

Sub: Advanced Data Structure

Rollno: 22481A0563

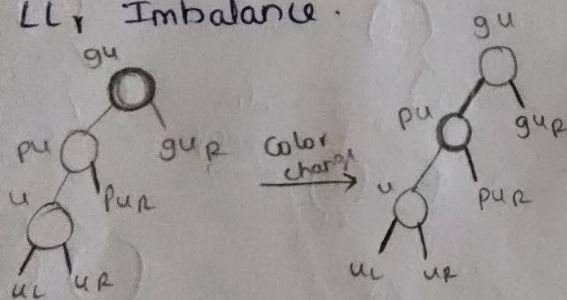
1. Define a Red-Black tree. Illustrate different imbalances in Red-black tree insertion operation. Start with an empty Red-black tree and insert 50, 10, 80, 90, 70, 60, 65, 62, 100, 40, 30, 25. After each insertion draw the updated tree color the nodes, links.

Definition: A Red black Tree is a Binary Search Tree in which every node is colored either red or black. In addition to this it has to satisfy the following properties:-

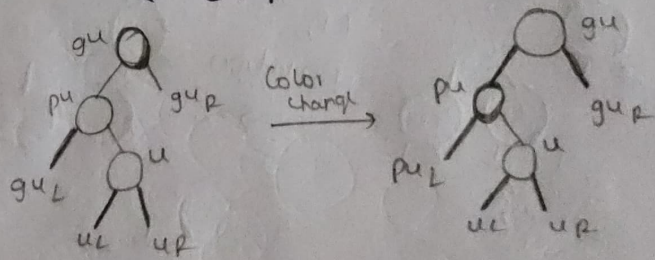
1. RB1: The Root and all external nodes are colored black.
2. RB2: No root-to-external node path has 2 consecutive red nodes.
3. RB3: All root-to-external node paths have same no. of black nodes.
4. RB₁[']: Pointers from an internal node to an external node are black.
5. RB₂[']: No root-external-node path has 2 consecutive red pointers.
6. RB₃[']: All root-external-node paths have same no. of black pointers.

Imbalances due to Insertion operation.

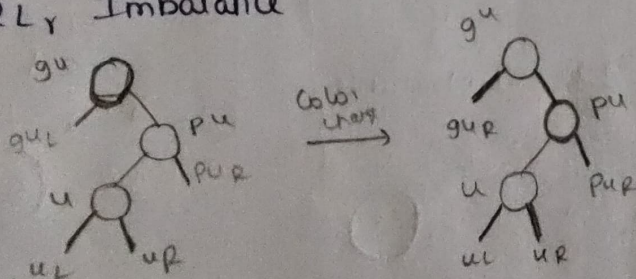
1. LL_r Imbalance.



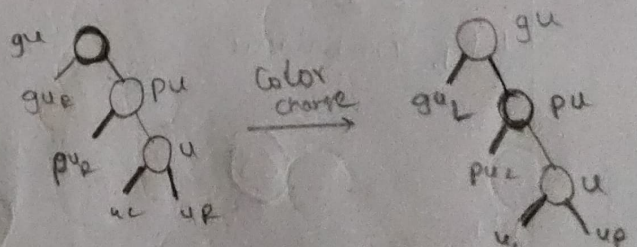
2. LR_r Imbalance.



