TKT20001 Tietorakenteet ja algoritmit (kevät 2019)

Kurssikoe 1 (4.3.2019)

Tentissä saa olla mukana käsin kirjoitettu yksi A4-kokoinen "lunttilappu", jonka molemmilla puolilla saa olla tekstiä. Muu lisämateriaali (laskimet, taulukot tms.) ei ole sallittua.

Vastaa kuhunkin tehtävään **erilliselle** konseptipaperille.

Kirjoita jokaisen paperin yläkulmaan **kurssin nimi, kokeen päivämäärä, oma nimesi ja opiskelijanumerosi**. Vaikka jättäisit johonkin tehtävään vastaamatta, palauta silti vastauspaperi kyseiseen tehtävään.

Tehtävissä 4 ja 5, joissa pyydetään algoritmia, voit käyttää luentojen ja kurssikirjan tyyppistä pseudokoodia tai muita ymmärrettäviä pseudokoodityylejä tai oikeaa ohjelmointikieltä, esim. Javaa. Jos käytät oikeaa ohjelmointikieltä, selitä erityisen hyvin, mitä ohjelmassasi tapahtuu, äläkä käytä mitään kielen erikoisia piirteitä. Myös sanallinen selitys ilman pseudokoodia kelpaa, kunhan se on rittävän selkeä ja yksityiskohtainen, että vaadittun aikavaativuuden yms. pystyy selvästi toteamaan. Näissä tehtävissä voit käyttää kaikkia kurssilla esitettyjä tietorakenteita ja algoritmeja, niiden tunnettuja aikavaativuuksia jne., kunhan sanot selvästi, mitä milloinkin käytät.

Vastaa kaikkien kysymysten kaikkiin kohtiin. Kokeen maksimipistemäärä on 20.

1. [*3 pistettä*] Mitä etuja pikajärjestämisellä on lomitusjärjestämiseen verrattuna? Entä toisin päin: mitä etuja lomitusjärjestämisellä on pikajärjestämiseen verrattuna? Millaisessa tilanteessa käyttäisit lisäysjärjestämistä; milloin taas et käyttäisi?

   Vastaa lyhyesti ja selkeästi vetoamalla kurssilla esitettyihin algoritmien ominaisuuksiin. Vastauksesi tähän tehtävään tulisi normaalikäsialalla kirjoitettuna mahtua helposti yhdelle sivulle.

2. [*3 pistettä*] Mitkä ovat hajautustaulun eri operaatioiden aikavaativuudet? Entä tasapainotetun hakupuun? Millaisissa tilanteissa tämän perusteella hajautustaulu on parempi? Entä milloin tasapainotettu hakupuu on parempi? Esitä myös *yleisellä tasolla ja lyhyesti* (parilla virkkeellä) perusajatus, miten näihin aikavaativuuksiin päädytään.

   Vastaa lyhyesti ja selkeästi vetoamalla kurssilla esitettyihin tietorakenteiden ominaisuuksiin. Vastauksesi tähän tehtävään tulisi normaalikäsialalla kirjoitettuna mahtua helposti yhdelle sivulle.

3. [*5 pistettä*] Mitkä ovat seuraavien algoritmien aikavaativuudet?

Jokaisessa kohdassa ilmoita aikavaativuus parametrin n funktiona käyttäen iso-O-merkintää ja *selitä* lyhyesti, miten päättelit aikavaativuuden. Selitykseksi riittää virke tai pari, joissa mainitset käyttämäsi yleiset periaatteet tms.; älä esitä tarkkoja matemaattisia todistuksia tms.

Algoritmit eivät tee mitään erityisen hyödyllistä. Kolme pistettä "..." tarkoittaa jotain vakioajassa tapahtuvaa laskentaa.

(a)
```
for i=1 to n
    ...
for i=1 to n
    for j=1 to n
        ...
```

(b)
```
for i=1 to n
    for j=1 to i
        for k=1 to j
            ...
```

(c)
```
for i=1 to n
    j = 0
    while j < n
        j = j + n/10 + 1
        ...
```

(d)
```
k = 0
summa = 0
while summa < n
    k = k + 1
    summa = summa + k
```

(e)
```
rekursiivinen(n)
    if n == 0
        return
    rekursiivinen(n-1)
    rekursiivinen(n-1)
    rekursiivinen(n-1)
```

4. [*5 pistettä*] Tarkastellaan tilannetta, jossa m hattua on numeroitu $1, \ldots, m$. Hattuihin voidaan lisätä ja niistä voidaan poistaa palloja. Lisäysten ja poistojen yhteislukumäärälle on annettu yläraja n. Aluksi jokainen hattu on tyhjä.

   Esitä tehokkaat algoritmit seuraavien operaatioiden toteuttamiseksi:

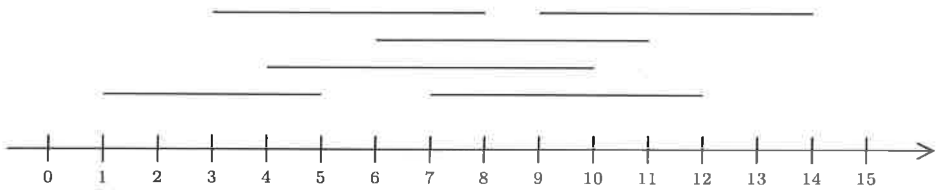
   - *lisää*(i): Lisää yksi pallo hattuun numero i.
   - *poista*(i): Poista yksi pallo hatusta i. Kuitenkin jos hattu i on tyhjä, niin poista yksi pallo numerojärjestyksessä seuraavasta ei-tyhjästä hatusta. Voit olettaa, että jokin seuraavista hatuista sisältää palloja.
   - *pallot*(i): Palauta hatussa i olevien pallojen lukumäärä.

   Täysien pisteiden saamiseksi kaikkien operaatioiden tulee toimia pallojen lukumäärän suhteen logaritmisessa ajassa $O(\log n)$ riippumatta siitä, kuinka suuri hattujen lukumäärä m on. Erityisesti aikavaativuuden $O(\log n)$ pitää päteä, kun m on hyvin paljon suurempi kuin n ja siis selvä enemmistö hatuista on tyhjiä.

5. [*4 pistettä*] On annettu joukko lukusuoran välejä $[a, b]$, $0 < a < b$. Tässä $[a, b]$ tarkoittaa suljettua reaalilukuväliä a:sta b:hen, siis pistejoukkoa $\{ x \in \mathbb{R} \mid a \leq x \leq b \}$. Tehtävänä on selvittää, onko jokin näistä väleistä jonkin toisen välin aito osaväli; ts. onko joukossa välit $[a, b]$ ja $[c, d]$, joilla $a < c < d < b$. Voit olettaa, että kaikki välien päätepisteet ovat erisuuria kokonaislukuja.

   Esitä ongelmaan tehokas ratkaisualgoritmi. Täysien pisteiden saamiseksi algoritmisi tulee toimia ajassa $O(n \log n)$, missä n on välien lukumäärä.

   Esimerkki 1: Jos on annettu välit $[1, 5]$, $[7, 12]$, $[4, 10]$, $[6, 11]$, $[3, 8]$ ja $[9, 14]$, niin vastaus on *ei*, sillä mikään väleistä ei sisälly kokonaan toiseen väliin:



   Esimerkki 2: Jos on annettu välit $[1, 6]$, $[8, 13]$, $[5, 7]$, $[9, 14]$, $[2, 11]$ ja $[4, 12]$, niin vastaus on *kyllä*, sillä väli $[5, 7]$ sisältyy väliin $[2, 11]$, ja myös väliin $[4, 12]$.