

Tietorakenteet ja algoritmit
Uusinta- ja erilliskoe 30.1.2019
Antti Laaksonen

Kokeen kesto: 3 h 30 min

OHJEET

Kokeessa on sallittua olla mukana:

- kaksipuolinen A4-kokoinen lunttilappu (käsin tehty tai tulostettu)
- laskin
- MAOL-taulukkokirja

Mahdolliset suoritustavat ovat:

- 1. kurssikokeen uusinta
- 2. kurssikokeen uusinta
- 1. ja 2. kurssikokeen uusinta
- erilliskoe

Uusintakokeessa suorituksessa otetaan huomioon syksyn 2018 kurssin tehtäväpisteet.
Erilliskokeessa suoritus perustuu vain koetehtäviin.

Huomaa, että uusintakokeissa ja erilliskokeessa on eri tehtävät. Jokaisessa suoritustavassa kokeen kesto on sama.

1. kurssikokeen uusintakoe

Vastaa näihin tehtäviin, jos haluat uusia syksyn 2018 kurssin 1. kurssikokeen.

Tehtävä 1 (4 pistettä)

Ilmoita jokaisen seuraavan algoritmin aikavaativuus O -merkinnällä ja perustele tämä lyhyesti. Merkintä ... tarkoittaa koodia, joka vie aikaa $O(1)$.

1. for i = 1 to n/2
 for j = 1 to n/3

...

2. for i = 1 to n/2

...

for i = 1 to n/3

...

3. for i = 1 to 1000

...

4. a = 1, b = n

while a < b

 a++

 b--

5. for i = 1 to n

 for j = 1 to i

...

6. i = 1

while i*i <= n

 i++

7. k = 1

for i = 1 to n*n

 while k < i

 k++

8. while n > 0

 n /= 2

 n /= 3

Tehtävä 2 (4 pistettä)

Vastaa lyhyesti jokaiseen seuraavaan kohtaan:

1. Anna esimerkki taulukosta, jossa ei ole yhtään inversiota.
2. Mitä etua taulukosta on taulukkolistaan verrattuna?
3. Mitä etua taulukkolistasta on taulukkoon verrattuna?
4. Mitä etua linkitetystä listasta on taulukkolistaan verrattuna?
5. Miksi hajautustaulusta ei voi löytää nopeasti pienintä avainta?
6. Anna esimerkki binääripuusta, joka ei ole binäärihakupuuta.
7. Anna esimerkki binäärihakupuusta, joka ei ole AVL-puu.
8. Mikä merkitys AVL-puun kierroilla on?

Tehtävä 3 (6 pistettä)

Annettuna on taulukko, jossa on n alkioita ja jokainen alkio on 0 tai 1. Monessako yhtenäisessä alitaulukossa on parillinen määrä ykkösiä?

Esimerkiksi taulukossa [0,0,1,1] oikea vastaus on 6, koska mahdolliset alitaulukot ovat [0], [0], [0,0], [1,1], [0,1,1] ja [0,0,1,1].

Suunnittele algoritmi, joka ratkaisee tehtävän ajassa $O(n)$.

Kuvaile ensin algoritmin toimintaidea 1–2 virkkeellä. Anna tämän jälkeen algoritmin koodi ja perustele, miksi se vie aikaa $O(n)$. Voit käyttää pseudokoodia tai haluamaasi ohjelmointikieltä, ja voit käyttää kurssilla esitettyjä tietorakenteita.

Tehtävä 4 (6 pistettä)

Sinulla on 10^9 hattua, jotka on numeroitu $1,2,3,\dots,10^9$. Aluksi jokainen hattu on tyhjä. Tehtäväsi on toteuttaa seuraavat operaatiot:

- `lisaa(x)`: lisää pallo hattuun, jonka numero on x
- `laske()`: ilmoita, monessako hatussa on yli 3 palloa

Tässä on esimerkki, miten operaatioita voi käyttää:

- `laske()` (antaa tuloksen 0)
- `lisaa(2)`
- `lisaa(2)`
- `lisaa(5)`
- `lisaa(2)`
- `laske()` (antaa tuloksen 0)
- `lisaa(2)`
- `laske()` (antaa tuloksen 1)

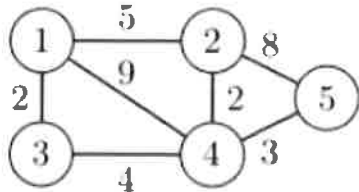
Sinun tulee keksiä toteutus, jossa molemmat operaatiot toimivat tehokkaasti. Kuvaile ensin operaatioiden toimintaidea 1–2 virkkeellä. Anna tämän jälkeen operaatioiden koodi ja perustele, miksi ne ovat tehokkaita. Voit käyttää pseudokoodia tai haluamaasi ohjelmointikieltä, ja voit käyttää kurssilla esitettyjä tietorakenteita.

