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 $\Theta(n^2)$.

   ```
   for( int i = 0; i < n; ++i ){}
   ```

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

   Ei ole olemassa, koska jokin on $\Theta(f(n))$ jos ja vain jos se on $O(f(n))$ ja $\Omega(f(n))$.

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

   Ei ole olemassa, koska $O(n)$ ilmaisee ylärajan. Jos jokin on $O(n)$, niin myös sitä nopeampi on $O(n)$.

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

   ```
   if( n % 2 ){ for( int i = 0; i < n; ++i ){}}
   ```

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`].

```
int j = 0;
for( int i = 0; i < A.size(); ++i ){
  if( ! T( A[i] ) ){ A[j++] = A[i]; }
}
A.resize(j);
```

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

   Se ei koskaan vaihda sellaisten alkioiden järjestystä, joilla on keskenään yhtäsuuret avaimet.

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?

   Täsmälleen yksi lapsi on yhdellä tai ei yhdelläkään solmulla. Jos jollakin solmulla on täsmälleen yksi lapsi, niin solmujen määrä on parillinen.

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

   Vahvuus: Lisämuistin tarve on vain vakio. Heikkous: Ei ole vakaa.

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

    $O(\log n)$ ja $O(1)$, missä n on alkioiden määrä keossa.

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

    $[6,4,5,2,2]$

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 }
```

    Rivillä 4 pitäisi lukea >= (tai >) eikä <=.

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)?

    Ison taulukon täyttämisestä tulisi hidasta. Se veisi aikaa $\Theta(n^2)$, koska joka kasvatuksessa kopioidaan koko vanha sisältö.

14. Minkä verran kasvavan taulukon muistialue tyypillisesti kasvaa kerrallaan?

    Uusia alkioita on tyypillisesti sama määrä kuin alkioita oli entuudestaan.

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

    Kukin alkuosan alkio on enintään yhtäsuuri kuin mikään muiden osien alkio. Kukin keskiosan alkio on enintään yhtäsuuri kuin mikään loppuosan alkio. Yhteensä osat sisältävät alkuperäiset alkiot, eikä muuta. Ainakin kaksi osaa on epätyhjiä.

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?

Perusversiolla se on $\Theta(n)$, mutta se putoaa arvoon $\Theta(\log n)$ järjestämällä osituksen tuottamista osataulukoista isoin rekursion sijaan silmukalla.

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

    Jokaisessa solmussa (paitsi lähtösolmussa) on linkki reitin edelliseen solmuun.

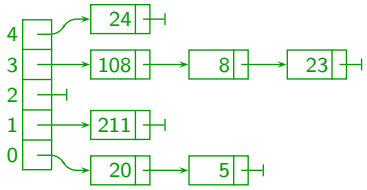
20. Miten A* reitinetsintäalgoritmi eroaa Dijkstran algoritmista?

    A* käyttää apuna arviota jäljellä olevasta matkasta maaliin.

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

    Lineaarinen lisämuistin tarve.

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.

    Alkion löytää avaimen perusteella keskimäärin ajassa $O(1)$.

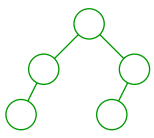
24. Kerro yksi punamustien puiden etu verrattuna hajautustauluihin.

    Alkioita voi selata avainten mukaisessa suuruusjärjestyksessä.

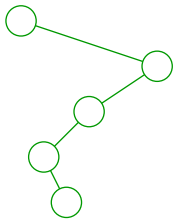
25. Miten käsite ”binäärihakupuu” eroaa käsitteestä ”binääripuu”?

    Binäärihakupuun jokaisessa solmussa on avain. Kunkin solmun vasemman alipuun solmujen avaimet ovat enintään yhtäsuuret ja oikean alipuun solmujen avaimet vähintään yhtäsuuret kuin solmun oma avain.

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.

```
int korkeus( solmu_os os ){
  if( !os ){ return -1; }
  int k = korkeus( os->vasen ), h = korkeus( os->oikea );
  if( h > k ){ k = h; }
  return k+1;
}
```

30. Minkä arvelet kurssilla olleen kaikkein hyödyllisintä tulevalle työurallesi?

loppu