Priority Queues (Heaps)

Q1

a. Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2, one at a time, into an initially empty binary heap.

[1]

Q2

Show the result of performing three deleteMin operations in the heap of the previous exercise.

[1]

Q3

A complete binary tree of N elements uses array positions 1 to N. Suppose we try to use an array representation of a binary tree that is not complete. Determine how large the array must be for the following:

a. a binary tree that has two extra levels (that is, it is very slightly unbalanced)

b. a binary tree that has a deepest node at depth 2 log N

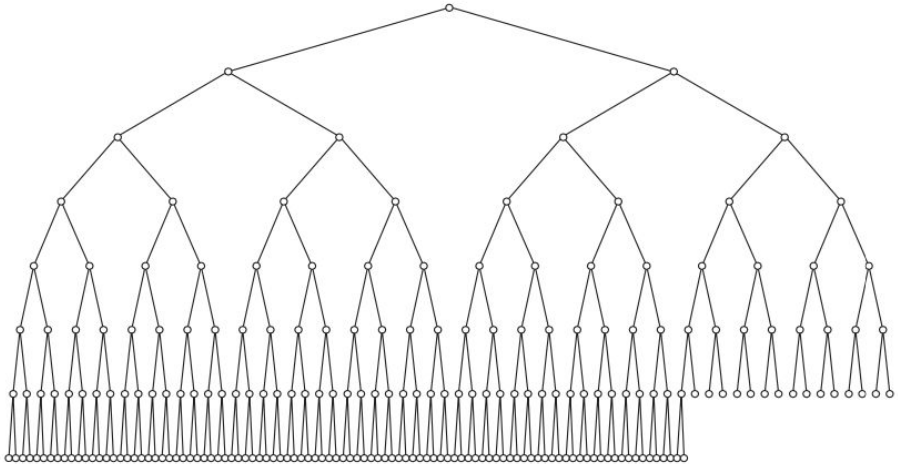
c. a binary tree that has a deepest node at depth 4.1 log N

d. the worst-case binary tree

[1]

Q4

How many nodes are in the large heap in Figure?



[1]

Q5

Prove that for the perfect binary tree of height h containing $2^{h+1} - 1$ nodes, the sum of the heights of the nodes is $2^{h+1} - 1 - (h+1)$.

Q6

Each deleteMin operation uses 2 log N comparisons in the worst case.

a. Propose a scheme so that the deleteMin operation uses only

$$\log N + \log \log N + O(1)$$

comparisons between elements.

This need not imply less data movement.

[1]

References

[1]Weiss - Data Structures and Algorithm Analysis in Java 3rd Edition