Exam no. 2

Wed next week nov 14

Problem with arrays

1.insertion //create push everything back and forth

2.deletion //deletion to fill that gap

3.you need to specify the array size //beginning to search size

4. search in O(n)

WE SOLVED SOME OF THE PROBLEMS BY USING LINKED LIST

1.good for insertion

2. good for deletion

3. not required to specify the number of nodes

4. still problem the searching. The runtime to search is big O(n)

We introduced binary search tree BST

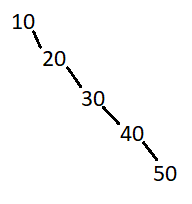
To resolve the searching process in linked list.

What is the runtime to search for item x in BST with n nodes. Runtime is O(log n base2)

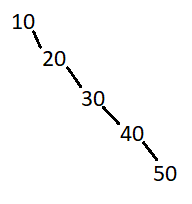
But, still there is a problem with BST.

The problem is if the original data is already sorted, that is:

Data: 10 20 30 40 50



The tree becomes a linked list and the runtime is O(n), doesn’t make it faster.

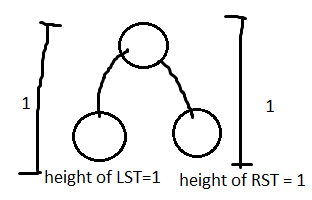


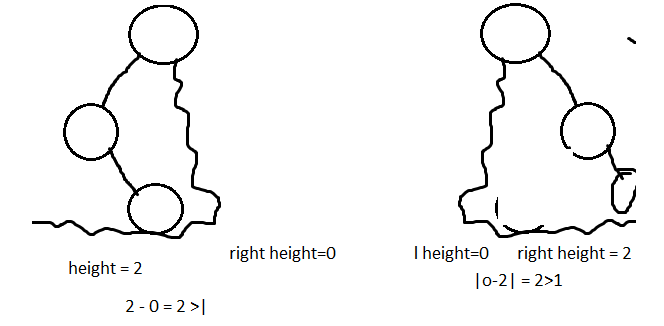
3 students solved this problem by creating an AVL tree.

AVL tree is a balanced BST. Tree wont look like

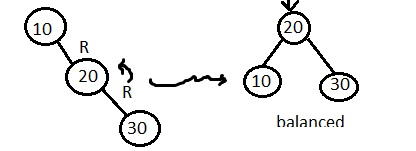
To construct an AVL tree, we should consider the following four special cases.

Note The tree is balanced if the difference between the height of left and right subtree is not more than 1.

