Vastaa tentin järjestäjän antamalle paperille (ei kysymyspaperille). Kirjoja, laskinta tms. ei saa olla tentissä. Kukin tehtävä on 1 tai 2 pisteen arvoinen. Muiden kuin koodaa tai piirrä -tehtävien mallivastaukset ovat melko lyhyet, 1, . . . , 4 riviä.

1. Kirjoita helppotajuinen pseudokoodi tai ohjelmanpätkä, joka toteuttaa kuvauksen, tai perustele, että sellaista ei ole olemassa: Sen suoritusaika kaikissa tapauksissa on $O(n^2)$, mutta ei ole kaikissa tapauksissa $\Theta(n^2)$.

2. Kirjoita tai perustele kuten tehtävässä 1: Sen suoritusaika kaikissa tapauksissa on $\Theta(n^2)$, mutta ei ole kaikissa tapauksissa $O(n^2)$.

3. Kirjoita tai perustele kuten tehtävässä 1: Sen suoritusaika hitaimmassa tapauksessa on $O(n)$, mutta ei ole kaikissa tapauksissa $O(n)$.

4. Kirjoita tai perustele kuten tehtävässä 1: Sen suoritusaika hitaimmassa tapauksessa on $\Theta(n)$, mutta ei ole kaikissa tapauksissa $\Theta(n)$.

5&6. Kirjoita tehokas pseudokoodi tai ohjelmanpätkä, joka poistaa taulukosta A tarpeettomat alkiot ja säilyttää muiden alkioiden järjestyksen ennallaan. Aliohjelma `T(x)` palauttaa `true` jos ja vain jos `x` on tarpeeton. Jos esimerkiksi A on [`t`, `i`, `e`, `t`, `o`, `r`, `a`, `k`, `e`, `n`, `n`, `e`] ja `T(x)` on `return x == 'e';` , niin A:n pitää lopuksi olla [`t`, `i`, `t`, `o`, `r`, `a`, `k`, `n`, `n`].

7. Mitä tarkoittaa, että järjestämisalgoritmi on vakaa (stable)?

8. Tarkastellaan binäärikekoa (binary heap). Vähintään ja enintään kuinka monella solmulla on täsmälleen yksi lapsi? Mitä solmujen määrästä voidaan päätellä, jos jollakin solmulla on täsmälleen yksi lapsi?

9. Kerro yksi kekojärjestämisen (heapsort) vahvuus ja yksi heikkous.

10. Kuinka paljon keosta poistaminen käyttää aikaa hitaimmillaan ja nopeimmillaan?

11. Mikä on lopputulos, kun keosta $[8,6,5,2,4,2]$ poistetaan yksi alkio?

12. Etsi ohjelmointivirhe oheisesta aliohjelmasta.

```
1  void poista_keosta(){
2    if( keko.size() < 1 ){ return; }
3    unsigned h = keko.size() - 1, i = 0, j = 1;
4    while( true ){
5      if( j+1 < h && keko[j+1] <= keko[j] ){ ++j; }
6      if( j >= h || keko[j] <= keko[h] ){ break; }
7      keko[i] = keko[j]; i = j; j = 2*i + 1;
8    }
9    keko[i] = keko[h]; keko.resize(h);
10 }
```

käännä

13. Jos kasvavan taulukon alkioille varattu muistialue loppuu kesken, niin ohjelman suoritusympäristö kasvattaa sitä automaattisesti. Miksi se ei kasvata sitä joka kerta aina samalla määrällä (esim. 100 alkiolla)?

14. Minkä verran kasvavan taulukon muistialue tyypillisesti kasvaa kerrallaan?

15&16. Pikajärjestämisen (Quicksort) käyttämä ositus jakaa taulukon kahteen tai kolmeen osaan. Mitä ominaisuuksia osituksen lopputuloksella täytyy olla?

17&18. Mikä on pikajärjestämisen perusversion huonoimman tapauksen lisämuistin kulutus, millaisella muutoksella sitä voi olennaisesti parantaa, ja kuinka suuri on parannetun version huonoimman tapauksen lisämuistin kulutus?

19. Miten Dijkstran algoritmi ylläpitää reitin etsinnän aikana tietoa, jonka avulla reitti voidaan lopuksi tulostaa?

20. Miten A* reitinetsintäalgoritmi eroaa Dijkstran algoritmista?

21. Mikä on lomitusjärjestämisen (merge sort) pahin heikkous?

22. Piirrä hajautustaulu, jonka hajautusfunktiona on x mod 5 eli `x % 5`, ja jossa on alkiot 211, 23, 5, 20, 24, 8 ja 108.

23. Kerro yksi hajautustaulujen etu verrattuna punamustiin puihin.

24. Kerro yksi punamustien puiden etu verrattuna hajautustauluihin.

25. Miten käsite ”binäärihakupuu” eroaa käsitteestä ”binääripuu”?

26. Binääripuun solmun korkeus on mahdollisimman pitkän solmusta johonkin lehteen vievän polun pituus. Piirrä binääripuu, jossa on viisi solmua, ja jonka juuren korkeus on mahdollisimman pieni.

27. Piirrä binääripuu, jossa on viisi solmua, ja jonka juuren korkeus on mahdollisimman suuri.

28&29. Osoitin `os` on tyyppiä `solmu_os` ja osoittaa binääripuun solmuun. Kirjoita pseudokoodi tai ohjelmanpätkä, joka palauttaa kyseisen solmun korkeuden.

30. Minkä arvelet kurssilla olleen kaikkein hyödyllisintä tulevalle työurallesi?

loppu