ICS621 Homework 2: Heaps

Choose one of the following.

Problem 6-2 from CLRS: Analysis of d-ary heaps. A **d-ary heap** is like a binary heap, but (with one possible exception) non-leaf nodes have d children instead of 2 children.

a) How would you represent a d-ary heap in an array?

b) What is the height of a d-ary heap of n elements in terms of n and d ?

c) Give an efficient implementation of EXTRACT-MAX in a d-ary max-heap. Analyze its running time in terms of d and n.

d) Give an efficient implementation of INSERT in a d-ary max-heap. Analyze its running time in terms of d and n.

e) Give an efficient implementation of INCREASE-KEY(A, i, k), which flags and error if $k < A[i]$, but otherwise sets $A[i] = k$ and then updates the d-ary max-heap structure appropriately. Analyze its running time in terms of d and n.

Problem 6-3 from CLRS: Young tableaus. An $m \times n$ **Young 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 we treat as nonexistent elements. Thus, a Young tableau can be used to hold $r \leq mn$ finite numbers.

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

b) Argue that an $m \times n$ Young tableau Y 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 nonempty $m \times n$ 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 $m \times (n-1)$ subproblem. (*Hint*: Think about MAX-HEAPIFY). Define $T(p)$, where $p = m + n$, to be the maximum running time of EXTRACT-MIN on any $m \times n$ Young tableau. Give and solve a recurrence for $T(p)$ that yields the $O(m + n)$ time bound.

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

e) Using no other sorting method as a subroutine, show how to use an $n \times n$ Young tableau to sort n^2 numbers in $O(n^3)$ time.

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