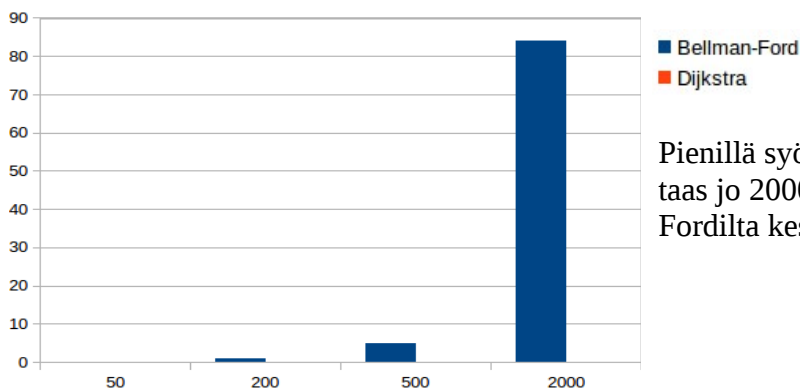


Testausdokumentti

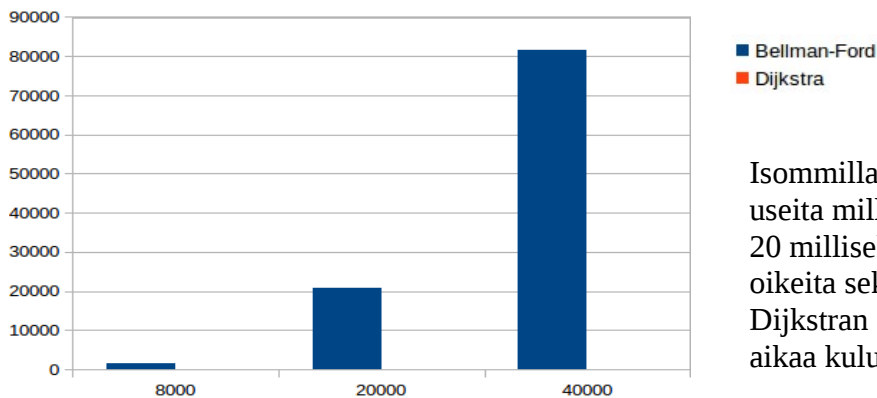
Manuaalisen testauksen avulla on käynyt selväksi, että Bellman-Ford on monta kertaa hitaampi kuin Dijkstra. Pienillä syötteillä Dijkstra käyttää yhden millisekunnin, kun taas Bellman-Ford kolmisenkymmentä. Annoin myös 200*200 solmun taulukon kummankin algoritmin ratkaistavaksi: Dijkstralla aikaa meni keskimäärin 19 millisekuntia lyhimmän reitin löytämiseen, kun Bellman-Ford käytti aikaa keskimäärin 81638 millisekuntia (20 toistoa).

Vielä alla pylväsdiagrammi, jossa seinätodennäköisyys oli 0.19, toistojen määrä 20 ja taulukon solmujen määrä on pylvästen alla ja korkeus taas kertoo hakujen keskiarvoajan millisekunteina.

Vielä alla pylväsdiagrammi, jossa seinätodennäköisyys oli 0.19, toistojen määrä 20 ja taulukon solmujen määrä on pylvästen alla ja korkeus taas kertoo hakujen keskiarvoajan millisekunteina.



Pienillä syötteillä Dijkstralla meni 0 ms, kun taas jo 2000 solmun taulukko vei Bellman-Fordilta keskimäärin 84 ms.



Isommilla syötteillä Dijkstrallakin alkaa kulua useita millisekunteja, mutta selvitään silti alle 20 millisekunnin. Bellman-Ford taas kuluttaa oikeita sekunteja ja minuuttejakin. Vasta Dijkstran saatua 250 000 solmun taulukko aikaa kului keskimäärin 188 millisekuntia.

Sovellusta on testattu manuaalisesti todella paljon, sillä yhdessä vaiheessa NullPointterin etsiminen vaati manuaalista testausta. Bellman-Ford toimi riippuen taulukon koosta, seinien määrästä ja sijainnista sekä lähtö- ja maalipisteestä. Oli siis labyrinttejä, joissa haku onnistui, kun lähtöpiste oli 0,0 ja maalipiste vastakkainen kulma, mutta aloitettaessa vastakkaisesta kulmasta tarkoituksena löytää reitti pisteeseen 0,0 haku epäonnistui.

Ohjelmaa on testattu eri kokoisilla taulukoilla. On testattu, kuinka iso seinätodennäköisyys aiheuttaa sen, että ratkaistavaa labyrinttiä ei meinata saada luotua ($t_n > 0.4$). On testattu, kuinka isoja labyrintteja kannattaa syöttää ja tultu siihen tulokseen, että dijkstra taipuu sadankin tuhannen solmun syötteellä, kunhan on aikaa, mutta Bellman-Fordilla ei kannata ratkaista kymmenien tuhansien alkioiden labyrinttejä. Isoilla syötteillä Bellman-Ford aiheuttaa muistinloppumisen. Lisäksi käyttöliittymää on testattu. JUnit-testit ovat erittäin vähäiset.