

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi, tutkinto-ohjelmasi, kurssikoodi ja kurssin nimi, päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*. Tentissä ei saa käyttää mitään ylimääräisiä apuvälineitä.

1) Kymmenen kysymystä (10 x 1p)

Tämä tehtävä on tentin pakollinen osa, josta on saatava vähintään 5p/10p, jotta loput tentistä tarkistetaan. Tämä tehtävä ei kuitenkaan yksistään riitä tentin läpäisyyn. Toisaalta viiteen pisteeseen ei edellytetä "täysin oikeaa vastausta" vaan oleellista on, että pystyt osoittamaan ymmärtäneesi tehtävän koodin toiminnan. Käytä siis aikaa perustelujen miettimiseen ja esittämiseen. Viittaa perusteluissa ohjelmakoodin rivinumeroihin, jos mahdollista.

Alla on annettu kaksi algoritmia (*bs1* ja *bs2*), jotka molemmat etsivät puolitushaulla järjestetystä taulukosta *table* alkion *x*. Lue ensin kaikki kysymyskohdat vastaamatta niihin ja sen jälkeen tutustu annettuihin koodinpätkiin erittäin huolella. Vastaa tämän jälkeen kaikkiin kysymyksiin ja käytä aikaa perustelujen pohtimiseen ja muotoilemiseen. Huomaa, että kaikissa kysymyksissä viitataan alla oleviin algoritmeihin ja, että vastaukset tulee perustella hyvin, tai siis *pisteet tulevat vain perusteluista!*

```
1 int bs1(int table[], int x) {
2     int low = 0;
3     int high = table.length - 1;
4     int mid;
5
6     while( low <= high )
7     {
8         mid = (low + high) / 2;
9
10        if (table[mid] < x)
11            low = mid + 1;
12        else if (table[mid] > x)
13            high = mid - 1;
14        else return mid;
15    }
16    return -1;
17 }
```

```
18 def bs2(table, x, low, high):
19     if low > high:
20         return -1
21     mid = (low + high) / 2
22     item = table[mid]
23     if item == x:
24         return mid
25     elif x < item:
26         return bs2(table,x,low,mid-1)
27     else:
28         return bs2(table,x,mid+1,high)
29
30
31
32
33
```

- Selitä* algoritmin *bs1* toiminta sanallisesti (ilman esimerkkiä). Huom! Pyri selittämään *miten* algoritmi ratkaisee ongelman. Älä selitä koodia rivi-riviltä.
- Selitä* nyt algoritmin *bs2* toiminta sanallisesti. Miten toiminta eroaa edellisestä?
- Anna esimerkki hausta*, jossa algoritmilla *bs1* etsitään alkion $x = 512$ taulukosta, jossa on alkion 1, 2, 4, 8, 16, 32, 64, 128, 256 ja 512. Vihje: Taulukoi mitä arvoja muuttujat *low*, *high* ja *mid* saavat ohjelman suorituksen edetessä. Mitä ohjelma palauttaa ja mitä laskennan tuloksena saadaan?
- Anna esimerkki hausta*, jossa algoritmilla *bs2* etsitään em. taulukosta arvoa $x = 100$. Vihje: taulukoi tässä muuttujat *low*, *high*, *mid* ja *item* kuten edellä. Mitä ohjelma palauttaa ja mikä on laskennan tulos tässä tapauksessa?
- Määrittele *bs1*:n ja *bs2*:n ns. "syötteen koko", ts. mistä muuttujista ja miten algoritmien suoritusajat riippuvat?
- Analysoi* *bs1*:n suoritusajaksi sen saaman syötteen koon n funktiona.
- Analysoi* *bs2*:n suoritusajaksi sen saaman syötteen koon n funktiona.
- Algoritmia *bs1* testattiin suurella aineistolla (haettiin aineiston pienintä alkion), jolloin sen suoritusajaksi saatiin noin 1 millisekunti. Tämän jälkeen aineiston koko kaksinkertaistettiin ja suoritusajaksi saatiin noin 2 millisekuntia. Jos aineisto jälleen kaksinkertaistettaisiin, niin kuinka pitkään arvioisit laskennan tällä kertaa kestävän? Perustele.
- Millaisia oletuksia ja reunaehtoja *bs2*:n virheetön ja tehokas suoritus asettaa taulukolle *table* ja taulukon sisältämille alkiolle?
- Perustele pitääkö väite paikkansa vai ei: *bs1* on tehokkaampi kuin *bs2*.
- Bonustehtävä: *Pohdi ja vertaile* *bs1*:n ja *bs2*:n muistinkäyttöä.

2) Terminologiaa (2p + 2p + 2p + 2p)

Määrittele seuraavat käsitteet (4 x 1p). Anna jokaisesta myös *esimerkki* (4 x 1p).

- Abstrakti tietotyyppi (ADT)
- Prioriteettijono
- Binäärikeko
- Kekoehto

3) Järjestämismenetelmät (3p + 3p + 2p)

Joudut valitsemaan algoritmin tehtävään, jossa tulee järjestää annettu aineisto. Mitä asioita (kriteereitä) huomioit tehdessäsi valintaa? Lue ensin koko tehtävänanto.

a) Valitse kolme (3) keskeistä kriteeriä joiden valossa tarkastelet tilannetta. *Perustele* miksi tai miten valitsemasi kriteerit liittyvät järjestämisiongelmaan.

b) Nimeä jokaisen kriteerin kohdalla erikseen ainakin yksi *algoritmi*, joka täyttää ko. kriteerin ja yksi joka ei täytä (algoritmien toimintaperiaatteita ei tarvitse selittää).

Anna b-kohdan vastauksesi matriisimuodossa, jossa sarakkeilla (3) on kriteerit ja niiden alapuolella riveillä (2) algoritmien nimiä, jotka täyttävät ja eivät täytä ko. kriteeriä.

c) Nimeä *algoritmi*, joka täyttää kaikki kriteereistäsi. Nimeä myös *algoritmi*, joka ei täytä kahta kriteereistäsi.

4) Tasapainotetut hakupuut ja hajautus (2p + 2p + 2p + 2p)

a) Selitä tasapainotettujen hakupuiden (*balanced search trees*) peruseriaatteet.

b) Selitä hajautuksen (*hashing*) peruseriaatteet.

c) Arvioi yleisesti hakurakenteisiin liittyvien operaatioiden aikakompleksisuutta hajautusmenetelmien yhteydessä. Mitkä operaatiot ovat tehokkaampia hajautusmenetelmissä kuin esim. hakupuissa? Entä mitkä operaatiot ovat tehottomampia hajautusmenetelmissä kuin esim. hakupuissa? Pohdi sekä keskimääräistä että pahinta tapausta.

d) Arvioi hajautusmenetelmien hyviä ja huonoja puolia. Minkälaisiin sovelluksiin kurssilla esitetyt perushajautusmenetelmät soveltuvat tai eivät sovellu hyvin?

5) Verkkoalgoritmit (4p + 4p)

a) Selitä joko Primin tai Dijkstran algoritmin toiminta sanallisesti. Käytä apuna sopivaa esimerkkiä.

b) Toimivatko em. algoritmit, jos painotetussa verkossa on negatiivisia särmäjä? Perustele näkemyksesi molempien algoritmien osalta erikseen. Jos jompi kumpi algoritmi toimii mielestäsi virheellisesti, anna esimerkki.