

Tietorakenteet ja algoritmit 811312A 14.5.2012

Tenttitulokset tulevat viimeistään torstaina 31.5.2012

1. Valitse ja tee toinen seuraavista tämän tehtävän vaihtoehdoista
vaihtoehto 1) Osoita induktiotodistuksella, että $1 + 2 + 2^2 + \dots + 2^n = 2^{n+1} - 1$
aina, kun $n \in \mathbb{N}$ ja $n \geq 0$.
vaihtoehto 2) Havainnollista seuraavan ”määritelmän”,
 $f(n) \leq c * g(n) \quad \forall n > n_0 > 0$, missä reaaliluku $c > 0$,
symboleille kompleksisuusluokkaan $O(g(n))$ liittyvät merkitykset
esimerkin $f(n) = 2n^2 + 1$ avulla.
2. Seuraava algoritmi laskee syötteenä saamansa taulukon alkioden keskiarvon.
Todista algoritmi oikeaksi.

Syöte: Taulukko $A[1, \dots, n]$, $n \geq 1$ **Tuloste:** Taulukon alkioden summa

```
SUMMA(A)
1.  s = 0
2.  i = 1
3.  ka = 0
4.  while i <= n do
5.      s = s+A[i]
6.      i = i+1
7.  end while
8.  ka = s/n
9.  return ka
```

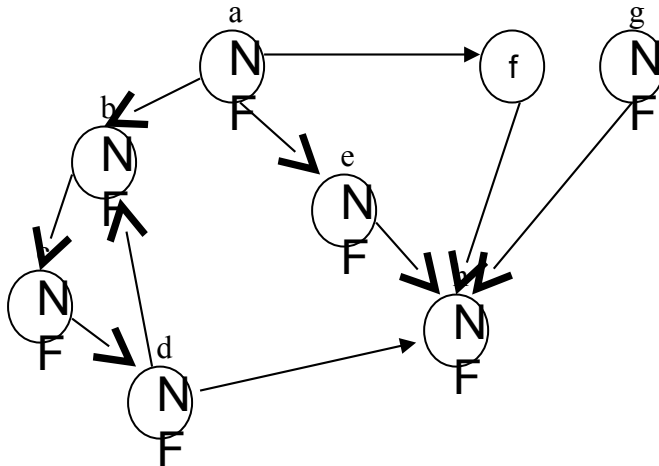
3. Seuraavaa lisäslajittelua kannattaa käyttää pienten taulukoiden lajitteluun. Määritä (ja perustele) algoritmin kompleksisuusluokka? (Voit tarkastella ainoastaan huonointa tapausta.)

Syöte: Taulukko $A[1, \dots, n]$, $n \geq 1$ **Tuloste:** Taulukon luvut järjestyksessä $A[1] \leq \dots \leq A[n]$

```
LISÄYS(A)
1.  for j = 2 to n do
2.      k = A[j]
3.      i = j-1
4.      while i > 0 && A[i] > k do
5.          A[i+1] = A[i]
6.          i = i-1
7.      end while
8.      A[i+1] = k
9.  end for
10. return
```

KÄÄNNÄ PAPERI

4. Suorita syvyysbushu seuraavalle suunnatulle verkolle ja luokittete verkon vält. Tarkista myös, onko verkossa sykliä.



5. Esitä vaihe vaiheelta miten algoritmi PARTITIO jakaa taulukossa A olevat alkioit alkujärjestyksestä 8, 10, 9, 2, 11, 3, 5, 7 kahteen osaan. Toimiiko algoritmi PARTITIO oikein?

Syöte: Taulukon osa $A[p, \dots, r]$, $p < r$
Tulostus: Taulukko $A[p, \dots, j, \dots, r]$, jossa indeksi j jakaa alkioit kahteen osaan siten, että taulukon osassa $A[p, \dots, j-1]$ alkioit ovat pienempiä tai yhtä suuria kuin $A[j]$ ja osassa $A[j+1, \dots, r]$ suurempia tai yhtä suuria kuin $A[j]$

```

PARTITIO(A, p, r)
1.  x = A[p]
2.  i = p
3.  j = r + 1
4.  while TRUE do
5.    repeat
6.      j = j - 1
7.    until A[j] <= x
8.    repeat
9.      i = i + 1
10.   until A[i] >= x
11.   if i < j then
12.     exchange A[i] ↔ A[j]
13.   else
14.     exchange A[p] ↔ A[j]
15.   return j
16. end i
  
```