

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. Oheisen algoritmin nimi on INSERTIONSORT. Mikä on sen suoritus aika  $\Theta$ -merkinnällä ilmaistuna, jos kaikki alkioit ovat yhtäsuuret? Perustelee.
 

```

1  for i := 1 to A.koko - 1 do
2    apu := A[i]; j := i
3    while j > 0 && A[j - 1].x > apu.x do
4      A[j] := A[j - 1]; j = j - 1
5    A[j] := apu
      
```
2. Oletetaan, että  $A$ :ssa on ensin  $\sqrt{n}$  ykköstä ja loput alkioit ovat nollia. Mikä on INSERTION-SORT:n suoritus aika? Ei tarvitse perustella.
3. Miten INSERTIONSORT huononee tai paranee, jos rivi 1 muutetaan muotoon **for**  $i := 0$  to  $A.koko - 1$  **do** ?
4. Miten INSERTIONSORT huononee tai paranee, jos rivi 3 muutetaan muotoon **while**  $j \geq 0$  &&  $A[j - 1].x > apu.x$  **do** ?
5. Miten INSERTIONSORT huononee tai paranee, jos rivi 3 muutetaan muotoon **while**  $j > 0$  &&  $A[j - 1].x \geq apu.x$  **do** ?
6. Anna INSERTIONSORT:n ulommalle silmukalle invariantti.
7. Anna INSERTIONSORT:n sisemmälle silmukalle invariantti.
8. Tarkastellaan binäärihekoa (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. Piirrä hajautustaulu, jonka hajautusfunktiona on  $x \bmod 5$  eli  $x \% 5$ , ja jossa on alkioit 211, 23, 5, 20, 24, 8 ja 108.
13. Kerro yksi hajautustaulujen etu verrattuna punamustiin puihin.
14. Kerro yksi punamustien puiden etu verrattuna hajautustauluihin.
15. Miten käsite "binäärihakupuu" eroaa käsitteestä "binääripuu"?
16. 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. Piirrä binääripuu, jossa on viisi solmua, ja jonka juuren korkeus on mahdollisimman suuri.

**käännä**

17&18. Minkä tehtävän Dijkstran algoritmi ratkaisee, ja miten se toimii?

Juoksukisan reitin varrelle tarvitaan  $h$  huoltopistettä. Koska ne tarvitsevat tilaa yms., niitä ei voi laittaa minne tahansa. Mahdollisia paikkoja on  $n - 2$  kappaletta, missä  $n \geq 2$ . Ne on ilmoitettu metreissä juoksureittiä pitkin taulukossa `sijainnit`, missä `sijainnit[0] = 0` on lähtö, `sijainnit[1]` on ensimmäinen mahdollinen paikka, ..., `sijainnit[n - 2]` on viimeinen mahdollinen ja `sijainnit[n - 1]` on maali. *Osuus* on lähdöstä ensimmäiseen huoltopisteeseen, huoltopisteestä seuraavaan tai viimeisestä huoltopisteestä maaliin. Pisin osuus halutaan minimoida.

19. Olkoon `sijainnit = [0,9,18,27,35,44,52]`. Jos huoltopisteitä voidaan perustaa enintään 3, niin mitkä ovat niiden etäisyydet lähdöstä?

20&21. Kirjoita yksinkertainen algoritmi, joka selvittää, onko mahdollista, että pisin osuus on pituudeltaan enintään  $h$ , kun huoltopisteiden määrä on enintään  $hpm$ .

Oheinen algoritmi valitsee  $hpm$  huoltopistettä välille lähtö, ..., paikka. Aluksi `lyhin[paikka][0]` on `sijainnit[paikka]` ja muut `lyhin[paikka][hpm]` ovat ääretön.

```
1 int paras( int paikka, int hpm ){
2     if( lyhin[ paikka ][ hpm ] == ääretön ){
3         for( int i = 0; i <= paikka; ++i ){
4             int max = sijainnit[ paikka ] - sijainnit[ i ];
5             int alku = paras( i, hpm - 1 );
6             if( alku > max ){ max = alku; }
7             if( max < lyhin[ paikka ][ hpm ] ){
8                 lyhin[ paikka ][ hpm ] = max; valittu[ paikka ][ hpm ] = i;
9             }
10        }
11    }
12    return lyhin[ paikka ][ hpm ];
13 }
```

22. Ilmoita algoritmin suoritus-aika  $n:n$  ja  $h:n$  funktiona  $O$ -merkinnällä.

23. Perustele edellinen vastauksesi.

24. Miten algoritmia voi nopeuttaa puolitus-haun avulla? Sanallinen kuvaus riittää, ei tarvita pseudo- eikä ohjelmakoodia.

25. Ilmoita edellisen kohdan mukaisesti nopeutettu suoritus-aika  $O$ -merkinnällä.

26&27. Kirjoita algoritmi, joka tulostaa huoltopisteiden etäisyydet lähdöstä.

28. Minkä arvelet kurssilla olleen kaikkein hyödyllisintä tulevalle työurallesi?

29&30. Mitä asiaa olisit halunnut kurssiin lisätä, ja miksi? Minkä asian voisi jättää kurssista pois, jotta saataisiin tilaa asialle, jota haluaisit lisätä?

**loppu**