Evan Cripe

Assignment Six

Minimum Spanning Tree

4/29/17

Abstract:

The problem at hand was to write an algorithm to solve the Minimum Spanning Tree (MSP) problem. This is similar to the Traveling Salesperson Problem except this algorithm is much faster. The only problem is that it does not find the absolute best solution, it only finds a good solution. In other words, it sacrifices the optimal solution in return for an algorithm that is far more efficient. For smaller numbers like 12, 13, 14, 15 or even 16, The traveling salesman problem done in lab 4 did not take that long, but there is not enough time in the universe to complete the TSP for 29 cities. This is where the MSP algorithm comes in handy. It goes through the 29 cities very fast (less than a second) and finds a solution, while not optimal, still good none the less. The algorithm starts at city 0 and looks for the closest city and goes to that city so long as that city has not been visited. If it has been visited, then it goes to the next closest city. This continues for all of the remaining cities until every city has been visited exactly once. This means that the time complexity is O(N^2) compared to the dreaded O(N!) time complexity for the TSP algorithm.

Outputs:

======= MSP for n = 12 =======
Path: 0, 5, 7, 4, 9, 1, 8, 10, 3, 11, 2, 6
Cost: 1351
======= MSP for n = 13 =======
Path: 0, 5, 7, 4, 9, 1, 8, 10, 12, 3, 11, 2, 6
Cost: 1476
======= MSP for n = 14 =======
Path: 0, 5, 7, 4, 9, 1, 8, 10, 12, 3, 11, 2, 6, 13
Cost: 1710
======= MSP for n = 15 =======
Path: 0, 5, 7, 4, 9, 1, 8, 10, 12, 14, 3, 11, 2, 6, 13
Cost: 1835
======= MSP for n = 16 =======
Path: 0, 5, 7, 12, 4, 11, 1, 8, 9, 3, 14, 2, 6, 13, 10, 15
Cost: 2929
======= MSP for n = 19 =======
Path: 0, 5, 7, 12, 4, 11, 1, 15, 8, 9, 3, 18, 14, 2, 17, 6, 13, 10, 16
Cost: 2618
======= MSP for n = 29 =======
Path: 0, 27, 23, 20, 5, 7, 12, 26, 4, 11, 1, 15, 8, 9, 19, 25, 3, 28, 18, 22, 14, 24, 2, 17, 6, 13, 10, 21, 16
Cost: 3834