Design & Analysis of Algorithms

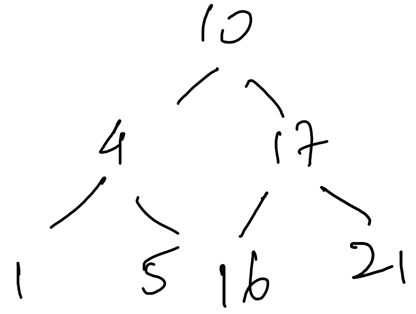
Pranav Umakant Pujar

1001965075

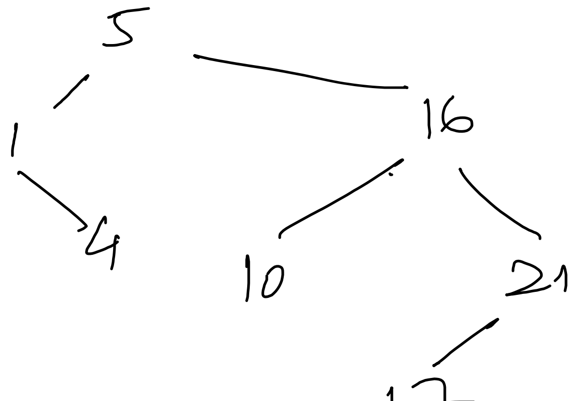
12.1-1) given array = { 1, 4, 5, 10, 16, 17, 21 }

BST constraint: left children must be ≤ parent
right child must be > parent

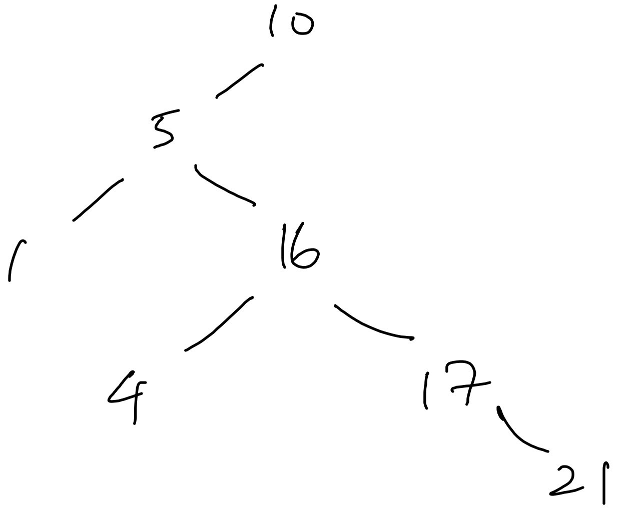
Tree of height 2: (3 layers, first layer is root)



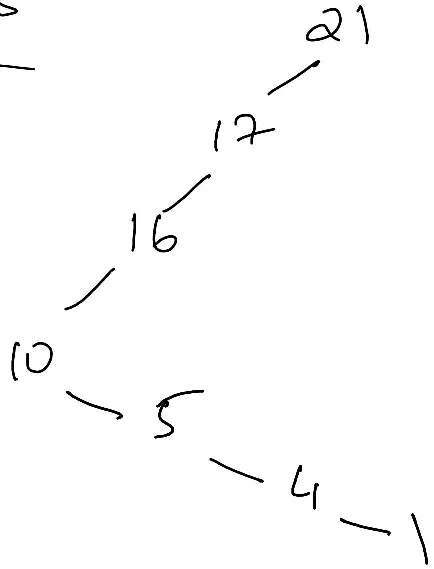
Tree of height = 3 :



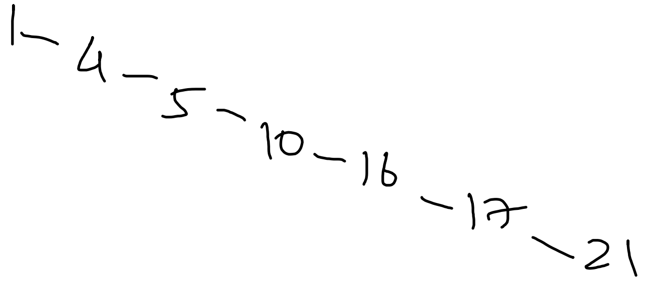
Tree of height 4 :



Tree of height 5 :



Tree of height 6 :



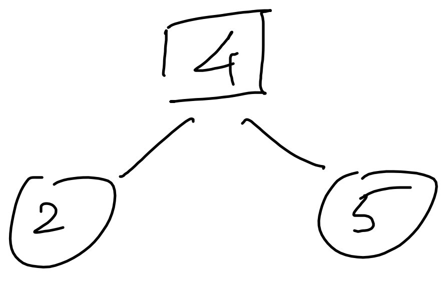
12.1-2) BST Property : All left children are lesser than the parent node in value and all right children are greater than the parent node in value. This means that an in-order DFS traversal through the BFS tree will return the sorted order of elements in ascending order.

