

Problem Set 2 (due Wednesday, February 2)

(**Reminder:** Problem 5 of Problem Set 1 is due with this problem set.)

1. (10 points) Towers of Hanoi

The Towers of Hanoi problem is a classic example of recursion. In the Towers of Hanoi problem, n disks of different sizes are piled on a peg in order of their size, with the largest at the bottom. There are two empty pegs. The problem is to move all the disks to the third peg by moving only one disk at a time and never placing a disk on top of a smaller one. The second peg may be used for intermediate moves.

The usual solution recursively moves all but the last disk from the starting peg to the spare peg, then moves the remaining disk (the largest one) from the start peg to the destination peg, and then recursively moves all the others from the spare peg to the destination peg. This recursive solution is described in the following recursive procedure.

HANOI(n , $start$, $destination$, $spare$)

1. if $n > 0$
2. HANOI($n - 1$, $start$, $spare$, $destination$)
3. **print** "move top disk from peg ", $start$, "to peg ", $destination$
4. HANOI($n - 1$, $spare$, $destination$, $start$)

Write a recurrence for the number of moves for n disks. Solve this recurrence.

2. (20 points) Recurrences Find the asymptotic solution (using the Θ -notation) for each of the following recurrences. Assume that $T(n)$ is constant for n sufficiently small.

(a) $T(n) = T(n/2) + 2n^2$.

(b) $T(n) = 81T(n/3) + n^4/3$.

(c) $T(n) = T(\sqrt{n}) + 1$.

(d) $T(n) = T(n/5) + T(n/3) + 2n$.

3. (10 points) FIFO queues and stacks

Exercise 6.5-6, page 142.

4. (10 points) Building a heap using insertion

Problem 6-1, page 142.

5. (10 bonus points) A fault-tolerant OR-gate

Assume we are given an infinite supply of two-input, one-output gates, most of which are OR gates and some of which are AND gates. Unfortunately the OR and AND gates have been mixed together and we can't tell them apart. For a given integer $k \geq 0$, we would like to construct a two-input, one-output combinational " k -OR" circuit from our supply of two-input, one output gates such that the following property holds: If at most k of the gates are AND gates then the circuit correctly implements OR. Assume for simplicity that k is a power of two.

For a given integer $k \geq 0$, we would like to design a k -OR circuit that uses the smallest number of gates. Design the best possible circuit you can and derive a Θ -bound (in terms of the parameter k) for the number of gates in your k -OR circuit.