

Report: Super Vector Mario

by Mikkel Bernt Buchvardt and Theodor Lars Nyholm Ommen

1 Results

The following table summarizes our result:

Input file	MST total weight
USA-highway-miles.txt	16598.0
tinyEWG-alpha.txt	181

The MST we found in tinyEWG-alpha.txt can be drawn like this:

2 Implementation details

We consider a map, where the coordinates come from loading the strings from the in-file into a String array.

We talked about two ways to implement a solution:

- starting in an S-field and pushing all legal fields from there to a queue and then pop the fields from the queue and push legal fields from these to the queue until a F-field is reached, counting each move and saving the smallest value for print in the end.
- doing a recursive method that starts in a S-field and calls itself on all legal fields until a F-field is reached.

Pros and cons on each method made us decide on the first.

The argument is as follows:

If we store all visited fields with the speed we have visited them with in a simple table, we can check if we have been in a specific position with a specific speed (given by two coordinates) - if this is the case we do not go to this field again. This works because the algorithm gets us to a specific field in the shortest possible route.

This is only true though if we use the queue method. Using the recursive method we might overshoot the target and get the first results of hitting an F on a longer route than the shortest (in fact this happens on all but the smallest maps). If we want to make the recursive method work we need to consider not only if we have been to a certain field with a certain speed but also if we have been there on a shorter route than the present.

In other words the recursive method takes more calculations, more memory and more time.