid.
https://moodle.ut.ee/pluginfile.php/16354/mod_resource/content/9/loeng6_7aasta2018.pdf (09.05.2019).
- [11] IT Kolledži aine „Algoritmid ja Andmestruktuurid“ koduleht.
<http://enos.itcollege.ee/~jpoial/algoritmid/index.html> (10.05.2019).
- [12] Aine „Algoritmid ja andmestruktuurid“ kirjeldus TLÜ õppeinfosüsteemis.
<https://ois2.tlu.ee/tluois/aine/IFI6083.DT> (10.05.2019).
- [13] Petuhhov, I. (2018). TLÜ aine „Algoritmid ja andmestruktuurid“ loengukonspekt.
http://www.cs.tlu.ee/~inga/alg_andm_18/hash_2010.pdf (09.05.2019).
- [14] Petuhhov, I. (2018). TLÜ aine „Algoritmid ja andmestruktuurid“ loenguslaidid.
http://www.cs.tlu.ee/~inga/alg_andm_18/Paiskmeetod_slaidid_2018.pdf (09.05.2019).
- [15] Petuhhov, I. (2018). TLÜ aine „Algoritmid ja andmestruktuurid“ ülesannete leht.
http://www.cs.tlu.ee/~inga/alg_andm_18/Ylesanded/ (09.05.2019)
- [16] Petuhhov, I. (2018). TLÜ aine „Algoritmid ja andmestruktuurid“ kordamisküsimused.
http://www.cs.tlu.ee/~inga/alg_andm_18/kordamiskysimused_18.txt (09.05.2019)
- [17] Tallinna Tehnikaülikooli aine „Algoritmid ja andmestruktuurid“ kirjeldus õppeinfosüsteemis.
https://ois.ttu.ee/portal/page?_pageid=37,674581&_dad=portal&_schema=PORTAL&link=345192D6B58A1474 (10.05.2019).

- [18] Leppikson, V. (2019). TalTechi aine „Algoritmid ja andmestruktuurid“ loenguslaidid. <http://www.tud.ttu.ee/im/Viktor.Leppikson/IAS0090%20slaidid.pdf> (10.05.2019).
- [19] Viktor Leppiksoni koduleht. <http://www.tud.ttu.ee/im/Viktor.Leppikson/> (10.05.2019).
- [20] Tallina Tehnikaülikooli aine „Algoritmid ja andmestruktuurid“ laborijuhend. <http://www.tud.ttu.ee/im/Viktor.Leppikson/IAS0090%20labori%20juhend%202018.pdf> (10.05.2019).
- [21] Houstoni Ülikooli aine „Data Structures“ koduleht. <http://www2.cs.uh.edu/~arjun/courses/ds/> (10.05.2019).
- [22] Florida Riikliku Ülikooli Arvutiteaduse Osakonna bakalaureuseõppe ainete kirjeldused. <https://www.cs.fsu.edu/academics/undergraduate-programs/undergraduate-course-descriptions/> (10.05.2019).
- [23] Florida Riikliku Ülikooli aine COP 4530 õppekava. <http://www.cs.fsu.edu/~duan/classes/cop4530/syllabus.htm> (10.05.2019).
- [24] Florida Riikliku Ülikooli aine COP 4530 loenguslaidid. <http://www.cs.fsu.edu/~myers/cop4530/notes/> (10.05.2019).
- [25] Florida Riikliku Ülikooli aine COP 4530 kodulehekülg. <http://www.cs.fsu.edu/~myers/cop4530/> (10.05.2019).
- [26] Browni Ülikooli koduleht. Aine „Introduction to Algorithms and Data Structures“ infoleht. <https://cs.brown.edu/courses/info/csci0160/> (10.05.2019).
- [27] Browni Ülikooli aine „Introduction to Algorithms and Data Structures“ koduleht. Lisamaterjalid. <http://cs.brown.edu/courses/csci0160/docs.html> (10.05.2019).
- [28]] Browni Ülikooli aine „Introduction to Algorithms and Data Structures“ koduleht. Loengumaterjalid. <http://cs.brown.edu/courses/csci0160/lectures.html> (10.05.2019).
- [29] Browni Ülikooli aine „Introduction to Algorithms and Data Structures“ 4. kodutöö. <http://cs.brown.edu/courses/csci0160/static/files/assignments/homeworks/hw4.pdf> (10.05.2019).
- [30] Browni Ülikooli aine „Introduction to Algorithms and Data Structures“ vaheeksami näidis. http://cs.brown.edu/courses/csci0160/static/files/assignments/practice_exams/practice_midterm.pdf (10.05.2019).
- [31] Browni Ülikooli aine „Introduction to Algorithms and Data Structures“ näidiseksam. http://cs.brown.edu/courses/csci0160/static/files/assignments/practice_exams/practice_final.pdf (10.05.2019).
- [32] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ koduleht. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/> (10.05.2019).
- [33] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ koduleht. Ressursside alamleht. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/resources/> (10.05.2019).
- [34] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ koduleht. Laborijuhendite alamleht. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/labs/> (10.05.2019).

