Lab 06

AVL Tree, Priority Queue & Heaps

Q1 - A - Adding to minheap

Draw the tree state of the min-heaps after adding the following elements into the tree.

Check lecture11_Priority Queue_Heap.pdf - Page 14-20

▪ When an element is added to a heap, it should be initially placed as the rightmost leaf (to maintain the completeness property).

▪ bubble up: To restore heap ordering, the newly added element is shifted ("bubbled") up the tree until it reaches its proper place.

Q1 - B - Removing from min-heap

Show the state of the min-heaps after calling the remove() method(discussed in the class) 3 times.

Check lecture11_Priority Queue_Heap.pdf - Page 21-28

▪ When the root is removed from a heap, it should be initially replaced by the rightmost leaf (to maintain completeness).

▪ Bubble down: To restore heap ordering, the new improper root is shifted ("bubbled") down the tree until it reaches its proper place.

Q2 - (Balanced Trees): Is It Balanced?

Your task is to complete two methods in the BST.java file: recIsBalanced() and height().

Check lecture9_tree_part4.pdf - Page 3

- Remember that a tree is balanced if and only if:
  a) the difference between the heights of the right and left subtrees is zero or one,
  b) AND each of the right and left subtrees are also balanced.

Height of each “node/subtree” is important.

Q3 - Priority Queue

Check lecture 11 - Slides 12-28