KUOPION YLIOPISTO

Tietojenkäsittelytieteen laitos

Tietorakenteet ja algoritmit

(1./)2. välikoe 03.11.2009

Huom: Voit toimia kahdella eri tavalla.

(a) Jos haluat suorittaa vain 2. välikokeen, tee silloin vain tehtävät 1–3.

(b) Jos haluat korvata 1. välikokeen tuloksesi (siis suorittaa lopputentin), niin vastaa kysymyksiin 1–5.

---

1. [Yhteensä **20p**] Vastaa *lyhyesti* seuraaviin kysymyksiin:

   (a) Vertaile kekojärjestämistä ja lomitusjärjestämistä: mitä hyviä ja huonoja puolia niissä on? Mitkä ovat aikavaativuudet ja aputilavaativuudet?

   (b) Luettele prioriteettijonon tarjoamat operaatiot. Kerro myös niiden aikavaativuudet, kun prioriteettijono on toteutettu binäärikekona (eli kasana).

   (c) Vertaile seuraavia algoritmeja: leveyssuuntainen läpikäynti, Dijkstran algoritmi, Bellman-Fordin algoritmi, Floydin algoritmi: kerro niiden aikavaativuudet, ja mainitse lyhyesti millaisille verkoille / ongelmille ne soveltuvat.

   (d) Mitä tarkoitetaan verkon vahvasti yhtenäisillä komponenteilla ja komponenttiverkolla?

2. [Yhteensä **10p**] On annettu kokonaisluku k ja taulukko $A[1 \ldots n]$, johon on talletettu (satunnaiseen järjestykseen) n kokonaislukua. Tehtävänä on etsiä taulukosta k pienintä lukua. Anna ongelmalle kolme *eri* ratkaisua, joiden aikavaativuudet ovat:

   (a) $O(nk)$

   (b) $O(n \log(n))$

   (c) $O(n + k \log(n))$.

   Kuhunkin kohtaan riittää lyhyt (mutta täsmällinen) sanallinen selostus, ja lyhyt perustelu miksi algoritmi toimii vaaditussa ajassa.

   Mikä algoritmeista on mielestäsi paras? Riipuuko valintasi jotenkin parametrin k arvosta?

3. [Yhteensä **10p**] Tee vain *toinen* seuraavista vaihtoehdoista (joko (a) tai (b)):

   (a) i. Suunnatun painotetun verkon solmujen u ja v välinen lyhin polku on p. Verkon jokaisen kaaren painoon lisätään 1; muuten verkko säilyy ennallaan Onko p välttämättä solmujen u ja v välinen lyhin polku myös muunnetussa verkossa? Perustele!

   ii. Suuntaamattomalla painotetulla verkolla on pienin virittävä puu T. Verkon jokaisen kaaren painoon lisätään 1; muuten verkko säilyy ennallaan. Onko T välttämättä pienin virittävä puu myös muunnetussa verkossa? Perustele!

   iii. Olkoon annettu suuntaamaton yhtenäinen painotettu verkko G ja lisäksi tämän verkon pienin virittävä puu T. Miten lasket *tehokkaasti* solmujen u ja v välisen lyhimmän polun verkossa G? Entä virittävässä puussa T? Onko tämä etäisyys puussa T pienenpi, sama vai suurempi, kuin verkossa G? Perustele!

   (b) Suunnattu painottamaton verkko $G = (V, E)$ on esitetty vieruslistoina. Jokaisella verkon kaarella on väri; *joko* punainen *tai* sininen. On annettu kaksi verkon solmua, s ja t. Tehtävänä on etsiä polku solmusta s solmuun t. Polun on kuitenkin oltava sellainen, että *kaikki* punaiset kaaret ovat *ennen* sinisiä tällä polulla. Toisin sanoen polun alkuosa muodostuu pelkistä punaisista kaarista, ja loppuosa pelkästään sinisistä kaarista. Jos tällaisia polkuja on useita, niin näistä vaihtoehdoista on palautettava *lyhin* (kaarien lukumäärällä mitattuna) polku (jos sellaisiakin on useita, niin niistä voi palauttaa minkä vain). Esitä tehtävälle tehokas ratkaisualgoritmi ja analysoi sen aikavaativuutta. Voit käyttää apuna mitä tahansa kurssilla esitettyjä algoritmeja ja niiden tunnettuja aikavaativuuksia. Täysien pisteiden saamiseksi algoritmin on lisäksi toimittava ajassa $O(|V| + |E|)$.

