

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, koulutusohjelmasi, kurssikoodi ja -nimi "T-106.1220 Tietorakenteet ja algoritmit T" tai "T-106.1223 Tietorakenteet ja algoritmit Y", päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*.

1. Järjestäminen ja algoritmianalyysi (2p + 1p + 1p + 2p + 2p + 2p)

Seuraavassa on esitetty eräs algoritmi, joka järjestää kokonaislukutaulukon array, jonka koko on size.

```
/* 1*/ MinMaxSort(int[] array, int size) {  
/* 2*/     int minValue, maxValue, minPos, maxPos;  
/* 3*/     for (int i = 1; i <= size/2; i++) {  
/* 4*/         minValue = maxValue = array[i];  
/* 5*/         minPos = maxPos = i;  
/* 6*/         for (int j = i+1; j <= size-i+1; j++) {  
/* 7*/             if (array[j] < minValue) {  
/* 8*/                 minValue = array[j];  
/* 9*/                 minPos = j;  
/*10*/            }  
/*11*/            if (array[j] >= maxValue) {  
/*12*/                maxValue = array[j];  
/*13*/                maxPos = j;  
/*14*/            }  
/*15*/        }  
/*16*/        if (i == maxPos)  
/*17*/            maxPos = minPos;  
/*18*/        array[minPos] = array[i];  
/*19*/        array[i] = minValue;  
/*20*/        array[maxPos] = array[size-i+1];  
/*21*/        array[size-i+1] = maxValue;  
/*22*/    }  
/*23*/ }
```

n/2
n/2
+ tarkasteltava ja pienin swap
issain viimeiseksi

Vastaa lyhyesti, analyttisesti ja perustellen em. algoritmia koskeviin kysymyksiin:

7

- Mikä on algoritmin toimintaperiaate?
- Järjestääkö algoritmi avaimet kasvavaan vai laskevaan järjestykseen?
- Onko algoritmi stabiili?
- Mikä on algoritmin suoritusaika iso O -notaatioissa?

Vastaa seuraaviin kohtiin *piirrosesimerkein ja selittämällä* mitä kuvassa tapahtuu.

- Anna esimerkki, jossa algoritmi toimii väärin, jos rivien 16 ja 17 vertailua ja sijoitusta ei tehtäisi.
- Osoita, että esimerkkiäsi vastaavassa tilanteessa algoritmi toimii oikein (ts. kun rivit 16 ja 17 suoritetaan).

2. Hakurakenteet (2p + 2p + 2p + 2p + 4p)

a) Määrittele käsite *hakurakenne* (dictionary).

b) Määrittele käsite *hajautusmenetelmä* (hashing method).

c) Selitä jonkin hajautusmenetelmän *toimintaperiaate*. Voit selittää toimintaperiaatteen yhdessä seuraavan kohdan esimerkin kanssa.

d) Talleta seuraavat avaimet määrittelemääsi *hajautusrakenteeseen* yksitellen tässä järjestyksessä. Esitä riittävästi välivaiheita, jotta hajautusrakenteen täyttyminen käy selvästi ilmi.

T I E T O R A K E N N E

e) Vertaile keskenään *tasapainotettuja hakupuita ja hajautusmenetelmiä*, kun niitä tarkastellaan hakurakenteen toteutusmenetelminä. Esitä mielestäsi *neljä* tärkeintä *kriteeriä* tätä vertailua varten. Esitä vastauksesi luettelona siten, että kussakin kohdassa esität ja perustelet ensin vertailukriteerin ja sen jälkeen vertailun tulokset. Tämän kohdan vastauksen pituus saa olla korkeintaan 1 sivu.

3. Verkot (3p + 3p + 2p + 2p)

a) Nimeä ja selitä sanallisesti jonkin sellaisen *algoritmin toiminta*, jonka avulla voidaan vierailla verkon kaikissa solmuissa.

b) Kirjoita em. algoritmi *pseudokoodina* tai jotakin lausekielistä ohjelmointikieltä kuten C tai Java muistuttavalla notaatiolla.

c) Analysoi algoritmisi *aikavaativuus* iso O -notaatioissa, jos verkossa on V solmua ja E kaarta.

d) Missä järjestyksessä algoritmisi vierailee alla olevan verkon solmuissa, jos lähtösolmu on A?

A: B E
B: A C D I
C: ~~B~~ ~~E~~ ~~F~~ ~~I~~ H
D: B G
E: C A
F: C H
G: D
H: F C
I: C B

9/19
A = 0
B 1
C 2
D 3
E 4
F 5
G 6
H 7
I 8
J 9
K 10
L 11
M 12
N 13
O 14
P 15
Q 16
R 17
S 18
T 0