

# Kisakoodarin käsikirja

Antti Laaksonen

6. marraskuuta 2016



# Sisältö

<b>Alkusanat</b>	<b>v</b>
<b>I Perusasiat</b>	<b>1</b>
1 Johdanto	3
2 Aikavaativuus	15
3 Järjestäminen	23
4 Tietorakenteet	33
5 Täydellinen haku	43
6 Ahneet algoritmit	51
7 Dynaaminen ohjelmointi	57
8 Tasoitettu analyysi	69
9 Välikyselyt	75
10 Bittien käsittely	87
<b>II Verkkoalgoritmit</b>	<b>95</b>
11 Verkkojen perusteet	97
12 Verkon läpikäynti	105
13 Lyhimmät polut	111
14 Puiden käsittely	121
15 Virittävät puut	127
16 Suunnatut verkot	135
17 Vahvasti yhtenäisyys	143

<b>18 Puukyselyt</b>	<b>149</b>
<b>19 Polut ja kierrokset</b>	<b>155</b>
<b>20 Virtauslaskenta</b>	<b>161</b>
 <b>III Uusia haasteita</b>	 <b>169</b>
<b>21 Lukuteoria</b>	<b>171</b>
<b>22 Kombinatoriikka</b>	<b>177</b>
<b>23 Matriisit</b>	<b>183</b>
<b>24 Todennäköisyys</b>	<b>187</b>
<b>25 Peliteoria</b>	<b>193</b>
<b>26 Merkkijonot</b>	<b>199</b>
<b>27 Neliöjuorialgoritmit</b>	<b>207</b>
<b>28 Lisää segmenttipuusta</b>	<b>211</b>
<b>29 Geometria</b>	<b>219</b>
<b>30 Pyyhkäisyviiva</b>	<b>225</b>



# **Osa I**

## **Perusasiat**



# **Osa II**

## **Verkkoalgoritmit**





# Luku 18

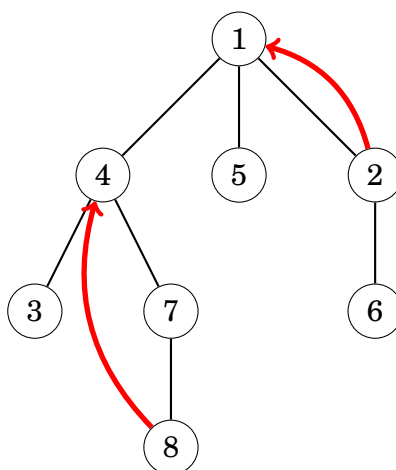
## Puukyselyt

Tyypillisiä puukyselyitä ovat puun polkuihin ja alipuihin liittyvät kyselyt. Tämä luku esittelee tekniikoita, joiden avulla on mahdollista toteuttaa puukyselyjä tehokkaasti. Usein esiintyvä idea on muuttaa puu jollakin tavalla taulukoksi, mikä helpottaa puun käsittelyä.

### 18.1 Tehokas nouseminen

*Tehtävä:* Annettuna on juurellinen puu, jonka solmut on numeroitu  $1 \dots n$  ja juurisolmu on 1. Tehtäväsi on vastata tehokkaasti kyselyihin muotoa ”mikä solmu on  $k$  askelta ylempänä solmua  $x$ ”.

Merkitään  $f(x, k)$  solmua, joka on  $k$  askelta ylempänä solmua  $x$ . Esimerkiksi seuraavassa puussa  $f(2, 1) = 1$  ja  $f(8, 2) = 4$ .



Suoraviivainen tapa laskea funktion  $f(x, k)$  arvo on kulkea puussa  $k$  askelta ylöspäin solmusta  $x$  alkaen. Tämän aikavaativuus on kuitenkin  $O(n)$ , koska on mahdollista, että puussa on ketju, jossa on  $O(n)$  solmua.

Kuten luvussa 16.3.1, funktion  $f(x, k)$  arvo on mahdollista laskea tehokkaasti ajassa  $O(\log k)$  sopivan esikäsittelyn avulla. Ideana on laskea etukäteen kaikki arvot  $f(x, k)$ , joissa  $k = 1, 2, 4, 8, \dots$  eli 2:n potenssi. Esimerkiksi yllä olevassa puussa muodostuu seuraava taulukko:

$x$	1	2	3	4	5	6	7	8
$f(x, 1)$	0	1	4	1	1	2	4	7
$f(x, 2)$	0	0	1	0	0	1	1	4
$f(x, 4)$	0	0	0	0	0	0	0	0
...								

