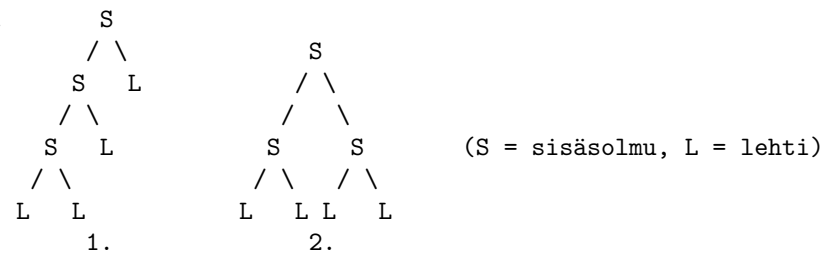
Tietorakenteet 2018
Harjoitukset 6, ratkaisut (Viikko 42)

1.-2. Katso valmiit Java-tiedostot malliratkaisukansiosta.

3.


Pahin tapaus on kohta 1. ylläolevassa kuvassa. Tällöin jokaisella tasolla on 2 solmua paitsi viimeisellä on vain 1 solmu (juuri). Maksimissaan korkeus on siis $(n-1)/2$ (alin taso on 0). Tapauksessa 2., kun puu on täydellinen, on solmuja eri tasoilla alkaen juuresta $1, 2, 4, 8, \ldots, 2^h$, joten puun solmujen lukumäärä on

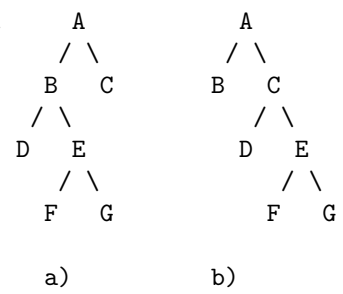
$$1 + 2 + 2^2 + 2^3 + \cdots + 2^h = \sum_{i=0}^{h} 2^i$$

Tällöin saadaan yhteys solmujen lukumäärän ja puun korkeuden välille:

$$n = \sum_{i=0}^{h} 2^i = 2^{h+1} - 1,$$

josta ratkaisemalla h saadaan

$$h = \log_2(n+1) - 1.$$

4.


5. Algoritmit 1–7. Tässä oletetaan, että käsitellään aidon binääripuun taulukkoesitystä (S), missä puuttuvat solmut on korvattu null-viitteillä (n). Lisäksi oletetaan, että taulukon indeksissä 0 on null-viite. Tällöin binääripuun juurisolmu on indeksissä 1. Koska tehtävän kannalta oleellisinta on tarkastella metodien toimintaperiaatetta binääripuun operaatioina, ei seuraavissa pseudokoodiesityksissä tarkisteta parametrien oikeellisuutta (esim. onko binääripuu S tyhjä, osoittaako välitetty indeksi taulukon ulkopuolelle, ovatko metodeissa lasketut indeksit taulukon ulkopuolella, jne.).

Algorithm 1 Palauttaa juurisolmun arvon.

```
root(S)
  return S[1]
```

Algorithm 2 Palauttaa k:nnen solmun vanhemman arvon.

```
parent(S,k)
  return S[⌊(k/2)⌋]
```

Algorithm 3 Palauttaa k:nnen solmun vasemman lapsen arvon.

```
leftChild(S,k)
  return S[2k]
```

Algorithm 4 Palauttaa k:nnen solmun oikean lapsen arvon.

```
rightChild(S,k)
  return S[2k + 1]
```

Algorithm 5 Palauttaa arvon true jos kyseessä on sisäsolmu.

```
isInternal(S,k)
  if S[k] != n and (leftChild(S,k) != n or rightChild(S,k) != n) then
    return true
  else
    return false
  end if
```

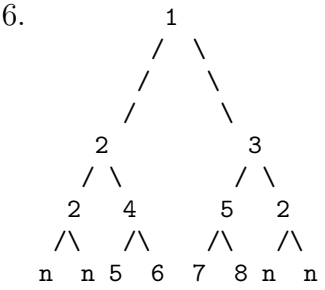
Algorithm 6 Palauttaa arvon true jos kyseessä on ulkosolmu.

```
isExternal(S,k)
  if S[k] != n and leftChild(S,k) = n and rightChild(S,k) = n then
    return true
  else
    return false
  end if
```

Algorithm 7 Palauttaa arvon true jos kyseessä on juurisolmu.

```
isRoot(S,k)
  if S[k] != n and parent(S,k) = n then
    return true
  else
    return false
  end if
```

6.



7.-8. Molemmissa kohdissa on talletettava itse taulukko (table) ja montako alkiota siinä on (currentElements).

Algorithm 8 removeMinElement ()

a)
```
if currentElements > 0 then
  result = table[0]
  for i = 1 to currentElements-1 do
    table[i-1] ←table[i]
  end for
  currentElements = currentElements-1
else
  virhe "Jono on tyhjä"
end if
return result
```

Algoritmi removeMinElement poistaa alkion taulukon alusta ja siirtää loppuja yhden pykälän nollaa lähemmäksi.

Algorithm 9 insertItem (x)

```
if currentElements < maxElements then
  i ← currentElements
  while i > 0 and table[i − 1] > x do
    table[i] ←table[i − 1]
    i ← i − 1
  end while
  table[i] ← x
  currentElements ← currentElements +1
else
  virhe "Jono täynnä"
end if
```

Algoritmi insertItem etsii lisäyspaikan lisättävälle avaimelle, tekee tilaa siirtämällä suurempia avaimia yhden eteenpäin ja lisää avaimen taulukkoon omalle paikalleen.

Algoritmi removeMinElement etsii taulukon pienimmän alkion kopioi sen talteen, korvaa sen taulukon viimeisellä alkiolla, pienentää taulukon kokoa yhdellä ja palauttaa pienimmän alkion. Näin voidaan tehdä, koska taulukko ei ole järjestyksessä.

Algorithm 10 removeMinElement ()

b)
```
if currentElements > 0 then
  minIndex ←0
  min ←table[0]
  for i ←1 to currentElements-1 do
    if table[i] < min then
      minIndex = i
      min = table[i]
    end if
  end for
  table[minIndex] ←table[currentElements-1]
  currentElements ←currentElements-1
else
  virhe "jono on tyhjä"
end if
return min
```

Algorithm 11 insertItem (x)

```
if currentElements < maxElements then
  table[currentElements] = x;
  currentElements = currentElements + 1
else
  virhe "jono on täynnä"
end if
```

Algoritmi insertItem lisää alkion taulukon loppuun (koko kasvaa yhdellä). Tämä voidaan tehdä, koska removeMinElement etsii koko taulukon läpi.