Homework 4

Due Wednesday, February 16th at 11:59 pm ET on Gradescope

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and pseudocode (*),
2. a proof of correctness,
3. an analysis of the asymptotic running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class, e.g. correctness of subroutines, running time of subroutines.

You should be as clear and concise as possible in your write-up of solutions.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand.

(*) It is fine if the English description concentrates on the high level ideas and doesn't include all the details. But the reader should not have to figure out your solution solely based on the pseudocode. You can also add comments to your pseudocode, in fact that is best practice. FYI we will share a document on good pseudocode style with you and it will also be discussed in labs and lecture.

Problem 1. *(10 points)*

In this problem we analyze what happens if you manipulate edges in a tree. Let $G(V, E)$ be a tree, that is, a connected, undirected graph that contains no cycles. Consider the following two scenarios.

1. You are given as input G and an edge (u, v) in the tree. You may assume that (u, v) is indeed an edge, you don't have to verify that. Let's delete (u, v) from G. Since it is a tree, the deletion results in the remaining graph being disconnected. Find an $O(n)$ algorithm that identifies the two disconnected components. The inputs are $G(V, E)$ and the vertices , u and v. The expected output is two sets of nodes (one for each component). Provide running time analysis and proof of correctness.

2. Now you are given the tree $G(V, E)$ as input again. (G here is in its original state, with the edge (u, v) in place.) Let x and y be two vertices that are NOT connected by an edge. Now, add the edge (x, y) to E. As we know, adding an edge to a tree creates a cycle. Design an $O(n)$ algorithm that detects cycles in the augmented graph and returns the vertices in the cycle. The input is $G(V, E)$, x and y and the output is the set of vertices. You may assume without verifying that initially x and y don't have an edge between them. Provide running time analysis and proof of correctness.

Problem 2. *(10 points)*

You are an engineer specializing in urban development. There are multiple towns in a certain region that are connected by roads. Some towns have a direct connection between them, but for others you may need to travel through multiple towns. Each road segment is represented by a tuple of the names of the settlements it connects, e.g. (Newton, Needham). The roads are considered bidirectional, the tuple (Newton, Needham) and (Needham, Newton) would represent the same road. (*The word "tuple" is a mathematical expression that refers to a pair of elements. We do not refer to the tuple data object in programming.*)

You are worried about frequent traffic jams. In order to minimize traffic, you decide to build some additional roads in such a way that there are multiple routes between any pair of cities. (It is possible that alternative routes have a different length.) For this you need to identify bottleneck roads. A road (town_1, town_2) is considered a bottleneck if this road is the only way to travel between these two towns.

Design an an algorithm that identifies all such bottleneck roads. The algorithm takes as input the tuples representing roads and returns all pairs of cities with a *direct* connection that is a bottleneck. (Note that you are only asked to indentify bottlenecks, we don't worry about designing the alternative routes.)

These kinds of problems often lend themselves to be represented as a graph problem. If you do so, then the first step in your algorithm is to create the corresponding graph.