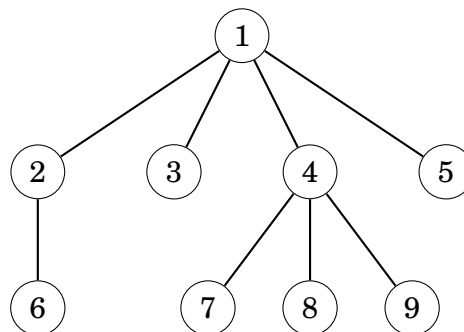
Taulukossa arvo 0 tarkoittaa, että nousemalla  $k$  askelta päättyy puun ulkopuolelle juurisolmun yläpuolelle.

Esilaskenta vie aikaa  $O(n \log n)$ , koska jokaisesta solmusta voi nousta korkeintaan  $n$  askelta ylöspäin. Tämän jälkeen minkä tahansa funktion  $f(x, k)$  arvon saa laskettua ajassa  $O(\log k)$  jakamalla nousun 2:n potenssin osiin.

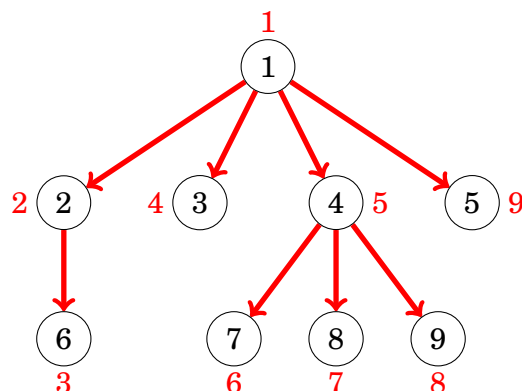
## 18.2 Solmutaulukko

Solmutaulukko sisältää juurellisen puun solmut siinä järjestyksessä kuin juuresta alkava syvyyshaku vierailee solmuissa.

Esimerkiksi puussa



syvyyshaku etenee



ja solmutaulukoksi tulee:

1	2	3	4	5	6	7	8	9
1	2	6	3	4	7	8	9	5

## 18.2.1 Alipuiden käsittely

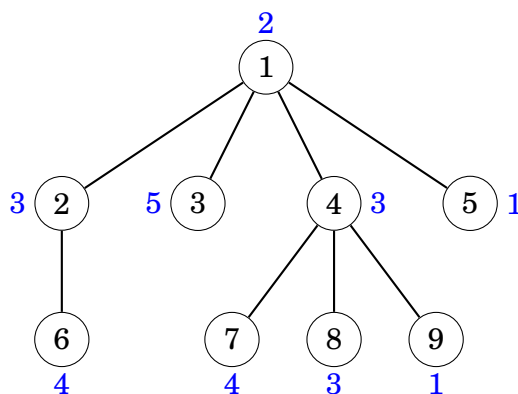
Solmutaulukossa jokaisen alipuun kaikki solmut ovat peräkkäin niin, että ensin on alipuun juurisolmu ja sitten kaikki muut alipuun solmut. Esimerkiksi äskeisessä taulukossa solmun 4 alipuuta vastaa seuraava taulukon osa:

1	2	3	4	5	6	7	8	9
1	2	6	3	4	7	8	9	5

Tämän ansiosta solmutaulukon avulla voi käsitellä tehokkaasti puun alipuihin liittyviä kyselyitä. Voimme ratkaista esimerkiksi seuraavan tehtävän:

*Tehtävä:* Annettuna on juurellinen puu, jossa on  $n$  solmua ja jokaisella solmulla on tietty arvo. Tehtäväsi on käsitellä kyselyt muotoa ”muuta solmun  $x$  arvoa” sekä ”laske arvojen summa solmun  $x$  alipuussa”.

Tarkastellaan seuraavaa puuta, jossa siniset luvut ovat solmujen arvoja. Esimerkiksi solmun 4 alipuun arvojen summa on  $3 + 4 + 3 + 1 = 11$ .



