Algorithms I

Tutorial [redacted] 7

October 28, 2016

Problem 1

Give an $O(|V| + |E|)$ time algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two nodes s and t, and returns the number of paths from s to t in G.

Problem 2

Prove or disprove: BFS and DFS algorithms on a undirected, connected graph $G = (V, E)$, produce the same tree if and only if G is a tree.

Problem 3

You are given an undirected graph where each edge has cost 1. A path p between u and v is a shortest path if the sum of cost of the edges on p is minimum among all the paths between u and v. Given two vertices u and v of the graph you need to find the cost of shortest path between u and v. Can this idea be generalised if all the edges have same positive cost c.

Problem 4

The *eccentricity* of a vertex v in a graph is the maximum distance from v to any other vertex in the graph. The *center* of a graph is the set of all vertices with minimum eccentricity. Your task is to find the center of a tree.

Problem 5

Consider an undirected graph $G = (V, E)$ with nonnegative edge weights $w_e \geq 0$. Suppose that you have computed a minimum spanning tree of G. Now suppose each edge weight is increased by 1: the new weights are $w_e = w_e + 1$. Does the cost of minimum spanning tree change? Give an example where it changes or prove it cannot change.

Problem 6

You are given a graph $G = (V, E)$ with positive edge weights, and a minimum spanning tree $T = (V, E')$ with respect to these weights; you may assume G and T are given as adjacency lists. Now suppose the weight of a particular edge $e \in E$ is modified from $w(e)$ to a new value $w'(e)$. You wish to quickly update the minimum spanning tree T to reflect this change, without recomputing the entire tree from scratch. There are four cases. In each case give a linear-time algorithm for updating the tree.

1. $e \notin E'$ and $w'(e) > w(e)$.
2. $e \notin E'$ and $w'(e) < w(e)$.
3. $e \in E'$ and $w'(e) < w(e)$.
4. $e \in E'$ and $w'(e) > w(e)$.