

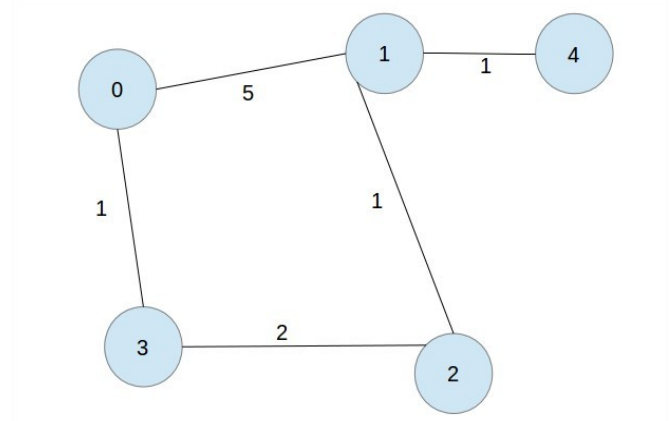
Reitinhaku Dijkstran algoritmia käyttäen Testaus

Ohjelmani testaus on pääosin toteutettu laajoilla automaattisilla JUnit -testeillä. Jokaista ohjelman luokkaa varten on toteutettu oma testausluokka, jossa testataan kaikkia luokan olennaisia metodeja. Setterien ja getterien testaus on jätetty vähemmälle, paitsi jos ne sisältävät jotakin erikoista. Automaattiset testit on laadittu siten että ne testaavat yksittäisten kutsujen lisäksi myös monien perättäisten kutsujen sarjoja. Minimikeosta olen automaattisilla testeillä testannut solmujen lisäämistä, poistoa ja etsimistä. Lisäämisessä olennaista on, että keko kasvaa tarpeen mukaan. Poistossa taas olennaista on, että oikea solmu palautuu, ts. kekoehdot on säilynyt lisäyksissä. Lista-tietorakenteen testeissä testasin myös lisäyksen kykyä kasvattaa listaa, ja että lisätyt solmut löytyvät listalta.

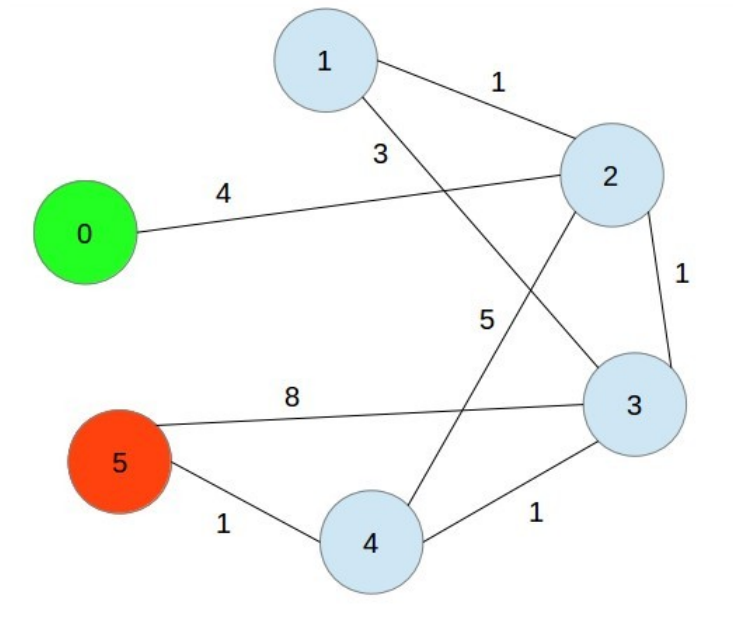
Automaattisen testauksen lisäksi ohjelmaa on ajettu erilaisilla testisyötteillä, aluksi valmiiksi "kovakoodatuilla" taulukoilla ja myöhemmin Ui:lle syötteitä antamalla.

Kaksi erilaista verkkoa testattuna syöttämällä ne käyttöliittymälle;

```
osku@osku-VirtualBox:~/tiralabra/dist$ java -jar Tiralabra.jar
Montako solmua?
5
Anna solmujen 0 ja 1 välinen etäisyys (0 = ei yhteyttä)
5
Anna solmujen 0 ja 2 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 0 ja 3 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 0 ja 4 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 1 ja 2 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 1 ja 3 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 1 ja 4 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 2 ja 3 välinen etäisyys (0 = ei yhteyttä)
2
Anna solmujen 2 ja 4 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 3 ja 4 välinen etäisyys (0 = ei yhteyttä)
0
Anna lähtösolmun numero (ensimmäinen solmu on solmu 0)
0
Anna maalisolmun numero (viimeinen solmu on solmujen lkm -1)
4
Lyhin reitti on: 0 3 2 1 4
Reitin pituus on: 5
osku@osku-VirtualBox:~/tiralabra/dist$
```



```
osku@osku-VirtualBox:~/tiralabra/dist$ java -jar Tiralabra.jar
Montako solmua?
6
Anna solmujen 0 ja 1 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 0 ja 2 välinen etäisyys (0 = ei yhteyttä)
4
Anna solmujen 0 ja 3 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 0 ja 4 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 0 ja 5 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 1 ja 2 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 1 ja 3 välinen etäisyys (0 = ei yhteyttä)
3
Anna solmujen 1 ja 4 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 1 ja 5 välinen etäisyys (0 = ei yhteyttä)
0
Anna solmujen 2 ja 3 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 2 ja 4 välinen etäisyys (0 = ei yhteyttä)
5
Anna solmujen 2 ja 5 välinen etäisyys (0 = ei yhteyttä)
8
Anna solmujen 3 ja 4 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 3 ja 5 välinen etäisyys (0 = ei yhteyttä)
1
Anna solmujen 4 ja 5 välinen etäisyys (0 = ei yhteyttä)
1
Anna lähtösolmun numero (ensimmäinen solmu on solmu 0)
0
Anna maalisolmun numero (viimeinen solmu on solmujen lkm -1)
5
Lyhin reitti on: 0 2 3 4 5
Reitin pituus on: 7
osku@osku-VirtualBox:~/tiralabra/dist$
```



Testaus on sujunut melkolailla ongelmitta sillä olen tehnyt automaattisia testejä yleensä sen mukaan kun olen kirjoittanut ohjelmaan uutta koodia, näin olen myös löytänyt mahdollisia virheitä testien avulla. Erityisesti testeistä oli hyötyä omien tietorakenteiden käyttöönoton yhteydessä, kun virheet löytyivät koodista helpommin niiden avulla.