OULUN YLIOPISTO Tietojenkäsittelytieteiden laitos

Tietorakenteet ja algoritmit 811312A 9.1.2012

Tenttitulokset tulevat viimeistään tiistaina 31.1.2012

1. Seuraavan algoritmin oletetaan laskevan syötteenä saamansa taulukon alkioiden tulon.

```
Input: Taulukko A[1,..,n], n >= 1
Output: Taulukon alkioiden tulo

TULO(A)
  t = 1
  j = 1
  while j <= n
      t = t * A[j]
      j = j + 1
  return t
```

a) Todista algoritmi oikeaksi.
b) Määritä algoritmin kompleksisuus.

2. Valitse vain toinen seuraavista kohdista (a tai b).

a) Esitä mitä määritelmän mukaan tarkoittaa merkintä $f(n) \in \Theta(n^2)$.

b) Onko $f(n) \in \Theta(n^2)$, kun

1) $f(n) = 2n^2$
2) $f(n) = n^3 - 2n + 1$?

Perustele vastauksesi.

3. Tee molemmat kohdat a ja b.

a) Seuraava algoritmi laskee rekursiivisesti Fibonacci-lukuja.

```
Syöte: Kokonaisluku n >= 0
Tulostus: n:s Fibonacci-luku
FIBO(n)
 if n <= 1
   return 1
 else
   return FIBO(n-1) + FIBO(n-2)
```

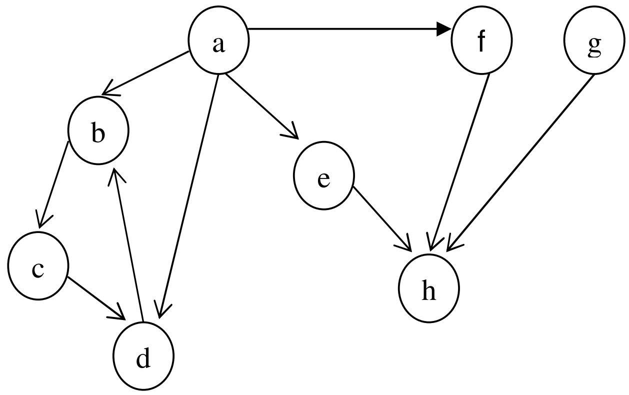
Tee pseudokoodi algoritmille, joka laskee niitä iteratiivisesti.

b) Selvitä mitä seuraava rekursiivinen algoritmi tekee.

```
Syöte: Kokonaisluku n >= 0
Tulostus: Kokonaisluku
JEP(n)
if n < 2
  luku = 2
else
  luku = JEP(n-1) + JEP(n-2)
return (luku-1)
```

KÄÄNNÄ PAPERI!

4. Suorita syvyyshaku seuraavalle suunnatulle verkolle ja luokittele verkon välit.
Tarkista myös, onko verkossa sykliä.



5. Painotetussa suunnatussa verkossa G = (V, E) on solmujen joukko
V = { s, a, b, c, d } ja välien painoarvot w(s, a) = 3, w(s, c) = 5, w(a, b) = 6, w(a, c) = 2, w(b, d) = 2, w(c, a) = 1, w(c, d) = 6, w(d, s) = 3, w(d, b) = 7.
Määritä algoritmia DIJKSTRA käyttäen solmusta s lyhimpien polkujen pituudet kaikkiin muihin solmuihin. Esitä myös etäisyys- ja edeltävyystaulukoiden alkioiden kehitys sekä lyhimpien polkujen puu.

Input: Suunnattu syklitön verkko G, jonka solmujen joukko on V, väleillä on painoarvofunktio w ja alkusolmuna on s..
Output: Etäisyystaulukossa d on lyhimmät etäisyydet solmusta s ja edeltävyystaulukossa π on jokaiselle solmulle edeltäjäsolmut lyhimmillä poluilla solmusta s.

```
DIJKSTRA(G, w, s)
INITIALIZE_SINGLE_SOURCE(G, s)
S = ∅    // ”S on valmiiksi laskettujen solmujen joukko”
Q = G.V
while Q ≠ ∅
    u = EXTRACT_MIN(Q)  // ”u:lla on joukon Q solmuista pienin arvo taulukossa d”
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

```
INITIALIZE_SINGLE_SOURCE(G, s)
for each vertex v∈ G.V
    d[v] = ∞
    π[v] = NIL
d[s] = 0
```

```
RELAX(u, v, w)
if d[v] > d[u] + w(u, v)
    d[v] = d[u] + w(u, v)
    π[v] = u
```