---

4. [Yhteensä **20p**] Vastaa *lyhyesti* (1-2 lauseella) seuraaviin kysymyksiin:

   (a) Väite: Jos hakupuuta ei ole tasapainoitettu, niin kaikki perus-operaatiot (annetun alkion / minimin / maksimin etsiminen, jne.) toimivat puun korkeuteen verrannollisessa ajassa. Totta vai ei? Jos ei, niin miksi ei?

   (b) Väite: Binäärihaku linkitetyssä listassa on tehokkaampi kuin peräkkäishaku. Totta vai ei? Jos ei, niin miksi ei?

   (c) Väite: Jos jokaiselle binääripuun solmulle x pätee: $x.left.key < x.key < x.right.key$, niin puu on hakupuu. Totta vai ei? Jos ei, niin mikä on oikea ehto?

   (d) Erään binääripuun solmujen avaimet ovat leveyssuuntaisessa järjestyksessä lueteltuna 3, 2, 5, 1. Saman puun avaimet jälkijärjestyksessä ovat 1, 2, 5, 3. Väite: voidaan päätellä, että puu on hakupuu. Totta vai ei? Jos ei, niin anna vastaesimerkki.

   (e) Väite: *Mikä tahansa* hajautusfunktio toimii huonosti *jollakin* syötteellä. Totta vai ei? Jos ei, niin miten voidaan valita sellainen funktio joka toimii hyvin kaikilla syötteillä?

5. [Yhteensä **20p**] Olkoon annettu kahteen suuntaan linkitetty lista L, joka sisältää n kappaletta kokonaislukuja. Listaa ei ole järjestetty.

   (a) [**6p**] Kirjoita algoritmi (pseudokoodi), joka poistaa listasta peräkkäiset samanarvoiset luvut, jättäen jäljelle vain ensimmäisen. Siis jos listassa on aluksi luvut 4, 4, 4, 2, 3, 3, 1, 5, 5, 2, 4, 4 (tässä järjestyksessä), niin lopputuloksena listassa on luvut 4, 2, 3, 1, 5, 2, 4 (tässä järjestyksessä). Algoritmisi saa käyttää vain $O(1)$ verran aputilaa. Mikä on aikavaativuus?

   (b) [**5p**] Kuten kohta (a), mutta nyt jokaisesta alkiosta säilytetään vain sen ensimmäinen esiintymä, riippumatta siitä, ovatko esiintymät peräkkäisiä vai ei. Tulos olisi siis 4, 2, 3, 1, 5. Saat vapaasti käyttää kursilla annettuja lista-operaatioita (mutta et mitään aputietorakenteita) kirjoittamatta niiden pseudokoodeja. Algoritmisi ei tarvitse olla erityisen tehokas. Mikä on sen aikavaativuus?

   (c) [**4p**] Kuten kohta (b), mutta saat käyttää aputietorakenteena hakupuuta. Lisäksi tiedetään, että eriarvoisia lukuja on listassa vain k kappaletta (k on siis tuloksen pituus, esimerkissämme $k = 5$; mutta lukuja on syötteessä yhteensä edelleen n kappaletta). Pseudokoodia ei tarvita, vaan lyhyt sanallinen selitys riittää. Nyt algoritmin pitäisi olla mahdollisimman tehokas. Mikä on aikavaativuus ja aputilavaativuus? Perustele lyhyesti.

   (d) [**3p**] Kuten kohta (c), mutta hakupuu korvataan hajautuksella. Mikä on nyt aikavaativuus ja aputilavaativuus? Perustele.

   (e) [**2p**] Jos tulos halutaankin järjestyksessä (eli siis esimerkissämme haluaisimme tuloksesi luvut järjestyksessä 1, 2, 3, 4, 5), niin mikä olisi tehokkaan algoritmin aikavaativuus käytettäessä apuna hakupuuta?