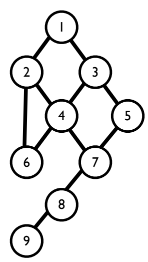
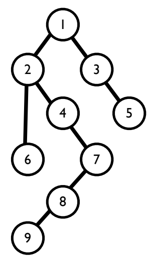
IE 512 Graphs, Networks, and Algorithms

Homework 0

There are two problems in this homework. Homework 0 will not be graded.



connected undirected graph



tree

Problem 0.1 An **undirected graph** G is a collection of nodes where some pairs of nodes are connected by edges. An edge connects two nodes, and the pair of nodes connected by an edge is called **adjacent** nodes.

- A **walk** is a sequence of nodes $(v_1, v_2, \ldots, v_k)$ such that every consecutive nodes are adjacent, that is connected by an edge, e.g. (1,3,5,7,4,3).
- A **path** is a walk where no node is repeated more than once, e.g. (1,3,5,7,4).
- Two nodes i and j are connected if G contains a path from i to j.
- An undirected graph is **connected** if and only if all pairs of nodes are connected.
- A **cycle** is a walk that ends where it started, e.g. (1,3,5,7,4,2,1).

A **tree** is an undirected graph of n nodes that is *i*) connected and *ii*) has no cycle. Show using a mathematical induction that the number of edges in a tree with n nodes is $n - 1$.

Problem 0.2 Given a sorted array of n real numbers, we want to find the relative position of a input real-valued key v in the sorted array. The "binary search" algorithm proceeds as follows.

- compare the input key with the value in the middle
- if the key is smaller, apply binary search recursively to the smaller half of the array
- if the key is larger, apply to the larger half
- repeat until the array has only one number

Show that, in the worst case, the total number of comparisons, $T(n)$, is given by a recurrence

$$T(n) \;\; = \;\; 1 + T(\lceil (n-1)/2 \rceil) \; .$$

Provide an upper bound on the total number of comparisons $T(n)$ as a function of n.