Heap

Heaps

- A heap (min-heap) :
  - Binary tree
  - Almost completely filled
    - All nodes are filled in, except the last level
    - may have some nodes missing toward the right
  - All nodes fulfill the heap property
    - The value of any node is less than or equal to the values of its descendants.
- The value of the root is the minimum of all all the values in the tree.

Heaps

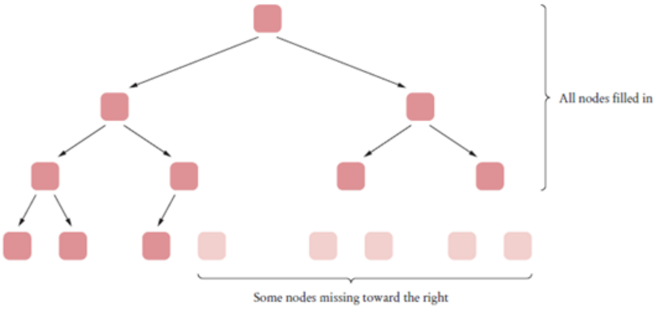


Figure 26 An Almost Completely Filled Tree

In an almost complete tree, all layers but one are completely filled.

Heaps

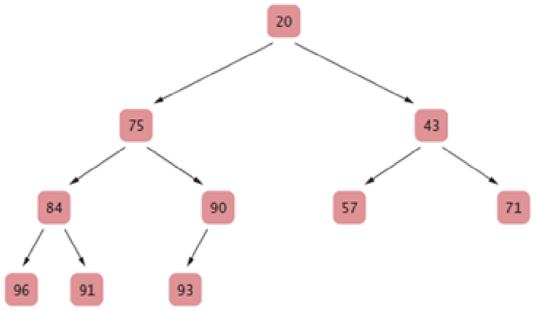


Figure 27 A Heap

- The value of every node is smaller than all its descendants.

Heaps

Differences from a binary search tree

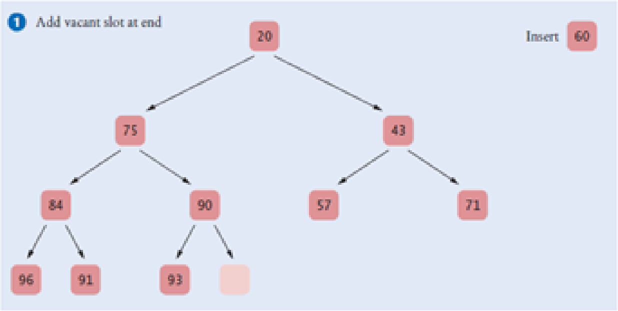
1. The shape of a heap is very regular.
    - Binary search trees can have arbitrary shapes.
2. In a heap, the left and right subtrees both store elements that are larger than the root element.
    - In a binary search tree, smaller elements are stored in the left subtree and larger elements are stored in the right subtree.

Heaps - Insertion

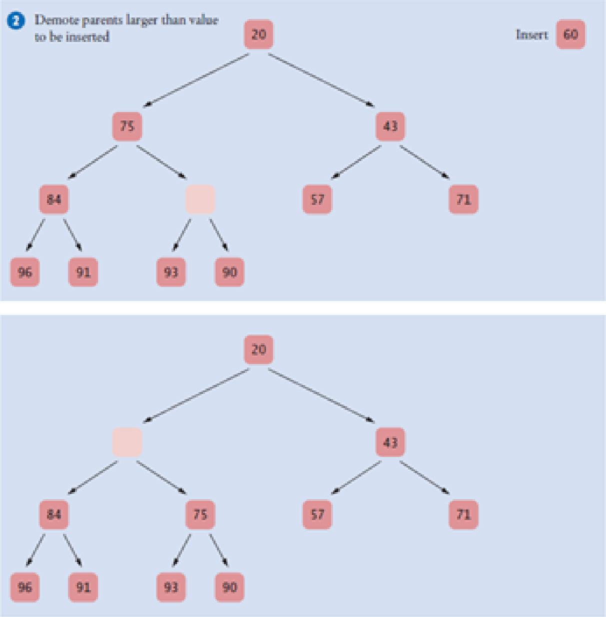
Algorithm to insert a node

1. Add a vacant slot to the end of the tree.
2. If the parent of the empty slot if it is larger than the element to be inserted:
    - Demote the parent by moving the parent value into the vacant slot,
    - Move the vacant slot up.
    - Repeat this demotion as long as the parent of the vacant slot is larger than the element to be inserted.
3. Insert the element into the vacant slot at this point,
    - Either the vacant slot is at the root
    - Or the parent of the vacant slot is smaller than the element to be inserted.

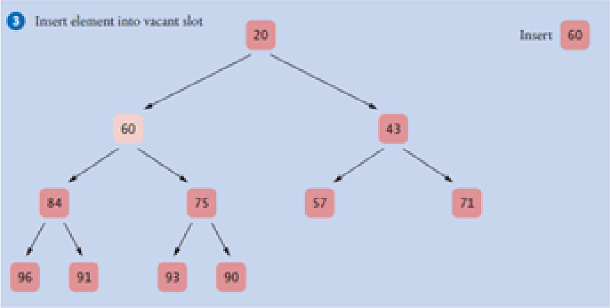
Heaps – Insertion Step 1



Heaps – Insertion Step 2



Heaps – Insertion Step 3


3 Insert element into vacant slot
Insert 60
20
60
43
84
75
57
71
96
91
93
90