Assignment 3 CSC263
Due: Nov 6, 2008

1. Suppose we are using a priority queue Q to schedule jobs on a CPU. Job requests together with their priorities are inserted into the queue. Whenever the CPU is free, the next job to execute is found by extracting the highest priority job in Q. Using a heap implementation for Q the scheduler can both extract the next job from Q and add a new job to Q in time $O(\log n)$ where n is the number of pending jobs in the queue.

   However, in practice, we want our scheduler to be able to do more than just insert jobs and extract the highest priority job. In particular, we want our scheduler to be able to remove a job x from the job queue (DELETE(x)) as well as change the priority of a job x to some new value k (CHANGE-PRIORITY(x, k)).

   Show how to add the operations DELETE(x) and CHANGE-PRIORITY(x, k) to the heap data structure so that (a) they both run in time $O(\log n)$ and (b) the resulting data structure after executing either these operations is still a heap. For both operations, you may assume that parameter x gives you the index corresponding to job x in the array representation of the heap.

2. Prove that any heap on n nodes has at most $\lceil n/2^{h+1} \rceil$ nodes of height h.

3. A *pennant* of height h is a tree consisting of 2^h nodes: a root with exactly one child, which is the root of a complete binary tree of the remaining $2^h - 1$ nodes. It also satisfies a priority condition: the element at any node is the maximum of all the elements in the subtree rooted at that node. In particular, the tree rooted at a pennant's root's child is a (max-)heap. Note that a pennant of height 0 consists of a single node.

   A *pennant forest* is a sequence of pennants $P_0, P_1, \ldots, P_m$ satisfying the following five properties:

   - height(P_{i-1}) $\leq$ height(P_i), for $0 < i \leq m$.
   - There are at most 2 pennants of any height.
   - There are at least $i + 1$ pennants of height at most i, for $0 \leq i \leq m$.
   - There are at most $i + 2$ pennants of height at most i, for $0 \leq i \leq m$.
   - The element at the root of $P_{i-1} \leq$ the element at the root of P_i, for $0 < i \leq m$.

   (a) Prove that if $P_0, P_1, \ldots, P_m$ is a pennant forest, then it contains at least 2^m nodes and fewer than 2^{m+1} nodes. **Hint:** Use induction.

   (b) Give an efficient algorithm that given a tree of height h satisfying the heap conditions, constructs a pennant forest $P_0, \ldots, P_h$ with the same nodes. **Hint:** Use recursion.

   (c) Give an efficient algorithm that, given a pennant forest $P_0, \ldots, P_m$, constructs a tree of height m with the same nodes satisfying the heap conditions.

   (d) Give an efficient algorithm for splitting a pennant of height h into two pennants of height $h - 1$.

   (e) Give an efficient algorithm for merging two pennants of height $h - 1$ into a pennant of height h.

   (f) Give an algorithm that takes a sequence $P_0, \ldots, P_m$ of pennants that satisfies the first four properties of a pennant forest and constructs a pennant forest with the same nodes. Your algorithm should run in $O(m^2)$ time. **Hint:** Use recursion.

   Briefly explain why your algorithms are correct, and state (with justifications) your algorithms' (worst-case) run-times.