Min-Heap Property: Every node has value that is lesser than or equal to that of its parent. The root node is the minimum value.

BST is already sorted in-order, so it is possible to retrieve a sorted sequence in O(n) time. However, a min heap must heapify (O(logn)) every time the "min" element is popped. This happens 'n' times, hence O(nlogn) is the minimum TC for this operation using a min-heap. There is no relative ordering maintained between the nodes at a particular level in a min-heap, unlike a BST.

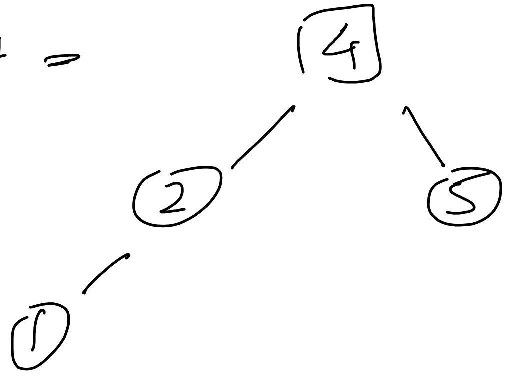
HW - Ch 13

1) a) given Red-Black Tree: (using ☐ = Black and O = red)



to insert '1', we can try inserting ① as a red node.

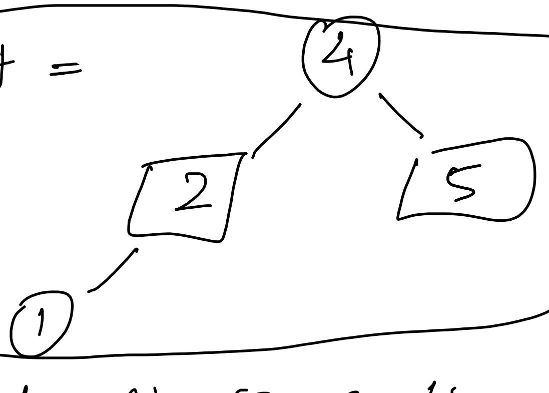
result =



this violates property that red node cannot have a red child.

to resolve this, lets convert 4 to a red node, since '4' is not root, this is possible.

result =



final check: No consecutive red nodes, black-height is maintained from root to all NIL nodes.

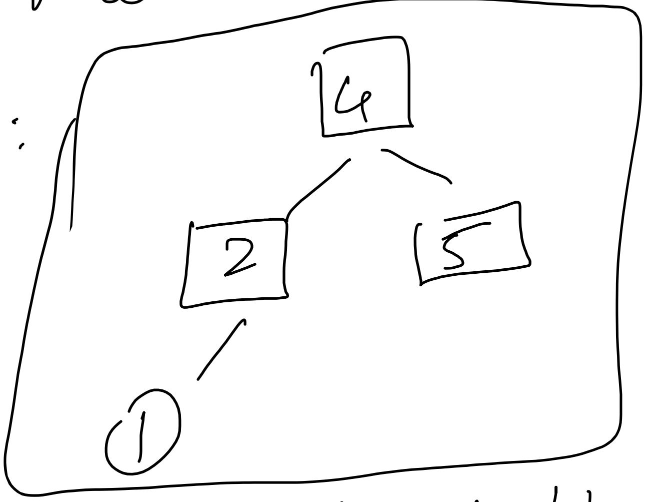
any parent of ④ will handle any further balancing.

This is Now a valid Red-Black tree.

b) Now, since 4 is the root, it will have to remain a black node.

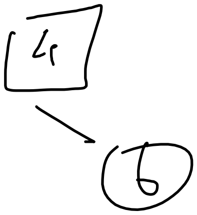
So, to add red node ①, we need to have its parent nodes be black. Hence, ⑤ and ② need to become $\boxed{5}$ and $\boxed{2}$ black nodes.

result:



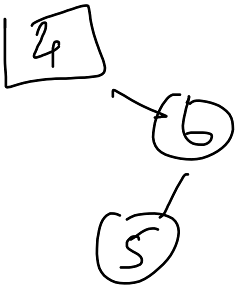
this is now a valid red-black tree.

c) given tree :



inserting red node ⑤:-

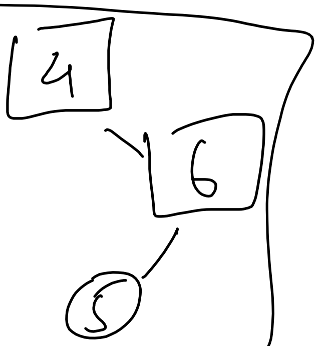
result:



this violates property that a red node cannot have another red node.

Changing parent of ⑥ to black:

result:



this tree satisfies the Red-black properties!