

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, tutkinto- tai koulutusohjelmasi, kurssikoodi ja kurssin nimi, päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*.

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 (`fact_1` ja `fact_2`), jotka laskevat kertoma-funktion (*factorial*). 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 väittämät voi perustella yhtä hyvin vastaamalla kyllä tai ei, joten pisteet tulevat vain *perusteluista!*

```
1 int fact_1(int n)                8 int fact_2(int n)
2 {                                9 {
3     if (n<2)                    10     int i, fact;
4         return 1;                11     fact = 1;
5     else                          12     for (i=1; i<=n; i++)
6         return fact_1(n-1)*n;    13         fact = fact*i;
7 }                                14     return fact;
                                   15 }
```

- Selitä* algoritmin 1 (ts. `fact_1`) toiminta *esimerkin avulla*.
- Selitä* algoritmin 2 (ts. `fact_2`) toiminta *esimerkin avulla*.
- Missä järjestyksessä ja kuinka monta kertolaskua* `fact_1` suorittaa? *Anna esimerkki*, kun algoritmia kutsutaan parametrillä $n=3$.
- Missä järjestyksessä ja kuinka monta kertolaskua* `fact_2` suorittaa? *Anna esimerkki*, kun algoritmia kutsutaan parametrillä $n=3$.
- Analysoi* algoritmin 1 suoritus aika sen saaman syötteen koon n funktiona.
- Analysoi* algoritmin 2 suoritus aika sen saaman syötteen koon n funktiona.
- Perustele* pitääkö väite paikkansa vai ei: algoritmi 1 on tehokkaampi kuin algoritmi 2.
- Perustele* pitääkö väite paikkansa vai ei: algoritmi 1 laskee saman funktion kuin algoritmi 2.
- Mikä olisi algoritmin 1 kertolaskujen suoritusjärjestys*, jos riviä 6 muutettaisiin muotoon ”return $n*fact_1(n-1)$ ”? *Anna esimerkki*.
- Algoritmi 2 käyttää *for*-silmukkaa. Voitaisiinko se korvata jollakin toisella silmukalla? Perustele joko miksi ei tai anna esimerkki miten (kirjoita algoritmi uusiksi).

Bonustehtävä:

- Pohdi ja vertaile* algoritmien 1 ja 2 muistinkäyttöä.

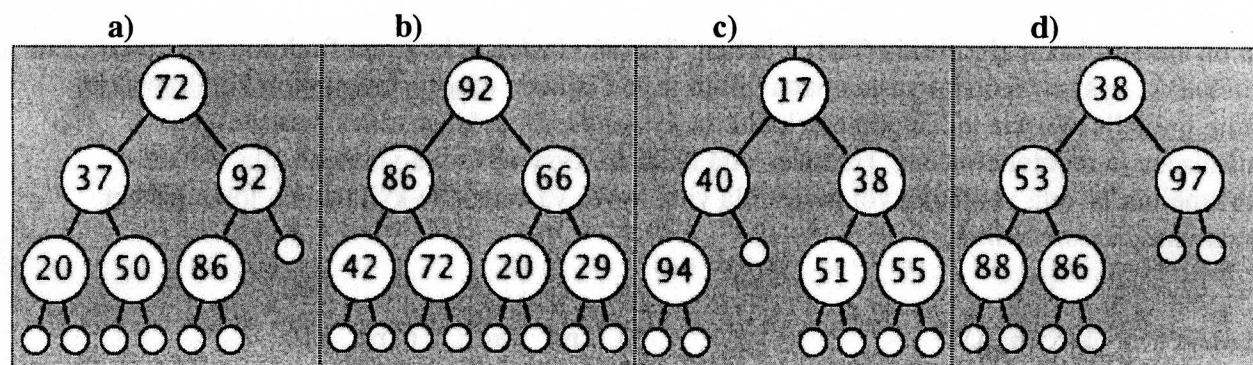
2) Terminologiaa (2p + 2p + 2p + 2p)

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

- Stabiili (*stable*) järjestämismenetelmä
- Paikoillaan tapahtuva (*in place*) järjestäminen
- Inversio (*inversion*)
- Valikointi (*selection problem*)

3) Prioriteettijonot (2p + 2p + 2p + 2p)

Seuraavissa kuvissa on neljä binääripuuta (*binary tree*) a, b, c ja d. Vastaa jokaisen kohdalla erikseen onko kuvan tietorakenne *minimikeko* (*MinHeap*) (kyllä/ei, 1p) ja *perustele* miksi (1p).



4) Tasapainotetut hakupuut (2p + 6p + 4p)

- Määrittele käsite *tasapainotettu hakupuut* (*balanced search tree*).
- Millaisia erilaisia tasapainotettuja hakupuuta on olemassa? Miten ne eroavat toisistaan? Käsittele tässä kolmea eri tietorakennetta.
- Anna esimerkki (*piirrä ja merkitse* (nimeä) välivaiheet ennen ja jälkeen *jokaisen tasapainotusoperaation*) jonkin tasapainotetun hakupuun toiminnasta, kun siihen lisätään alkioit $11, 5, 2, 18, 7, 4, 3, 6, 10, 5, 6, 5$ tässä järjestyksessä. Samanarvoiset alkioit sijoitetaan aina oikeanpuoleiseen haaraan.

5) Verkkoalgoritmit (7p + 3p)

- Kirjoita algoritmi*, joka etsii annetun suuntaamattoman verkon yhtenäiset osaverkot (*connected sub-graphs*).
- Analysoi* algoritmisi suoritus aika.

Voit esittää algoritmin jollakin tunnetulla ohjelmointikielellä tai käyttää vapaampaa pseudokieliesitystä. Selitä algoritmin toiminta kuitenkin myös sanallisesti.

6) Palaute (2p)

Anna asiallista *kurssipalautetta* täyttämällä kurssin kotisivun kautta löytyvä palautelomake 18.5. mennessä, niin saat tenttiin 2 lisäpistettä. Lisäpisteillä ei voi kuitenkaan nostaa hylättyä arvosanaa hyväksytyksi.