TARTU ÜLIKOOL
Arvutiteaduse instituut

Jüri Kiho

ALGORITMID JA ANDMESTRUKTUURID

Ülesannete kogu

TARTU 2005

T A R T U Ü L I K O O L
Arvutiteaduse instituut

Jüri Kiho

ALGORITMID JA ANDMESTRUKTUURID

ÜLESANNETE KOGU

TARTU 2005

Käesoleva ülesannete kogu väljaandmist on toetanud Eesti Infotehnoloogia Sihtasutus projekti "Tiigriülikool" raames.

Toimetaja: Peeter Laud

ISBN 9949-11-033-5

Tartu Ülikooli Kirjastus
http://www.tyk.ut.ee
Tellimus nr. 151

SISUKORD

Saateks

Käesolev ülesannete kogu kujutab endast lisamaterjali ainele *Algoritmid ja andmestruktuurid*.

Ülesannete rühmitusviis järgib võimalust mööda põhiõpiku [1] jaotisteks liigendust. Teemade valikul on eeskätt silmas peetud algajat õppurit: ülesannete raskusaste on madal, käsitlus on sihitud praktilisele tööle põhimõistetega. Viimases jaotises esitatakse mõningate (linnukesega ✓ märgitud) ülesannete näidislahendused.

Ülesannete kogu võiks olla abivahendiks nii üliõpilastele (mõistete kinnistamine, ettevalmistus eksamiks) kui ka õppejõududele (kontrollülesannete komplekteerimine, uute ülesannete koostamine).

Autor on tänulik kaasabi eest materjali ettevalmistamisel Reimo Palmile, kellelt pärinevad ka mõned kogusse lülitatud ülesanded.

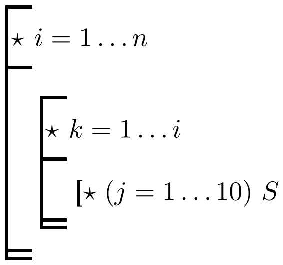
Autor
kiho@ut.ee

§ 1. Funktsiooni asümptootiline hinnang

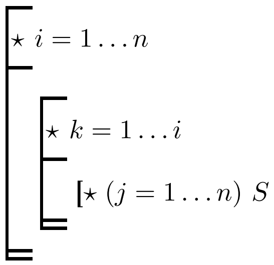
1. Tõestada omadused $1^0 - 9^0$ ([1], lk 15).
2. Näidata, et kui k on nullist erinev konstant, siis $kf(n)$ on $\Theta(f(n))$.
3. Millised O-relatsiooni omaduste ([1], lk 15) analoogid kehtivad ka Θ-relatsiooni kohta? Tõestada need Θ-relatsiooni omadused.
4. Näidata, et tabeli 1.1 ([1], lk. 12) esimeses veerus iga funktsioon on O talle järgnevast funktsioonist ega ole O talle eelnevast funktsioonist.
5. Näidata, et $12n \log n - 120 \log n$ on $\Omega(n \log n)$.
6. Näidata, et $12n \log n + 120 \log n$ on $\Omega(n \log n)$.
7. Näidata, et $\log_3 n + \sqrt{n}$ on $\Theta(\sqrt{n})$.
8. Näidata, et $1500n + n \log n$ on $\Theta(n \log n)$.
9. Leida võimalikult lihtne Θ-hinnang
   (a) funktsioonile $2n^2 + 3(\log n)^3$; ✓
   (b) funktsioonile $3n^2 + 5n \log n^5$.
10. Kas 2^n on $O(3^n)$?
11. Kas 2^n on $\Theta(3^n)$?
12. Näidata, et $n!$ ei ole $O(2^n)$. (Vt. Stirlingi valem [1], lk. 71.)

§ 2. Algoritmi ajaline keerukus

1. Hinnata joonistel 1 ja 2 esitatud algoritmide ajalist keerukust, kui alamalgoritmi S ajaline keerukus on $O(1)$.



Joonis 1. Kolmekordne tsükkel I.



Joonis 2. Kolmekordne tsükkel II.

2. Kirjutada mitterekursiivne algoritm, mis kontrollib, kas antud sõnes kõik sümbolid on paarikaupa erinevad. Hinnata selle algoritmi ajalist keerukust.
3. Kirjutada rekursiivne algoritm järjendi summa leidmiseks, mis jagab järjendi neljaks enam-vähem võrdseks osaks ja siis tagastab nende osade summade summa. Hinnata selle algoritmi ajalist keerukust põhiteoreemi kasutades. ✓
4. Hinnata algoritmi ajalist keerukust, kui lahendusaeg T avaldub
   (a) kujul $T(n) = 9T(n/3) + n$;
   (b) kujul $T(n) = 2T(n/2) + n^2$.

§ 3. Puu ja kahendpuu

1. Joonistada mingi 12-tipuline puu. Kirjutada see puu üles nii vasaku kui ka parema suluesitusena ning koostada sellele puule vastava kahendpuu tippude loetelud ees-, kesk- ja lõppjärjestuses.
2. Olgu tasemed kahendpuus nummerdatud $0, 1, 2, \ldots$. Mitu tippu ja mitu lehte on täielikus kahendpuus, mille viimase taseme number on h? Tõestada need valemid.
3. Eeldades, et kahendpuu T igal tipul leidub väli $.x$ (ehk x-väli),
   (a) kirjutada algoritm, mis T iga tipu x-väljale omistab etteantud konstandi;
   (b) kirjutada algoritm, mis T iga tipu x-väljale omistab sellest tipust algava kahendpuu tippude arvu;
   (c) kirjutada algoritm, mis T iga tipu x-väljale omistab selle tipu tasemenumbri puus T; ✓

(d) kirjutada algoritm, mis T iga tipu x-väljale omistab sellest tipust algava kahendpuu juure astme (0, 1 või 2);

(e) kirjutada algoritm, mis T iga tipu x-väljale omistab sellest tipust algava kahendpuu juure iseloomustuse: 0 – alluvateta, 1 – ainult vasak alluv, 2 – ainult parem alluv, või 12 – mõlemad alluvad;

