Design and Analysis of Algorithms **out:** January 17, 2017

Problem Set 0

Lecturer: Daniel A. Spielman **due:** Never

This is just to test your background

I wrote these problems to help you decide if you know enough discrete math to take this course. You should be able to solve these problems after a little thought. These problems are asking you to prove things that should be "intuitively obvious", but which take some work to make formal. If you find them confusing, I suggest discussing them with the course staff.

I plan to distribute solutions to these on Thursday.

Problem 1: Defining a Tree

Let $G = (V, E)$ be a graph. A *simple path* in G is a sequence of vertices $v_0, \ldots, v_k$ in which each vertex occurs at most once and such that each consecutive pair of vertices, (v_i, v_{i+1}), is in E. We often describe this as a path between v_0 and v_k or from v_0 to v_k. The qualifier "simple" means that no vertex appears twice.

A *simple cycle* is the same as a simple path, except that $v_0 = v_k$.

A graph T with n vertices is called a tree if it satisfies any two of the following three properties:

a. T has no simple cycles.

b. T has $n - 1$ edges.

c. T is connected (that is, there is a simple path between every pair of vertices).

Prove that each two of these properties implies the third.

Problem 2: Re-defining a Tree

Here's another definition of a tree: a tree is a connected graph such that for every two vertices x and y, there is exactly one simple path in the graph from x to y. Prove that this definition is equivalent to the definition from Problem 1.

Problem 3: Another property of trees.

Let T be a tree. Prove that T contains a v vertex of degree 1. Let (u, v) be the unique edge involving v. Prove that if we remove v and edge (u, v) from T, then the graph that remains is also a tree.