

Praktikum 8

Kahendotsimispuud

Harjutusülesanded

- KOP_1. Kahendotsimispuu valmistamine
- KOP_2. Kahendotsimispuu kontroll
- KOP_3. AVL-puu kontroll
- KOP_4. Otsimine kahendotsimispuus
- KOP_5. Lisamine kahendotsimispuusse
- KOP_6. Eemaldamine kahendotsimispuust
- KOP_7. AVL-puu tasakaalustamine

Iseseisev töö nr 5

Harjutusülesanded

KOP_1. Kahendotsimispuu valmistamine

Kirjutada ja testida funktsioon järgmise ülesande lahendamiseks.

Antud: arv n , $n > 0$.

Tulemus: n -tipuline juhu-kahendotsimispuu.

Funktsiooni võimalik skeem:

- Luua n -tipuline juhu-kahendpuu kp
- Luua n -elemendiline juhuarvude sorteeritud järjend a (näiteks juhuarvudest lõigult [1; 99])
- Omistada arvud järjendist a kahendpuu kp tippude märgenditeks tippude keskjärjekorras (vt ül KP_4)

KOP_2. Kahendotsimispuu kontroll

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: kahendpuu, milles tipumärgenditeks arvud.

Tulemus: kontrollitud, kas antud kahendpuu on kahendotsimispuu.

Vt [Kiho 2003, teoreem lk 31]

KOP_3. AVL-puu kontroll

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: kahendpuu.

Tulemus: kontrollitud, kas antud kahendpuu on AVL-puu.

KOP_3-1. Rikkekohad

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.

Antud:

- kahendpuu kp , milles iga tipuga on seotud kõrguseväli h (vt ül KP_7)
- kp tipp t
- tipu t ülemus (Λ , kui t on kp juurtipp) /Java: $\Lambda = \text{null}$ /

Tulemus:

kp tasakaalu rikkekohtade list paaridest $\langle v, y \rangle$, kus

v on AVL-rikkega tipp, st v vasaku ja parema alampuu kõrgused erinevad rohkem kui ühe võrra
 y on selle tipu (v) ülemus (või Λ , kui v on kp juurtipp)

Vihje: kahendpuu läbimine toimetada eesjärjestuses.

KOP_3 skeem:

- Omistada kp tippudele kõrguseväljad (ül KP_7-1)
- Tulemus: kp on AVL-puu, kui kp on kahendotsimispuu (ül KOP_2) ja kp rikkekohtade list on tühi

KOP_4. Otsimine kahendotsimispuus

Kirjutada ja testida rekursiivne funktsioon järgmise ülesande lahendamiseks.

Antud: *kop* - mittetühi kahendotsimise puu, *t* - selle tipp ja arv *k*, mille kohta (ja ülemust) otsida alates tipust *t*.

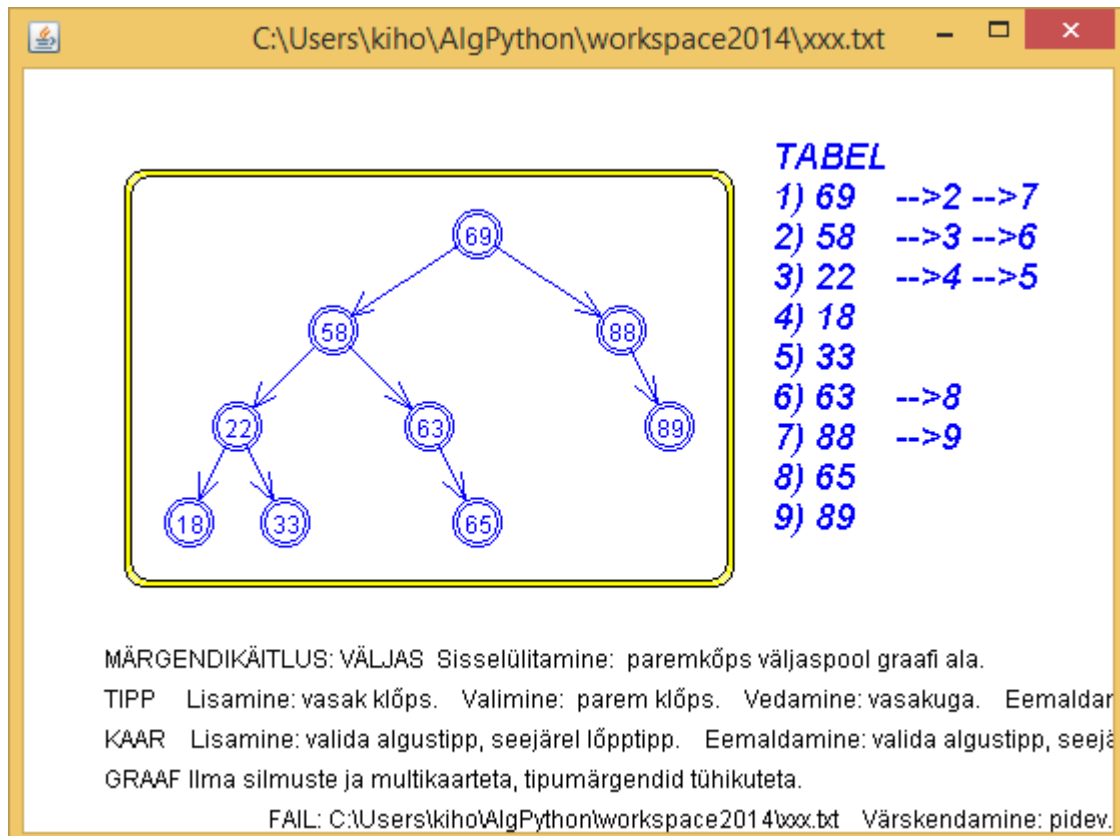
Tulemus: paar $\langle t1, t2 \rangle$, kus