(f) kirjutada algoritm, mis T iga tipu x-väljale omistab sellest tipust algava kahendpuu kõrguse puus T;

(g) kirjutada algoritm, mis leiab antud tipust algava kahendpuu kõrguse puus T (x-välju kasutamata). ✓

4. Eeldades, et puu P igal tipul leidub väli $.x$ (ehk x-väli)

   (a) kirjutada algoritm, mis iga tipu x-väljale omistab sellest tipust algava alampuu tippude arvu; ✓

   (b) kirjutada algoritm, mis iga tipu x-väljale omistab selle tipu tasemenumbri puus P;

   (c) kirjutada algoritm, mis iga tipu x-väljale omistab sellest tipust algava alampuu juure astme, st. tema alampuude arvu puus P;

   (d) kirjutada algoritm, mis iga tipu x-väljale omistab sellest tipust algava alampuu kõrguse puus P.

5. Kirjutada algoritm, mis antud kahendpuu korral konstrueerib vastava tavapuude metsa.

6. Kirjutada algoritm, mis antud metsa korral konstrueerib vastava kahendpuu.

7. Puu P on antud vastava kahendpuuna T;

   (a) kirjutada algoritm, mis iga tipu x-väljale omistab selle tipu tasemenumbri puus P; ✓

   (b) kirjutada algoritm, mis iga tipu x-väljale omistab sellest tipust algava alampuu juure astme, st. tema alampuude arvu puus P;

   (c) kirjutada algoritm, mis iga tipu x-väljale omistab sellest tipust algava alampuu kõrguse puus P; ✓

   (d) kirjutada algoritm, mis iga tipu x-väljale omistab tippude arvu sellest tipust algavas puu P alampuus:

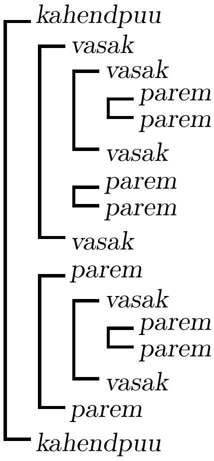
   (e) kirjutada algoritm, mis iga tipu x-väljale omistab sõlmede (ehk sisetippude) arvu sellest tipust algavas puu P alampuus:

(f) kirjutada algoritm, mis iga tipu x-väljale omistab lehtede arvu sellest tipust algavas puu P alampuus.

8. Kirjutada puu joonistamise algoritm, kui puu on antud vastava kahendpuuna. Eeldatakse, et iga tipu (t) korral on antud selle tipu koordinaadid joonisel, vastavalt väljadel $t.x$ ja $t.y$; olemas on protseduurid $joonistadaTipp(t)$ ja $joonistadaServ(t, t')$.

§ 4. Kahendotsimise puu

1. Konstrueerida võimalikult madal kahendpuu, mille tippudeks on sümbolid Teie ees- ja perekonnanimest ning mille tippude keskjärjestus annaks Teie ees- ja perekonnanime.
2. Lisada eelmises ülesandes konstrueeritud kahendpuu tippudele paarikaupa erinevad arvulised võtmed nii, et moodustuks kahendotsimise puu nende võtmete järgi. Joonistada kahendotsimise puu, mis on saadud sellest puust juurtipu eemaldamisel.
3. Kirjutada algoritm kontrollimaks, kas antud kahendpuu on kahendotsimise puu. ✓
4. Kirjutada algoritm kontrollimaks, kas antud kahendpuu on AVL-puu. ✓
5. Joonistada
   (a) 11-tipuline AVL-puu;
   (b) AVL-puu, mis saadakse sellest pärast juurtipus oleva kirje eemaldamist (vajadusel tasakaalustamist rakendades).
6. Joonisel 3 kujutatud kahendpuu struktuurile vastavalt
   (a) joonistada see kahendpuu;
   (b) lisada üks tipp ja kirjutada igasse tippu võtmeväärtus nii, et saadud puu T' oleks AVL-puu;
   (c) joonistada kahendotsimise puu T'', mis on saadud puust T' juurtipu kirje eemaldamisel;
   (d) tasakaalustada puu T''.
7. Esitada konkreetne näide AVL-puu tasakaalustamisest kahekordse pöördega.
8. AVL-puust tipu eemaldamisel saadi tasakaalustamata puu, mille kõik lehed on samal tasemel;
   (a) joonistada sellise puu näide;



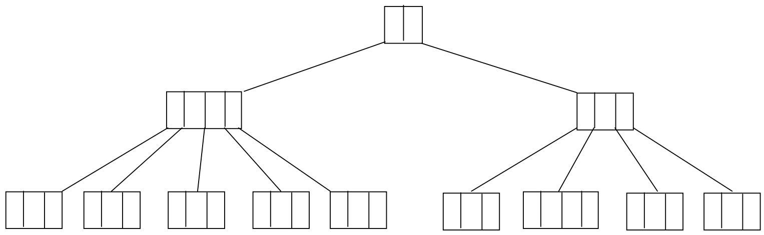
Joonis 3. Kahendpuu hierarhilise struktuurina.

   (b) tasakaalustada see puu;
   (c) joonistada saadud kahendpuule vastav harilik puu (mets).

9. AVL-puust tipu eemaldamisel saadi tasakaalustamata puu, mille kõik lehed on eri tasemetel;
   (a) joonistada sellise puu näide;
   (b) tasakaalustada see puu;
   (c) joonistada saadud kahendpuule vastav harilik puu (mets).

