Homework #4
Introduction to Algorithms/Algorithms 1
600.363/463

Due on: Friday, April 13, 11:59 a.m. (NOON)

Where to submit: On blackboard, under student assessment
Late submissions: will NOT be accepted

Format: Please start each problem (1 through 18) on a new page.
Please type your answers; handwritten assignments will not be accepted.

March 30, 2012

1

Show that if a graph's edges all have distinct weights, the MST is unique.

2

Give an algorithm for a maximum spanning tree.

3

Develop a version of Prim's algorithm or of Kruskal's algorithm for the minimum spanning forest for graphs that are not necessarily connected.
Hint: Use the strongly connected components algorithm.

4

Suppose you implement the disjoint-set data structure using union-by-rank but without path compression. Give a sequence of m union and find operations whose running time is $\Omega(m \lg(n))$.

5

Let T be a MST of a graph G. Given a connected subgraph H of G, show that $T \cap H$ is contained in some MST of H.

6

For each of the following statements, either prove its correctness or give a counterexample:

1. If a graph $G = (V, E)$ has more than $|V| - 1$ edges and there is a unique heaviest edge, then G cannot be part of a MST.
2. If G has a cycle with a unique heaviest edge e, then e cannot be a part of the MST.
3. If the lightest edge e is unique, then e must be a part of every MST.

7

State whether the following statement is true or false, providing either a proof or a counterexample:

Adding a constant weight to the weight of every edge does not change the outcome for the single source shortest-path algorithm.

8

Is the path between two nodes in a MST necessarily the shortest path between them? Prove or give a counterexample.

9

Does either of Prim or Kruskal's algorithm work if we allow negative weights?

10

Modify Dijkstra's algorithm by using a *max-priority queue* and extracting *Max* every time. Will it produce paths with a maximum weight (single-source longest paths)?

11

Design a linear-time algorithm that takes a connected undirected graph and checks whether there is an edge that can be removed while still leaving the graph connected.

12

Prove that if an undirected graph has k components then it has at least $|V| - k$ edges.

13

Suppose that for a graph $G = (V, E)$, every vertex v has a weight c in addition to the edge weights, where

$$\mathrm{c}(v) \geq 0 \quad \text{for} \quad v \in V$$

The weight of a path is now defined as the sum of all the weights (both edges and vertices, including the ends). Design an algorithm to solve this modified version of the single-source shortest paths problem.

14

Given a directed weighted graph G and a tree T with root s, check in a time $\mathrm{O}(V + E)$ that T is a single-source shortest path tree with source s.

15

Consider a directed graph whose only negative edges are those that leave s; will Dijkstra's algorithm work when started at s?

16

Let d be a degree of a vertex v. (Please refer to CLRS for a definition). Prove that:

1. $$\sum_{a \in V} d(a) = 2\,|E|$$
2. the number of vertices with odd degree is even.

17

Prove that in any connected undirected graph, there exists a vertex whose removal leaves the graph connected.

18 Bonus question

Given a DAG, check if there is a path that touches every vertex exactly once.