College of Computer & Information Science — Fall 2014
Northeastern University — Handout 1
CS7800: Advanced Algorithms — 3 September 2014

Problem Set 1 (due Monday, September 15)

Instructions:

- The assignment is due at the beginning of class on the due date specified. Late assignments will not be accepted.
- We encourage you to attempt and work out all of the problems on your own. You are permitted to study with friends and discuss the problems; however, *you must write up you own solutions, in your own words.*
- If you do collaborate with any of the other students on any problem, please list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is prohibited.
- We require that all homework submissions be neat, organized, and *typeset.* You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions.

1. (10 points) Fault-tolerant gates

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.

2. (10 points) Selection of multiple keys

In class, we studied the problem of selection of an element of a given rank from an unsorted list. In this problem, we consider the problem of simultaneously selecting keys corresponding to multiple (say k) ranks from an unsorted list of n elements. It is not hard to see that one can solve the above problem using the single selection algorithm in $O(nk)$ time. Can we do better?

Formally, the input to the problem 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}.

(a) Give an $O(n \log k)$-comparison divide and conquer algorithm for the problem.

(b) 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.)

3. (10 points) Selection from two databases

You are interested in selecting elements of a given rank from the union of two separate databases. One database has m numbers and the other has n numbers, and you may assume that all the values are different. The only access you have to each database is through a query where you specify a rank k, and the chosen database returns the kth smallest element that it contains.

Give an algorithm that finds an element of a given rank k from the union of the two databases, while making $O(\log(\min\{m, n\}))$ queries.

4. (10 points) Finding a local maximum in a grid

You are given an $n \times n$ grid, each cell of which contains a distinct integer. Each cell can be identified by the pair (r, c), $1 \leq r, c \leq n$, where r gives the row number and c gives the column number. Your access to grid is through a *probe*, by which you determine the integer stored in a given cell.

We say that (i, j) is a *neighbor* of a given cell (r, c) if and only if $|i - r| + |j - c| = 1$. We say that a cell (r, c) is a *local maximum* if the integer at (r, c) is greater than the integers stored at each of its neighboring cells.

Give an algorithm that determines a local maximum in a grid using $O(n)$ probes.