For general homework policies and our suggestions, please see the policy document.

For this homework, you can assume that addition, subtraction, multiplication and division of two numbers can be done in $O(1)$ time. (Side question: Is this assumption justified?)

1. (40 points) Exercise 3 in Chapter 3: The algorithm described in Section 3.6 (and in class) for computing a topological ordering of a DAG repeatedly finds a node with no incoming edges and deletes it. This will eventually produce a topological ordering, provided that the input graph really is a DAG.

Suppose that we’re given an arbitrary graph that may or may not be a DAG. Extend the topological ordering algorithm so that, given an input directed graph $G$, it outputs one of two things: (a) a topological ordering, thus establishing that $G$ is a DAG; or (b) a cycle in $G$, thus establishing that $G$ is not a DAG. The runtime of your algorithm should be $O(m + n)$ for a directed graph with $n$ nodes and $m$ edges.

Give a modification to the algorithm that computes the correct output: 20 pts

Prove the correctness of your modified algorithm: 10 pts

Prove the runtime of your modified algorithm: 10 pts

2. (45 points) Exercise 5 in Chapter 3: A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less that the number of leaves.

Base case: 15 pts

Induction: 30 pts

3. (15 points) Give an algorithm to compute $x^n$ in $\Theta(\log n)$ time and give a proof of the runtime of your algorithm.

(As always you can assume that a single multiplication takes $O(1)$ time.)

Correct algorithm: 10 pts

Proof of runtime: 5 pts