Ideana on luoda solmutaulukko, joka sisältää jokaisesta solmusta kolme tietoa: (1) solmun tunnus, (2) alipuun koko ja (3) solmun arvo. Esimerkiksi yllä olevasta puusta syntyy seuraava taulukko:

1	2	3	4	5	6	7	8	9
1	2	6	3	4	7	8	9	5
9	2	1	1	4	1	1	1	1
2	3	4	5	3	4	3	1	1

Tästä taulukosta alipuun solmujen arvojen summa selviää lukemalla ensin alipuun koko ja sitten sitä vastaavat solmut. Esimerkiksi solmun 4 alipuun arvojen summa selviää näin:

1	2	3	4	5	6	7	8	9
1	2	6	3	4	7	8	9	5
9	2	1	1	4	1	1	1	1
2	3	4	5	3	4	3	1	1

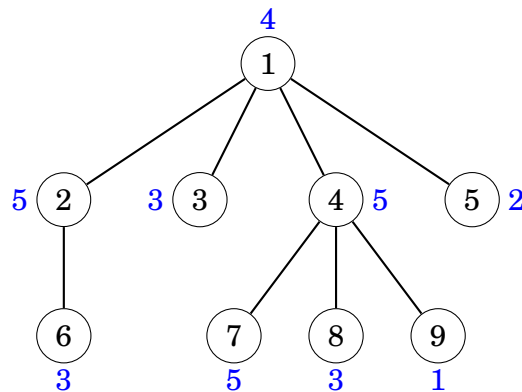
Viimeinen tarvittava askel on tallentaa solmujen arvot binääri-indeksipuu-  
hun tai segmenttipuuhun. Tällöin sekä alipuun arvojen summan laskeminen  
että solmun arvon muuttaminen onnistuvat ajassa  $O(\log n)$ , eli pystymme vas-  
taamaan kyselyihin tehokkaasti.

### 18.2.2 Polkujen käsittely

Solmutaulukon avulla voi myös käsitellä tehokkaasti polkuja, jotka kulkevat  
juuresta tiettyyn solmuun puussa. Näin on seuraavassa tehtävässä:

*Tehtävä:* Annettuna on juurellinen puu, jossa on  $n$  solmua ja jokaisella sol-  
mulla on tietty arvo. Tehtäväsi on käsitellä kyselyt muotoa ”muuta solmun  
 $x$  arvoa” sekä ”laske arvojen summa juuresta solmuun  $x$ ”.

Esimerkiksi seuraavassa puussa polulla solmusta 1 solmuun 8 arvojen sum-  
ma on  $4 + 5 + 3 = 12$ .



Ideana on muodostaa samanlaiset taulukot kuin alipuiden käsittelyssä mut-  
ta tallentaa solmujen arvot erikoisella tavalla: kun taulukon kohdassa  $k$  olevan  
solmun arvo on  $a$ , kohdan  $k$  arvoon lisätään  $a$  ja kohdan  $k + c$  arvosta vähenne-  
tään  $a$ , missä  $c$  on alipuun koko.

Esimerkiksi yllä olevaa puuta vastaa seuraava taulukko:

1	2	3	4	5	6	7	8	9	10
1	2	6	3	4	7	8	9	5	–
9	2	1	1	4	1	1	1	1	–
4	5	3	–5	2	5	–2	–2	–4	–4

Esimerkiksi solmun 3 arvona on  $-5$ , koska se on solmujen 2 ja 6 alipuiden jälkeinen solmu, mistä tulee arvoa  $-5-3$ , ja sen oma arvo on 3. Yhteensä solmun 3 arvo on siis  $-5-3+3=-5$ . Huomaa, että taulukossa on ylimääräinen kohta 10, johon on tallennettu vain juuren arvon vastaluku.

Nyt solmujen arvojen summa polulla juuresta alkaen selviää laskemalla kaikkien taulukon arvojen summa taulukon alusta solmuun asti. Esimerkiksi summa solmusta 1 solmuun 8 selviää näin:

	1	2	3	4	5	6	7	8	9	10
1	1	2	6	3	4	7	8	9	5	-
9	9	2	1	1	4	1	1	1	1	-
4	4	5	3	-5	2	5	-2	-2	-4	-4

Summaksi tulee  $4+5+3-5+2+5-2=12$ , mikä vastaa polun summaa  $4+5+3=12$ . Tämä laskentatapa toimii, koska jokaisen solmun arvo lisätään summaan, kun se tulee vastaan syvyysshaussa, ja vähennetään summasta, kun sen käsittely päättyy.

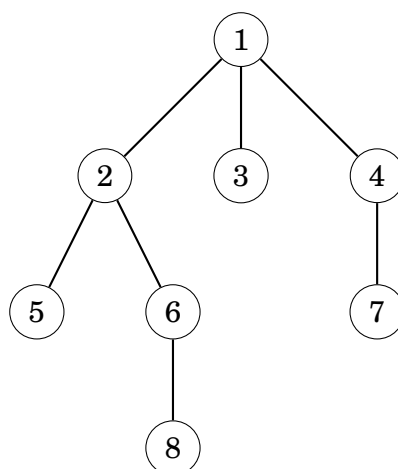
Alipuiden käsittelyä vastaavasti voimme tallentaa arvot binääri-indeksipuuhun tai segmenttipuuhun ja sekä polun summan laskeminen että arvon muuttaminen onnistuvat ajassa  $O(\log n)$ .

