CS140: Algorithms
Fall 1999
Assignment 4, 10/8/99
Due Fri, 10/15/99

This assignment is due at the start of class on Fri., 10/15/99. No late homework will be accepted. Your solutions will be graded for clarity, conciseness, and correctness.

1. The tower of Hanoi puzzle is a standard example of a nontrivial problem that has a simple recursive solution. There are n disks of distinct sizes. Each disk has a hole in the center so the disks can be stacked on a vertical peg. Three vertical pegs are provided. The disks are initially stacked on peg 1. They are ordered by size with the smallest disk on the top.

   The objective is the move the stack of disks from peg 1 to peg 2. But arbitrary moves are not allowed. Rather, in any single step the disk at the top of one of the stacks can be moved to the top of another stack. Moves must satisfy the additional constraint that a disk cannot be placed on top of a smaller disk. So while you may have up to three stacks at any one time, the disks of each stack must be ordered by increasing size from top to bottom.

   (a) Design a recursive algorithm to move the stack from peg 1 to peg 2.

   (b) Write a recurrence relation for the number of moves made by your algorithm. Solve the recurrence relation.

   (c) Is your algorithm optimal in that it uses the minimum number of moves?

2. A *vertex cover* of a graph $G = (V, E)$ is a subset of vertices W such that every edge of the graph is incident to at least one vertex of W. V is a vertex cover of G but, typically, we are interested in finding small vertex covers of G. The problem of finding the smallest vertex cover of a graph is NP-hard. (You don't have to know what that means now.) But the problem is much easier when the input is restricted to trees or, more generally, forests. Give a recursive algorithm that finds a smallest vertex cover of an input forest F.