Project 3: Optimizing a fuel network

# Introduction

For this project my objective was to supply multiple fuel stations from multiple depots. The primary objective was to optimize the routes the tankers should take so that none of the tanks had to drive a far distance.

# Approach

I was given two parameters, a map of all of the connections between vertex and a list of all of the stations and depots. I read in these files and then implemented Dijkstra's algorithm to find the best route.

**Dijkstra's algorithm**

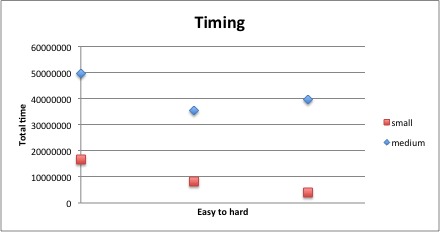
I used Dijkstra's algorithm to find the shortest path between the depots and stations. I reused my methods from lab 10 and lab 8 to read the file in, construct the graph and find the quickest paths.

# Methods

I only optimized the quickest route to an individual station. Part of the requirement was to minimize the longest route but if I had already found the shortest route I could not make any routes shorter. One issue that cropped up was there were some stations that were not connected to any depots so I don't display any route when that issue happens.

# Data and Analysis

I analyzed two different variables. I examined how the timing increased as the 'difficulty' of the routing became harder and as the size increased. Also I examined the range of different lengths. One issue I noticed was that for the large data samples the path lengths were 0 and all from the same vertex. This did not appear to be correct so I did not include this information in my analysis. Also I had an issue where a station was not connected to any depots so if that happens no data is returned for that station.



My data shows a surprising trend; the difficulty of the search doesn't seem to make much difference in the time to complete the optimization. Fortunately it is clear that the larger the graph the more time it takes to optimize every station.

Another surprising trend was that as the graph size increased the average distance between a station and it's closest depot decreased. This could be because there are more options so it is more likely for a station to be next to a depot. Unsurprisingly the difficulty of optimizing a route did not change the average length of a path.

# Conclusion

In conclusion, the more vertex there are the longer it takes to solve the quickest route for all stations. Also it appears that, assuming that the connections are random, the more connections there are the shorter the average path length will be.

# References

Project description: https://moodle.lafayette.edu/pluginfile.php/141173/mod\_resource/content/1/p3.pdf

Excel tutorials:

http://stackoverflow.com/questions/15124103/excel-how-can-i-make-a-scatter-plot-which-colors-by-a-third-column

http://www.excelbanter.com/showthread.php?t=117549

VBA tutorials:

http://stackoverflow.com/questions/12933279/how-to-comment-and-uncomment-blocks-of-code-in-the-office-vba-editor

http://www.cpearson.com/excel/declaringvariables.aspx

http://stackoverflow.com/questions/17194105/how-can-i-color-dots-in-a-xy-scatterplot-according-to-column-value

http://stackoverflow.com/questions/15981802/changing-the-colors-of-the-specific-dots-in-scatterplot-vba-excel/15982217#15982217

Bash tutorials:

http://www.tldp.org/LDP/Bash-Beginners-Guide/html/sect\_07\_01.html

Java api:

https://docs.oracle.com/javase/7/docs/api/

http://docs.oracle.com/javase/6/docs/api/

Book:

Data Structures & Problem Solving Using Java

Timing tutorial:

http://cs.lafayette.edu/~liew/courses/cs150/lab/labs/lab02f

Graph construction:

http://cs.lafayette.edu/~liew/courses/cs150/lab/labs/lab10b/

File reading:

http://cs.lafayette.edu/~liew/courses/cs150/lab/labs/lab08e/

Quickest path between 2 vertex:

http://stackoverflow.com/questions/17480022/java-find-shortest-path-between-2-points-in-a-distance-weighted-map

Implementation of Dijkstra's algorithm:

http://en.literateprograms.org/Dijkstra%27s\_algorithm\_(Java)

http://en.literateprograms.org/index.php?title=Special%3aDownloadCode/Dijkstra%27s\_algorithm\_%28Java%29&oldid=15444