

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. Perustele, että $[P, I, K, A, N, A]$ ei ole keko, jossa on suurin ylinnä.
2. Piirrä keko $[P, M, K, D, G, E]$ puuna.
3. Mikä on lopputulos, kun keosta $[P, M, K, D, G, E]$ poistetaan yksi alkio?

Arvioi algoritmin suoritusaikaa $n:n$ funktiona sekä O - että Ω -merkinnällä. Perustele vastauksesi. Taulukko A indeksoidaan $0, \dots, n-1$.

4&5. `i = 0; while( i < n && A[i] != x ){ i = i+1; }`

6&7. `i = 0; tulos = 0;
while(i < n){ if(A[i] > 0){ tulos += A[i]; } i = 2*i+1; }`

Alla on esitetty valintajärjestäminen C++-koodina. Olkoon $n = A.size()$.

```
1 void Valintajarjesta( taulukko & A ){  
2     for( unsigned i = 0; i+1 < A.size(); ++i ){  
3         unsigned p = i;  
4         for( unsigned j = i+1; j < A.size(); ++j ){  
5             if( A[j].x < A[p].x ){ p = j; }  
6         }  
7         alkio apu = A[i]; A[i] = A[p]; A[p] = apu;  
8     }  
9 }
```

8. Mitä $i:n$ arvosta tiedetään rivin 3 alussa?
 9. Mitä $i:n$, $j:n$ ja $p:n$ arvoista tiedetään rivin 5 alussa?
 10. Mitä $i:n$ ja $p:n$ arvoista tiedetään rivin 7 alussa?
- 11&12. Anna valintajärjestämisen ulommalle silmukalle invariantti!

UNION-FIND-tietorakenteen alkiossa on osoitin e kohti edustajaa ja kokonaisluku k tallettamassa korkeusarvion. Edustajan e -osoitin osoittaa edustajaan itseensä. Olkoot a ja b osoittimia alkioihin.

- 13&14. Kirjoita aliohjelma $EDUSTAJA(a)$, joka palauttaa osoittimen $a:n$ edustajaan. Yksi piste tulee siitä, että aliohjelma palauttaa oikean tuloksen ja toinen siitä, että se toteuttaa UNION-FIND:iin kuuluvan nopeutuskikan.
15. Kirjoita aliohjelma $SAMASSA(a, b)$, joka palauttaa true tai false sen mukaan, ovatko a ja b samassa joukossa (eli samassa graafin komponentissa).
- 16&17. Kirjoita aliohjelma $YHDISTÄ(a, b)$, joka yhdistää joukot, joihin a ja b kuuluvat. Yksi piste tulee toimivasta yhdistämisestä ja toinen nopeutuskikasta.

Käännä

Sekalaisia kysymyksiä

18. Miten Dijkstran algoritmi ylläpitää reitin etsinnän aikana tietoa, jonka avulla reitti voidaan lopuksi tulostaa?
19. Millä algoritmilla voidaan selvittää, onko suunnatussa graafissa silmukoita? Algoritmin nimi riittää vastaukseksi.
20. Mikä etu Bellman–Ford-algoritmilla on kurssin muihin reitinetsintäalgoritmeihin verrattuna?
21. Edmonds–Karp-algoritmi löytää maksimi vuon. Kuvaile lyhyesti, miten se toimii.
22. Miksi Edmonds–Karp-algoritmi tarvitsee monimutkaisemman esityksen graafille kuin esimerkiksi Dijkstran algoritmi?
23. Miten hajautustaulun kantataulukon koko kannattaa valita (eli sen taulukon, jota indeksoidaan hajautusfunktion tuloksella)?
24. Minkä toiminnon tasapainotettu binäärihakupuu tarjoaa paljon tehokkaammin kuin hajautustaulu?

DNA-sekvenssi voidaan esittää merkkijonona, jossa ei esiinny muita merkkejä kuin A, C, G ja T. Eliölajien sukulaisuutta voi mitata laskemalla niiden DNA-sekvenssien välisten erojen määrän oheisella aliohjelmalla. Olkoot `n1 = dna1.size()` ja `n2 = dna2.size()`. Alkuperäinen kutsu on `etaisyys( n1, n2 )`;

```
1 unsigned etaisyys( unsigned i1, unsigned i2 ){
2     if( i1 == 0 ){ return i2; }
3     if( i2 == 0 ){ return i1; }
4     unsigned tulos = etaisyys( i1-1, i2-1 );
5     if( dna1[ i1-1 ] != dna2[ i2-1 ] ){ ++tulos; }
6     tulos = std::min( tulos, 1 + etaisyys( i1-1, i2 ) );
7     tulos = std::min( tulos, 1 + etaisyys( i1, i2-1 ) );
8     return tulos;
9 }
```

25. Kirjoita mahdollisimman tiukka ehto, jonka luvut `i1` ja `i2` toteuttavat aina rivin 2 alussa.
26. Perustele, että `etaisyys` ei indeksoi taulukoita `dna1` ja `dna2` laittomasti eikä kutsu itseään laittomilla indekseillä.
27. Perustele, että DNA-sekvenssien AGAA ja GAC välinen etäisyys on 2.
28. Miksi `etaisyys` on hidas jo melko lyhyillä DNA-sekvensseillä?
- 29&30. Kirjoita olennaisesti nopeampi aliohjelma, joka laskee saman asian kuin `etaisyys`. Oleta, että sopiva taulukko on valmiiksi luotu ja alustettu.

loppu