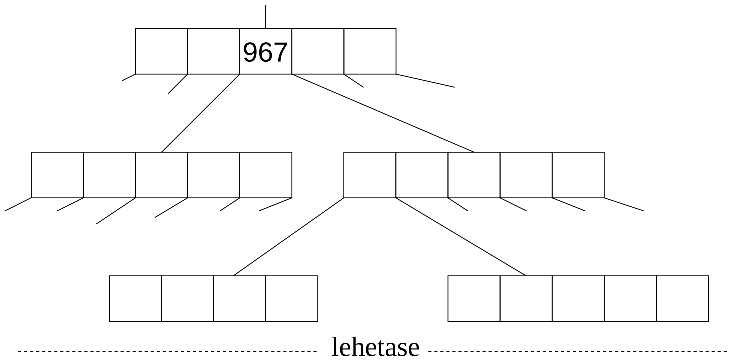
§ 5. B-puu

1. Mitu kirjet minimaalselt ja mitu kirjet maksimaalselt saab olla kolmandat järku B-puu
   (a) juurtipus;
   (b) vahetipus?
2. Mitu kirjet minimaalselt ja mitu kirjet maksimaalselt saab olla üheksandat järku B-puu
   (a) juurtipus;
   (b) vahetipus?
3. Konstrueerida viietipulise seitsmendat järku B-puu näide. ✓
4. Joonisel 4 kujutatud B-puu struktuuris täita tühjad võtmekohad sobivate väärtustega. Mis võiks olla selle B-puu järk?



Joonis 4. B-puu näitestruktuur.

5. Joonisel 5 kujutatud üheksandat järku B-puu fragmendis
   (a) täita tühjad võtmekohad sobivate väärtustega;
   (b) kirjeldada, kuidas eemaldatakse kirje võtmega 967;
   (c) joonistada pärast eemaldamist saadud fragment.



Joonis 5. B-puu fragment.

§ 6. Järjestikpaigutus ja seotud paigutus

1. Esitada magasini realisatsioon
   (a) massiivina;
   (b) päisega lihtahelana.
2. Esitada järjekorra realisatsioon

(a) massiivina;

(b) päisega lihtahelana.

3. Esitada kolme muutujaga hulkliikme realisatsioon päisega ringahelana, kus põhioperatsiooniks on hulkliikmete liitmine.
4. Kirjeldada "hõreda" maatriksi kujutamist ristahelatena. Operatsioonideks on ridade liitmine ja veergude liitmine. ✓

§ 7. Paisksalvestus

1. Leidke oma kaasüliõpilaste matriklinumbrite seast selline matriklinumber k'', mis annab kollisiooni Teie enda matriklinumbriga k', kui paiskfunktsiooniks on

   (a) $h(k) = k \mod 503$;

   (b) $h(k) = \lfloor k^2/2^{13} \rfloor \mod 503$.

2. Joonistada 7-realine paisktabel pärast tabelis antud kirjete lisamist topeltpaiskamise meetodil (tabelis toodud järjekorras), kasutades paiskfunktsiooni h ja teisese paiskfunktsiooni h_1 tabelis antud väärtusi.

Nr.	k	$info$	$h(k)$	$h_1(k)$
1.	785	Mart Must	1	6
2.	750	Sulev Sinine	0	1
3.	613	Peeter Punane	3	2
4.	801	Vello Valge	1	5
5.	730	Rein Roosa	3	2

§ 8. Kahendkuhi

1. Kirjutada algoritm kontrollimaks, kas antud kirjete massiiv on kahendkuhi. ✓
2. Kirjutada rekursiivne algoritm antud massiivi kuhjastamiseks ülesviimist (*viia_üles* [1], lk. 55) kasutades.
3. Kirjutada mitterekursiivne algoritm antud massiivi kuhjastamiseks ülesviimist kasutades.
4. Kirjutada rekursiivne algoritm kahendkuhjana antud massiivi mittekahanevalt sorteerimiseks.
5. Kirjutada algoritm sellise kirje võtmiseks kahendkuhjast, mille võti on suuruselt teine.

6. Kirjutada algoritm järgmise ülesande lahendamiseks: kui antud kahendkuhi ei ole täielik kahendpuu, siis eemaldada viimasel tasemel olevad tipud.
7. Kirjutada algoritm järgmise ülesande lahendamiseks: eemaldada antud kahendkuhjast tipud, mis asuvad $\lfloor h/3 \rfloor$ viimasel tasemel, kus h on kahendkuhja kõrgus.

§ 9. Binomiaalpuu. Binomiaalkuhi

1. Mitu tippu on binomiaalpuus, mille kõrgus on 8?
2. Mitmendat järku binomiaalpuudest koosneb 168-kirjeline binomiaalkuhi?
3. Arvutage n=(Teie sünniaasta)+(Teie sünnikuu)+ (Teie sünnipäev kuus). Millist järku binomiaalpuud esinevad n-kirjelises binomiaalkuhjas?
4. Joonistada binomiaalkuhi,
   (a) milles on kirjed võtmetega $1, 2, \ldots, 23$;
   (b) milles on kirjed võtmetega $1, 2, \ldots, 26$;
   (c) mis saadakse eelmises alamülesandes (b) tehtud kuhjast suurima puu juurtipu eemaldamisel.
5. Joonistada binomiaalkuhi, milles on 13 kirjet ja vähima võtmeväärtusega kirje asub juurahela suurima astmega tipus;
   (a) kirjeldada, kuidas toimub vähima võtmeväärtusega kirje võtmine sellest binomiaalkuhjast;
   (b) joonistada pärast vähima võtmeväärtusega kirje võtmist saadud binomiaalkuhi.

§ 10. Sorteerimine

1. Kirjutada pistemeetodi rekursiivne algoritm.
2. Olemas on algoritm

   piste(k, m)

   - - - Antud: (globaalsena) massiiv $a_1, a_2, \ldots, a_n$;
   indeksid k ja m $(1 \leqslant k \leqslant m \leqslant n)$

   - - - Tulemus: element a_m pistetud kohale k massiivis a

(a) kirjutada kahendpistemeetodi rekursiivne algoritm, mis kasutab alamalgoritmi *piste*() ja milles pistekoha otsimine toimub vastava funktsiooni abil;

(b) kirjutada vastav pistekoha otsimise mitterekursiivne algoritm;

(c) kirjutada vastav pistekoha otsimise rekursiivne algoritm.

