

Tunnikontroll II aines "Algoritmid ja
andmestruktuurid,,
20. november 2018

Lahendused: variandid A ja B

A1

```
int väiksemaidAlamaid(Kahendpuu kp, Tipp t, Tipp t_ylem)»
»
» Antud: kahendpuu kp, selle tipp t ja tipu t ülemus 't_ylem';
» . erijuhul t_ülem = null, kui t on kp juurtipp,
» . t = null, kui on olematu (st puuduv) tipp
» Tulemus: tagastatakse kp selliste tippude arv, mille võti on väiksem
» . oma ülemuse võtmest (puu juur loetakse ka selliseks tipuks)
»
» if (t == null)»
»
» return 0;»
»
» int x = väiksemaidAlamaid(kp, kp.vasakAlluv(t), t);»
» int y = väiksemaidAlamaid(kp, kp.paremAlluv(t), t);»
»
» if (t_ylem == null) kui t ülemus puudub
»
» return x + y + 1; (t on juurtipp, tingimus täidetud)
»
» int ülemInfo = Integer.parseInt(t_ylem.märgend());»
» int tInfo = Integer.parseInt(t.märgend());»
»
» if (tInfo < ülemInfo) kui tingimus t ja t_ylem vahel täidetud
»
» return x + y + 1;»
»
» return x + y;»
```

A2

```
boolean onAlluvateKeskmisegaTipp(Kahendpuu kp, Tipp t, Tipp t_ylem)»
»
» Antud: kahendpuu kp, selle tipp t ja tipu t ülemus 't_ylem';
» . erijuhul t_ylem = null, kui t on kp juurtipp,
» . t = null, kui on olematu (nt mingi tipu puuduv alluv)
» Tulemus: tagastatakse true, kui kahendpuus kp leidub
» . kahe alluvaga tipp, mille võti on
» . alluvate aritmeetiline keskmine;
» . false, kui ei leidu
»
» if (t == null)»
»
return false;»

Tipp vas = kp.vasakAlluv(t);»
Tipp par = kp.paremAlluv(t);»

» if (onAlluvateKeskmisegaTipp(kp, vas, t))»
»
return true; leidus vasakul

» if (onAlluvateKeskmisegaTipp(kp, par, t))»
»
return true; leidus paremal

» -- alampuudes ei leidu.
» if (vas != null && par != null)
return false;»

» if (vas != null && par != null) kui t on 2 alluvaga
»
int tipus = Integer.parseInt(t.märgend());»
int vasakul = Integer.parseInt(vas.märgend());»
int paremal = Integer.parseInt(par.märgend());»
return (1.0* (vasakul + paremal)/2 == 1.0*tipus);»
```

A3

```
int pikimTee(Kahendpuu kp, Tipp t)»
» Antud: kahendpuu kp ja selle tipp t;
» . t = null, kui on olematu (nt mingi tipu puuduv alluv)
» Tulemus: tagastatakse pikima tee pikkus tipust t mõnda lehttippu;
» tee pikkus: teel olevate tippude võtmete summa
    if (t == null)»
    return 0;»
    int pikimVas = pikimTee(kp, kp.vasakAlluv(t));»
    int pikimPar = pikimTee(kp, kp.paremAlluv(t));»
    return Math.max(pikimVas, pikimPar) + Integer.parseInt(t.märgend());»
```

