ESO207A/211: Data Structures and Algorithms

Homework *1* **Due: September 3 2014**

Instructions.

1. Start each problem on a new sheet. Write your name, Roll No., the course number, the problem number, the date and the names of any students with whom you collaborated.
2. For questions in which algorithms are asked for, your answer should take the form of a short write-up. The first paragraph should summarize the problem you are solving and what your results are (time/space complexity as appropriate). The body of the essay should provide the following:
   (a) A description of the algorithm in English and/or pseudo-code, where, helpful.
   (b) At least one worked example or diagram to show more precisely how your algorithm works.
   (c) A proof/argument of the correctness of the algorithm.
   (d) An analysis of the running time of the algorithm.

   Remember, your goal is to communicate. *Full marks will be given only to correct solutions which are described clearly.* Convoluted and unclear descriptions will receive *low marks.*

Problem 1. Unimodal Search (Divide-and-Conquer). An array $A[1 \dots n]$ of numbers is unimodal if there exists some index $k \in \{1, 2, \dots, n\}$ such that $A[1] \le A[2] \le \dots \le A[k]$ and $A[k] \ge A[k+1] \ge \dots A[n]$. The element $A[k]$ is the maximum element of the array and it is the unique "locally maximum" element surrounded by elements $A[k-1]$ and $A[k+1]$, that are both not larger than itself. An example unimodal array: $A = \boxed{1\ 4\ 5\ 11\ 8\ 7\ 1}$. Here, $A[4] = 11$ is the maximum element.

Give an algorithm that given a unimodal array $A[1 \dots n]$, finds the maximum element in $O(\log(n))$ time. (a) Prove the correctness of your algorithm, and (b) prove the bound on its running time.

Problem 2. Young Tableau (Heaps). An $m \times n$ tableau is an $m \times n$ matrix such that the entries of each row are in sorted order from left to right and the entries of each column are in sorted order from top to bottom. Some of the entries of a Young tableau may be ∞ which are treated as non-existent elements.

a. Draw a 4×4 Young tableau containing the elements $\{9, 16, 3, 2, 4, 8, 5, 14, 12\}$.

b. Show that an $m \times n$ Young tableau is empty if $Y[1, 1] = \infty$. Argue that Y is full (contains mn elements) if $Y[m, n] < \infty$.

c. Give an algorithm to implement EXTRACT-MIN on a non-empty $m \times n$ Young tableau that runs in $O(m + n)$ time. Your algorithm should use a recursive subroutine that solves an $m \times n$ problem by recursively solving either an $(m-1) \times n$ or an $m \times (n-1)$ problem.

d. Show how to insert a new element into a non-full $m \times n$ Young tableau in $O(m + n)$ time.

e. Give an $O(m + n)$ time algorithm to determine whether a given number is stored in a given $m \times n$ Young tableau.