3. Üliõpilaste arv teaduskonnas on $n < 500$. Üliõpilaste kirjed tuleb sorteerida võtme (1000×(sünniaasta -1900)+ 100×sünnikuu + (sünnipäev kuus)) järgi. Kumb sorteerimismeetoditest, kas loendamismeetod või kimbumeetod, on eelistatavam? Miks?
4. Ühe aastakäigu kutsealuste arv on $n > 5000$. Kutsealuste kirjed tuleb sorteerida võtme (100×sünnikuu + (sünnipäev kuus)) järgi. Kumb sorteerimismeetoditest, kas loendamismeetod või kimbumeetod, on eelistatavam? Miks?

§ 11. Sõnetöötlus

1. Leida prefiksfunktsiooni väärtused sõne $s = 'aaaaab'$ jaoks.
2. Konstrueerida sõne, mille prefiksfunktsiooni väärtusteks on 0 1 0 1 2 3 0 1 2 3 4 0.
3. Loendada, mitu korda võrreldakse sümboleid sõne

   $s = 'aabcaabcd'$

   esinemise otsimisel tekstis

   $t = 'aabcaabcaaabcaabcdbaaaaaabab'$

   (a) Knuth-Morris-Pratti algoritmis (prefiksfunktsiooni leidmist arvestamata);

   (b) lihtsas otsimisalgoritmis ([1], joonis 5.1).

   Soovitus: teksti t alla järgnevatesse ridadesse kirjutage otsisõne s sobiva nihkega; otsisõnes kriipsutage alla vastaval sammul võrdlemisele tulevad sümbolid.
4. Antud sõnepaaride s ja t jaoks konstrueerida maatriks c vastavalt algoritmile *kavandada* ([1], lk. 90). Kriipsutada alla c elemendid, mis vaadatakse läbi nende sõnede pikima ühissõne leimisel.

   (a) $s = 'aabcbdba'$; $t = 'abbdcbba'$.

   (b) $s = 'aabcbdbaa'$; $t = 'abbdcbba'$.

   (c) $s = 'aabcbdbaab'$; $t = 'abbdcbab'$.

(d) Sõne s saamiseks kirjutage oma ees- ja perekonnanimi järjestikku, ilma tühikuteta ja ainult väiketähtedega, ning võtke sellest 10 esimest tähte. Kui tähti oli vähem kui 10, siis lisage lõppu vajalik arv algusest võetud tähti. Näiteks, *Jüri Kiho* jaoks $s =$ *'jürikihojü'*.
Sõne t saate sõne s transponeerimisel (sõne s tähed vastupidises järjekorras). Näiteks, *Jüri Kiho* jaoks
$t =$ *'üjohikirüj'*.

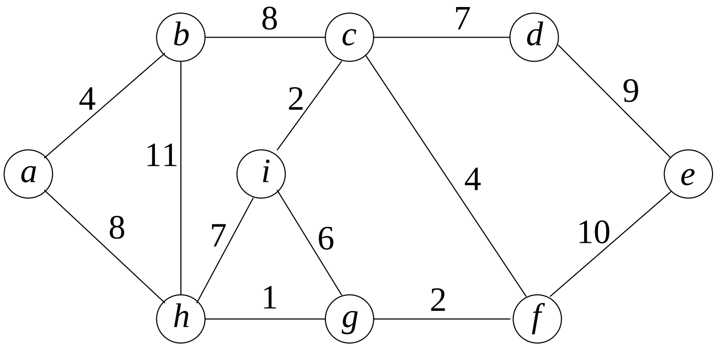
§ 12. Graafitöötlus

1. Kirjutada graafi sügavuti läbimisel põhinev algoritm, mis leiab kõik tipud antud graafis $G = (V, E)$, mis on saavutatavad antud tipust $a \in V$. ✓
2. Sõnastada üks originaalne graafitöötlusülesanne, mille lahendamine võiks põhineda graafi sügavuti läbimisel.
3. Kirjutada eelmises ülesandes sõnastatud graafitöötlusülesande lahendusalgoritm.
4. Kirjutada graafi sügavuti läbimise skeemil põhinev algoritm antud tippude a ja b vahelise ühe tee väljastamiseks (st. teel esinevate tippude nimede trükkimiseks).
5. Kirjutada graafi sügavuti läbimise skeemil põhinev algoritm antud tippude a ja b vaheliste kõigi teede väljastamiseks (st. nendel teedel esinevate tippude nimede trükkimiseks).
6. Olgu antud geograafiliste punktide graaf, mille iga kaarega (x, y) on seotud väli $(x, y).l$ – langus liikumisel punktist x punkti y. *Ohutuimaks teeks* kahe punkti vahel nimetatakse sellist teed, millel suurim lokaalne langus (punktist järgmisse punkti) on minimaalne. Kirjutada graafi laiuti läbimisel põhinev algoritm, mille käigus leitakse ohutuim tee antud punktist a antud punkti b.
7. Olgu antud geograafiliste punktide graaf, mille iga tipuga on seotud väli $.h$ – vastava punkti kõrgus merepinnast.
   (a) Kirjutada graafi laiuti läbimisel põhinev algoritm, mille käigus leitakse selline tee antud punktist a antud punkti b, mis läheb läbi võimalikult kõrge punkti.

(b) Kirjutada graafi sügavuti läbimisel põhinev rekursiivne algoritm, mis kontrollib, kas kahe antud punkti vahel leidub samakõrgustee, st. tee, millel asuvad punktid on kõik ühe ja sama kõrgusega.

(c) Kirjutada graafi laiuti läbimisel põhinev algoritm leidmaks sellist teed antud punktist a antud punkti b, mille kõrgeim tipp on võimalikult madalal.

8. Leida joonisel 6 kujutatud graafi minimaalne toes Kruskali meetodil, kirjutades seejuures üles ka igal sammul tekkivad tipuklassid.



Joonis 6. Servamaksumustega graaf.

