

Tietorakenteet, ensimmäinen kurssikoe 28.2.2011

Vastaa tehtävät 1, 2 ja 3 erillisille konsepteille. Kirjoita jokaiseen konseptiin kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero. Vastaukset palautetaan tehtäväkohtaisiin pinoihin.

Vaikka jättäisit johonkin tehtävään vastaamatta, tulee vastauspaperi siinäkin tapauksessa palauttaa.

Tehtävissä joissa pyydetään algoritmia, voit käyttää luentomonisteen pseudokoodityylin lisäksi muitakin "ymmärrettäviä" pseudokoodityylejä tai oikeista kielistä Javaa, Pythonia, C:tä, C++:aa tai jotain muuta selkeälukuista imperatiivista kieltä. Älä kuitenkaan käytä kielten eksoottisia piirteitä vaan pidä algoritmit pseudokoodimaisina.

Kokeen tulos ilmestyy kurssisivulle huomiseen klo 16 mennessä

1. (8 p) Määritä seuraavien algoritmien aikavaativuus ja tilavaativuus O-merkintää käyttäen. **Perustele vastauksesi muutamalla lauseella!**

- (a) Algoritmi `pieninJaSuurin` tulostaa taulukon pienimmän ja suurimman luvun. Taulukossa on yhteensä n lukua.

```
pieninJaSuurin(taulukko)
    pienin = taulukko[1]
    for i = 2 to taulukko.length
        if taulukko[i] < pienin
            pienin = taulukko[i]
    print "Pienin: " + pienin
    suurin = taulukko[1]
    for i = 2 to taulukko.length
        if taulukko[i] > suurin
            suurin = taulukko[i]
    print "Suurin: " + suurin
```

- (b) Algoritmi `tulostaParilliset` tulostaa listan kaikki parilliset luvut. Lista on yhteen suuntaan linkitetty lista, jonka avaimet ovat kokonaislukuja. Algoritmille annetaan viite listan ensimmäiseen solmuun. Listassa on yhteensä n solmua.

```
tulostaParilliset(solmu)
    if solmu == NIL
        return
    if solmu.key % 2 == 0
        print solmu.key
    tulostaParilliset(solmu.next)
```

- (c) Algoritmi `kolmenSumma` tutkii, onko taulukossa t kolmea lukua, joiden summa on s . Sama taulukon luku saa esiintyä summassa monta kertaa. Taulukossa on yhteensä n lukua.

```
kolmenSumma(t, s)
    for a = 1 to t.length
        for b = 1 to t.length
            for c = 1 to t.length
                if t[a] + t[b] + t[c] == s
                    return true
    return false
```

- (d) Algoritmi `tulostaBinaarina` tulostaa positiivisen kokonaisluvun binäärimuodossa. Esimerkiksi jos n on 20, algoritmi tulostaa luvun 10100.

```

tulostaBinaarina(n)
  if n == 0
    return
  tulostaBinaarina(n / 2)
  print n % 2

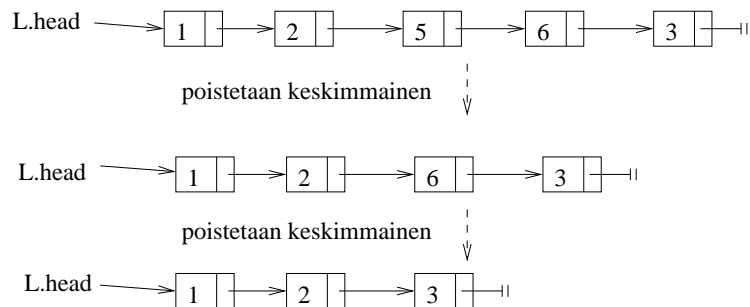
```

2. (6 p) Tee algoritmi, joka poistaa yhteen suuntaan linkitetyltä tunnussolmulliselta listalta keskimmäisenä olevan solmun. Jos solmuja on parillinen määrä, kelpaa kumman tahansa "kahdesta keskimmäisestä" poistaminen, eli esim. 6:n solmun tapauksessa joko kolmannen tai neljännen poistaminen kelpaa. Voit olettaa, että operaatiota suoritettaessa lista ei ole tyhjä.

Analysoi algoritmisii aika- ja tilavaativuus.

Et voi tässä tehtävässä olettaa että listallasi on mitään valmiita operaatioita. Tiedät vaan, että lista L tuntee ensimmäisenä olevan solmun $L.head$ ja jokainen listasolmu s tuntee sitä seuraavana olevan solmun $s.next$ ja viimeisenä listalla olevan solmun seuraavaksi on asetettu NIL eli "ei mitään".

Seuraavassa esimerkki algoritmin toiminnasta (parillisen pituiselta listalta on esimerkissä poistettu keskimmäisistä jälkimäinen):



3. (8 p) Binäärihakupuut

- (a) (1p) Selitä lyhyesti miten (tasapainottamattomasta) binäärihakupuusta poistetaan alkioita. Älä kirjoita vastaukseen tarkkaa pseudokoodia, lyhyt tekstuaalinen selitys riittää.

Poista allaolevasta puusta juuressa oleva alkio 5.

- (b) (3p) Selitä lyhyesti miten AVL-puuhun lisätään alkioita.

Lisää allaolevaan puuhun (jolle ei ole tehty a-kohdan poisto-operaatiota) ensin alkio 9. Ja tuloksena olevaan puuhun alkio 6.

Näytä puun lopputilanne kunkin lisäyksen tultua suoritetuksi sekä välitilanteet, joissa tehdään jokin tasapainotusoperaatio.

- (c) (4p) Anna algoritmi, joka tulostaa binääripuun alimmalla tasolla olevien lehtisolmujen sisällön.

Analysoi algoritmisii aika- ja tilavaativuus.

Esim. allaolevan puun tapauksessa tulostettaisiin 8 ja 14. Arvoa 2 ei tulosteta, sillä vaikka kysessä on lehtisolmu, ei se sijaitse alimmalla tasolla.

