Stamatios Morellas
COM S 311 – Homework 4
Recitation: Tuesdays @ 1:10pm
10/25/19

Stamati Morellas

COM S 311 - Homework 4

10/25/19

(1) Steps of algorithm:

1. Create and initialize adjacency list G and G^2
2. For every vertex u in V, traverse the adjacency list for every adjacent vertex.
3. During the traversal, find a vertex V such that V is adjacent to V^1 and V^1 is adjacent to u.
4. If a vertex v is found with value a and the length of the combined edge sequence is a-1, add vertex v as a node in G^2

   Upper bound: $O(m \cdot n)$
5. Repeat until all vertices are found

Algorithm: (brute force)

```
for x=0 to V.size {
    for y=0 to V.size {
        G2[x][y] = 0
        for z=0 to V.size {
            if (G[x][z] == 1 and G[z][y] == 1)
                G2[x][y] == 1
                break;
        }
    }
}
```

Runtime: $O(n^3)$

(3.)

To prove that every DAG has a sink, we will prove this by contradiction.

Assume there is no sink in the DAG, this means that there is at least one edge coming out of every node. Then every time the graph is traversed, there will be a node with an outgoing edge from it regardless of the starting point. Since the graph has a finite amount of nodes, there will always be one node pointing to an already visited node in the traversal. This means that a cycle exists in the graph, which is contradictory to the specifications of a DAG.

Therefore, every DAG must have a sink.

Topological ordering of DAG:

1. Create an array with length being the numbers of nodes in the graph G
2. Find the end node and put it at the end of the array
3. Delete that node from G
4. Repeat this until there are no more nodes left in G and the first node in the array has no incoming edges

(2)

- Set Discovered[s] = true
- Initialize L[] array with s element
- While L[i] not empty, initialize an empty list L[i+1]
- For each node in L[i], increase the count for vertex v if it is not discovered.
- Add v to the list
- Repeat

(4.)

In order for G' to be a DAG, it must be a graph that does not contain any cycles. If G' did contain any cycles, then then that would mean that there is more than 1 $u \in S_i$ and $v \in S_j$ such that $\langle u, v \rangle \in E$. This would then mean that either S_i or S_j is not actually a strongly-connected component. Since we are given the list of strongly-connected components and we will always know that it is true, then this cannot be right.

$\therefore$ G' is a DAG