§ 13. Planimeetria

1. Kirjutada algoritmid kontrollimaks punkti kuuluvust
   (a) kolmnurka; ✓
   (b) nelinurka.

2. Eeldusel, et on olemas kontrollimisalgoritm
   $$kuulubKolmnurka(punkt, A, B, C)$$
   kus A, B, C on kolmnurga tipupunktid,

(a) kirjutage algoritm punkti nelinurka kuulumise kontrollimiseks (nelinurk ei pruugi olla kumer);

(b) kirjutage lineaarse keerukusega algoritm punkti kumerasse hulknurka kuulumise kontrollimiseks.

3. Kirjutada algoritm järgmise ülesande lahendamiseks.
   Antud on neli punkti ($p1$, $p2$, $p3$, $p4$) tasandil. Analüüsida tuleb kinnist murdjoont $M = p1$-$p2$-$p3$-$p4$-$p1$.
   Tulemus: tagastatakse
   - 0, kui M ei ole nelinurk,
   - 1, kui M on mittekumer nelinurk,
   - 2, kui M on trapets,
   - 3, kui M on rööpkülik.

4. Kirjutada lihtne ($O(n)$) algoritm, mis antud punktihulgast elimineerib kumerasse kattesse mittekuuluvaid punkte.

5. Kirjutada detailne algoritm kontrollimaks punkti kuuluvust hulknurka, kui hulknurga tipud on antud r i n g a h e l a n a (loomulikus järjekorras).

lõikuvad(p, q)
- - - *Antud: lõik otspunktidega p ja q ning (globaalsena) horisontaalne kontrolllõik otspunktidega t ja* $(1000, t.y)$
- - - *Tulemus: tagastatakse 1, kui lõigul $p \dots q$ ja kontrolllõigul leidub ühine punkt; vastasel korral tagastatakse 0*

$p.y = q.y$ **?**
- - - *lõik $p \dots q$ on horisontaalne (või $p = q$)*
$\longleftarrow$ $(p.y = t.y)$
- - - *lõik $p \dots q$ ei ole horisontaalne ja asub sirgel s võrrandiga*
- - - $X := (p.x - q.x) \times (Y - q.y)/(p.y - q.y) + q.x$
$k := (p.x - q.x) \times (t.y - q.y)/(p.y - q.y) + q.x$
- - - *k on lõikudele vastavate sirgete lõikepunkti abstsiss, st.*
- - - *sirgel s asuva sellise punkti abstsiss, mille ordinaat on $t.y$*
$\longleftarrow$ $(t.x \leqslant k \leqslant 1000 \wedge (p.x \leqslant k \leqslant q.x \vee q.x \leqslant k \leqslant p.x))$

Joonis 7. Lõikumise erikontroll.

6. Olgu punkti hulknurka kuulumise praktilise ülesande korral teada, et vaadeldavate punktide koordinaadid ei ole absoluutväärtuselt suuremad kui 999. Kas algoritmis *punkt_ hulknurgas* ([1], lk. 117) võib siis alamalgoritmi *lõikumine* asemel kasutada joonisel 7 toodud algoritmi *lõikuvad*? Põhjendada oma arvamust.

§ 14. Algoritmi korrektsus

1. Leida omistamisdirektiivi nõrgim eeltingimus P joonisel 8.

- - - P
$x := x^2 - y$
- - - $x < y$

Joonis 8. Omistamisdirektiiv.

2. Tõestada joonistel 9 – 13 leiduvate algoritmide korrektsus.

- - - $n > 5$
$s := 0.5$
$k := 2$
$k < n$ **?**
$m := k + 1$
$s := s + \dfrac{1}{km}$
$k := m$
- - - $s = 1 - \dfrac{1}{n}$

Joonis 9. Lihtmurdude summa.

- - - $\sqrt[3]{3} \leqslant n^m \leqslant \sqrt[2]{3}$
$m := 1 - m$
$n := n + 1$
$\alpha := 1 - 2n$
$b := 1 + \cos^2 \alpha$
$b := b - \log_3 (n^2 + \alpha) + sin^2\alpha - 2$
$x := \log_3 (n^2 + \alpha)^2$
$y := mb$
$x := x + 2y$
$x := x/2$
- - - $0,5 \leqslant x \leqslant 1$

Joonis 10. Lineaarne algoritm.

- - - $0 \leqslant k < n$
$m := k + 1$
$t := m + 1$
$t \leqslant n$ **?**
$m := m * t$
$t := t + 1$
- - - $m = \dfrac{n!}{k!}$

Joonis 12. Faktoriaalide jagatis.

- - - $0,5 \leqslant a \leqslant 0,7$ *ja*
- - - $0,5 \leqslant \log_3 n \leqslant 0,7$
$n := n + 1$
$\alpha := 1 - 2n$
$a := 1 - a$
$b := 2 + \cos^2 \alpha$
$b := b - \log_3 (n^2 + \alpha) + sin^2\alpha - 3$
$x := \log_3 (n^2 + \alpha)^2$
$y := ab$
$x := x + 2y$
$x := x/2$
- - - $0,5 \leqslant x \leqslant 1$

Joonis 11. Lineaarne algoritm.

- - - $n > 5$
$m := 6$
$t := 7$
$t \leqslant n$ **?**
$m := m * t$
$t := t + 1$
$m := m \times 120$
- - - $m = n!$

Joonis 13. Suurem faktoriaal.

§ 15. Varia

1. Leida valem, mille järgi arvutada joonisel 14 kujutatud n-kordse tsükli ajaline keerukus antud m ja n korral (eeldades, et tsükli sisu S ajaline keerukus on $O(1)$). ✓

$$
\begin{array}{l}
\star\ i_1 = 1, 2, \ldots, m \\
\star\ i_2 = 1, 2, \ldots, i_1 \\
\star\ \ldots \\
\star\ i_{n-1} = 1, 2, \ldots, i_{n-2} \\
\star\ i_n = 1, 2, \ldots, i_{n-1} \\
\quad S
\end{array}
$$

Joonis 14. Mitmekordsete tsüklite pere üldkuju.

