Purdue University | Hambrusch
CS 580 | Spring 2010

Assignment 3

Due: Thursday, February 11, 2010 (before class)

Midterm Exam: Wednesday, March 10, 8-10pm, Lawson B155

1) (15 pts.)
(i) Describe and analyze an algorithm that determines the smallest and second smallest element in a set containing n elements. Your algorithm should perform $n + \lceil \log n \rceil - 2$ comparisons.
(ii) Array A contains n elements in arbitrary order, $n = 2^k$. Describe an efficient algorithm that reports the $2k + 1$ elements of rank $n/2 - k, n/2 - k + 1, \ldots, n/2, n/2 + 1, \ldots, n/2 + k - 1, n/2 + k$.

2) (20 pts.) A research project needs a data structure D that supports the operations min, extract-min, max, extract-max, and insert(x). You know n, the estimated maximum number of elements in D, but you do not know anything about the values of the elements. The implementation can use only one array A of size $n + 1$, plus a constant number of additional variables. One array location should contain one data element. A student (who has taken 580 last year) proposes the following heap-like structure D:

- The root of the tree contains no element.
- The left subtree is a heap with min value at the root (min-heap), and the right subtree is a heap with max value at the root (max-heap).
- Each element of the min-heap is less than or equal to the element in the same position in the max-heap.
- If there are k, $k \leq n$, elements in D, they are in array locations 2, ..., $k + 1$.
- If an element in the min-heap does not have a corresponding element in the max-heap, then the parent of the missing element is larger than the element in the min-heap. This can happen only for leaves in the min-heap.

(i) How fast can the minimum and maximum element be reported? How is the index of the "corresponding element" determined? Give a time bound.

(ii) The student claims that operations extract-min, extract-max and insert(x) can each be implemented in $O(\log n)$ time. Either describe how each can be implemented, analyze the time complexity, and address the correctness, or prove that the structure does not support the operations as claimed.

3) (15 pts.) You are given n intervals, where the i-th interval is represented by the pair (l_i, r_i), $1 \leq i \leq n$, representing the two endpoints on the real line.

A subset T of intervals is called a **covering set** if the intervals in T cover the n given intervals; that is, any real value that is contained in some interval is also contained in some interval in T. The size of a covering set T is the number of intervals in T. Describe and analyze an efficient algorithm to compute a covering set of minimum size.