CS420/520: Graph Theory with Applications to CS, Winter 2014

Homework 1*

Due: Tue, Feb/4/14

Homework Policy: Each student should submit his/her own set of solutions, independently. You are allowed to discuss the homework with other students, however, you need to indicate their names in your submission. Also, you are allowed to use other sources, but you must cite every source that you use.

Problem 1. Throughout the lecture, we assumed that no two edges in the input graph have equal weights, which implies that the minimum spanning tree is unique. In fact, a weaker condition on the edge weights implies MST uniqueness.

(a) Describe an edge-weighted graph that has a unique minimum spanning tree, even though two edges have equal weights.

(b) Prove that an edge-weighted graph G has a unique minimum spanning tree if the following condition holds:

- For any partition of the vertices of G into two subsets, the minimum-weight edge with one endpoint in each subset is unique.

(c) Describe and analyze an algorithm to determine whether or not a graph has a unique minimum spanning tree.

Problem 2. Consider the following algorithm for finding the smallest element in an unsorted array:

```
RandomMin A[1..n]
  min ← ∞
  for i ← 1 to n in random order do
    if A[i] < min then
      min ← A[i]          (*)
    end if
  end for
  return min
```

[*] Problems 1, 2 and 4 are from Jeff Erickson's lecture notes, and problem 3 is from Uri Zwick's lecture notes.

(a) In the worst case, how many times does RandomMin execute line $(*)$?

(b) What is the probability that line $(*)$ is executed during the nth iteration of the for loop?

(c) What is the *exact* expected number of executions of line $(*)$?

Problem 3. The randomized linear time algorithm of Karger, Klein and Tarjan for finding a minimum spanning tree performs two steps of the Boruvka algorithm and uses sampling steps in which each edge of the graph is chosen, independently, with probability 1/2.

(a) What would be the running time of the algorithm in terms of m, n and k if it performs k steps of the Boruvka algorithm instead, where k is a positive integer?

(b) What would be the expected running time of the algorithm in terms of m, n and p if the sampling probability were changed to p, where $0 \leq p \leq 1$?

Problem 4.

(a) Describe and analyze a modification of Shimbel's shortest-path algorithm that actually returns a negative cycle if any such cycle is reachable from s, or a shortest-path tree if there is no such cycle. The modified algorithm should still run in $O(mn)$ time.

(b) Describe and analyze a modification of Shimbel's shortest-path algorithm that computes the correct shortest path distances from s to every other vertex of the input graph, even if the graph contains negative cycles. Specifically, if any walk from s to v contains a negative cycle, your algorithm should end with $dist(v) = -\infty$; otherwise, $dist(v)$ should contain the length of the shortest path from s to v. The modified algorithm should still run in $O(mn)$ time.