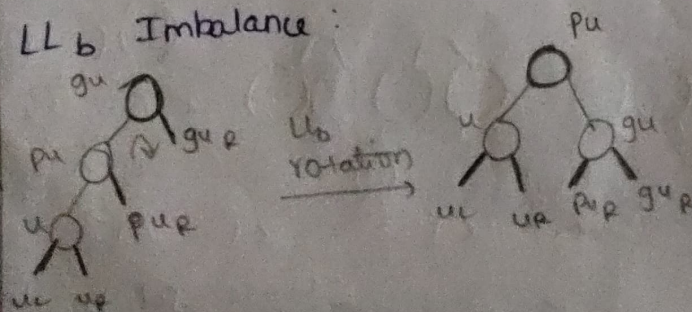
3. RL_r Imbalance



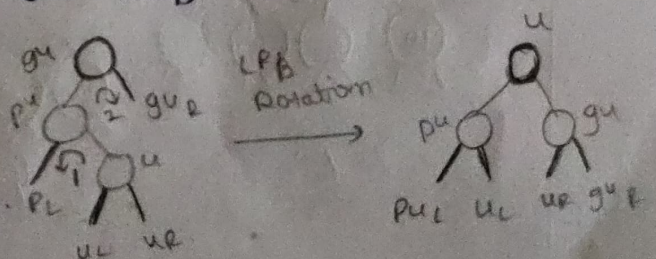
4. RR_r Imbalance



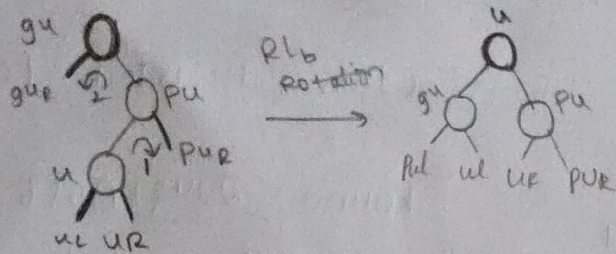
5. LL_b Imbalance:



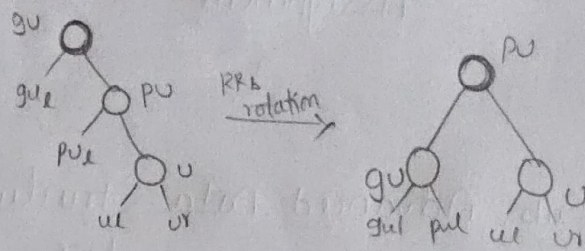
6. LR_b Imbalance.



7. RL_b Imbalance:



8. RR_b Imbalance:

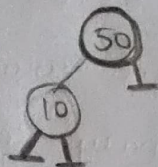


Insertion:

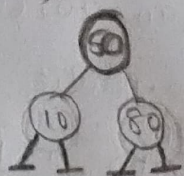
1) 50



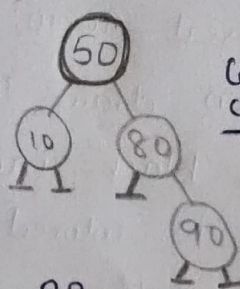
2) 10



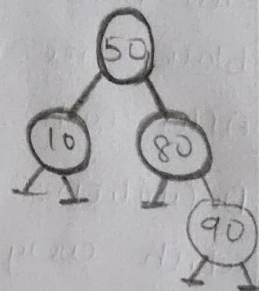
3) 80



4) 90



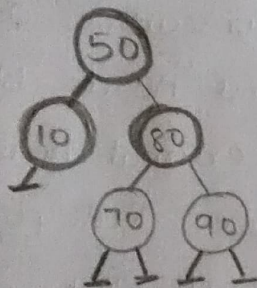
Color change



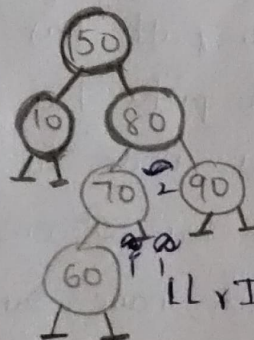
RR_r Imbalance

5)

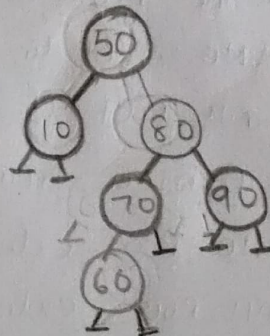
70



6) 60



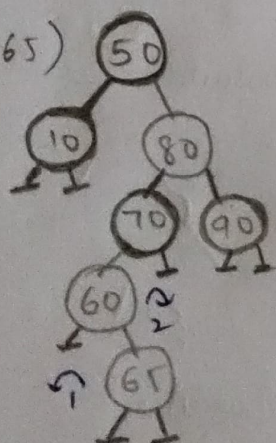
Rotation
Color change



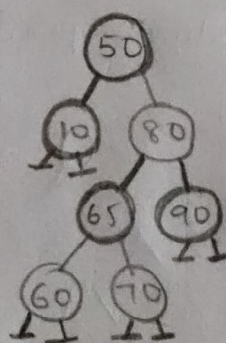
LL_r Imbalance

7)

