

# CS420/520: Graph Theory with Applications to CS, Winter 2015

## Homework 1\*

Due: Tue, Feb/10/15

**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.** It is known that if no two edges in a graph have equal weights then the graph has a unique minimum spanning tree. 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.**

- (a) Describe and analyze a modification of Bellman-Ford'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 Bellman-Ford'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.

---

\*Problems 1 and 2 are from Jeff Erickson's lecture notes, and problem 5-(a) is from Uri Zwick's lecture notes.

**Problem 3.** Let  $G = (V, E)$  be a directed graph, and let  $\ell : E \rightarrow \mathbb{R}$  be a length function. The length of the edges may be negative, but there is not negative cycle in the graph. Use potential functions to describe (and analyze) an  $O(mn + n^2 \log n)$  time algorithm that computes the shortest paths between all pairs of vertices of  $G$ . (Hint: the high level idea is mentioned in class).

**Problem 4.** Let  $G = (V, E)$  be a directed acyclic graph (DAG), and let  $\ell : E \rightarrow \mathbb{R}$  be a length function. Design and analyze an algorithm to compute the longest path between a given pair of vertices  $s, t \in V$ .

**Problem 5.** Let  $G = (V, E)$  be an undirected graph, and let  $\Delta$  be the maximum vertex degree in  $G$ . Suppose that  $\Delta$  is a constant.

- (a) Design and analyze a  $O(n)$  time  $\frac{1}{2}$ -approximation algorithm for the maximum matching problem in  $G$ ; an algorithm that finds a matching with cardinality at least  $1/2$  of the cardinality of the maximum matching.
- (b) (**Extra credit**) Now, let  $\alpha$  be a positive integer constant. Design (and analyze) a  $O(n)$  time  $\frac{\alpha}{\alpha+1}$ -approximation algorithm for the maximum matching problem in  $G$ . Describe the running time of your algorithm as a function of  $\alpha$ ,  $\Delta$  and  $n$ .