

Vastaa tentin järjestäjän antamalle paperille. 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, tyypillisesti 1, ..., 3 riviä.

1. Millä nimellä alla oleva algoritmi (ja sen kaltaiset) tunnetaan?

```
1  int ala = 0, yla = A.size();
2  while( ala < yla ){
3      int vali = (ala + yla)/2;
4      if( A[vali] >= x ){ yla = vali; }else{ ala = vali+1; }
5  }
```

2. Mikä ehto A:n täytyy toteuttaa, jotta algoritmi tekisi mitä sen pitää tehdä?
3. Miksi rivillä 1 lukee `yla = A.size()`; eikä `yla = A.size() - 1`;
4. Oletetaan, että riviltä 4 poistetaan ”+1”. Anna esimerkki syötteestä, jolla algoritmi toimii väärin, ja kerro miten se toimii väärin.

Oletetaan, että keossa on suurin ylinnä.

5. Mitkä palvelut keko tarjoaa nopeasti?
6. Piirrä [V, U, O, R, O, T, E, L, L, A] puuna ikään kuin se olisi keko.
7. Perustele, että [V, U, O, R, O, T, E, L, L, A] ei ole keko.
8. Jos keon korkeus on h , niin kuinka monta solmua siinä on vähintään ja enintään?

Sekalaisia kysymyksiä

9. Kirjoita mikä tahansa ohjelmanpätke, jonka suoritusaika on hitaimmillaan $\Theta(n^2)$ ja nopeimmillaan $\Theta(n)$.
10. Kirjoita tälle invariantti: `i = 0; while( i < n && A[i] < 0 ){ ++i; }`
11. Miksi ihmisen nimen ensimmäisen kirjaimen käyttö hajautusfunktiona on huono ajatus?
12. Hajautusfunktiona on $n \bmod 5$. Piirrä hajautustaulu, jossa on 4211, 13, 6, 2025, 128 ja 256.
- 13&14. Miksi hajautuslistojen määrän ei pidä olla liian suuri ja miksi ei liian pieni?
15. Mitä tarkoittaa ahne algoritmi?
16. Mitä tarkoittaa dynaaminen ohjelmointi?
17. Kerro lyhyesti, miten kasvava taulukko (esimerkiksi C++:n vector) toimii.

Käännä

Olkoon $|V|$ suunnatun graafin solmujen määrä ja $|E|$ kaarten määrä. Seuraavaksi on alla olevaa algoritmia koskevia kysymyksiä. Muuttujat on alustettu järkevästi.

```
1 solmut[ lahto ].etaisyys = 0;
2 unsigned kierros = 1; bool muuttui = true;
3 while( kierros <= solmut.size() && muuttui ){
4     ++kierros; muuttui = false;
5     for( unsigned kk = 0; kk < kaaret.size(); ++kk ){
6         kaarityyppi & KK = kaaret[ kk ]; solmutyyppi & SS = solmut[ KK.karki ];
7         if( solmut[ KK.hanta ].etaisyys == aareton ){ continue; }
8         pituustyyppi uusi = solmut[ KK.hanta ].etaisyys + KK.pituus;
9         if( uusi < SS.etaisyys ){
10             SS.etaisyys = uusi; SS.reitti = kk; muuttui = true;
11         }
12     }
13 }
```

18. Piirrä graafi, jossa on 5 solmua ja 8 kaarta, jossa lahto:sta on reitti jokaiseen solmuun, ja jolla algoritmi on hidas. Kerro, miksi se on hidas.
 19. Piirrä graafi, jossa on 5 solmua ja 8 kaarta, jossa lahto:sta on reitti jokaiseen solmuun, ja jolla algoritmi on nopea. Kerro, miksi se on nopea.
 20. Ilmoita algoritmin suoritusaika O -merkinnällä ja Ω -merkinnällä.
 21. Perustele tehtävän 20 O -vastauksesi.
 22. Perustele tehtävän 20 Ω -vastauksesi.
 23. Miten algoritmin lopetettua voidaan päätellä, onko solmusta lahto reittiä solmuun maali?
 24. Kirjoita rekursiivinen aliohjelma, joka tulostaa mahdollisimman lyhyen reitin solmujen numerot lahto ensin ja maali viimeisenä. Saat luottaa, että mahdollisimman lyhyt reitti on olemassa.
 25. Piirrä graafi, jossa ei ole mahdollisimman lyhyttä reittiä lähdestä itseensä.
 26. Miten algoritmin lopetettua voidaan päätellä, onko solmusta lahto mahdollisimman lyhyttä reittiä solmuun maali?
- Binäärihakupuun solmussa on kentät vasen, oikea, avain ja (solmusta alkavan alipuun) koko. Aluksi ss osoittaa juurta. Jos seuraavissa tehtävissä mainittua solmua ei ole olemassa, niin algoritmin jälkeen ss:n pitää osoittaa ei minnekään.
27. Kirjoita algoritmi, jonka jälkeen ss osoittaa puun viimeistä solmua.
 28. Kirjoita algoritmi, jonka jälkeen ss osoittaa solmua, jonka avain on x.
 - 29&30. Kirjoita algoritmi, jonka jälkeen ss osoittaa solmua, jonka edellä puussa on ii solmua.

loppu