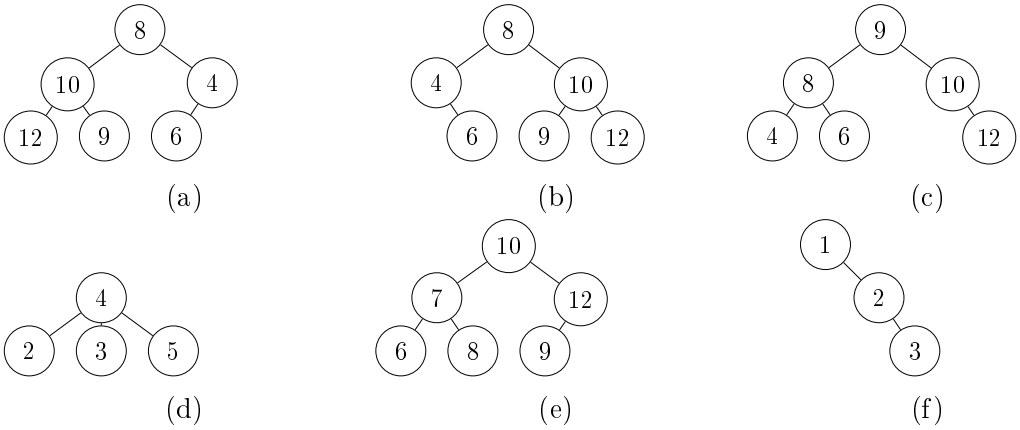
Tehtävä 2: Aineistotehtävä

Kysymys 1

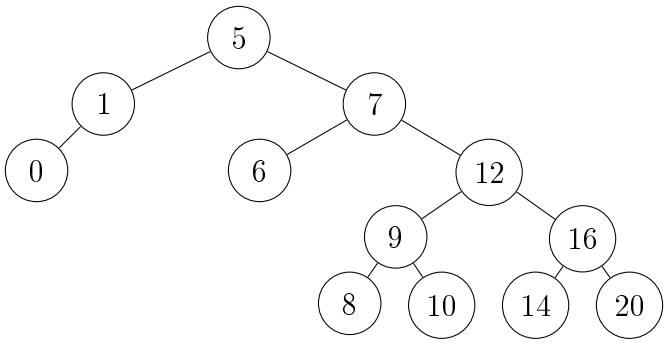


- (a) ei ole tehtävänannon määritelmän mukainen, koska esimerkiksi solmun 8 vasemman lapsen solmu on 10. Tämä rikkoo ehtoa, jonka mukaan solmun vasemmasta alipuusta löytyvien avainarvojen tulee olla solmun avainarvoa pienempiä.
- (c) ei ole binäärinen hakupuu, koska solmun 8:n oikea lapsi on 6, vaikka solmun oikeassa alipuussa olevien avainten tulisi olla solmun avainta suurempia.
- (d) ei ole binäärinen hakupuu, koska juurisolmulla on 3 lasta.
- (e) ei ole binäärinen hakupuu, koska juuren (avain 10) oikeasta alipuusta löytyy sen avainarvoa pienempi avainarvo (avain 9).

Kysymys 2

a) Kutsu käy läpi tässä järjestyksessä solmut: 12, 7, 2
b) Kutsu käy läpi tässä järjestyksessä solmut: 12, 16, 14, 15
c) Kutsu käy läpi tässä järjestyksessä solmut: 12, 7, 9, 10

Kysymys 3



Kysymys 4
Binäärisen hakupuun *korkeus* on askelten määrä kuljettaessa juurisolmusta alimpana olevaan lehtisolmuun.

a) Ensimmäiseen kerrokseen puussa mahtuu 1 solmua, toiseen 2, kolmanteen 4 jne., eli puun korkeuden kasvaessa uusien kerroksien koot kasvavat kahden potensseissa. 10:n korkuiseeen puuhun mahtuu $2^0+2^1+2^2+...+2^{10} = 2^{11}-1 = 2047$ solmua.
b) $2^{h+1}-1$.
c) Pahimmassa tapauksessa algoritmi käy läpi puun juuresta alimpaan lehtisolmuun, suorittaen kullakin tasolla vakiomäärän operaatioita. Tarvittavien operaatioiden määrä on samaa suuruusluokkaa, kuin puun korkeus. Koska puu on mahdollisimman tehokkaasti muodostettu, b-kohdan perusteella voimme antaa karkeaksi arvioksi puun koolle $h \approx \log_2(n)$, (tarkalleen ottaen $h = \lceil \log_2(n+1) - 1 \rceil$) ja tarvittaviksi operaatiomääriksi pienet vakiot kerrottuna tällä.

