ASSIGNMENT 7

DUE: Wednesday November 8, 7 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies. In particular, note that “giving” an algorithm includes justifying correctness and run time.

1. [10 marks] Depth First Search and Topological Sort. Let \( G \) be a directed acyclic graph. We want a linear time algorithm to decide if there is a directed path in \( G \) that goes through every vertex exactly once. For each of the following approaches, either prove or disprove that it works correctly.

   (a) [5 marks] Run DFS from some vertex. If the DFS tree is a path, then output the path, otherwise output FAIL.

   (b) [5 marks] Run DFS to find a topological sort of the graph and test if that ordering is a directed path. If it is, then output the path, otherwise output FAIL. (Recall that we get a topological sort by ordering the vertices in decreasing order of finish time.)

2. [10 marks] Path length. Given an undirected graph \( G \) with non-negative weights on the edges and a start node \( s \), there may be more than one shortest path from \( s \) to \( v \). In this case we prefer the shortest path with the fewest edges. Define the hop number, \( h(v) \) to be the minimum number of edges in a shortest path from \( s \) to \( v \). Define \( h(s) \) to be 0. Give an \( O(m \log n) \) time algorithm to compute the hop numbers of all the vertices of \( G \). Here \( n \) is the number of vertices and \( m \) is the number of edges.

3. [10 marks] MST. Suppose you have an edge weighted undirected graph \( G \) and a minimum spanning tree \( T \). Let the weight function be \( w : E \to \mathbb{R}^\geq 0 \). One edge \( e = (u, v) \) changes its weight and we want to update the minimum spanning tree. There are 4 cases. In each case give a linear time algorithm to update \( T \). You may assume that \( T \) is stored as adjacency lists.

   (a) \( e \not\in T \) and its weight goes up

   (b) \( e \not\in T \) and its weight goes down

   (c) \( e \in T \) and its weight goes down

   (d) \( e \in T \) and its weight goes up