65

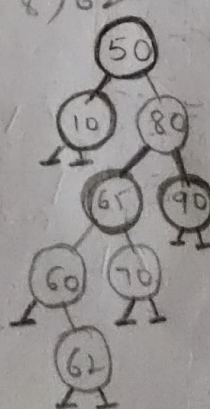


LR_b
Rotation

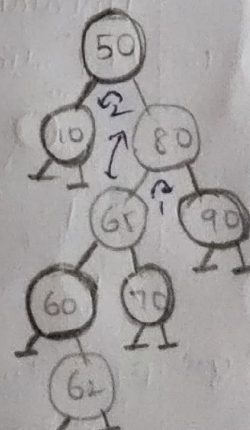


LR_b Imbalance

8) 62



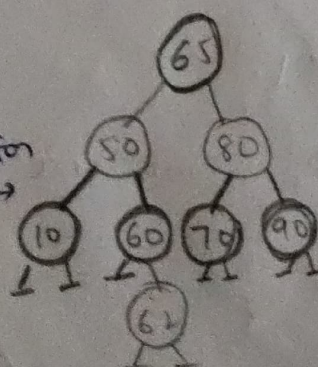
Color change



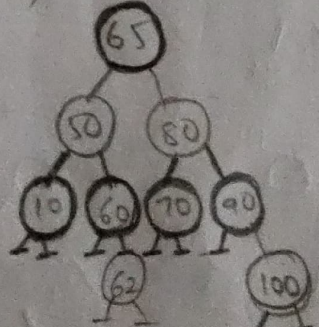
LR_r Imbalance

RL_b Imbalance

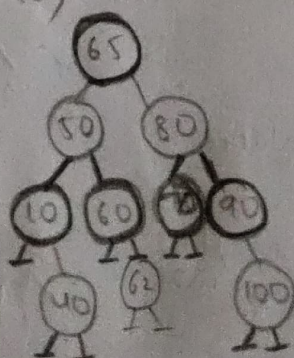
RL_b
Rotation



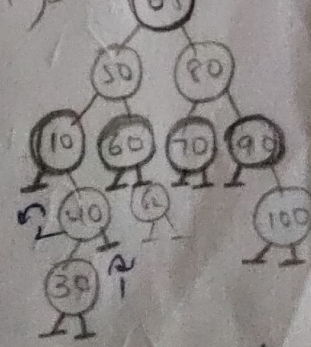
9) 100



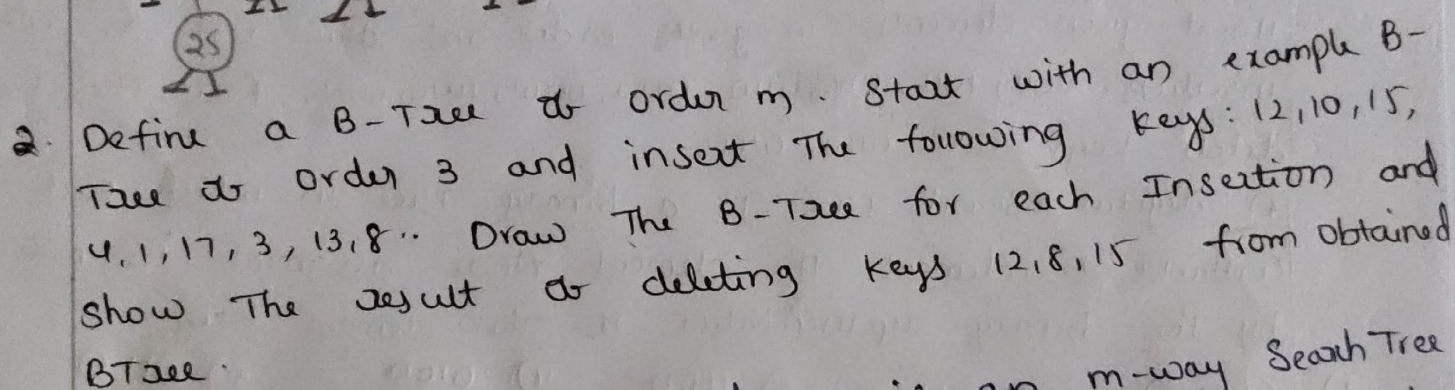
10) 40



11) 30



RL_b Imbalance

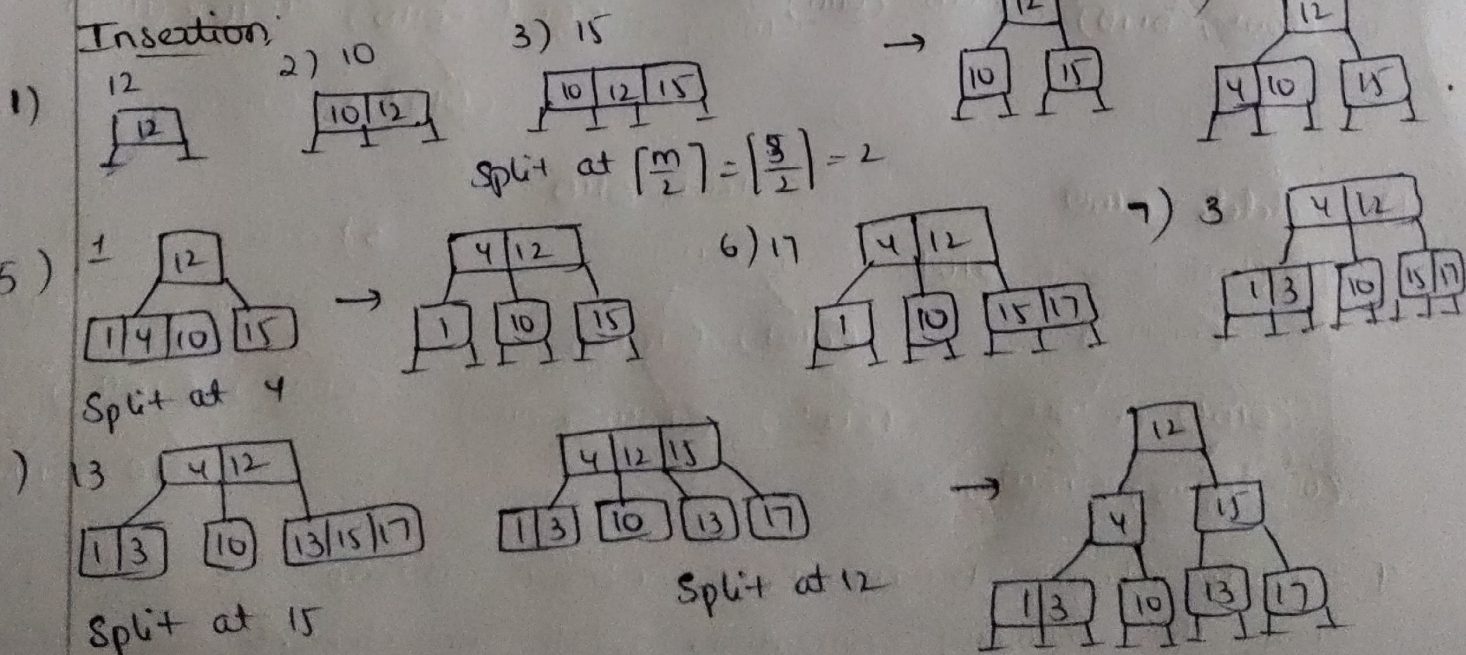


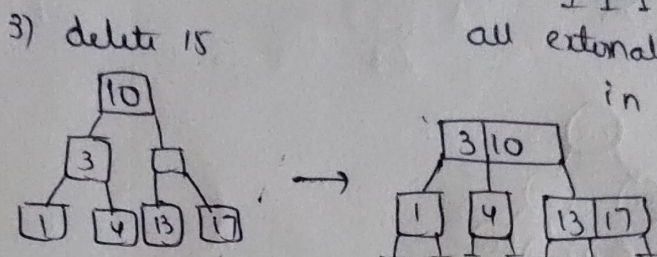