2. Automaat koostab ühikruutudest malelauda mõõtmetega $2^n \times 2^n$ nii, et teeb valmis 4 võrdset värvitud ruudustikku ja seejärel paneb need kokku. On teada, et nii ühe ühikruudu värvimine kui ka ruuduplokkide kokkutõstmine võtab ühe ajaühiku. Kui kaua aega kulub terve malelaua valmistamiseks?
3. On teada, et 5 objekti järjestamiseks piisab 7 võrdlusest, kuigi tavalised algoritmid kulutavad 8. Konstrueerida algoritm (otsustuste puu), mis suudab 5 objekti sorteerida 7 võrdlemisega.
4. Defineerida tõenäosuslik algoritm.
5. Millistest osadest koosneb andmestruktuuri algebraline spetsifikatsioon?

Vastuseid

§1. Ülesanne 9a

Tugineme O-relatsiooni omadustele $1^0, 2^0, 3^0, 4^0, 6^0, 8^0$ ([1], lk. 15).
Esimene liidetav:
$1^0 \Rightarrow 2n^2$ on $O(n^2)$.
Teine liidetav:
$1^0 \Rightarrow 3(\log n)^3$ on $O(\log n)^3$.
$8^0 \Rightarrow \log n$ on $O(\sqrt[3]{n})$.
$6^0 \Rightarrow (\log n)^3 = (\log n \log n \log n)$ on $O(n)$.
$4^0 \Rightarrow n$ on $O(n^2)$.
Kokkuvõttes,
$3^0 \Rightarrow 3(\log n)^3$ on $O(n^2)$.
Mõlemad liidetavad on $O(n^2)$, seega
$2^0 \Rightarrow 2n^2 + 3(\log n)^3$ on $O(n^2)$.
Teiselt poolt, n^2 on $O(2n^2 + 3(\log n)^3)$,
sest $n^2 \leqslant 2n^2 + 3(\log n)^3$ iga $n \geqslant 0$ korral.
Seega $2n^2 + 3(\log n)^3$ on $\Theta(n^2)$.

§2. Ülesanne 3

Nõuetekohane algoritm on esitatud joonisel 15.
Olgu $T(n)$ operatsioonide arv, mis sooritatakse n-elemendilise järjendi elementide summa arvutamisel selle algoritmi järgi. Tulemuse (s) arvutamiseks (kui $n > 1$) rakendatakse sama algoritmi neli korda (neli korda lühemate osade korral). Kõigi ülejäänud operatsioonide koguarv on aga $O(1)$. Seega
$T(n) = 4T(n/4) + f$, kus f on $O(1)$.
Täidetuks osutub põhiteoreemi ([1], lk. 20) esimese väite eeldus: võttes $a = b = 4$ ning $\varepsilon = 1$, saame $\log_4 4 = 1$, $n^{\log_b a - \varepsilon} = n^0 = 1$; seega f on $O(n^{\log_b a - \varepsilon})$. Põhiteoreemi esimese väite ($T(n)$ on $\Theta(n^{\log_b a})$) kohaselt antud juhul $T(n)$ on $\Theta(n)$.

§3. Ülesanne 3c

Joonisel 16 olev algoritm saab algandmetena nii kahendpuu kui ka juurele omistatava tasemenumbri. Näiteid selle rakendamisest konkreetse kahendpuu T_0 tippudele tasemenumbrite omistamiseks:
$tasemed(T_0, 0)$; - - - juure tasemeks loetakse 0;
$tasemed(T_0, 1)$; - - - juure tasemeks loetakse 1.

$summa(a, x, u)$

--- *Antud: järjend a ja selle kahe elemendi indeksid x ning u*
--- *Tulemus: arvutatakse ja tagastatakse $a_x + a_{x+1} + \ldots + a_u$*
$n := u - x + 1$ --- *liidetavate arv*
$n = 0$ **?**
$\longleftarrow$ (0)
$n = 1$ **?**
$\longleftarrow$ (a_x)
--- *kolm "vahepunkti":*
$z := x + \lfloor n/2 \rfloor$
$y := x + \lfloor n/4 \rfloor$
$w := u - \lfloor n/4 \rfloor$
--- *tulemus nelja osa summade summana:*
$s := summa(a, x, y - 1) + summa(a, y, z - 1) +$
$+ summa(a, z, w - 1) + summa(a, w, u)$
$\longleftarrow$ (s)

Joonis 15. Järjendi summa.

$tasemed(T, m)$
--- *Antud: kahendpuu T*
--- *ja selle juure soovitav tasemenumber m*
--- *Tulemus: T iga tipu x-väljale omistatud vastav tasemenumber*

T ei ole tühi **?**
$T.juur.x := m$
$tasemed(T_v, m + 1)$
$tasemed(T_p, m + 1)$

Joonis 16. Tasemenumbrite omistamine.

§3. Ülesanne 3g

Algoritm on esitatud joonisel 17.

$k\tilde{o}rgus(T)$
- - - *Antud: kahendpuu T*
- - - *Tulemus: tagastatakse antud kahendpuu kõrgus*

T on tühi **?**
$\leftarrow$ (0)
$\leftarrow$ $(\max(k\tilde{o}rgus(T_v), k\tilde{o}rgus(T_p)) + 1)$

Joonis 17. Kahendpuu kõrguse leidmine.

§3. Ülesanne 4a

Algoritm on esitatud joonisel 18.

$tippudeArv(P)$
- - - *Antud: mittetühi puu P*
- - - *Tulemus: puu P iga tipu x-väljale omistatud sellest*
- - - *tipust algava alampuu tippude arv*

$P.juur.x := 1$

$\star$ puu P juure iga alampuu P' korral

$tippudeArv(P')$
$P.juur.x := P.juur.x + P'.juur.x$

Joonis 18. Tippude arv puus.

§3. Ülesanne 7a

Algoritm on esitatud joonisel 19.

