Homework #6
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2014

Due on: Tuesday, March 11th, 5pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On blackboard, under student assessment
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

March 4, 2014

Problem 1 (20 points)

Given a connected undirected graph $G = (V, E)$, call a vertex $v \in V$ *vulnerable* if removing v (and all edges that touch vertex v) from graph G would result in G being disconnected.

(i) **15 points** Suppose that we run DFS on graph G starting at node $v \in V$. The resulting DFS tree (it's a tree, not a forest, because G is connected) is rooted at v. Prove that v is vulnerable if and only if v has more than one child in the DFS tree.

(ii) **5 points** Explain how you would use this fact to determine in $O(|V| + |E|)$ time whether or not vertex v is vulnerable in G (Of course, we could also just remove v from G and run DFS on the remaining graph to check for connectivity, but that isn't the point of this problem).

Problem 2 (20 points)

Problem 2.1 (5 points)

Prove that every directed acyclic graph (DAG) has at least one vertex with no entering edges. That is, for any DAG $G = (V, E)$, there exists a node $v \in V$ for which there **do not** exist any edges of the form $(u, v) \in E$.

Problem 2.2 (5 points)

Find a necessary and sufficient set of conditions for a DAG to have a unique topological sort. That is, find a set of statements $S_1, S_2, \ldots, S_m$ such that directed acyclic graph G has a unique topological sort if and only if $S_1, S_2, \ldots, S_m$ are all true. These conditions must be non-trivial and must be properties of the graph itself– i.e., the answer "a graph has a unique topological sort if and only if every time we run the topological sort algorithm the result is the same" is not a valid answer, nor is, say, "a graph has a unique topological sort if and only if it satisfies all of the conditions necessary and sufficient for it to have a topological sort".

Problem 2.3 (10 points)

Write a non-recursive version of DFS. That is, write a new version of depth-first search that doesn't need to call itself. Prove that your algorithm is correct.

Optional exercises

Solve the following problems and exercises from CLRS: 23.2-7, 23.2-1, 23-1, 23-4.