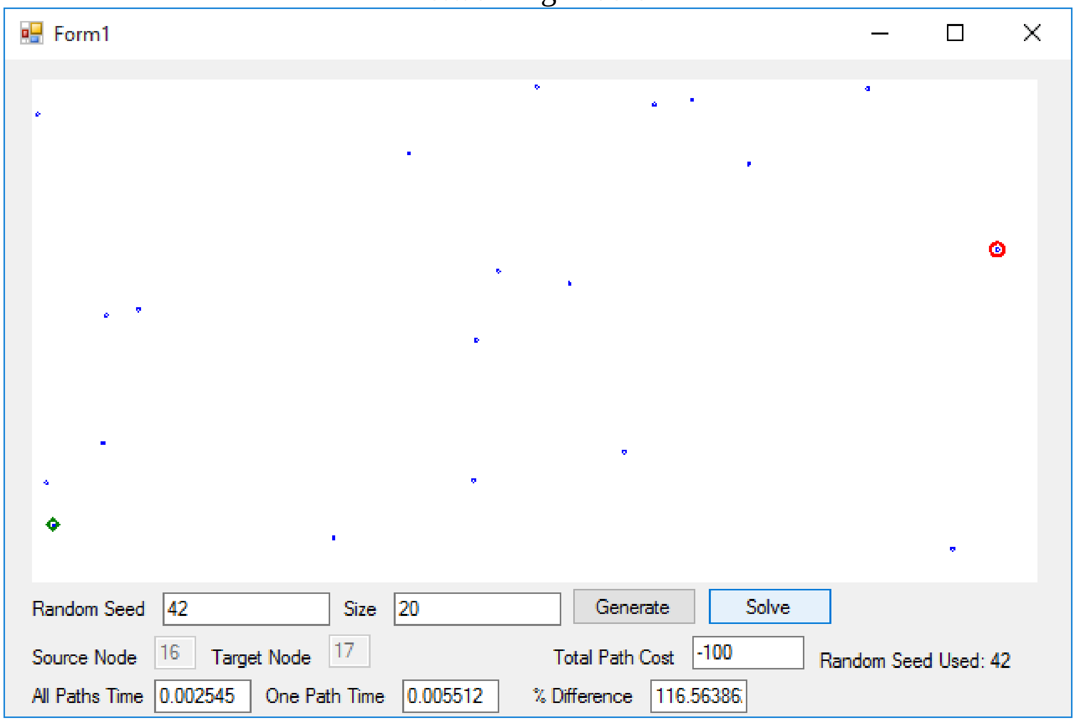
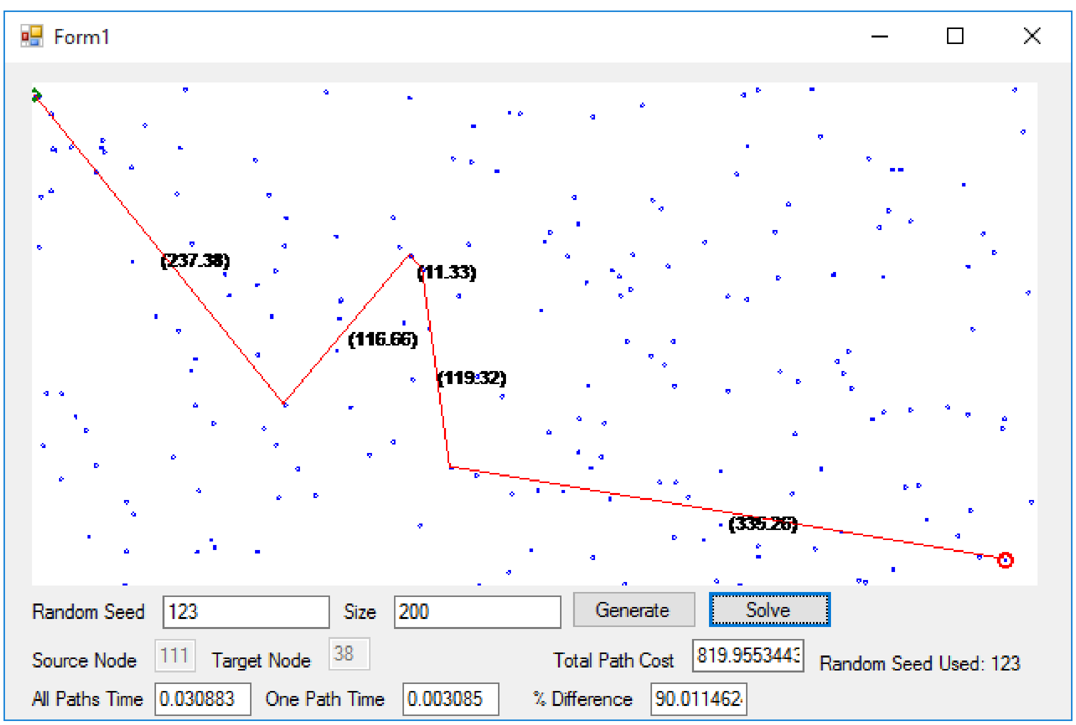
Anthony Constantino
Project 3 Write Up

Networking Problem



In the graph above, there is no solution.



To solve this problem, I used a SortedSet data structure. I chose this data structure because of its tree like, which resembles that of a Red-black tree structure, and because of its fast access to with methods: find, add, insert and remove[1]. All of which have speeds of O(log n), allowing us to achieve optimal speeds, when creating the fastest path.

The reason for each of these functions are O(log n) is because of that tree like structure. First, with the DeleteMin function, we remove the top node from the tree which is O(1) and then the nodes on one side of the tree bubble up the tree based on their distances O(log n). When a node's distance from the starting point is updated that node is then rebased in the tree and bubbles as far up the tree as it can. Making it O(log n) because the most it will go up in the tree if from a leaf to the root. With an insert, the node is inserted and then will bubble up the tree as far as it can rebase on its distance value O(log n).

For the empirical time complexity of the two algorithms, I would say the one-path algorithm is "typically" 9 times faster than the all-path algorithm. As the number of points goes up you would imagine that the one-path would be significantly faster than all-path because it doesn't go through every single node, but that's not the case. While the all-path increases by size n and one-path does not, there is not always a really nice path from the start node to the end node. The one-path still tends to have to go through a good amount of nodes before he discovers the end node.

[1] http://www.growingwiththeweb.com/2013/02/what-data-structure-net-collections-use.html