CS1102: Data Structure and Algorithms

Tutorial 9 Heaps
Week of 29 March 2010

Question 1: Heap Operations

a. Create a max-heap by inserting the following integers into an initially empty heap (one by one):
   6, 2, 4, 1, 8, 5, 3, 7, 9
   Draw some diagrams to show the evolution of this heap.

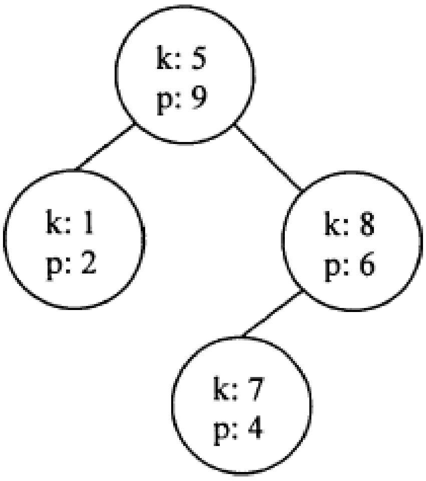
b. Next, assume array {6, 2, 4, 1, 8, 5, 3, 7, 9} represents a complete binary tree, draw the tree and some other diagrams to show the "heapification" of this tree into a max-heap.

c. What are the differences between the above two approaches in constructing a heap from a given set key values?

d. Next, perform two deleteMax() operations from the "heapify heap". Show the final results.
   Note that performing deleteMax() n times is simply the heap sort algorithm.

Question 2: Treap

A **treap** is a **binary tree** where each node has a **key** and a **priority**. The **keys** follow a **binary search tree property** and the **priorities** follow the **heap property**. See the example below



a. Is a treap a heap?

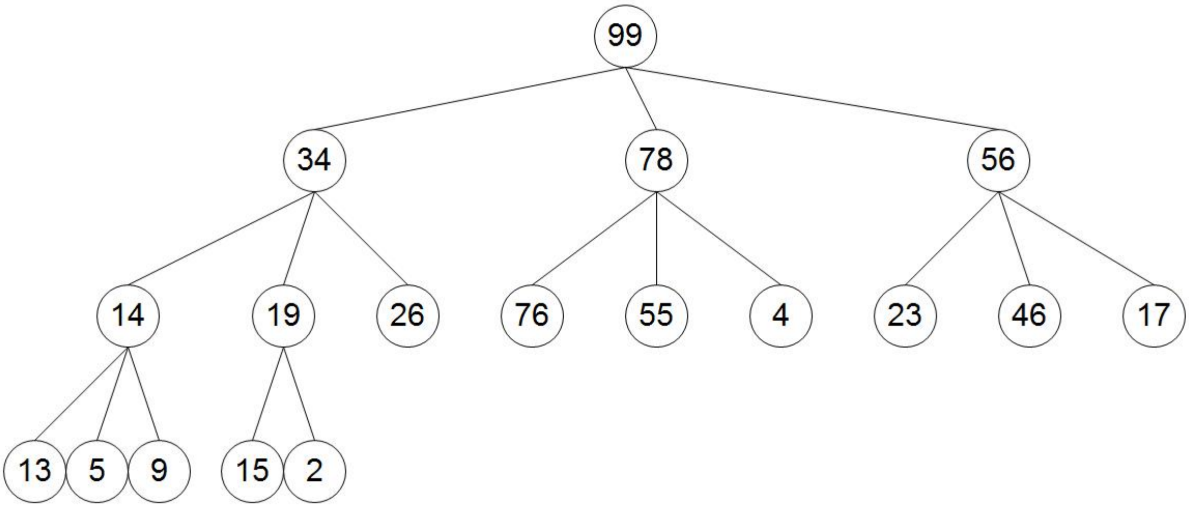
b. Is a treap a binary search tree?

c. You are given a set of n pairs (k = keys, p = priority) and you are asked to find a corresponding treap. For instance, if n = 4 and the pairs are (5, 9), (7, 4) , (8, 6), (1, 2) then a correct answer is given in the picture above. How many such treaps are possible in general? Justify your answer briefly. Assume that there is no duplicate key or priority. Namely, you won't see any input containing (1,2) and (1,3); neither will you see any input containing (2,1) and (3,1).

d. Design an algorithm (no programming required) to transform a treap from an array representation to a tree structure. You can use the following treap as an instance:
The treap contains 6 nodes with labels (k = 2, p = 5), (k = 5, p = 2), (k = 3, p = 1), (k = 4, p = 7), (k = 9, p = 4), (k = 8, p = 3).

Question 3: D-ary Heap

A d-ary heap is like a binary heap, with all but the last non-leaf node having d children instead of only 2 children. A ternary (d=3) max-heap is shown below to give you an idea.



a. How would you represent a d-ary heap in an array? Given a child node with index childId, how to locate the index of its parent? Given a parent node, how to locate the indices of its children?
b. What is the height of a d-ary heap of n elements in terms of n and d?
c. Give an implementation of a d-ary max-heap class. We mainly focus on the method add() and removeMax(). Analyze the running time of these two methods in terms of n and d.
d. Analyze the running time of heap sort using your d-ary heap in terms of n and d.