

Kirjoita jokaiseen paperiin oma nimi, oppilasnumero, tutkinto-ohjelma, kurssikoodi ja kurssin nimi, päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*. Numeroi palauttamasi paperit juoksevilla numeroinnilla. 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 (pow1 ja pow2), jotka molemmat laskevat potenssifunktion (x^n) kokonaisluvuille x ja n . 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 def pow1(x, n):
2     if (n == 0):
3         return 1
4     if (n == 1):
5         return x
6     if (n % 2 != 0): // pariton
7         return pow1(x*x, n//2)*x
8     else:
9         return pow1(x*x, n//2)
10
11 def pow2(x, n):
12     p = 1
13     for i in range(0, n):
14         p = p * x
15     return p
```

- Selitä* algoritmin pow1 toiminta sanallisesti (ilman esimerkkiä). Huom! Pyri selittämään *miten* algoritmi toimii yleisesti. Älä selitä koodia rivi-riviltä.
- Selitä* algoritmin pow2 toiminta sanallisesti (ilman esimerkkiä). Miten se eroaa edellisestä?
- Millä ohjelmariveillä ja kuinka monta kertolaskua pow1 suorittaa? Anna esimerkki, kun suoritetaan pow1(2, 9).
- Millä ohjelmariveillä ja kuinka monta kertolaskua pow2 suorittaa? Anna esimerkki, kun suoritetaan pow2(2, 9).
- Analysoi algoritmin 1 aikavaatimus sen saaman syötteen koon n funktiona.
- Analysoi algoritmin 2 aikavaatimus sen saaman syötteen koon n funktiona.
- Perustelee pitääkö väite paikkansa vai ei: algoritmi 1 on tehokkaampi kuin algoritmi 2.
- Perustelee pitääkö väite paikkansa vai ei: algoritmi 1 laskee saman funktion kuin algoritmi 2.
- Mikä olisi algoritmin 1 kertolaskujen suoritusjärjestys, jos riviä 7 muutettaisiin muotoon
”return x * pow1(x*x, n//2); else”? Anna esimerkki.

- j) Algoritmi 2 käyttää *for*-silmukkaa. Voitaisiinko se korvata jollakin toisella silmukalla? Perustele joko miksi ei tai anna esimerkki miten (kirjoita algoritmi uusiksi).

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

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

- a) Abstrakti tietotyyppi (ADT, Abstract Data Type)
- b) Rekursio (Recursion)
- c) Binääripuu (Binary tree)
- d) Prioriteettijono (Priority queue)

3) Lomitusjärjestäminen (2p + 2p + 2p + 2p + 2p = 10p)

- a) *Selitä* sanallisesti miten *lomitus (merging)* toimii. *Havainnollista* selitystäsi käyttämällä sopivia piirroksia.
- b) *Selitä* sanallisesti miten *lomitusjärjestäminen (mergesort)* toimii. *Havainnollista* selitystäsi käyttämällä sopivia piirroksia.
- c) Soveltuuko lomitusjärjestäminen *taulukoille (arrays)* paremmin kuin *linkitetyille listoille (linked lists)*? *Perustele* vastauksesi (vihje: kysymykseen ei ole yhtä oikeaa kyllä/ei-vastausta, pyri hakemaan vähintään kaksi erilaista näkökulmaa, pisteet tulevat vain perusteluista).
- d) *Analysoi* lomitusjärjestämisen *aikavaatimus* pahimmassa tapauksessa em. selityksen perusteella.
- e) Tarkastellaan listaa, jossa on seuraavat alkiot: 3, 1, 6, 2, 5, 7, 4, 8, 9, 0. *Esitä välivaiheittain* miten järjestäminen etenee lomitusjärjestämisessä.

4) Hakurakenteet (2p + 2p + 2p + 4p = 10p)

- a) *Määrittele* käsite *hakurakenne (search structure)*. Anna *esimerkkejä* hakurakenteista.
- b) *Selitä* tasapainotettujen hakupuiden (*balanced search trees*) *peruseriaatteet*.
- c) *Selitä* hajautusmenetelmien (*hashing*) *peruseriaatteet*.
- d) *Analysoi* hakurakenteiden keskeisten operaatioiden aikakompleksisuudet tasapainotettujen hakupuiden ja hajautusmenetelmien osalta. *Vertaile* tasapainotettuja hakupuita ja hajautusmenetelmiä keskenään eri operaatioiden keskimääräisen sekä pahimman tapauksen aikavaatimusten perusteella.

5) Ajankäyttö (0p)

Arvioi tenttiin vastaamiseen käyttämäsi aika noin 15 minuutin tarkkuudella.