Problem Set 1 (due Wednesday, September 22)

1. (10 points) Visibility

A fundamental problem in computer graphics is to find visible surfaces of a collection of objects. Consider the following simplified problem in two dimensions. We have a set of n lines $y = a_i x + b_i$, $1 \leq i \leq n$, where a_i and b_i are integers for $1 \leq i \leq n$. Suppose you are walking along the line $y = \infty$ and looking "down" at this set of lines. Line i is *visible* in an interval I of reals if $a_i x + b_i > a_j x + b_j$ for $x \in I$, $j \neq i$, $1 \leq j \leq n$. (Informally, line i is visible in I if line i above all of the other lines in interval I of the x-axis.)

Give an $O(n \log n)$ time deterministic algorithm, that takes as input n lines (i.e., the (a, b) pairs) and determines for each line, all of its visible intervals. Point out how this information yields the curve for the visible surface of the given set of lines.

2. (10 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.

3. (3 + 3 + 4 = 10 points) Matrix Multiplication

The problem of squaring an $n \times n$ matrix is clearly a special case of multiplying two arbitrary $n \times n$ matrices. It turns out, in fact, that the complexity of both of these problems in terms of the number of arithmetic operations is identical, ignoring multiplicative constants.

(a) Prove the above fact. (*Hint:* Show how the product BC can be extracted from the square of a suitably defined larger matrix.)

We next consider whether multiplying two matrices A and B is harder than checking, given A, B, and C, whether $AB = C$.

(b) Let v be an n-dimensional vector whose entries are independently and randomly chosen to be 0 or 1 (each with probability 1/2). Prove that if M is a non-zero $n \times n$ matrix, then $\Pr[Mv = 0] \leq 1/2$.

(c) Give a randomized $O(n^2)$ algorithm that, given three $n \times n$ matrices A, B, and C, returns the correct answer with probability at least 0.99. What kind of error may your algorithm make?

4. (10 points) Lower bound for multi-selection

Let the *multi-selection problem* be defined as follows. The input is a set S of n distinct keys drawn from some totally ordered universe, and a set of $k+1$ integers r_i such that $1 = r_0 < r_1 < \cdots < r_k = n+1$. The output is a partition of S into k sets $S_0, \ldots, S_{k-1}$ such that S_i is the set of all keys with ranks greater than or equal to r_i and strictly less than r_{i+1}.

Show that in the comparison based model, every algorithm incurs $\Omega(n \log k)$ comparisons on its worst-case instance. (*Hint:* Use the approach used for placing a lower bound on the number of comparisons needed for sorting. See Section 8.1 of text.)

Note that this is a tight bound. You are encouraged to devise an $O(n \log k)$-comparison divide and conquer algorithm for the problem (you do not need to submit this).

5. (10 points) Fault-prone networks

Designers of communication networks prefer graphs that are well-connected and fault-tolerant in the sense that there are multiple (hopefully, disjoint) paths between any two pairs of vertices. In this problem, we consider a much-ignored class of directed graphs, in which there is at most one simple path from any vertex to any other vertex. Using depth-first search, design an $O(mn)$ time algorithm to determine whether a given directed graph with n vertices and m edges satisfies the preceding property.