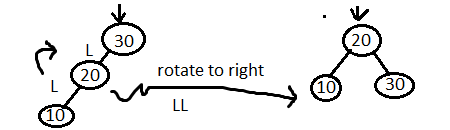




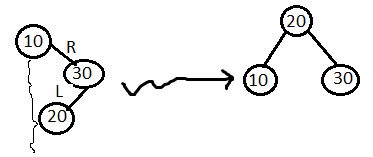
Case 1) Data: 10,20,30

rotate left

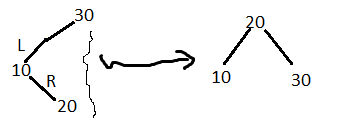
Case 2) Data: 30, 20 ,10



Case 3) 10,30,20

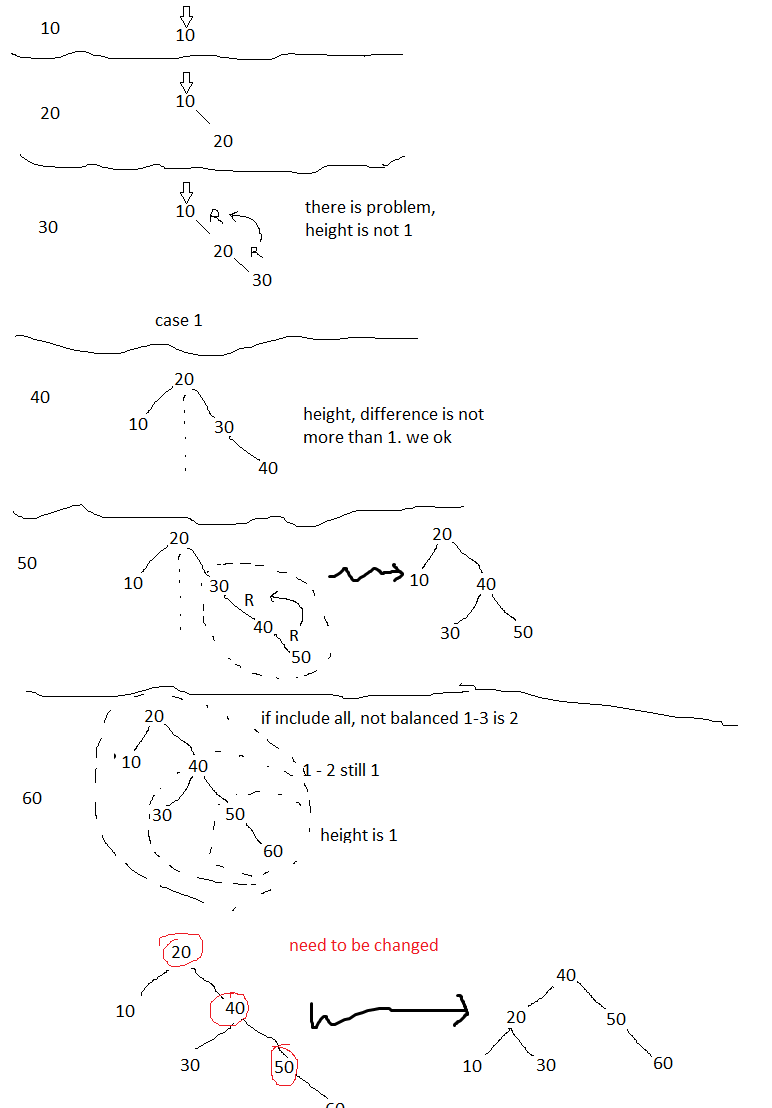


Case 4) Data: 30, 10, 20

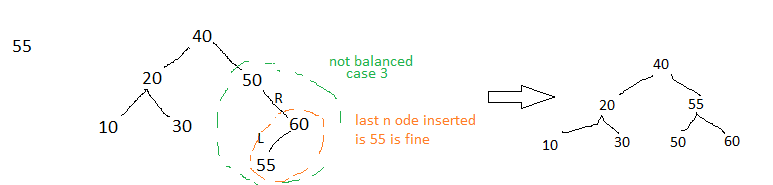


EXAMPLES TO IMPLEMENT CASES

Ex. Insert 10, 20, 30, 40, 50, 60 into an AVL-tree

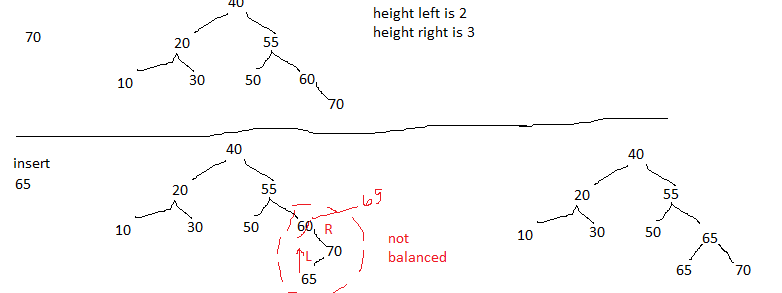


Example, add 55 to the tree above



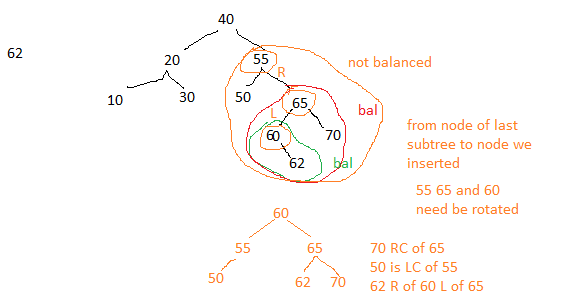
At this point, anything you insert will be balanced

Test insert 70



At this point anything added will make both unbalanced

Ex. Add 62



Quiz:

Insert name of months from Jan to Dec into an AVL-tree turn in your answer on Monday