§3. Ülesanne 7c

Algoritm on esitatud joonisel 20. Paneme tähele, et kui $u.x = 1$ enne algoritmi rakendamist tipu u alampuudele (st. esimesele alampuule),

$tasemedPuus(T, m)$
--- *Antud: kahendpuu T ja selle juure soovitav tasemenumber m*
--- *Tulemus: iga tipu x-väljale omistatud selle tipu tasemenumber*
--- *kahendpuule T vastavas tavapuus*

T ei ole tühi **?**
$T.juur.x := m$
$tasemedPuus(T_v, m + 1)$
$tasemedPuus(T_p, m)$

Joonis 19. Tasemed kahendpuus.

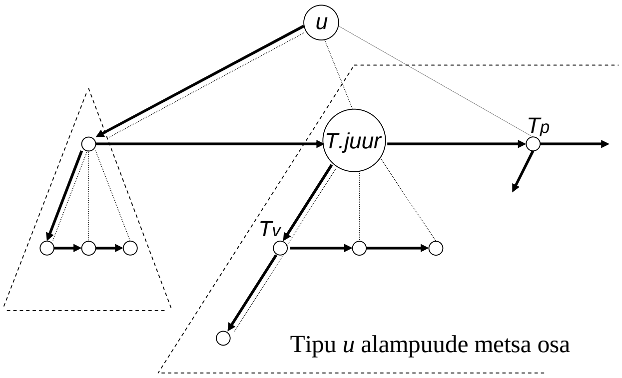
siis pärast algoritmi täitmist on välja $u.x$ väärtuseks tipust u algava tavapuu kõrgus (tipu u alampuude kõrgustest suurim pluss 1).

Antud ülesande lahendamiseks konkreetse kahendpuu T_0 korral tuleb algoritmi rakendada fiktiivset tippu kasutades:
$k\tilde{o}rgusedPuus(T_0, fiktiivneTipp)$

$k\tilde{o}rgusedPuus(T, u)$
--- *Antud: mingi tavapuu tipu u alampuude metsa (osa)*
--- *kujutav kahendpuu T (vt. joonis 21);*
--- *Tulemus: 1) T iga tipu x-väljale omistatud sellest tipust*
--- *algava alampuu kõrgus tavapuus;*
--- *2) tingimuslikult muudetud välja $u.x$ väärtus*

T ei ole tühi **?**
$T.juur.x := 1$
$k\tilde{o}rgusedPuus(T_v, T.juur)$
[? $(T.juur.x + 1 > u.x)$ $u.x := T.juur.x + 1$
$k\tilde{o}rgusedPuus(T_p, u)$

Joonis 20. Kahendpuuna antud tavapuu kõrguse leidmine.



Joonis 21. Puu ja vastav kahendpuu.

§4. Ülesanne 3

Algoritm on esitatud joonisel 22. Antud kahendpuu läbitakse keskjärjestuses, kontrollides, et ettetulevate tippude võtmed (x-väljade väärtused) ei kahaneks.

§4. Ülesanne 4

Algoritm on esitatud joonisel 23.

§5. Ülesanne 3

Sobiv B-puu on esitatud joonisel 24.

§6. Ülesanne 4

I. Mõisted.
Hõredas maatriksis on rõhuva enamuse elementide väärtuseks null (vt. nt. joonis 25).
Ridade liitmise operatsiooni $liitaRida(k, l)$ all mõistame (globaalsena antud) maatriksi k-nda rea igale elemendile l-nda rea vastavast vee-

```
onKahendotsimisePuu(T)
  --- Antud: kahendpuu T ja globaalsena muutuja xEelmine,
  ---        mille väärtuseks on T juurele keskjärjestuses eel-
  ---        neva tipu võti  (esimesel rakendamisel minimaal-
  ---        ne, st. −∞)
  --- Tulemus: tagastatakse 1, kui T on kahendotsimise puu,
  ---          0 vastasel korral

 T on tühi ?
<—— (1)
 ¬onKahendotsimisePuu(T_v) ?
<—— (0)
   töödelda juur
  T.juur.x < xEelmine ?
<——— (0)
   xEelmine := T.juur.x
   töödelda juur
 ¬onKahendotsimisePuu(T_p) ?
<—— (0)
<—— (1)
```

Joonis 22. Kahendotsimise puu kontroll.

rust võetud väärtuse liitmist. Nt. pärast operatsiooni $liitaRead(1,4)$ rakendamist maatriksile joonisel 25 on selle esimeseks reaks

9 0 0 3 0 0 0 0 0 0 0 0 0 0 0.

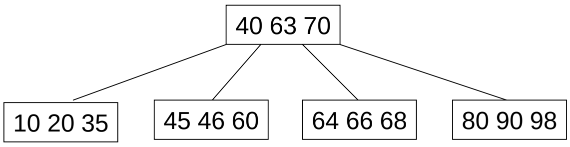
Veergude liitmine defineeritakse analoogselt (seda operatsiooni käesolevas näidislahenduses täpsemalt ei käsitleta).

II. Andmete kujutusviis.

Kasutatakse päisega ringahelaid; ahelate lülideks on maatriksi nullist erinevad elemendid (vt. joonis 25). Iga lüli kuulub kahte ahelasse, nii rea- kui ka veeruahelasse. Reaahela lülid on seotud viidavälja r kaudu, veeruahela lülid aga viidavälja v kaudu. Peale viidaväljade on lülis veel elemendi väärtuseväli (a) ning elemendi indeksite väljad (i ja j). Eeldades, et maatriksi mõõtmed (tihti) ei muutu, on otstarbekohane

onAVLpuu(T)
--- *Antud: kahendpuu T*
--- *Tulemus: tagastatakse 1, kui T on AVL puu,*
--- *0 vastasel korral*