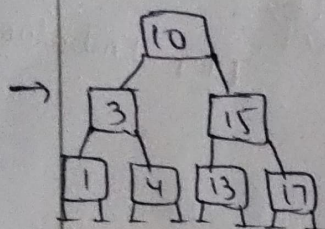
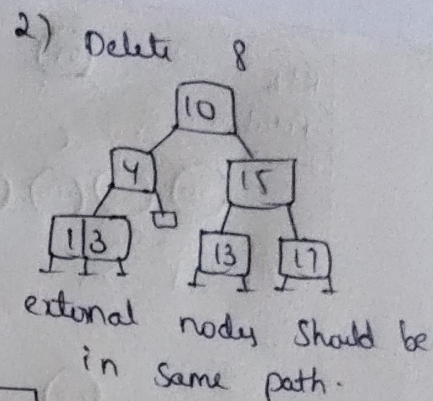
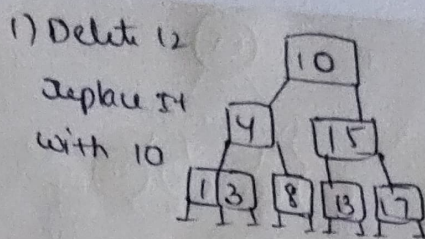
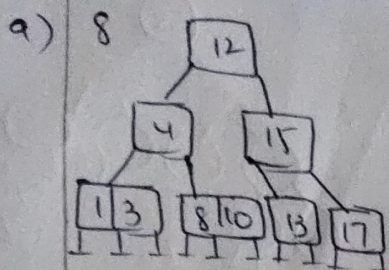
Definition: A B-tree of order m is an m -way Search Tree extended

If The B-tree is not empty The corresponding extended Tree satisfies the following properties

1. The root has atleast 2 children.
2. All internal nodes other than root have atleast $\lceil \frac{m}{2} \rceil$ children
3. All external nodes are at same level.

