TheoryPrep **(Due: 7/26/20)**

Assignment #8: Algorithms

Name: Student name(s)

We've taken some problems/solutions from the course texts and CS 121/124/125 material. Some problems are marked as (Challenge) or (Additional Problems). Do the rest first, and if you have time, return to these problems. It's **much more important** to have a rigorous understanding of the core problems and how to prove them than to simply finish all the additional problems.

Problem 1: Algorithms

Learning goal: This problem set is different from previous ones. The goal for this problem set is to help you prepare for 124, and one of the main difficulties in 124 is that the problems are longer. This means you'll need to use your longer proof writing skills (lemmas, breaking down proofs into smaller parts, applying multiple proof techniques in one proof, et cetera).

Thus, Pset 8 has only one problem (taken from 124–but don't let this scare you since you did a couple of 124 problems last week!). 124 problems often follow the structure below

1. Learn about an algorithm in class. See its proof of runtime and correctness (a proof of correctness means it correctly gives the answer to a problem, like yielding a topological sort).
2. You are given a problem that you can solve with the algorithm from class (or a slightly modified version).
3. Give the pseudocode for the algorithm (you can abstract away the function that you learned in class–so on this problem set you could just "get the Topological Sort" or "return the SCC graph").
4. Give some runtime bound $f(n)$ and prove that your code runs in $O(f(n))$.
5. Prove that it correctly solves the problem.

You can follow the above outline for this problem set

(a) In one of your math classes, the professor wanted to show that the following 4 statements are equivalent:

(1) $A \subseteq B$

(2) $A \cap B = A$

(3) $A \cup B = B$

(4) $\bar{B} \subseteq \bar{A}$ (where $\bar{A}$ denotes the complement of A)

Since time was running out in lecture, the professor decided to only show the proofs that (1)⇒(2), and that (3)⇔(4), and she asked you to think about how to finish the equivalence proof when you return home. Seeing as how proving implications is time-consuming, you wanted to prove as few as possible to show equivalence of all statements, and you realized that you could do so by further showing just two more statements: namely (2)⇒(3), and (3)⇒(1). Since you're also taking CS 124, you decided to challenge yourself by trying to develop an algorithm for the following generalized variant:

Given n statements that you want to show are logically equivalent, and given proofs of implications of some subset S of the $n(n-1)$ possible implications, output a minimum number of extra implications one needs to prove that, when combined with those already proven in S, implies equivalence of all n statements. You should give an algorithm (pseudocode/python level is fine, but you should explain how to use the data structures you choose to do the operations you want your program to do), give its runtime (you can do so in terms big-oh of n and $|S|$), and prove that it returns the correct result. Again, the begin{lstlisting} environment may be helpful for writing code.

(b) (Additional Question) Design an efficient algorithm to find the *longest* path in a directed acyclic graph given in adjacency list representation. (Partial credit will be given for a solution where each edge has weight

1; full credit for solutions that handle general real-valued weights on the edges, including *negative* values; you can assume the adjacency list representation is a list of lists of tuples (u, i) where u is the destination of the edge an i is the weight of the edge). You should again give an algorithm (pseudocode/python level is fine but it should be granular enough to use the adjacency list representation), give its runtime (in terms of the number of vertices and edges in the graph), and prove that it returns the correct result.

Problem 2: Review

Learning goal: The goal for this section is to reflect on how you can improve your proof skills!

(a) Select the problem from a previous week that you think you would learn the most from redoing. Copy the problem and your instructor's feedback below, rewrite the proof, and add a short reflection on the changes you made.

(b) (Additional Question) Choose another problem you received feedback on and copy the problem and feedback you received, rewrite the proof, and write a sentence about why you made your changes (so do Part A again) OR solve a problem you left blank from a previous week (try to pick from a week you felt least secure on).

Problem 3: Feedback (Mini Q)

Purpose: Your instructor will have a survey link for you to fill out. This will be really helpful to us if we run this course (or other versions) in the future, so please fill it out authentically!