t1 on tipp, milles arv *k* leiti (või Λ , kui arvu ei leitud)

t2 on *t1* ülemus (või Λ , kui *t1* on juurtipp);

juhul kui arvu ei leitud ($t1 = \Lambda$), siis *t2* on tipp, millele arv *k* sobiks alluvaks.

Testi tulemuse näide



The screenshot shows a window titled "C:\Users\kiho\AlgPython\workspace2014\xxx.txt". Inside, there is a binary search tree diagram with nodes containing numbers. The root node is 69. Its left child is 58, and its right child is 88. Node 58 has children 22 and 63. Node 22 has children 18 and 33. Node 63 has child 65. Node 88 has child 89. A yellow box highlights the tree diagram. To the right of the tree is a table titled "TABEL" with 9 rows of test cases. Below the table, there is a block of text in Estonian providing instructions for using the workspace.

TABEL		
1) 69	-->2	-->7
2) 58	-->3	-->6
3) 22	-->4	-->5
4) 18		
5) 33		
6) 63	-->8	
7) 88	-->9	
8) 65		
9) 89		

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Eemaldamine: vasak klõps.
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algustipp, seejärel lõpptipp.
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.
FAIL: C:\Users\kiho\AlgPython\workspace2014\xxx.txt Värskendamine: pidev.

Otsitav arv: 63

Leitud arv ja selle ülemus: 63 58

Otsitav arv: 75

Otsitavat arvu ei ole, koha ülemus: 88

Otsitav arv: 69

Otsitav arv leitud juurtipus (ülemus: null)

KOP_5. Lisamine kahendotsimispuusse

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: kahendotsimispuu *kop* (arvud-võtmed unikaalsed) ja lisatav arv *k*.

Tulemus: tipp arvuga *k* lisatud antud kahendotsimispuusse;

kui arv *k* juba leitud, siis ei tehta midagi.

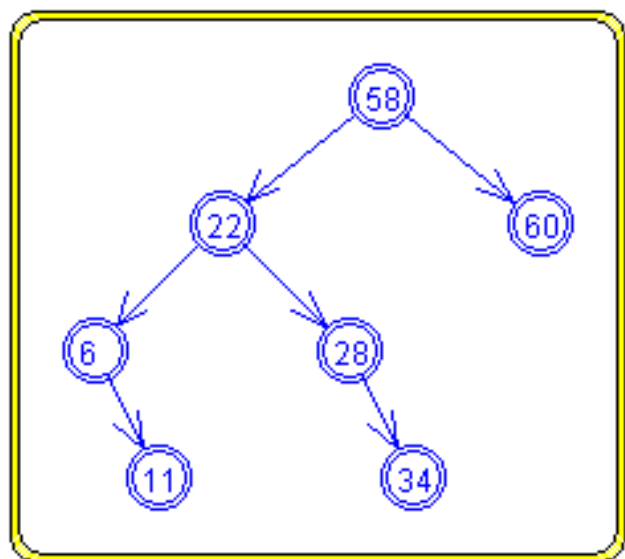
Java

```
Tipp t = new Tipp(k + "[0;0]");  
-- tehtud uus tipp t märgendiga k ja (suva)koordinaatidega x = 0; y = 0
```

