CS325: Analysis of Algorithms, Fall 2016

Practice Assignment 3*

Due: Tue, 11/22/16

Homework Policy:

1. Students should work on practice assignments individually. Each student submits to TEACH one set of *typeset* solutions, and hands in a printed hard copy in class or slides it under my door before the midnight of the due day. The hard copy will be graded.
2. Practice assignments will be graded on effort alone and will not be returned. Solutions will be posted.
3. The goal of the assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
4. You are allowed to discuss the problems with others, and you are allowed to use other resources, but you *must* cite them. Also, you *must* write everything in your own words, copying verbatim is plagiarism.
5. More items might be added to this list. ☺

Problem 1. Let G = (V, E) be an arbitrary connected graph with weighted edges. Suppose that the weights of edges are distinct. Prove or disprove (a), (b) and (c).

(a) For any partition of the vertices V into two disjoint subsets, the minimum spanning tree of G includes the minimum-weight edge with one endpoint in each subset.

(b) For any cycle in G, the minimum spanning tree of G excludes the maximum weight edge in that cycle.

(c) The minimum spanning tree of G includes the minimum-weight edge in every cycle in G.

Problem 2. A boolean formula is in disjunctive normal form (or DNF) if it consists of a disjunction (Or) of several terms, each of which is the conjunction (And) of one or more literals. For example, the formula

$$(\overline{x} \wedge y \wedge \overline{z}) \vee (y \wedge z) \vee (x \wedge \overline{y} \wedge \overline{z})$$

is in disjunctive normal form. DNF-SAT asks, given a boolean formula in disjunctive normal form, whether that formula is satisfiable.

(a) Describe a polynomial-time algorithm to solve DNF-SAT.

*Some problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on Minimum Spanning Trees and NP-hardness is recommended.

(b) What is the error in the following argument that P=NP?

Suppose we are given a boolean formula in conjunctive normal form with at most three literals per clause, and we want to know if it is satisfiable. We can use the distributive law to construct an equivalent formula in disjunctive normal form. For example,

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) \Leftrightarrow (x \wedge \overline{y}) \vee (y \wedge \overline{x}) \vee (\overline{z} \wedge \overline{x}) \vee (\overline{z} \wedge \overline{y}).$$

Now we can use the algorithm from part (a) to determine, in polynomial time, whether the resulting DNF formula is satisfiable. We have just solved 3SAT *in polynomial time. Since* 3SAT *is NP-hard, we must conclude that P=NP!*

Do *not* submit solutions for the following problems, they are just for practice.

Practice Problem A. Consider a path between two vertices s and t in an undirected weighted graph G. The bottleneck length of this path is the maximum weight of any edge in the path. The bottleneck distance between s and t is the minimum bottleneck length of any path from s to t. (If there are no paths from s to t, the bottleneck distance between s and t is ∞.)

Describe an algorithm to compute the bottleneck distance between every pair of vertices in an arbitrary undirected weighted graph. Assume that no two edges have the same weight.

Practice Problem B. Consider the following problem, called BoxDepth: Given a set of n axis-aligned rectangles in the plane, how big is the largest subset of these rectangles that contain a common point?

(a) Describe a polynomial-time reduction from BoxDepth to MaxClique.

(b) Describe and analyze a polynomial-time algorithm for BoxDepth. [Hint: $O(n^3)$ time should be easy, but $O(n \log n)$ time is possible.]

(c) Why don?t these two results imply that P=NP?