B1

```
int häidTippe(Kahendpuu kp, Tipp t, Tipp t_ylem)»
»
» Antud: kahendpuu kp, selle tipp t ja tipu t ülemus 't_ylem'
» . erijuhul t_ülem = null, kui t on kp juurtipp
» . t = null, kui on olematu (nt mingi tipu puuduv alluv)
» Tulemus: tagastatakse kp selliste tippude arv, mille võti on sama,
» . mis naabritel (puuduva naabri -- alluva/ülemuse võti loetakse samaks)
»
» if (t == null)»
»
» return 0;»
»
Tipp vas = kp.vasakAlluv(t);»
Tipp par = kp.paremAlluv(t);»
»»
int x = häidTippe(kp, vas, t);»
int y = häidTippe(kp, par, t);»
int tulem = x + y;»
»»
» tipp t võib ise ka veel "hea" olla:
int tInf = Integer.parseInt(t.märgend());»
» naabrite Inf (puuduva korral := tInf):
int t_ylemInf = (t_ylem != null)? Integer.parseInt(t_ylem.märgend()) : tInf;»
int t_vasInf = (vas != null)? Integer.parseInt(vas.märgend()) : tInf;»
int t_parInf = (par != null)? Integer.parseInt(par.märgend()) : tInf;»
»
» if (tInf == t_ylemInf && t_ylemInf == t_vasInf && t_vasInf == t_parInf)»
»
» return tulem + 1;»
»
return tulem;»
```

B2

```
boolean onVäikesedLehttipud(Kahendpuu kp, Tipp t, Tipp t_ylem)»
```

```
» Antud: kahendpuu kp, selle tipp t ja tipu t ülemus 't_ylem'  
» . erijuhul t_ylem = null, kui t on kp juurtipp  
» . t = null, kui on olematu (nt mingi tipu puuduv alluv)  
» Tulemus: tagastatakse  
» . true, kui kp iga lehttipu võti on väiksem oma ülemuse võtmest,  
» . false vastasel korral
```

```
if (t == null)»
```

```
return true;»
```

```
Tipp vas = kp.vasakAlluv(t);»
```

```
Tipp par = kp.paremAlluv(t);»
```

```
»»
```

```
if (lonVäikesedLehttipud(kp, vas, t))»
```

```
return false; vasakus kõik lehttipud ei ole nagu vaja
```

```
if (lonVäikesedLehttipud(kp, par, t))»
```

```
return false; paremas kõik lehttipud ei ole nagu vaja
```

```
» vasak ja parem alampuu on selliste lehttipudega nagu vaja
```

```
» tipp t ise? :
```

```
if (vas == null && par == null)
```

```
return true;»
```

```
if (vas == null && par == null) kui t on lehttip
```

```
if (t_ylem == null)»
```

```
return true; ainuke tipp t -- on lehttip (ja tingimuse loeme kehtivaks)
```

```
if (Integer.parseInt(t.märgend()) >= Integer.parseInt(t_ylem.märgend()))»
```

```
return false;»
```

B3

```
boolean onKahanevTeeLehttippu(Kahendpuu kp, Tipp t)»
```

```
» Antud: kahendpuu kp ja selle tipp t
```

```
» . t = null, kui on olematu (nt mingi tipu puuduv alluv)
```

```
» Tulemus: true, kui teel tipust t mingisse lehttipu on tippude võtmed kahanevad
```

```
» false, vastasel korral (sellist kahanevate võtmetega teed ei leidu)
```

```
if (t == null)»
```

```
return false; tühjas alampuus ei ole mingit teed
```

```
»»
```

```
Tipp vas = kp.vasakAlluv(t);»
```

```
Tipp par = kp.paremAlluv(t);»
```

```
if (vas == null && par == null) kui t on lehttipu
```

```
return true; t ise ainukesena sellisel teel
```

```
»»
```

```
boolean onVasakul = onKahanevTeeLehttippu(kp, vas);»
```

```
boolean onParemal = onKahanevTeeLehttippu(kp, par);»
```

```
if (!onVasakul && !onParemal)»
```

```
return false; pole lootust
```

```
if (onVasakul && Integer.parseInt(vas.märgend()) < Integer.parseInt(t.märgend()))
```

```
if (onParemal && Integer.parseInt(par.märgend()) < Integer.parseInt(t.märgend()))
```

```
return false;»
```

```
if (onVasakul && Integer.parseInt(vas.märgend()) < Integer.parseInt(t.märgend()))»
```

```
return true; tipust t vasakule saab, vähenevalt
```

```
if (onParemal && Integer.parseInt(par.märgend()) < Integer.parseInt(t.märgend()))»
```

```
return true; tipust t paremale saab, vähenevalt
```