Many Campers Sort Piles Instructor: Padraic Bartlett

Homework 2: Many Campers Sort Piles

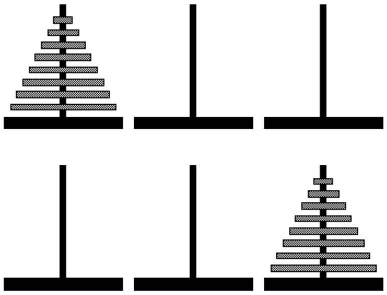
Week 4 *Mathcamp 2012*

Attempt all of the problems that seem interesting, and let me know if you see any typos! (+) problems are harder than the others. (++) problems are currently open.

1. Take the following lists, and apply quicksort and mergesort to put them in the right order:
   - $(8, 7, 6, 5, 4, 3, 2, 1)$.
   - $(3, 1, 4, 1, 5, 9, 2)$.

2. The Towers of Hanoi is the following puzzle: Start with 3 rods. On one rod, place n disks with radii $1, 2, \ldots n$, so that the disk with radius n is on the bottom, the disk with radius $n-1$ is on top of that disk, and so on/so forth.

   The goal of this puzzle is to move all of the disks from one rod to another rod, obeying the following rules:
   - You can move only one disk at a time.
   - Each move consists of taking the top disk off of some rod and placing it on another rod.
   - You cannot place a disk A on top of any disk B with radius smaller than A.



Find a recursive algorithm for solving this puzzle! How long does it take to complete your solution? Suppose that you can perform a move once every second, and you can perform moves until the heat death of the universe (10^{100} years, say.) What is the largest puzzle you can solve?

3. Consider the following algorithm (Stoogesort[1]!) for sorting a list: Take as input a list $L = (l_1, \ldots l_n)$.

   - If your list contains one or two elements, sort it by just looking at the list.
   - Otherwise, the list contains ≥ 3 elements. Let $M = \lceil 2/3 \rceil$.
   - Stoogesort the list $(l_1, \ldots l_m$.
   - Stoogesort the list $(l_{n-m}, \ldots l_n)$.
   - Stoogesort the list $(l_1, \ldots l_m$.

   Prove that this algorithm sorts any list.

[1] Named after the comedy routines of the Three Stooges; specifically, the ones where each stooge hits the other two.