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

TENTTI
18.8.2008

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

1) Koodin analyysi (1p + 2p + 3p + 2p + 2p)

Seuraavassa on annettu *pikajärjestämisalgoritmin* rekursiivinen osa, joka järjestää taulukossa `array` olevat alkiot suuruusjärjestykseen.

Tutustu koodiin huolella ja lue ensin kaikki kysymykset. Vastaa sen jälkeen kysymyksiin. 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 osoittaa *ymmärtäneensä* koodin toiminnan. Käytä siis aikaa myös perustelujen esittämiseen.

```
1 quicksort(array:Array, left:integer, right:integer)
2   if (right-left) ≥ 1
3     split = partition(array,left,right)

4     if (split-1)-left > 0
5       quicksort(array, left, split-1)
6     if right-(split+1) > 0
7       quicksort(array, split+1, right)

8   endif
9 end
```

a) Määrittele algoritmin "syötteen koko" (N) sen saamien parametrien funktiona.

b) Rivillä 3 algoritmi kutsuu aliohjelmaa `partition (array,left,right)`, joka jakaa parametrinä annetun taulukon osan `array[left..right]` alkiot "pieniin" ja "suuriin" *jonkin* pivot-alkion suhteen. Lopuksi pivot-alkion indeksi palautetaan paluuarvona ja sijoitetaan muuttujaan `split`. Selitä sanallisesti *jokin* partitiointimenetelmä, joka soveltuu pikajärjestämismenetelmälle. Voit myös käyttää pseudokoodia ja kuvia selityksesi tukena.

c) *Selitä* `quicksort`*-algoritmin toiminta* käyttäen apuna sopivaa esimerkkiä. Voit esimerkiksi piirtää *rekursiohistoriapuun*, josta ilmenee millä parametreillä algoritmi kutsuu itseään rekursiivisesti ja missä tilassa taulukko `array` on kunkin rekursiokutsun jälkeen.

d) Analysoi kuvaamasi partitiointimenetelmän aikavaativuus iso O –notaatiossa. Perustele.

e) Pohdi ennen kuin vastaat! *Kuinka monta rekursiokutsua* algoritmi tekee, kun sitä kutsutaan syötteellä, jonka koko on *N* (suuruusluokka riittää vastaukseksi). Perustele.

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

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

a) Abstrakti tietotyyppi (ADT)
b) Pino
c) Valikointi (Selection)
d) Stabiili järjestämismenetelmä

3) Hakurakenteet (2p + 2p + 2p + 2p +2p)

Vertaile tasapainotettuja hakupuita (balanced search trees) ja hajautusrakenteita (hashing) keskenään. Jäsennä vastauksesi seuraavasti:

a) *Määrittele* abstrakti tietotyyppi *hakurakenne* (dictionary).
b) *Nimeä* ainakin *yksi tasapainotettu hakupuu* ja *yksi hajautusrakenne*.
c) *Mitkä operaatiot* (vähintään 2) *voidaan toteuttaa* tasapainotetuissa hakupuissa tehokkaammin kuin hajautusrakenteissa? Perustele vastauksesi.
d) *Mitkä operaatiot* (vähintään 2) *voidaan toteuttaa* hajautusrakenteissa tehokkaammin kuin tasapainotetuissa hakupuissa? Perustele vastauksesi.
e) *Miten* jokin tasapainotettu hakupuu ja jokin hajautusrakenne voitaisiin *yhdistää* siten, että saataisiin aikaan vielä tehokkaampi hakurakenne?

4) Prioriteettijonot (2p + 2p + 4p)

a) *Määrittele* käsite *prioriteettijono (priority queue)*.

b) *Mainitse nimeltä* jokin algoritmi, joka käyttää prioriteettijonoa aputietorakenteenaan. *Selitä lyhyesti* mihin ko. algoritmi sitä tarvitsee.

c) *Esitä välivaiheittain* miten lineaarisessa ajassa toimiva *BuildHeap*-algoritmi (joka tunnetaan myös nimillä *FixHeap* ja *Bottom-Up Heap Construction*) toimii, jos se saa syötteenään taulukon, jossa on alkiot 7, 2, 25, 1, 10, 23, 14, 20, 3, 5, 6, 15 ja 13. Kekoehto on "isä pienempi kuin lapsensa". **Huom!** Kyseessä ei ole tapaus, jossa alkiot lisätään rakenteeseen yksi kerrallaan.

5) Pienimmän virityspuun etsiminen (3p + 3p)

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.