- [35] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ koduleht. Harjutusülesannete alamleht. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/selfstudy/> (10.05.2019).
- [36] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ 2018.05.31 toimunud eksami ülesanded. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/files/exam/exam2018-1.pdf> (10.05.2019).
- [37] Chalmersi Tehnoloogiaülikooli aine „Data Structures“ 2018.8.24 toimunud eksami ülesanded. <http://www.cse.chalmers.se/edu/year/2018/course/DIT961/files/exam/exam2018-2.pdf> (10.05.2019).
- [38] Bergeni Ülikooli aine „Algoritmar, datastruktuur og programmering“ loenguslaidid. <https://mitt.uib.no/courses/12780/files/folder/lecturenotes> (10.05.2019).
- [39] Bergeni Ülikooli aine „Algoritmar, datastruktuur og programmering“ õppekava. <https://mitt.uib.no/courses/12780/assignments/syllabus> (10.05.2019).
- [40] Bergeni Ülikooli aine „Algoritmar, datastruktuur og programmering“ kolmas kodutöö. <https://github.com/torsteins/inf102f18-mandatory2> (10.05.2019).
- [41] Helsingi Ülikooli aine „Tietorakenteet ja algoritmit“ loenguslaidid. https://moodle.helsinki.fi/pluginfile.php/1865262/mod_resource/content/23/tira_k kaikki.pdf (10.05.2019).
- [42] Helsingi Ülikooli aine „Tietorakenteet ja algoritmit“ koduleht. <https://courses.helsinki.fi/fi/tkt20001/119558719> (10.05.2019).
- [43] Helsingi Ülikooli aine „Tietorakenteet ja algoritmit“ 5. kodutöö. https://moodle.helsinki.fi/pluginfile.php/1916053/mod_resource/content/6/week05.pdf (10.05.2019).
- [44] Washingtoni Ülikooli aine „Data Structures and Algorithms“ koduleht. Õppekava. <https://courses.cs.washington.edu/courses/cse373/19wi/syllabus/> (10.05.2019).
- [45] Washingtoni Ülikooli aine „Data Structures and Algorithms“ 10. loengu slaidid. <https://courses.cs.washington.edu/courses/cse373/19wi/files/lectures/slides/lecture10.pdf/> (10.05.2019).
- [46] Washingtoni Ülikooli aine „Data Structures and Algorithms“ 11. loengu slaidid. <https://courses.cs.washington.edu/courses/cse373/19wi/files/lectures/slides/lecture11.pdf/> (10.05.2019).
- [47] Washingtoni Ülikooli aine „Data Structures and Algorithms“ koduleht. <https://courses.cs.washington.edu/courses/cse373/19wi/> (10.05.2019).
- [48] Washingtoni Ülikooli aine „Data Structures and Algorithms“ 3. kodutöö. <https://courses.cs.washington.edu/courses/cse373/19wi/homework/3/> (10.05.2019).