2. kurssikokeen uusintakoe

Vastaa näihin tehtäviin, jos haluat uusia syksyn 2018 kurssin 2. kurssikokeen.

Tehtävä 1

Miten Primin algoritmi käsittelee seuraavan verkon? Näytä välivaiheet ja lopputulos.



Tehtävä 2

Vastaa lyhyesti jokaiseen seuraavaan kohtaan:

1. Anna esimerkki verkosta, jonka jokaisen solmun aste on pariton.
2. Anna esimerkki verkosta, joka ei ole puu eikä yhtenäinen.
3. Anna esimerkki verkosta, joka on yhtenäinen mutta ei puu.
4. Anna esimerkki verkosta, joka on sykliton mutta ei puu.
5. Anna esimerkki verkosta, jonka jokaisessa virittävässä puussa on kolme kaarta.
6. Anna esimerkki verkosta, jonka kaikki solmut ovat samassa vahvasti yhtenäisessä komponentissa.
7. Miten syvyysshaun voisi toteuttaa ilman rekursiota?
8. Anna esimerkki verkkoalgoritmista, joka perustuu dynaamiseen ohjelmointiin.

Tehtävä 3

Pelaat peliä, jossa on n huonetta. Pelissä on kaksisuuntaisia tunneleita, joista jokainen yhdistää kaksi huonetta. Jokaisessa huoneessa on yksi kolikko.

Saat päättää pelin alussa, mistä huoneesta lähdet liikkeelle. Tämän jälkeen voit kerätä kaikki kolikot huoneista, joihin pääset tunneleita pitkin.

Selosta, miten voit mallintaa ongelman verkkona. Miten voit selvittää tehokkaasti, mistä huoneesta sinun kannattaa aloittaa peli ja montako kolikkoa pystyt keräämään enintään?

Tehtävä 4

Annetaan taulukko, jossa on n kokonaislukua. Lähdet liikkeelle taulukon ensimmäisestä alkioista ja sinun tulee päästä viimeiseen alkioon. Joka vuorolla voit hypätä kaksi tai kolme askelta eteenpäin (eli kohdasta k kohtaan $k+2$ tai $k+3$). Pistemääräsi on summa luvuista, jotka keräät matkasi aikana. Mikä on suurin pistemäärä, jonka voit saavuttaa?

Suunnittele tehtävään-dynaamisen ohjelmoinnin algoritmi, joka toimii ajassa $O(n)$.

Kuvaile ensin algoritmin toimintaidea muutamalla virkkeellä. Anna tämän jälkeen algoritmin koodi ja perustele sen aikavaativuus. Voit käyttää pseudokoodia tai haluamaasi ohjelmointikieltä, ja voit käyttää kurssilla esitettyjä tietorakenteita.

Erilliskoe

Vastaa näihin tehtäviin, jos haluat suorittaa kurssin erilliskokeena ilman tehtävapistettä.

Tehtävä 1 (10 pistettä)

Järjestä seuraavat algoritmin aikavaativuudet hitaimmasta tehokkaimpaan:

$O(\log^2 n)$, $O(n \log n)$, $O(n^{1.9})$, $O(\sqrt{n})$, $O(1)$, $O(\sqrt{n} \log n)$, $O(n^2)$, $O(n / \log n)$, $O(\log n)$, $O(n)$

Tehtävässä ei vaadita perusteluja, vaan riittää, että ilmoitat järjestyksen.

Tehtävä 2 (15 pistettä)

AVL-puu on aluksi tyhjä, ja sitten siihen lisätään luvut järjestyksessä 5, 4, 9, 2, 6, 1, 7, 8, 3. Näytä kuvasarja, jossa on puun rakenne jokaisen luvun lisäämisen jälkeen.

Tehtävä 3 (15 pistettä)

Bellman-Fordin algoritmin avulla voimme havaita, jos suunnatussa painotetussa verkossa on negatiivinen sykli (eli syklin kaarten painojen summa on negatiivinen). Näytä esimerkki suunnatusta painotetusta verkosta, jossa on negatiivinen sykli. Näytä sitten, miten Bellman-Fordin algoritmi käsittelee tämän verkon, ja miten voimme päätellä, että siinä on negatiivinen sykli.

Tehtävä 4 (20 pistettä)

Annettuna on yhtenäinen, suuntaamaton verkko, jossa on n solmua ja m kaarta. Solmut on numeroitu 1, 2, 3, ..., n ja verkko on annettu vieruslistaesityksenä.

Tehtäväsi on suunnitella tehokas algoritmi, joka valitsee jokaiselle kaarelle suunnan niin, että tuloksena olevassa suunnatussa verkossa ei ole suunnattua sykliä.

Selosta ensin algoritmin toiminta ja perustele sitten, miksi se toimii oikein ja tehokkaasti. Täysien pisteiden saaminen tehtävästä edellyttää, että algoritmisi toimii ajassa $O(n + m)$.