

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, tutkinto-ohjelmasi, 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 (fib1 ja fib2), jotka molemmat laskevat n :nen Fibonaccin luvun. 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 fib1(int n) {
2     if (n <= 2)
3         return 1;
4     else
5         return fib1(n-1) +
6             fib1(n-2);
7 }
8 def fib2(n):
9     x = 1
10    y = 1
11    z = 1
12    while n > 2:
13        x = y
14        y = z
15        z = x + y
16        n = n - 1
17    return z

```

- Selitä* algoritmin `fib1` toiminta sanallisesti. Huom! Pyri selittämään *miten* algoritmi ratkaisee ongelman. Älä selitä koodia rivi-riviltä.
- Selitä* algoritmin `fib2` toiminta sanallisesti. Miten ratkaisu eroaa edellisestä?
- Millä ohjelmariveillä ja kuinka monta* yhteenlaskua `fib1` suorittaa? *Anna esimerkki*, kun algoritmia kutsutaan parametrilla $n=6$.
- Millä ohjelmariveillä ja kuinka monta* yhteenlaskua `fib2` suorittaa? *Anna esimerkki*, kun algoritmia kutsutaan parametrilla $n=6$.
- 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: Molemmat algoritmit laskevat funktion $f(n) = f(n-1) + f(n-2)$, jossa $f(1) = f(2) = 1$.
- Perustele* pitääkö väite paikkansa vai ei: Molempien algoritmien suoritus aika kasvaa eksponentiaalisesti parametrin n funktiona.
- Pohdi ja vertaile* algoritmien 1 ja 2 muistinkäyttöä.
- Bonustehtävä: Miten *kultainen leikkaus* liittyy Fibonaccin lukuihin?

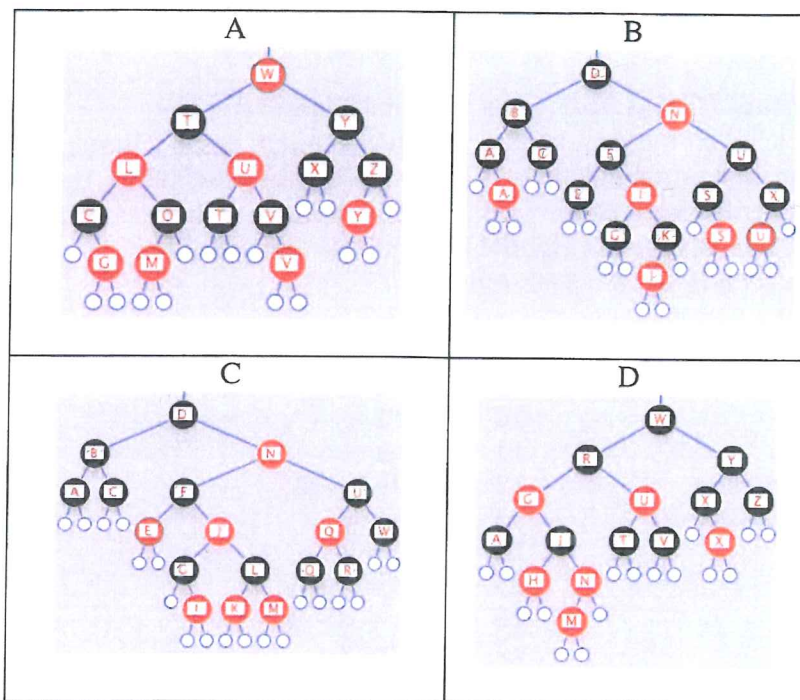
2) Terminologiaa (2p + 2p + 2p + 2p)

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

- Hajautusfunktio
- Yhteentörmäysten käsittelymekanismi
- Valikointi (Selection)
- Stabiili järjestämismenetelmä

3) Hakurakenteet: Punamusta puu (4p + 1p + 3p)

- Määrittele punamusta puu (4 sääntöä).
- Mikä kuvien vaihtoehdoista on validi punamusta puu?
- Mitä punamustan puun sääntöjä muut vaihtoehdot rikkovat?



4) Prioriteettijonot (8p)

Kirjoita essee aiheesta *prioriteettijonot*. Määrittele esseessä käsite prioriteettijono, anna esimerkkejä sellaisten toteutuksista ja vertaile toteutuksia keskenään. Anna myös esimerkkejä miten tai mihin prioriteettijonoja voi soveltaa.

5) Lyhimpien polkujen virityspuun etsiminen (4p + 4p)

a) Esitä jonkin reitinhakualgoritmin toiminta, jolla voidaan laskea lyhimpien polkujen virityspuu. Mikä on esittämäsi algoritmin aikakompleksisuus, jos graafissa on V solmua ja E kaarta? Perustele tuloksesi.

b) Tarkastellaan painotettua suuntaamatonta verkkoa, johon kuuluvat solmut A-F ja särmät AB(2), AD(2), AE(6), BC(4), BE(3), CE(1), CF(4), DE(1), EF(3). Särmien painot on esitetty särmän jälkeen sulussa. Esitä miten annetulle verkolle voidaan löytää lyhin reitti solmusta A solmuun F edellä kuvaamaasi algoritmia soveltamalla.