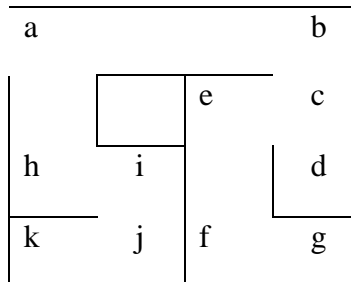


811312A Tietorakenteet ja algoritmit, 2018 - 2019, Harjoitus 6

Harjoituksen aiheet ovat verkkojen leveys- ja syvyyshakualgoritmit

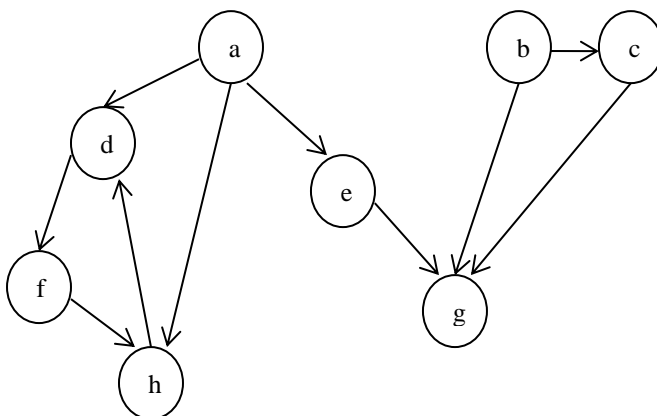
Tehtävä 6.1 Hae leveyshakualgoritmia käyttäen lyhin polku seuraavan sokkelon läpi, kun sokkeloon mennään vasemmasta yläkulmasta ja poistutaan oikeasta alakulmasta. Muodosta verkko (solmu jokaiseen sokkelon risteyskohtaan, jotka on alla merkitty kirjaimilla) ja suorita algoritmi yksityiskohtaisesti.



Tehtävä 6.2 Kuten luennoista muistetaan, syvyyshakualgoritmi on seuraavan lainen:

DFS (G) 1. for each u in V 2. $color[u] = WHITE$ 3. $p[u] = NIL$ 4. $time = 0$ 5. for each u in V 6. if $color[u] == WHITE$ 7. $DFS_VISIT(u)$ 8. return	DFS_VISIT(u) 1. $color[u] = GRAY$ 2. $time = time + 1$ 3. $d[u] = time$ 4. for each v in $Adj[u]$ 5. if $color[v] == WHITE$ 6. $p[v] = u$ 7. $DFS_VISIT(v)$ 8. $color[u] = BLACK$ 9. $time = time + 1$ 10. $f[u] = time$ 11. return
---	--

Suorita syvyysshaku seuraavalle suunnatulle verkolle lähtien solmusta a ja luokittele verkon välit puuväleihin, eteneviin väleihin, takautuviin väleihin ja poikittaisväleihin. Tarkista myös, voiko algoritmin suorituksen aikana havaita verkossa sykliä.



Ohjelmointitehtävä seuraavalla sivulla ->

Ohjelmointitehtävä

Tehtävä 6.3 Suunnittele ja implementoi (joko C- tai Python-kielellä) muokattu **leveyshakualgoritmi** niin, että algoritmi havaitsee, onko verkko **kaksijakoinen** (**bipartite**). Kaksijakoiseksi sanotaan suuntaamatonta verkkoa, jonka solmut halutaan värittää punaisella ja sinisellä värillä niin, että verkon jokaisen välin toinen solmu on punainen ja toinen sininen. Algoritmin tulisi myös tehdä väritys, mikäli se on mahdollinen. Alla olevassa linkissä on koodi, jossa on toteutettu leveyshakualgoritmi. Sovella ohjelmaasi seuraaviin verkkoihin

