

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, koulutusohjelmasi, kurssikoodi ja -nimi "T-106.1223 Tietorakenteet ja algoritmit Y", päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*.

**1. Järjestäminen ja algoritmianalyysi (1p + 2p + 3p + 4p + 2p)**

Seuraavassa on annettu *pikajärjestämisalgoritmin* rekursiivinen osa, joka järjestää taulukossa `array` (jonka koko on  $N$ ) olevat alkiot suuruusjärjestykseen.

```
1 quicksort(array:Array, left:integer, right:integer)
2   if |left-right| > 1
3     split = partition(array,left,right)

4     if (split-1)-left > 0
5       quicksort(array, left, split-1)
6     if right-(split+1) > 0
7       quicksort(array, split+1, right)

8   endif
9 end
```

- a) Määrittele algoritmin "syötteen koko" ( $N$ ) sen saamien parametrien funktiona.
- b) *Kuinka monta kertaa algoritmi kutsuu itseään rekursiivisesti* (suuruusluokka riittää vastaukseksi)? Perustele.
- c) *Selitä* algoritmin *toiminta* käyttäen apuna sopivaa esimerkkiä.
- d) Rivillä 3 algoritmi kutsuu funktiota `partition(array, left, right)`. Selitä sanallisesti *jokin* partitiointimenetelmä, joka soveltuu pikajärjestämismenetelmälle. Voit myös käyttää pseudokoodia ja kuvia selityksesi tukena.
- e) Analysoi kuvaamasi partitiointimenetelmän aikavaativuus iso  $O$  -notaatiossa.

## 2. Binäärikeko (1p + 1p + 4p + 3p + 3p)

Seuraava taulukko T esittää binäärikekoa (lukuarvot esittävät alkioden prioriteettejä):

$T = [99, 95, 96, 90, 67, 59, 57, 80, 75, 64, 60, 50, 33, 54, 52, 73]$

- Mikä on T:n kekoehto (heap-order property)?
- Piirrä T binääripuuna (binary tree representation).
- Selitä sanallisesti miten suurimman alkion poisto binäärikeosta tapahtuu.
- Esitä kolme (3) kertaa peräkkäin suurimman alkion poisto (DeleteMax) T:stä.
- Lisää (Insert) alkuperäiseen kekkoon T alkiot prioriteeteiltään 96, 95 ja 94.

**Huom!** Anna välivaiheissa vastauksesi (myös) taulukkoesitysmuodossa aina kun proseduuri (percolate up/down) muuttaa taulukkoa.

## 3. Hakurakenteet (2p + 2p + 2p + 2p + 2p)

Vertaile tasapainotettuja hakupuita (balanced search trees) ja hajautusrakenteita (hashing) keskenään. Jäsennä vastauksesi seuraavasti:

- Määrittele abstrakti tietotyyppi (abstract data type) hakurakenne (dictionary).
- Nimeä ainakin yksi tasapainotettu hakupuu ja yksi hajautusrakenne.
- Mitä operaatioita tasapainotetut hakupuut tukevat paremmin kuin hajautusrakenteet? Perustele vastauksesi.
- Mitä operaatioita hajautusrakenteet tukevat paremmin kuin tasapainotetut hakupuut? Perustele vastauksesi.
- Miten jokin tasapainotettu hakupuu ja jokin hajautusrakenne voitaisiin yhdistää siten, että saataisiin aikaan vielä tehokkaampi hakurakenne?

## 4. Verkot (4p + 4p + 4p)

Määrittele seuraavat käsitteet. Anna jokaisesta myös esimerkki. Selitä lyhyesti miten kyseessä oleva laskennallinen ongelma voidaan ratkaista (käytä em. esimerkkiäsi ja kuvaile sen avulla lähtötilanne, mainitse jokin algoritmi, jolle se voidaan antaa syötteenä ja kerro lyhyesti miten ko. algoritmi tuottaa halutun lopputuloksen).

- Virityspuu
- Minimaalinen virityspuu
- Lyhimmin poluin virittävä puu