Testi tulemuse näide

Lisatav arv: 59
On KOP = true

C:\Users\kiho\AlgPython\workspace2014\xxx.txt



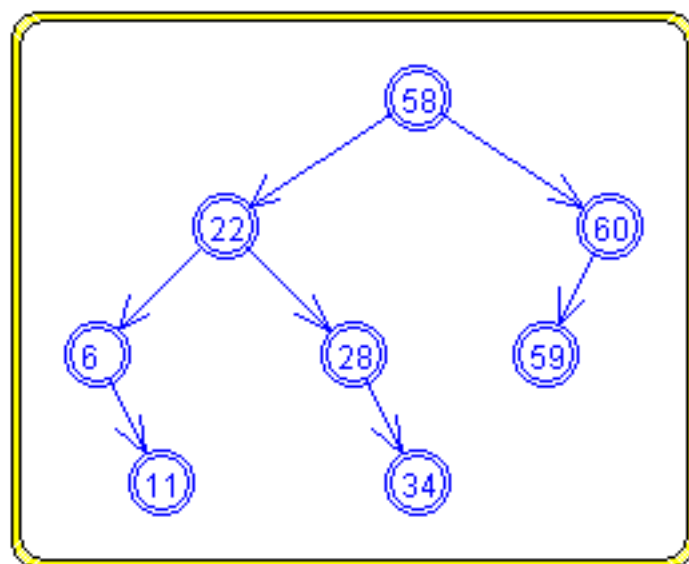
TABEL

- 1) 58 -->2 -->5
- 2) 22 -->3 -->4
- 3) 6 -->6
- 4) 28 -->7
- 5) 60
- 6) 11
- 7) 34

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremköps väljaspool graafi
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasaku
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida
GRAAF ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:\Users\kiho\AlgPython\workspace2014\xxx.txt Värskendamine:

C:\Users\kiho\AlgPython\workspace2014\xxx.txt



TABEL

- 1) 58 -->2 -->5
- 2) 22 -->3 -->4
- 3) 6 -->6
- 4) 28 -->7
- 5) 60 -->8
- 6) 11
- 7) 34
- 8) 59

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremköps väljaspool graafi ala.

TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Ee
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algustip
GRAAF ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:\Users\kiho\AlgPython\workspace2014\xxx.txt Värskendamine: pidev.

KOP_6. Eemaldamine kahendotsimispuust

KOP_6-1. Tipu eemaldamine [Kiho 2003, lk 31]

Kirjutada ja testidameetod järgmise ülesande lahendamiseks.

Antud: kop - kahendotsimispuu, t - selle tipp, y - tipu t ülemus (Δ , kui t on kop juurtipp).

Tulemus: tipp t eemaldatud kahendotsimispuust kop .

KOP_6-2. Arvu eemaldamine

Kirjutada ja testida funktsioon järgmise ülesande lahendamiseks.

Antud: kop - kahendotsimispuu, k – otsitav arv.

Tulemus: tipp arvuga k (kui leidus) eemaldatud kahendotsimispuust kop .

KOP_7. AVL-puu tasakaalustamise võte

Kirjutada ja testida funktsioon järgmise ülesande lahendamiseks.

Antud:

- kahendotsimispuu *kop* AVL-rikkega ühes tipus, nagu [Kiho 2003, Joonis 2.4 või selle peegeldus]
- kolm tippu – rikkega tipp *a* (allaviidav), selle alluv *b* (ülesviidav), tipu *a* ülemus *y* (Δ , kui *a* on *kop* juurtipp).

Tulemus: antud kahendotsimispuus *kop* sooritatud tasakaalustamise võte.

