Teknillinen korkeakoulu
Tietotekniikan laitos
T-106.1220/1223 Tietorakenteet ja algoritmit T/Y

TENTTI
10.8.2009

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, tutkinto- tai koulutusohjelmasi, kurssikoodi ja kurssin nimi, päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*.

1) Kymmenen kysymystä (10 x 1p)

Tämä tehtävä on *tentin pakollinen osa*, josta on saatava vähintään 5p/10p, jotta loput tentistä tarkistetaan. Tämä tehtävä ei kuitenkaan yksistään riitä tentin läpäisyyn. Toisaalta viiteen pisteeseen ei edellytetä "täysin oikeaa vastausta" vaan oleellista on, että pystyt osoittamaan *ymmärtäneesi* tehtävän *koodin* toiminnan. Käytä siis aikaa perustelujen miettimiseen ja esittämiseen. Viittaa perusteluissa ohjelmakoodin rivinumeroihin, jos mahdollista.

Liitteessä A on pikajärjestämisalgoritmi (quicksort, vastaavasta TRAKLA2-tehtävästä), joka järjestää annetun taulukon nousevaan suuruusjärjestykseen. Lue ensin kaikki kysymyskohdat vastaamatta niihin ja sen jälkeen tutustu koodiin erittäin huolella. Vastaa tämän jälkeen kaikkiin kysymyksiin ja käytä aikaa perustelujen pohtimiseen ja muotoilemiseen. Huomaa, että kaikissa kysymyksissä viitataan liitteen A algoritmiin ja, että väittämät voi perustella yhtä hyvin vastaamalla kyllä tai ei, joten pisteet tulevat vain *perusteluista*!

a) Merkitään algoritmin saaman *syötteen kokoa* N:llä, kun algoritmia kutsutaan järjestämättömälle taulukolle NrrNy. Määrittele N algoritmin saamien parametrien NNrt ja rrt ht funktiona.
b) Mitä algoritmi tekee riveillä 26-27? Entä riveillä 29-31?
c) Algoritmia kutsutaan taulukolle [2, 1, 9, 0, 5, 2, 3]. Mikä on taulukon alkioiden järjestys rivin 7 suorituksen jälkeen? Piirrä myös välivaiheita.
d) Mitä algoritmi tekee riveillä 10 ja 12?
e) *Perustele* pitääkö väite paikkansa vai ei: algoritmin saama syöte ja tuloste ovat yksi ja sama taulukko NrrNy.
f) *Perustele* pitääkö väite paikkansa vai ei: jos järjestettävän *taulukon koko* on aidosti pienempi kuin 3, sitä ei kannata järjestää tällä quicksort-algoritmilla.
g) *Perustele* pitääkö väite paikkansa vai ei: algoritmi toimii ajassa "NNNog/N".
h) Onko algoritmi *stabiili*? Miksi?
i) *Analysoi* algoritmin suoritusaika, kun sitä kutsutaan taulukolle, jonka koko Nn N, ja joka on aluksi *laskevassa suuruusjärjestyksessä*.
j) *Analysoi* algoritmin suoritusaika, kun sitä kutsutaan taulukolle, jonka koko Nn N, ja jonka *kaikki alkiot* ovat *yhtä suuria*.

Bonustehtävä (tee tämä ainoastaan jos aikaa jää tehtävien 2-4 jälkeen):

k) *Kuinka monta kertaa* algoritmi kutsuu aliohjelmaa NNrtrtron (jos syötteen koko on N)? Suuruusluokka perusteluineen riittää vastaukseksi. Anna suuruusluokka *iso O –notaatiossa*.

2) Terminologiaa (2p + 2p + 2p + 2p)

Määrittele seuraavat *käsitteet* (4 x 1p). *Anna* jokaisesta myös *esimerkki* (4 x 1p).

a) Abstrakti tietotyyppi (ADT)
b) Prioriteettijono
c) Binäärikeko
d) Kekoehto

3) Binääriset hakupuut (1p + 1p + 3p + 3p)

a) *Määrittele* käsite *binäärinen hakupuu*.
b) *Piirrä* esimerkki täydellisestä binäärisestä hakupuusta, jonka korkeus on 4.
c) Millaisia erilaisia binäärisiä hakupuita on olemassa? Miten ne eroavat toisistaan? *Käsittele* tässä kohtaa ainakin *kolmea eri tietorakennetta*.
d) *Vertaile* edellisen kohdan tietorakenteiden lisäys-operaation analyyttisiä ominaisuuksia *iso O –notaatiolla* kolmen eri aikakompleksisuusmitan suhteen: paras tapaus, keskimääräinen tapaus ja pahin tapaus. *Piirrä matriisi*, jossa on sarakkeina em. tapaukset ja riveillä valitsemasi hakupuut.

4) Pienimmän virityspuun etsiminen (4p + 4p)

a) Esitä jonkin algoritmin toiminta, jolla voidaan tuottaa verkon pienin virityspuu. Mikä on esittämäsi algoritmin aikakompleksisuus, jos graafissa on V solmua ja E kaarta? Perustele tuloksesi.

b) Tarkastellaan painotettua suuntaamatonta verkkoa, johon kuuluvat solmut A-F ja särmät AB(2), AD(2), AE(6), BC(4), BE(3), CE(1), CF(4), DE(1), EF(3). Särmien painot on esitetty särmän jälkeen suluissa. Esitä miten annetulle verkolle voidaan löytää pienin virityspuu soveltamalla a-kohdassa kuvaamaasi algoritmia.