

# Minimum spanning tree algorithms

Glen Lloyd-MacGilp

2394934

## Proposal

### Motivation

The minimum spanning tree is a structure used in solving certain types of combinatorial optimization problems. Example application areas include network design such as roads, telephone, cable laying, electrical, and so on. There are 2 popular greedy algorithms for acquiring the minimum spanning tree of a graph: (i) Kruskal's algorithm and (ii) Prim's algorithm. Another older algorithm is Boruvka's algorithm which will also be included in the study.

### Aims

The aim of this project is to implement these algorithms and carry out an empirical analysis on the implementation, to identify which of the algorithms perform better on randomly generated or real-world datasets. I will also use these algorithms to approximate the travelling salesman problem and produce an approximate optimal tour of all the distilleries in Scotland.

### Progress

- Researched algorithms and papers with similar aims to learn more and develop an in depth understanding of the algorithms and my tasks
- Created a GitHub repository and keep up to date code on it
- Made decisions on data structures and language for my project
- Created naïve and basic implementations of each algorithm
- Created more optimized implementations of each algorithm by using appropriate data structures to accelerate them
- Created a graph generator that takes in parameters to create pseudo random graphs based on inputs
- Made algorithms read graphs in from a file in the format that the graph generator outputs them
- Performed correctness testing on algorithms to ensure accuracy
- Started analysis and comparative study on algorithms

## Problems and risks

### Problems

- Had a problem with implementing DFS to detect cycles for Kruskal's algorithm. Taking a step back and starting again solved this problem however it ate up 2 weeks of my time
- Function to read files in could have an error (only in edge cases) in it and I need to do more testing to ensure it's working properly for all graphs

### Risks

- Algorithms could not be working properly as I have so far only tested them with smaller graphs and quickly with larger graphs. Mitigation: perform more testing on normal and extreme data and test the algorithms on a larger graph with solutions published with it

### Plan

- Over Christmas: Perform more in depth correctness testing for algorithms and function to read graphs. Keep working on analysis between algorithms and start working on my travelling salesman estimation.
- Week 1 – 3: Finish all coding. Travelling salesman estimation finished and analysis finished.
- Week 3 – 10: Write dissertation. First draft submitted to supervisor 3 weeks before deadline

Aim to have dissertation completely finished 1 week before the deadline.