## 18.3 Alin yhteinen esivanhempi

Solmujen  $a$  ja  $b$  alin yhteinen esivanhempi (*lowest common ancestor*) on mahdollisimman alhaalla puussa oleva solmu, jonka alipuuhun kuuluvat molemmat solmut  $a$  ja  $b$ . Luonteva tehtävä on:

*Tehtävä:* Annettuna on puu, jossa on  $n$  solmua. Tehtäväsi on vastata kyselyihin ”mikä on solmujen  $a$  ja  $b$  alin yhteinen esivanhempi”.

Esimerkiksi puussa



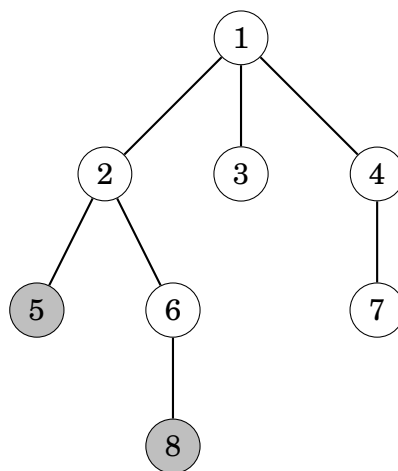
solmujen 5 ja 8 alin yhteinen esivanhempi on solmu 2 ja solmujen 3 ja 4 alin yhteinen esivanhempi on solmu 1.

Tutustumme seuraavaksi kahteen menetelmään alimman yhteisen esivanhemman selvittämiseen.

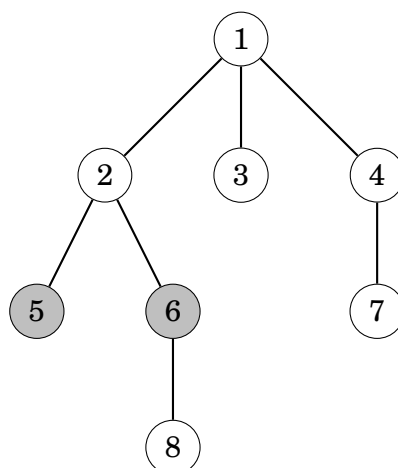
### Menetelmä 1

Yksi tapa ratkaista tehtävä on hyödyntää tehokasta nousemista puussa. Tällöin alimman yhteisen esivanhemman etsiminen muodostuu kahdesta vaiheesta. Ensin nouseaan alemmasta solmusta samalle tasolle ylemmän solmun kanssa, sitten nouseaan rinnakkain kohti alinta yhteistä esivanhempaa.

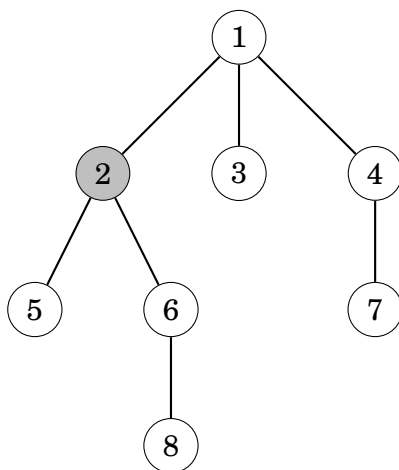
Tarkastellaan esimerkkinä solmujen 5 ja 8 alimman yhteisen esivanhemman etsimistä:



Solmu 5 on tasolla 3, kun taas solmu 8 on tasolla 4. Niinpä nousemme ensin solmusta 8 yhden tason ylemmäs solmuun 6:



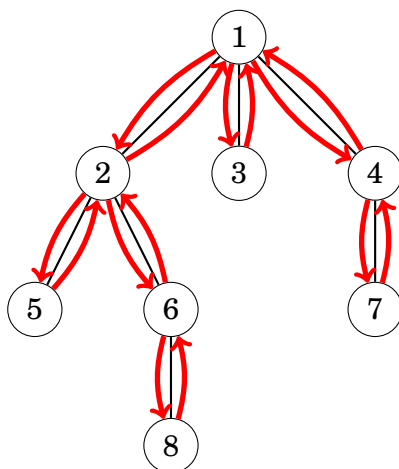
Tämän jälkeen etsitään pienin määrä tasoja, jotka nousemalla ylöspäin päädyimme samaan solmuun. Tässä tapauksessa riittää nousta yksi taso ylöspäin solmuun 2. Niinpä solmujen alin yhteinen esivanhempi on 2:



Menetelmä vaatii  $O(n \log n)$ -aikaisen esikäsittelyn, jonka jälkeen minkä tahansa kahden solmun alin yhteinen esivanhempi selviää ajassa  $O(\log n)$ , koska kumpikin vaihe nousussa vie aikaa  $O(\log n)$ .

