1 Exercise 22.2-7

Do Exercise 22.2-7 on p. 602. You do not have to give a rigorous proof of the correctness of your algorithm, but you do need to give an intuitive explanation why your algorithm works. Remember to analyze the running time.

2 Topological sort

Do Exercise 22.4-2 (on p. 614).

3 Querying on a tree

You are given a rooted binary tree $T = (V, E)$, along with a designated root node $r \in V$. The size of $V$ is $n$. Recall that a node in a tree keeps track of its descendants and its parent node. Also recall that node $u$ is said to be an ancestor of node $v$ in the rooted tree, if the path from $r$ to $v$ in $T$ passes through $u$.

A commonly performed querying on the trees is: given two nodes $u$ and $v$, is $u$ an ancestor of $v$? You wish to preprocess the tree so that queries of this form for any two nodes $u$ and $v$ can be answered in constant time. The preprocessing itself should take linear time. That is, you can spend $O(n)$ time before any query arrives; then you must answer each query like “is node $u$ an ancestor of node $v$?” in constant time.

4 Semi-connected graphs

Do Exercise 22.5-7 (on p.621). To get full credit, your algorithm should run in time $O(|V| + |E|)$.

10% Extra Credits

There are many graph algorithmic problems that you can practice what you learned in class. In the following, I list a few additional problems. If you want, you can select one of these problems, write up a solution and submit for 10% extra credits (that is, you get 110 points for this assignment if you do a perfect job). Well, some of these problems may be a little harder than the above problems and 10% does not seem much. But it may be fun if you are looking for some challenges. No solution will be posted for these problems. Again, you do not have to work on these problems, which are mainly intended for students who are looking for more challenges.

1. Do Exercise 22.2-8 (p. 602).

2. Do Exercise 22.3-13 (p. 612). Note you need to justify why your algorithm works.

3. Let $s$ and $t$ be two nodes in an undirected graph $G$ such that the distance between them (in terms of number of edges) is strictly greater than $n/2$, where $n$ is the number of nodes in $G$. Show that there is a node $v$ (which is not $s$ or $t$) such that all $s$-$t$ paths contain $v$. 