- [49] Washingtoni Ülikooli aine „Data Structures and Algorithms“ koduleht. Eksamid. <https://courses.cs.washington.edu/courses/cse373/19wi/exams/> (10.05.2019).
- [50] Washingtoni Ülikooli aine „Data Structures and Algorithms“ 2017. aasta talve vaheksam. <https://courses.cs.washington.edu/courses/cse373/19wi/files/exams/midterm-practice/cse373-17wi-practice1.pdf> (10.05.2019).
- [51] Washingtoni Ülikooli aine „Data Structures and Algorithms“ 2014. aasta talve eksam. <https://courses.cs.washington.edu/courses/cse373/19wi/files/exams/finals-practice/cse373-14wi-final.pdf> (10.05.2019).
- [52] Harvardi Ülikooli aine „Data Structures and Algorithms“ koduleht. Õppekava. <http://sites.fas.harvard.edu/~cs124/cs124/syllabus.html> (10.05.2019).
- [53] Harvardi Ülikooli aine „Data Structures and Algorithms“ 13. loeng. <http://sites.fas.harvard.edu/~cs124/lecs/lec13.pdf> (10.05.2019).
- [54] Harvardi Ülikooli aine „Data Structures and Algorithms“ 9. praktikumi ülesanded. <http://sites.fas.harvard.edu/~cs124/sections/section9.pdf> (10.05.2019).
- [55] Harvardi Ülikooli aine „Data Structures and Algorithms“ 7. kodutöö. <http://sites.fas.harvard.edu/~cs124/psets/pset7.pdf> (10.05.2019).
- [56] MIT aine „Introduction to Algorithms“ loengumaterjalid ja käsitletavat teemad. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-notes/> (10.05.2019).
- [57] MIT aine „Introduction to Algorithms“ koduleht. Õppekava. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/syllabus/> (10.05.2019)
- [58] MIT aine „Introduction to Algorithms“ 8. loeng. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/MIT6_006F11_lec08.pdf (10.05.2019).
- [59] MIT aine „Introduction to Algorithms“ 9. loeng. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/MIT6_006F11_lec09.pdf (10.05.2019).
- [60] MIT aine „Introduction to Algorithms“ 10. loeng. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/MIT6_006F11_lec10.pdf (10.05.2019).
- [61] MIT aine „Introduction to Algorithms“ 5. kodutöö. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/assignments/MIT6_006F11_ps4.pdf (10.05.2019).
- [62] MIT aine „Introduction to Algorithms“ eksam. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/exams/MIT6_006F11_final.pdf (10.05.2019).
- [63] California Ülikooli Berkeleys aine „Data Structures“ koduleht. <http://inst.eecs.berkeley.edu/~cs61b/fa18/index.html> (10.05.2019).

- [64] California Ülikooli Berkeleys aine „Data Structures“ koduleht. Aine info alamleht.
<http://inst.eecs.berkeley.edu/~cs61b/fa18/about.html> (10.05.2019).
- [65] California Ülikooli Berkeleys aine „Data Structures“ 24. loeng.
<http://inst.eecs.berkeley.edu/~cs61b/fa18/materials/lectures/lect24.pdf> (10.05.2019).
- [66] California Ülikooli Berkeleys aine „Data Structures“ kuhjade ja paisktabelite harjutusülesanded.
<http://inst.eecs.berkeley.edu/~cs61b/fa18/materials/disc/discussion9.pdf> (10.05.2019).
- [67] California Ülikooli Berkeleys aine „Data Structures“ 6. kodutöö.
<http://inst.eecs.berkeley.edu/~cs61b/fa18/materials/hw/hw6/index.html> (10.05.2019).
- [68] Princetoni Ülikooli aine „Algorithms and Data Structures“ loengukava.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/lectures.php> (10.05.2019).
- [69] Princetoni Ülikooli aine „Algorithms and Data Structures“ õppekava.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/syllabus.php> (10.05.2019).
- [70] Princetoni Ülikooli aine „Algorithms and Data Structures“ kodutööd.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/assignments.php>
(10.05.2019).
- [71] Princetoni Ülikooli aine „Algorithms and Data Structures“ küsitlused.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/quizzes.php> (10.05.2019).
- [72] Princetoni Ülikooli aine „Algorithms and Data Structures“ paisksalvestus harjutusülesanded.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/lectures/study/34HashTables.html> (10.05.2019).
- [73] Princetoni Ülikooli aine „Algorithms and Data Structures“ eksamiülesanded.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/exams.php> (10.05.2019).
- [74] Princetoni Ülikooli aine „Algorithms and Data Structures“ 2017. aasta sügise vaheksam.
<https://www.cs.princeton.edu/courses/archive/spring19/cos226/exams/mid-f17.pdf>
(10.05.2019).

Lisad

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, _____ Alfred Saidlo _____,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
_____ Paisktabeliprintsiibi õpetamine ülikoolikursustel _____,
(*lõputöö pealkiri*)

mille juhendaja on _____ Ahti Peder _____,
(*juhendaja nimi*)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, alates **pp.kk.aaaa** kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Alfred Saidlo

10.05.2019