Tyler Goodwin

C202

Program Assignment 6

December 2, 2016

With this program we are challenged with the task to solve a Traveling Salesperson Problem with a Stack (Shortest Path Tree). We are to use a non-recursive algorithm, and use lab assignment five as a starter. We are to create a adjacency matrix using the given files in lab five.

Along with lab five we are given an algorithm to create the method which we will use to find the shortest path tree. We are to return the shortest path from starter city to the last city.

The algorithm we are to use to create the method states that we are to create an empty stack and an empty array. In the array we are to store the number of cities visited. Along with these we are to add city 0 to the stack we created and we then set a variable named closestCity to zero. After this we create a true and false statement for minFlag, and we show the start city. After this we check to see if the stack is empty, if it is not we make our current city the top value of the stack and we set the minimum distance. While we are still checking if the stack is empty, we also check the remaining cities and return the minimum distance from the current city to city one, make the closest city now city 1, and make minFlag true. If the minFlag is true we then set closest city to visited, add closest city to the stack, output the closest city, set minFlag to false, then we return true. Lastly, we output the top element of the stack.

After finishing the implementation of lab five and the new algorithm I ran to see the difference between the two. Lab five took much longer to run than programming assignment six, and also the path results varied. Lab five could not finish running 19 and 29 for me, but programming assignment six finished almost instantly. The concluding result was that programming assignment six was much faster than lab five at finding the shortest path, but as for determining the cheapest cost it was ineffective.

My observations at the beginning were that they would be the same as I did not think in terms of shortest path, not lowest cost. The outputs I got seemed incorrect to me because of this, after checking I see that many others got the same results so I concluded that these must be the correct paths. After thinking a little I noticed why they would not be the same results, this is because lab five took into account cost of each path, so it took the cheapest rout rather than the fastest. While programming assignment six only looked for the shortest path and returned that path no matter the cost. I was impressed by this I believe I may attempt to implement this using a trip to Alaska from Indiana, I have driven it many times but have always wondered what exactly the shortest path would be.

Outputs (netbeans):

```
Please type in the number of cities:
```

```
12
Please type in the file name:
tsp12.txt
0 5 3 8 4 1 11 6 7 10 9 2
2690
Time taken: 503885

Please type in the number of cities:
13
Please type in the file name:
tsp13.txt
0 5 3 8 4 1 11 6 7 10 9 2 12
2779
Time taken: 529522

Please type in the number of cities:
14
Please type in the file name:
tsp14.txt
0 5 3 8 4 1 13 11 6 7 10 9 2 12
3521
Time taken: 551430



Please type in the number of cities:
15
```

```
Please type in the file name:
tsp15.txt
0 5 3 8 4 1 13 14 12 2 9 10 7 6 11
3874
Time taken: 707583

Please type in the number of cities:
16
Please type in the file name:
tsp16.txt
0 5 11 8 4 1 9 3 14 13 10 15 12 7 6 2
5935
Time taken: 598043

Please type in the number of cities:
19
Please type in the file name:
tsp19.txt
0 5 11 8 4 1 9 3 14 18 15 12 7 6 10 13 17 16 2
6812
Time taken: 1807646



Please type in the number of cities:
29
```

```
Please type in the file name:

tsp29.txt

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

11258

Time taken: 2549256
```