| **CSE521: Algorithms** | October 28, 2022 |
|---|---|
| Homework 2 | |
| *Anup Rao* | Due: November 6, 2022 |

Read the fine print[1]. Each problem is worth 10 points:

1. Give an algorithm to detect whether a given undirected graph has a cycle. If the graph has a cycle, your algorithm should output the cycle. The algorithm should run in time $O(m+n)$, where $m$ is the number of edges and $n$ is the number of vertices. HINT: Argue that if there is a cycle in a particular connected component, then some edge of that cycle must not be a part of the shortest path tree. Show that the breadth first search algorithm can be modified to detect any edge that is not part of the shortest path tree and so find a cycle.

2. You are given two sorted lists of integers of length $m$ and $n$. Give an $O(\log m + \log n)$ time algorithm for computing the $k$'th smallest integer in the union of the lists.

3. Here's a problem that occurs in automatic program analysis. For a set of variables $x_1, \ldots, x_n$, you are given some equality constraints, of the form $x_i = x_j$ and some disequality constraints, of the form $x_i \neq x_j$ . Is it possible to satisfy all of them? For instance, the constraints $x_1 = x_2, x_2 = x_3, x_3 = x_4, x_1 \neq x_4$ cannot be satisfied. Give an efficient algorithm that takes $m$ constraints over $n$ variables, and outputs whether they can be satisfied or not in time $O(m \log n)$. HINT: Use the union-find data structure.

4. You are given a graph $G$ with $n$ vertices and $m$ edges, and a minimum spanning tree $T$ of the graph. Suppose one of the edge weights $w(e)$ of the graph is updated. Give an algorithm that runs in time $O(m)$ to test if $T$ still remains the minimum spanning tree of the graph. You may assume that all edge weights are distinct both before and after the update. HINT: If $e \in T$, consider the cut obtained by deleting $e$ from $T$. If $e \notin T$, consider the cycle formed by adding $e$ to $T$.

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.