Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit - Testaustdokumentti

Jerry Mesimäki

1. Mitä testattiin ja miten testattiin
Algoritmianalysaattorin testaaminen tapahtui pääasiallisesti koodaamisen aikana main-luokkaan kirjoitetuilla yksinkertaisilla testeillä. Labyrinttigeneraattoria luodessani minulla oli käytössä metodi, joka yksinkertaisesti tulosti labyrintin helposti ymmärrettävään muotoon. Generaattoin syvyyssuuntaista brute-forcea varten tein metodin, joka tulosti luolaston ja kertoi onko se ratkaistavissa vai ei. Tämän jälkeen tarkastin luolaston ratkaistavuuden itse. Todettuani tarpeeksi monta kertaa, että DFS todella osaa antaa oikean vastauksen oli testaus siltä osin suoritettu.

Algoritmien testaaminen taas tapahtui siten, että käytin leveyssuuntaista brute-force -metodia luodessani Bellman-Fordia, joka löysi hyvin hitaasti lyhimmän reitin. Tarkastin samalla myös käsipelillä toimivatko molemmat pienissä luolastoissa. Tämä taas vaati luolaston tulostelua mm. solmujen etäisyyksien tai kuljettavuuden perusteella. Kun olin mielestäni havainnut algoritmin toimivaksi pystyin käyttämään sitä implementoidessani A*:ia ja Dijkstraa. Kun ne saivat esim. 10000 luolaston loopissa aina saman tuloksen bellmanin kanssa pystyin olettamaan, että ne antavat oikeita ratkaisuja.

Minimikekoa testasin lisäämällä ja poistamalla sieltä erilaisia random-generoituja solmuja. Heapify-metodia pystyi myös helposti testaamaan isostakin keosta pienellä koodinpätkällä, joka kyselee sattumanvaraisilta solmuilta onko se oikeaa ja vasenta lastaan pienempi.

2. Minkälaisilla syötteillä testaus tehtiin
Yksikkötestit testaavat niin pieniä kuin suuriakin luolastoja, koodamisen lomassa väliaikaiset testimetodit taas olivat lähinnä sellaisia, että niiden käsittelemiä asioita pystyi simuloimaan kynällä ja paperilla.

3. Testien toistaminen
Ohjelma antaa käyttäjälle vapaat kädet kokeilla algoritmien nopeuksia ja yksikkötestejä voi halutessaan ajaa, sillä ne löytyvät projektikansiosta.