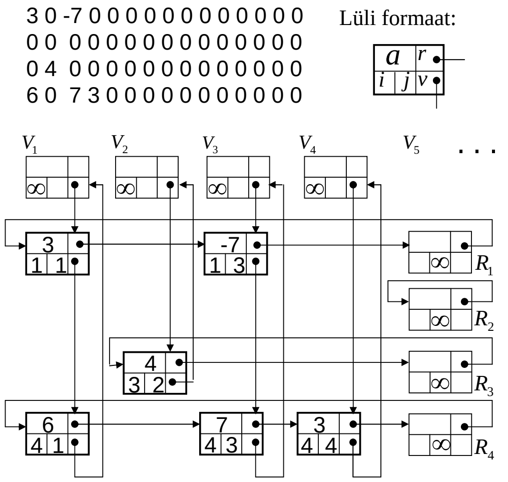
T on tühi **?**
$\longleftarrow$ (1)
$xEelmine := -\infty$
$\neg onKahendotsimisePuu(T)$ **?** --- *vt. joonis 22*
$\longleftarrow$ (0)
$\neg onAVLpuu(T_v)$ **?**
$\longleftarrow$ (0)
$\neg onAVLpuu(T_p)$ **?**
$\longleftarrow$ (0)
$\longleftarrow$ $(|kõrgus(T_v) - kõrgus(T_p)| < 2)$ --- *vt. joonis 17*



Joonis 24. Viietipulise seitsmendat järku B-puu näide.

ahelate päised paigutada järjestikuselt: reaahelate päiste massiivina (R) ja veeruahelate päiste massiivina (V). Reaahela päises leiavad kasutamist viidaväli (r) ja veeruindeksi väli (j); viimase väärtuseks on sobiv määrata suur arv (∞), mis ei saa olla tegelikuks veeruindeksiks. Veeruahela päises leiavad kasutamist viidaväli (v) ja reaindeksi väli (i); viimase väärtuseks on sobiv määrata suur arv (∞), mis ei saa olla tegelikuks reaindeksiks.
III. Operatsiooni realisatsioon.
Ülalkirjeldatud andmekujutusele vastavalt realiseerib ridade liitmise algoritm joonisel 26. Algoritmi täpsem spetsifikatsioon on järgmine:

- - - Antud: globaalsena maatriks ristahelates, vt. joonis 25
- - - ja maatriksi kahe rea numbrid k, l
- - - Tulemus: maatriksi k-nda rea elementidele liidetud l-nda
- - rea elemendid



Joonis 25. Hõre maatriks ja selle esitus ristahelatena.

§8. Ülesanne 1

Algoritm on esitatud joonisel 27.

§12. Ülesanne 1

Algoritm on esitatud joonisel 28.

§13. Ülesanne 1a

Algoritm on esitatud joonisel 29. Kasutatakse alamalgoritmi *vp* ([1], lk. 115).

$liitaRida(k, l)$

- - - *globaalsena antud maatrikisi l-nda rea liitmine k-ndale*

$pp := R_k$; $p := pp.r$; - - - *pp, p – jooksev viidatandem real k*

$q := R_l$; - - - *q – jooksev viit real l*

$q := q.r$; - - - *q edasi (võtta uus liidetav)*

$q.j = \infty$ **?**

$q.j < p.j$ **?**

$s :=$ <uus lüli, kus $s.a = q.a$, $s.i = k$, $s.j = q.j$ >

$pp.r = s$; $s.r = p$ - - - *s lülitada k-nda rea ahelasse*

- - - *s lülitada veeru $q.j$ ahelasse:*

[$\star$ ($t := V_{q.j}$; $t.v.i \leqslant k$; $t := t.v$)

$tt := t.v$; $t.v := s$; $s.v := tt$; - - - *s pannakse t järele*

$pp := s$

$q.j = p.j$ **?**

$p.a := p.a + q.a$

$p.a = 0$ **?**

- - - *elimineerida null-väärtusega lüli p:*

$pp.r := p.r$; - - - *reaahelast*

[$\star$ ($t := V_{p.j}$; $t.v \neq p$; $t := t.v$)

$t.v := t.v.v$; - - - *veeruahelast*

$pp = p$

$p := p.r$; - - - *p edasi*

- - - $q.j > p.j$

$pp = p$; $p = p.r$; - - - *pp, p edasi*

Joonis 26. Ristahelates maatriksi ühe rea liitmine teisele reale.

onKahendkuhi()
--- *Antud: massiiv a (globaalsena)*
--- *Tulemus: tagastatakse 1, kui a on kahendkuhi,*
--- *0 vastasel korral*

$\star\ i = 2, \ldots, n$

$a_i > a_{ülemus(i)}$ **?**

$\leftarrow$ (0)

$\leftarrow$ (1)

Joonis 27. Kahendkuhja kontroll.

§15. Ülesanne 1

Antud mitmekordse tsükli ajaline keerukus avaldub binoomkordajana $C(m + n - 1, n)$.

```
saavutatavus(G, v)
 - - - Antud: orienteeritud graaf G ja selle tipp v, mis on
 - - -        saavutatav mingist fikseeritud lähtetipust a;
 - - -        graafi G mõne tipu korral on juba leitud
 - - -        saavutatavus tipust a: nendel tippudel on
 - - -        x-välja väärtuseks 1, ülejäänutel (veel) 0
 - - - Tulemus: graafi G kõigi tippude korral määratud
 - - -          saavutatavus, st. iga tipu x-välja
 - - -          väärtuseks omistatud saavutatavuse
 - - -          tunnus (1 või 0)

 v.x := 1
 ⋆ ∀w, w ∈ Gv   - - - tipu v iga naabri w korral
   w.x = 0 ?
     saavutatavus(G, w);
```

Joonis 28. Saavutatavus graafis.

```
kuulubKolmnurka(p, A, B, C)
 - - - Antud: punkt p ja kolmnurk tippudega A, B, C
 - - - Tulemus: vastus küsimusele
 - - -          kas antud punkt kuulub antud kolmnurka

 vastus := lõikumine(p, p, A, B) ∨ lõikumine(p, p, B, C) ∨
           lõikumine(p, p, C, A) ∨
           (vp(p, A, B) = vp(p, B, C) = vp(p, C, A))
←— (vastus)
```

Joonis 29. Punkti kolmnurka kuulumise kontroll.

KIRJANDUS

[1] J. Kiho. *Algoritmid ja andmestruktuurid.* Kolmas, parandatud ja täiendatud trükk. TÜ, 2003, 147 lk.