Testi tulemuse näide

AVL-rikkega tipu (a) nr: 2
selle alluva tipu (b) nr: 4
tipu a ülemuse nr: 1

Antud

Tasakaalustatud

C:\Users\kiho\AlgPython\workspace2014\kop1.txt

TABEL

1) 35	-->2 -->8
2) 8	-->3 -->4
3) 5	
4) 20	-->5 -->6
5) 10	
6) 30	-->7
7) 25	
8) 70	-->9 -->15
9) 55	-->10 -->13
10) 45	-->11 -->12
11) 40	
12) 50	
13) 65	-->14
14) 60	
15) 85	-->16 -->18
16) 80	-->17
17) 75	
18) 95	-->19
19) 90	

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Eemaldamine: parem klõps valitud tipul (2 x parem klõps)
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algustipp, seejärel lõpptipp.
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:\Users\kiho\AlgPython\workspace2014\kop1.txt Varskendamise: pidev.

C:\Users\kiho\AlgPython\workspace2014\yyy.txt

TABEL

1) 35	-->4 -->8
2) 8	-->3 -->5
3) 5	
4) 20	-->2 -->6
5) 10	
6) 30	-->7
7) 25	
8) 70	-->9 -->15
9) 55	-->10 -->13
10) 45	-->11 -->12
11) 40	
12) 50	
13) 65	-->14
14) 60	
15) 85	-->16 -->18
16) 80	-->17
17) 75	
18) 95	-->19
19) 90	

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Eemaldamine: parem klõps valitud tipul (2 x parem klõps)
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algustipp, seejärel lõpptipp.
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:\Users\kiho\AlgPython\workspace2014\yyy.txt Varskendamise: pidev.

```
Amadeus - workspace2015s\KOP_7.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2015oCDB

Kahendpuu alla_üles(Kahendpuu kop, Tipp v, Tipp w, Tipp vy) sooritada tasakaalustamise võte
»
»
» spetsifikatsioon
boolean juurtipus = (vy==null);»
Tipp keskmine;»
»
» if (kop.paremAlluv(v) == w) Joonis 2.4
»
» keskmine = kop.vasakAlluv(w); II
»
» else Joonis 2.4 peegel
» keskmine = kop.paremAlluv(w);»
»
» korraldada ümber mõned kaared:
»
»
» Kaar k;»
» if (!juurtipus)
» k = kop.kaar(v, w);»
» k.seadaLõpptipp(v);»
» k.seadaAlgustipp(w);»
»
» if (keskmine != null)»
»
» k = kop.kaar(w, keskmine);»
» k.seadaAlgustipp(v);»
» seada "keskmine" v ja w vahele (topeltvõtte puhul vajalik):
» nihutadaH(kop, keskmine, (v.x()+w.x())/2 - keskmine.x());»
»
» if (juurtipus)»
»
» » w jääb juureks, tuua esikohale tippude seas:
» kop.esikohale(w);»
»
return kop;»

void nihutadaH(Kahendpuu kp, Tipp t, int xnihe)
```

```
Amadeus - <noname59>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

spetsifikatsioon
»
» » Antud: kop - kahendotsimispuu,
» . tipp v - allaviidav,
» . tipp w - ülesviidav,
» . vy - tipu v ülemus või null , kui kui v on kop juurtipp.
» . Joonistel [Kiho 2003, joonis 2.4 või selle peegeldus] ja
» . [Kiho 2003, joonis 2.5 või selle peegeldus]
» . tähistab allaviidavat tippu alla-nooleke ja
» . ülesviidavat tippu üles-nooleke.
» Tulemus: puus kop sooritatud tasakaalustamise võte,
» . vt nt [Kiho 2003, joonis 2.4]
» . tagasatakse (muudetud) kop

spetsifikatsioon
```

```
Amadeus - <noname60>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

void nihutadaH(Kahendpuu kp, Tipp t, int xnihe) abiks
»
» » Antud: kop - kahendpuu, t - selle tipp,
» . xnihe - nihke suurus;
» Tulemus: antud kp tipust t algava alampuu tippude
» . x-koordinaadile liidetud xnihe
»
» if (t != null)»
»
» Tipp vas = kp.vasakAlluv(t);»
» Tipp par = kp.paremAlluv(t);»
» nihutadaH(kp, vas, xnihe);»
» nihutadaH(kp, par, xnihe);»
» t.seadaKoordinaadid(t.x()+xnihe, t.y());»
```

Iseseisev töö nr 5

Esitamise tähtaeg rühmal J. Liivi 2-206, E12 -14: **7. november 2015, kell 12.15**

Esitamise tähtaeg rühmal J. Liivi 2-207, T 14 -16: **1. november 2015, kell 14.15**