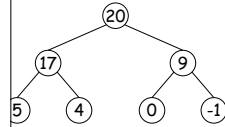


## Priority Queues, Heaps

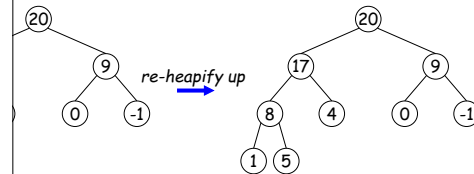
are defined by operations "add," "find largest," "remove largest," and "insert." They are used for scheduling long streams of actions to occur at various times, for sorting (keep removing largest). Implementation is the **heap**, a kind of tree. (The same term is used to describe the pool of storage an operator uses. Sorry about that.)

## Example: Inserting into a simple heap

1 20

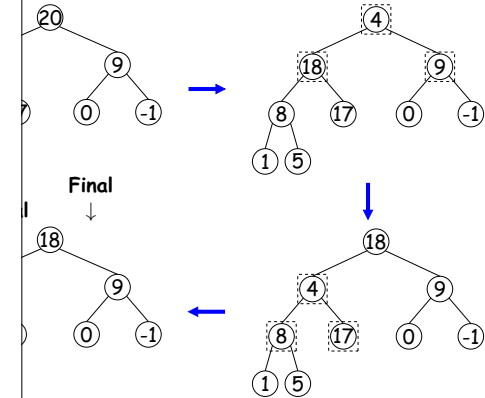


boxes show where heap property violated



## Removing Largest from Heap

**Algorithm:** Move bottommost, rightmost node to top, then swap as needed (swap offending node with larger child) to restore heap property.



## CS61B Lecture #23

Priority Queues (Data Structures §6.4, §6.5)

Heaps (§6.2)

Implementations: SortedSet, Map, etc.

Hashing (Data Structures Chapter 7).

## Heaps

A heap is a binary tree that enforces the

**Heap Property:** Labels of **both** children of each node are less than or equal to the node's label.

The root node has the largest label.

The heap property is a binary search property, which allows us to keep tree

operations always valid to put the smallest nodes anywhere at the bottom of the tree.

A heap can be made **nearly complete**: all but possibly the last level filled with as many keys as possible.

Insertion of new value and deletion of largest value are both  $O(\lg N)$  in worst case.

Min-Heaps are basically the same, but with the minimum value at the root and children having larger values than their parents.

## Heap insertion continued

