

Tik-106.4100 Algoritmien suunnittelu ja analyysi, syksy 2011

Tentti 5.9.2012

Ei apuvälineitä eikä laskimia

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi kurssin koodi ja nimi sekä tentin päivämäärä, nimesi, opiskelijanumerosi ja tutkinto-ohjelmasi sekä palauttamiesi paperien kokonaismäärä.

- a) (3p) Mitkä seuraavista väittämistä pitävät paikkansa ja mitkä eivät? Perustele lyhyesti! (Lyhyt sanallinen perustelu riittää, tarkkoja matemaattisia todistuksia ei vaadita.)
 - $3n^3 + 5n \log n \in \Omega(n^2)$
 - $3n^2 + 2n \in O(n^3)$
 - $5n^2 + 5n \log n \in \Theta(n^3)$
- b) (3p) Selitä, mitä tasoitettu vaativuus (amortized complexity) tarkoittaa ja miten se voidaan laskea pankkitili- eli potentiaalimenetelmällä. Suositeltu vastauksen pituus on noin yksi sivu.

- a) (3p) Ratkaise seuraava rekursioyhtälö, kun n on viiden potenssi. Anna täsmällinen ratkaisu (suuruusluokka ei riitä).

$$T(n) = \begin{cases} 1, & \text{kun } n = 1 \\ 2T(n/5) + 3n & \text{kun } n > 1 \end{cases}$$

- b) (3p) Arvaa hyvä ratkaisu seuraavalle palautuskaavalle ja todista ratkaisusi oikeaksi induktiolla (c_1 ja c_2 ovat vakioita).

$$T(n) \leq \begin{cases} c_1, & \text{kun } n = 1 \\ 3T(n-1) + c_2 & \text{kun } n > 1 \end{cases}$$

- a) (2p) Kerro lyhyesti, mihin UNION-FIND-rakenteita käytetään. Esitä myös esimerkki algoritmista, joka käyttää apuna UNION-FIND-rakennetta ja selitä, mihin rakennetta käytetään ko. algoritmissa. Itse algoritmia ei tarvitse kuvata yksityiskohtaisesti vaan riittää, että UNION-FIND-rakenteen käyttötarkoitus ko. algoritmissa käy selväksi.
- b) (4p) Esitä kaksi Fibonacci-keon operaatiota, joilla tasoitettu aikavaativuus on eri kuin pahimman tapauksen aikavaativuus. Anna näiden operaatioiden molemmat aikavaativuudet ja kerro, mihin nämä aikavaativuustulokset perustuvat (tarkkoja matemaattisia johtoja ei tarvitse esittää).

- (6 p) Olet lähdössä pitkälle matkalle (esim. USAn itärannikolta länsirannikolle). Aloitat matkasi tien alusta. Tien varrella on n hotelleja, joiden etäisyydet tien alusta ovat a_1, a_2, \dots, a_n siten että $a_1 < a_2 < \dots < a_n$. Voit pysähtyä yöksi vain näihin hotelleihin (mutta voit valita, mihin niistä pysähdyt) ja viimeisen päivämatkan jälkeen sinun pitää pysähtyä hotelliin a_n .

Ilman tapauksessa haluaisit matkustaa joka päivä 300 km. Tämä ei ole kuitenkaan mahdollista, koska hotellit eivät ole tasan 300 km:n päässä toisistaan. Jos matkustat yhden päivän aikana x kilometriä, niin tämän päivän tappioksi lasketaan $(300 - x)^2$. Haluat laatia matkaohjelmasi siten, että minimoit kaikkien matkustuspäivien tappioiden summan.

Laadi dynaamista ohjelmointia käyttävä algoritmi, joka laskee, missä hotelleissa sinun kannattaa yöpyä, jotta matkapäivien yhteistappio on mahdollisimman pieni. Älä esitä algoritmiasi pseudokoodina, vaan esitä ensin dynaamisessa ohjelmoinnissa käytettävät rekursiiviset lausekkeet ja selitä sen jälkeen, missä järjestyksessä lausekkeiden arvot lasketaan, jotta laskenta sujuisi tehokkaasti. Selitä lopuksi, miten laskennan perusteella voidaan päätellä oikeat hotellit.

Ratkaisusta ei saa täysiä pisteitä, jos se ei käytä dynaamista ohjelmointia.

- (6p.) Verkko $G = (V, E)$ on kaksijakoinen, jos verkon solmut voidaan jakaa kahteen joukkoon V_1 ja V_2 siten, että minkään kahden samaan joukkoon kuuluvan solmun välillä ei ole kaarta. (Esimerkiksi jos $u \in V_1$ ja $v \in V_1$, niin u :n ja v :n välillä ei ole kaarta verkossa G .) Kirjoita pseudokoodi algoritmillemme joka tutkii, onko annettu suuntaamaton verkko kaksijakoinen. Algoritmin aikavaativuuden tulee olla $O(|V| + |E|)$.

Vinkki: Suuntaamaton verkko on kaksijakoinen jos ja vain jos verkon solmut voidaan värittää käyttämällä vain kahta eri väriä niin, että kaikilla kaarilla $(u, v) \in E$ solmulla u on eri väri kuin solmulla v .