Thomas Wiles

Programming Assignment 6

Traveling Salesperson – Minimum Spanning Tree

May 2, 2017

Abstract: The goal of the project is to solve the traveling salesman problem, meaning the shortest path from a starting point that traverses through each city and returns to the original. The most comprehensive way to find the shortest path is to examine every single combination, as seen in lab 4. However, this is immensely time consuming as the possibilities reach N! amounts. Yet there is a convenient way to approximate this using stacks. Taking the current city, one can find the shortest path to the next city in a reasonable time, since at worst there are N comparisons to make. While this may not give the shortest total path, it provides an approximation. Most importantly, this approximation is calculated in a very short amount of time. When the original recursive TSP was calculated, it required 18 minutes to analyze 12, 13, 14, 15, 16, and 19 cities. The case for 29 cities was never reached. The new algorithm required 1 second to analyze all of the situations and give an approximate minimum path for each of them. In the situations for smaller cities, the new algorithm is fairly accurate. For 12 and 13 cities, it gave distances of 920 and 929 respectively. The recursive algorithm gave distances of 821 and 864 as the absolute minimum. For the larger situations, its accuracy was very poor. For 16 and 19 cities, it gave distances of 1623 and 1660 respectively. The recursive algorithm gave 1222 for 16, and never quite finished 19 but the best result found so far was 1467. So for the larger situations there are significant deviations between the two methods. Still, at least the new algorithm finished each of the calculations. It was even able to give a result for 29 cities, which was 2005 distance units. As such, the new algorithm provides a much faster method for approximating the minimum distance when traveling, but cannot give the true minimum because it does not run through every single possibility. This just exemplifies the struggle between desire for accuracy and time requirements.

```
run:
 ****************
TSP for 12 cities.
 ****************
0, 5, 3, 8, 4, 1, 11, 6, 7, 10, 9, 2,
Distance:  920
Duration:  0.103 sec


 ****************
TSP for 13 cities.
 ****************
0, 5, 3, 8, 4, 1, 11, 6, 7, 10, 9, 2, 12,
Distance:  929
Duration:  0.20 sec


 ****************
TSP for 14 cities.
 ****************
0, 5, 3, 8, 4, 1, 13, 11, 6, 7, 10, 9, 2, 12,
Distance:  1025
Duration:  0.15 sec


 ****************
TSP for 15 cities.
 ****************
0, 5, 3, 8, 4, 1, 13, 14, 12, 2, 9, 10, 7, 6, 11,
Distance:  994
Duration:  0.45 sec
```

****
TSP for 16 cities.
****
0, 5, 11, 8, 4, 1, 9, 3, 14, 13, 10, 15, 12, 7, 6, 2,
Distance: 1623
Duration: 0.17 sec

****
TSP for 19 cities.
****
0, 5, 11, 8, 4, 1, 9, 3, 14, 18, 15, 12, 7, 6, 10, 13, 17, 16, 2,
Distance: 1660
Duration: 0.14 sec

****
TSP for 29 cities.
****
0, 27, 5, 11, 8, 4, 20, 1, 19, 9, 3, 14, 18, 24, 6, 22, 26, 23, 7, 15, 12, 17, 13, 21, 16, 10, 28, 25, 2,
Distance: 2005
Duration: 0.73 sec

BUILD SUCCESSFUL (total time: 1 second)