Ural Sarp Sipahi 28093

1) We can think of the tourist problem as a single source shortest path problem. The recursive formulation of the tourist problem can be written down as follows:

(S0 = the initial distances between the source station to each station
Sk = the final distances between between the source station to each station
W=the distance between one station pair)

$S0(s,v) = \infty$ for $s \neq v$

(The initial distances between all vertices except the source vertex, is set to infinity)

$Sk(s,s) = 0$ for every $k = 0 \ldots V-1$

$Sk(s,v) = \min(\{ Sk-1(s,u) + w(u,v) \} \{ Sk(s,v) \})$ for $(u,v) \in E$

2) The pseudocode of the algorithm is as follows:

d = the array that holds the information of the distances of each station to the source station

d [s] * V = "Inf" #initially set all the values to infinity.

d [0] = 0 #0 is the source station's index. Set it to zero since it's distance to itself is 0.

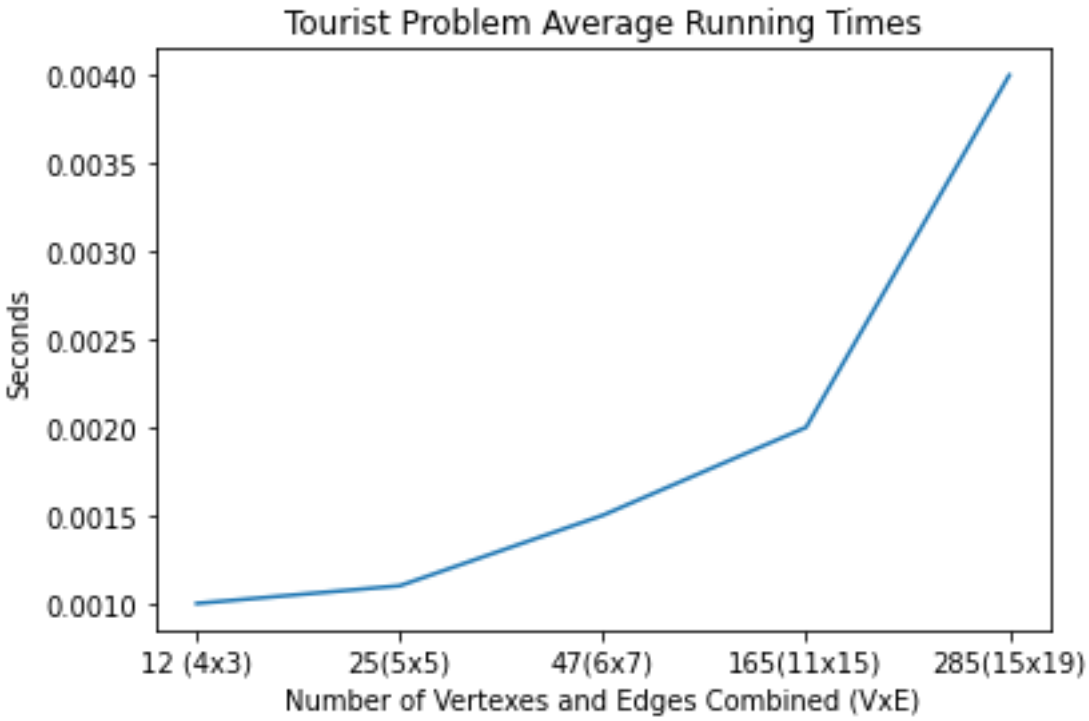
```
For k = 1 to |V-1|                          #O(V)
    For each edge (u,v)                     #O(E)
        If( d[v] > d[u] + w[u,v] )
            d[v] = d[u] + w[u,v]
```

3) As for the asymptotic time and space complexities:

Asymptotic time complexity of the algorithm is O(V * E). This is due to the two for loops above. We "relax" each edge depending on the if condition above. In the worst case we may need to relax all edges V-1 times.

As for the space complexity, we are using 1 array for keeping the distances between the source vertex to each other vertex. Therefore it's space complexity is O(V). In addition, in my algorithm I've put 3 more additional arrays for printing purposes. Each of these arrays contain V elements. Therefore each of them costs O(V) space complexity. In total it is O(4V) = O(V) space complexity.

4)



The results are as expected. As we can see from the graph as we increase the number of vertices and edges the running times increase gradually. These results support the fact that this algorithm has a running time of O(V * E).