Kysymys 5

Mikäli solmun s oikea alipuu ei ole tyhjä, on seuraaja oikean alipuun se solmu, jolla on pienin arvo. Jos taas oikea alipuu on tyhjä, löytyy seuraaja nousemalla ylös puussa niin kauan, kunnes löytyy ensimmäinen esivanhempi, jonka vasempaan alipuuhun s sisältyy. Jos tällaista solmua ei ole olemassa, on solmussa s puun suurin avainarvo, eikä seuraajaa ole olemassa. Seuraavassa perustelu menetelmän oikeellisuudesta.

TAPAUS 1: Jos s:llä on oikea alipuu, on seuraaja kyseisen alipuun minimi.

- Määritelmän mukaan seuraaja ei voi olla s:n vasemmassa alipuussa.
- Olkoon y s:n (esi)vanhempi.
  - Jos s on y:n oikeassa alipuussa, on voimassa avain[y] $<$ avain[s], eikä siis y (tai yksikään sen vasemman alipuun solmuista) voi olla etsitty seuraaja.
  - Jos s on y:n vasemmassa alipuussa, määritelmän mukaan avain[s]$<$avain[y]. Mille tahansa s:n oikean alipuun solmulle z taas on voimassa avain[z]$>$avain[s] ja toisaalta avain[z]$<$avain[y]. Siis avain[s] $<$ avain[z] $<$ avain[y], minkä vuoksi y ei selvästikään voi olla etsitty seuraaja. Luonnollisestikaan seuraaja ei tällöin voi olla myöskään y:n oikeassa alipuussa.
  - Edellisten perusteella, mikään s:n esivanhemmista ei voi olla sen seuraaja. Lisäksi, suhteessa kuhunkin esivanhempaan, s ja sen seuraaja löytyvät aina samasta alipuusta. Täten seuraajan on pakko löytyä s:n oikeasta alipuusta.

TAPAUS 2: Jos s:llä ei ole oikeaa alipuuta, on seuraaja s:n lähin esivanhempi, jonka vasemmassa alipuussa s on.

- Määritelmän mukaan seuraaja ei voi olla s:n vasemmassa alipuussa.
- Olkoon y s:n (esi)vanhempi.
  - Jos s on y:n oikeassa alipuussa, on voimassa avain[y] $<$ avain[s], eikä siis y (tai yksikään sen vasemman alipuun solmuista) voi olla etsitty seuraaja.
  - Jos s on y:n vasemmassa alipuussa, määritelmän mukaan avain[s]$<$avain[y]. Selvästikään seuraaja ei voi löytyä y:n oikeasta alipuusta. Jos y:n ja s:n välisellä polulla puusta löytyy solmu z jonka vasemmassa alipuussa s on, on oltava voimassa avain[s] $<$ avain[z] $<$ avain[y].

- Edellisten perusteella seuraajan on oltava solmua s lähin niistä esivanhemmista, jonka vasemmassa alipuussa s sijaitsee.

TAPAUS 3: s:llä ei ole oikeaa alipuuta, eikä ole olemassa esivanhempaa, jonka vasemmassa alipuussa s on. Tällöin s:n avain on puun maksimiarvo, eikä seuraajaa ole määritelty.

Algoritmi voidaan formaalisti määritellä esim. seuraavasti:

PieninArvo(solmu x)

1. **niin kauan kuin** vasen[x]$\neq$ NIL **suorita riviä** 2
2. **aseta** x $\leftarrow$ vasen[x]
3. **tulos on** x

EtsiSeuraaja(solmu x)

1. **jos** oikea[x] $\neq$ NIL **niin tulos on** PieninArvo(oikea[x])
2. **muutoin suorita seuraavat rivit**
3. **aseta** y $\leftarrow$ vanhempi[x]
4. **niin kauan kuin** y$\neq$NIL **ja** x=oikea[y] **suorita rivejä** 5-6
5. **aseta** x $\leftarrow$ y
6. **aseta** y $\leftarrow$ vanhempi[y]
7. **tulos on** y