## Menetelmä 2

Toinen tapa ratkaista tehtävä perustuu solmutaulukon käyttämiseen. Ideana on jälleen järjestää solmut syvyysshaun mukaan:



Erona aiempaan solmu lisätään kuitenkin solmutaulukkoon mukaan *aina*, kun syvyyshaku käy solmussa, eikä vain ensimmäisellä kerralla. Niinpä solmu esiintyy solmutaulukossa  $x + 1$  kertaa, missä  $x$  on solmun lasten määrä, ja solmutaulukossa on yhteensä  $2n - 1$  solmua.

Tallennamme solmutaulukkoon kaksi tietoa: (1) solmun tunnus ja (2) solmun taso puussa. Esimerkkipuuta vastaavat taulukot ovat:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	5	2	6	8	6	2	1	3	1	4	7	4	1
1	2	3	2	3	4	3	2	1	2	1	2	3	2	1



Tämän taulukon avulla solmujen  $a$  ja  $b$  alin yhteinen esivanhempi selviää etsimällä taulukosta alimman tason solmu solmujen  $a$  ja  $b$  välissä. Esimerkiksi solmujen 5 ja 8 alin yhteinen esivanhempi löytyy seuraavasti:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
node	1	2	5	2	6	8	6	2	1	3	1	4	7	4	1
depth	1	2	3	2	3	4	3	2	1	2	1	2	3	2	1

↑

Solmu 5 on taulukossa kohdassa 3, solmu 8 on taulukossa kohdassa 6 ja alimman tason solmu välillä 3...6 on kohdassa 4 oleva solmu 2, jonka taso on 2. Niinpä solmujen 5 ja 8 alin yhteinen esivanhempi on solmu 2.

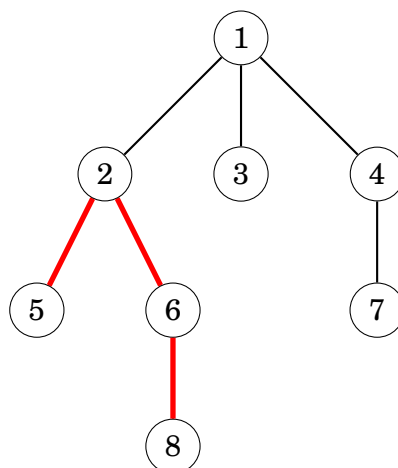
Alimman tason solmu välillä selviää ajassa  $O(\log n)$ , kun taulukon sisältö on tallennettu segmenttipuuhun. Myös aikavaativuus  $O(1)$  on mahdollinen, koska taulukko on staattinen, mutta tälle on harvoin tarvetta. Kummassakin tapauksessa esikäsitteily vie aikaa  $O(n \log n)$ .

## Solmujen etäisyydet

Myös seuraava tehtävä palautuu alimman yhteisen esivanhemman hakuun:

*Tehtävä:* Annettuna on juurellinen puu, jonka solmut on numeroitu  $1 \dots n$ . Tehtäväsi on vastata kyselyihin ”laske solmujen  $a$  ja  $b$  etäisyys puussa”.

Valitaan ensin mikä tahansa solmu puun juureksi. Tämän jälkeen solmujen  $a$  ja  $b$  etäisyys on  $d(a) + d(b) - 2 \cdot d(c)$ , missä  $c$  on solmujen alin yhteinen esivanhempi ja  $d(s)$  on etäisyys puun juuresta solmuun  $s$ . Esimerkiksi puussa



solmujen 5 ja 8 alin yhteinen esivanhempi on 2. Polku solmusta 5 solmuun 8 kulkee ensin ylöspäin solmusta 5 solmuun 2 ja sitten alaspäin solmusta 2 solmuun 8. Solmujen etäisyydet juuresta ovat  $d(5) = 3$ ,  $d(8) = 4$  ja  $d(2) = 2$ , joten solmujen 5 ja 8 etäisyys on  $3 + 4 - 2 \cdot 2 = 3$ .



# **Osa III**

## **Uusia haasteita**

