HW4

Chengqi Huang chengqihuang@gatech.edu

Problem 1

Part(a)

Each intersection of the city is represented by vertices v_i, each street in the city is represented by an edge e_i, and since all streets are one-way, all edges are directed with the same direction of the road. Create this graph G = (V,E), were V includes all vertices, and all edges are included in E. now the problems becomes for each u, v in V, whether it exist a path from u to v and we need to get this problem resolved in linear time.

SCC algorithm is a good idea. We simply need to see if the graph G is one SCC on its own, if yes, then the mayor is right. The run time of SCC is O(m+n) which is linear time.

Part(b)

Still we use the same formation of the problem, but this time we need to see if the town hall is within a sink SCC. If the town hall is not in a sink SCC, then there's path to reach other intersections that are out of this SCC, but cannot drive back to the townhall. If the town hall is within a sink SCC, then people can reach every other intersections in this SCC, and they are able to go back to the town hall. The total run time of the algorithm is still O(m+n), linear time.

Problem 2

Tree is connected, acyclic graph. So for any cycle C in graph G, if the input edge e is part of a cycle, and e is greater than any other edges in this cycle, then e won't be part of the MST. With this statement, we have below algorithm:

Step 1:
Run DFS, using vertices u as a starting point. Only include edges with weight $c_i<c_e$.
Step 2:
While finish running DFS, if u and v are connected, that means there's cycle with e(u,v) being the largest edge of the cycle. In this case, e is not part of the MST
Step 3:
If u and v are not connected with each other, e is part of some MST because E is not the greatest weight edge in any cycle.

Runtime:
The algorithm's runtime equals to the run time of DFS algorithm, which is O(n+m)