Homework 2 COM S 311

Alec Meyer

October 16, 2020

Question 1

```
initialize all nodes as unvisited;
DFSmod(v);

DFSmod(node v)
{
        dfs but mark seen nodes as visited;
}
```

Since we DFS for a log(n) amount of v the runtime will be: O(log(m + n))

Question 2

```
Use Dijkstra's Shortest path algorithm


      distDjikstra[V];
          distDjikstra[V];
          distDjikstra[s] = 0;
          call Djikstra(s, distDjikstra, T);

          distDjikstraG[s] = 0;
          call Djikstra(s, distDjikstraG, G);
          for( vertex x to V)
              if(distDjikstraG[x] not equals distDjikstraSPT[v])
                  return false;
      return true;
```

Minimum Distance of v from S in pi ¡ and ¿ Minimum Distance of v from S in G, therefore the algorithm is correct.

This will be the same runtime as Djikstra's algorithm so the runtime is: O(v + ElogV)

Question 3

```
T = {};
for all v in V:
          key[v] = INFINITY;
```

```
pi[v] = NULL;
Initialize priority queue Q ;
pick vertex r as root;
key[r] = 0;
while Q is not empty:
        u = EXTRACT-MIN(Q)
        if pi[u] != NULL:
                T = T in {(pi[u], u)}
                for each neighbour v of u:
                        if v in Q and w(u, v) < key[v]:
                                key--;
                                pi[v] = u;
```

it takes O(n) to build the sets, O(1) to decrease key, and O(1) to use the find. Therefore the final runtime is still O(m + n)

Question 4

```
array arr = C_i + p_i + a_i
max = arr[0]

for i=1 i<n i++:

    if max < arr[i]

        max = arr[i]

return curMax
```

This implementation will iterate through all vehicle times to paint and assemble. It will then find the max value which will be the total time taken to paint and assemble all vehicles.