

GraphVisualizer määrittelydokumentti

Aihe

Työn aiheena on laatia Java-ohjelma, joka piirtää näytölle verkon saamansa syötteen perusteella. Tarkoituksena olisi myös mahdollistaa verkon muokkaaminen graafisella käyttöliittymällä.

Verkko voi olla suunnattu tai suuntaamaton, painotettu tai painottamaton, syklinen tai sykliton.

Ohjelma lisäksi havainnollistaa visuaalisesti erilaisia verkkoalgoritmeja, kuten esimerkiksi leveyssuuntainen haku, Dijkstran algoritmi ja Primin algoritmi.

Syötteet ja niiden käyttö

Ohjelma ottaa syötteenään luettelon verkon solmuista ja kaarista tekstitiedostona, jonka ohjelma tulkitsee kaksiulotteiseksi taulukoksi (vierusmatriisiesitys) tai vieruslistaesitykseksi. Lisäksi käyttäjä voi näppäimistöllä ja hiirellä manipuloida verkkoa sen visuaalisessa esitysmuodossa.

Tarvittavat tietorakenteet ja algoritmit

Verkon säilyttämiseen muistissa tarvitaan jonkinlainen tietorakenne, joka pitää sisällään verkon kaaret ja solmut. Tähän on kaksi vaihtoehtoa: vieruslista- tai vierusmatriisiesitys. Kummatkin tarvitsevat taulukon, ja jos verkon kokoa voi muuttaa ohjelman suorituksen aikana, taulukon pitää olla dynaaminen. Tämä voidaan toteuttaa siten, että staattinen taulukko täytyttyään kopioi alkiot uuteen, tuplasti isompaan taulukkoon.

Vieruslistaesitystä varten tarvitaan (yhteen suuntaan) linkitetty lista kunkin solmun kaarien tallettamiseksi. Tähän voidaan kuitenkin käyttää myös kahteen suuntaan linkitettyä listaa ilman vaikutusta ohjelman suorituskykyyn.

Leveyssuuntaista läpikäyntiä varten tarvitaan jono, joka voi sekin hyödyntää kahteen suuntaan linkitettyä listaa. Myös syvyysuuntainen läpikäynti on helppo toteuttaa samalla linkitetyllä listalla.

Dijkstran algoritmin toteuttamiseksi tarvitaan kekorakenne, joka käyttää aiemmin esiteltyä dynaamista taulukkoa.

Tavoitteena olevat aika- ja tilavaativuudet

Tietorakenteet

Kahteen suuntaan linkitetty lista

Tilavaativuus:	$O(n)$
Search:	$O(n)$
Insert:	$O(1)$
Delete:	$O(1)$

Dynaaminen taulukko

Tilavaativuus:	$O(n)$
Insert:	$O(n)$
Delete:	$O(n)$

Jono

Tilavaativuus:	$O(n)$
Enqueue:	$O(1)$
Dequeue:	$O(1)$
IsEmpty:	$O(1)$

Keko

Tilavaativuus:	$O(n)$
Insert:	$O(\log n)$
DelMax:	$O(\log n)$
SetKey:	$O(\log n)$

Verkkoalgoritmit

Kuvataan Solmujen lukumäärää $|V|$ merkillä n , ja kaarten lukumäärää $|E|$ merkillä m .

	Aikavaativuus	Tilavaativuus
Leveyssuuntainen läpikäynti:	$O(n+m)$	$O(n)$
Dijkstran algoritmi:	$O((n+m) \log n)$	$O(n)$
Primin algoritmi:	$O(m \log n) / O(n^2)$	$O(n)$

Lähteet

Tietorakenteet ja algoritmit -kurssimateriaali

<http://www.cs.helsinki.fi/u/mikahutt/tira.pdf>

<http://bigocheatsheet.com/>