

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

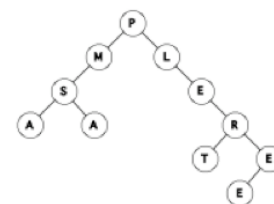
1) Kymmenen kysymystä (10 x 1p)

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 (traverse1 ja traverse2), jotka ovat binääripuun läpikäyntialgoritmeja. Lue ensin kaikki kysymyskohdat vastaamatta niihin ja sen jälkeen tutustu annettuihin koodinpätkiin 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 void traverse1(node t) {
2     if (t != NULL) {
3         visit(t);
4         traverse1(t->left);
5         traverse1(t->right);
6     }
7 }
8
9
10
11 void traverse2(node root) {
12     stack.push(root);
13     while (stack.notEmpty()) {
14         node next = stack.pop();
15         visit(next);
16         if (next->right != NULL)
17             stack.push(next->right);
18         if (next->left != NULL)
19             stack.push(next->left);
20     }
21 }
```

- Selitä* algoritmin 1 (traverse1) toiminta *esimerkin avulla*.
- Selitä* algoritmin 2 (traverse2) toiminta *esimerkin avulla*.
- Missä järjestyksessä* traverse1 vierailee kuvan 1 solmuissa?
- Missä järjestyksessä* traverse2 vierailee kuvan 1 solmuissa?
- Analysoi* algoritmin 1 suoritus aika sen syötteenä saaman puun koon (n solmua) funktiona.
- Analysoi* algoritmin 2 suoritus aika sen syötteenä saaman puun koon (n solmua) funktiona.
- Perustele* pitääkö väite paikkansa vai ei: algoritmi 1 on tehokkaampi kuin algoritmi 2.
- Perustele* pitääkö väite paikkansa vai ei: algoritmi 1 suorittaa vähemmän vertailuja (*if-lauseet*) kuin algoritmi 2.
- Mikä* olisi algoritmi 1:n tuottama puun läpikäyntijärjestys, jos rivit 3 ja 4 vaihtaisivat paikkaa? *Anna esimerkki*.
- Algoritmi 2 käyttää *while*-silmukkaa. Voitaisiinko se korvata jollakin toisella silmukalla? Perustele joko miksi ei tai anna esimerkki miten (kirjoita algoritmi uusiksi).



Kuva1: Binääripuu

Bonustehtävä:

- Pohdi ja vertaile* algoritmien 1 ja 2 muistinkäyttöä.

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

Seuraavassa määrittely- ja piirrostehtävässä sinun tulee jatkaa aina siitä kuvasta, joka on edellisen kuvan lopputulos ja piirtää uusi kuva. Tehtävänäsi on määrittellä kursivoitu käsite sanallisesti ja antaa esimerkki piirtämällä.

- Sykli. Piirrä minimäärällä solmuja suuntaamatonta verkkoa, jossa on sykli.
- Epäyhtenäinen verkko. Lisää solmuja ja kaaria siten, että saat epäyhtenäisen verkon.
- Leikkaussolmu. Tee verkosta yhtenäinen lisäämällä siihen yksi tai useampi leikkaussolmu.
- Kahdesti yhtenäinen verkko. Lisää solmuja ja kaaria siten, että saat kahdesti yhtenäisen verkon.

3) Järjestäminen (3 x 3p + 1p)

Valittaessa järjestämismenetelmää erääseen sovellukseen tutkittiin kolmea eri menetelmää, joiden perusversiot löytyivät valmiina käytetyn ohjelmointikielen kirjastoista: lisäysjärjestäminen, kekojärjestäminen ja lomitusjärjestäminen. Valintakriteereitä oli myös kolme: menetelmän tuli olla stabiili, se ei saanut olla rekursiivinen, eikä se saanut käyttää ylimääräistä muistitilaa. Analysoi annetut menetelmät annettujen kriteereiden valossa. Anna vastauksesi 3*3 matriisina, jossa riveillä on eri menetelmät ja sarakkeilla kriteerit. Merkitse kunkin menetelmän kohdalle täyttääkö se (perusversio) annetun kriteerin vai ei. Mitä menetelmään suosittelet tässä tapauksessa?

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

- Määrittele käsite tasapainotettu hakupuun (*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) Essee (10p)

Kirjoite essee joko aiheesta *prioriteettijonot* tai aiheesta *hajautusmenetelmät*. Esseen suosituspituus on alle 2 sivua.