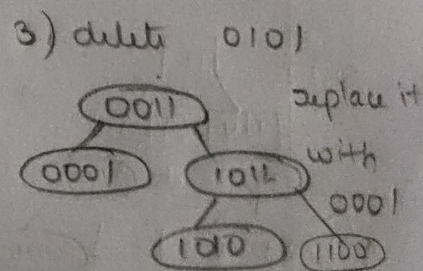
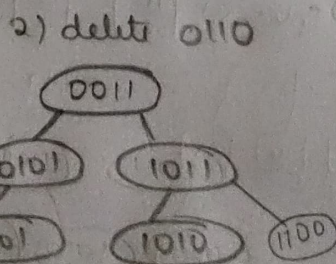
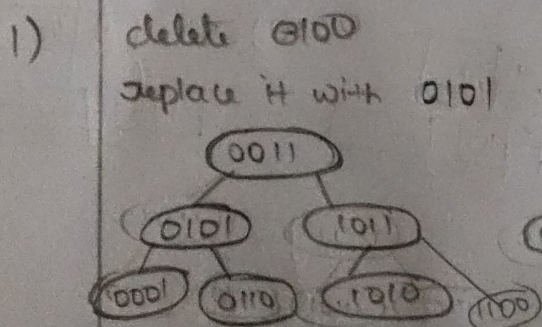
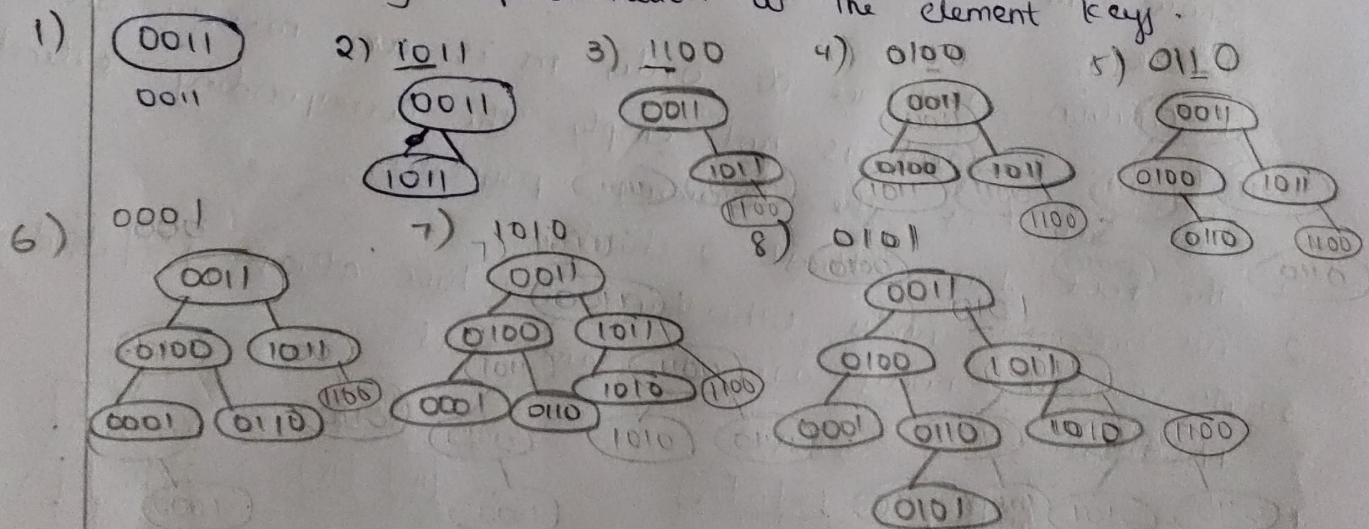
Insertion





3. write about digital Search Tree. Start with an empty DST insert given elements in given order and draw DST for every insertion: 0011, 1011, 1100, 0100, 0110, 0001, 1010, 0101. From The Tree delete keys: 0100, 0110, 0101.

A digital Search Tree is a binary Tree in which each node contains 1 element. The element Node assignment is determined by The binary representation of The element key.



4. write about Binary Trie. obtain Binary Trie after performing The following operations with The elements given below. Insert: 10010, 00100, 01001, 01010, 10100, 11100, 10111

Delete : 01010, 10010

Binary Tree is a Binary Tree That has Two kinds of Node.

1. Branch Node: It has Left and Right child pointers and doesn't have data field.

2. Element Node: It has no child pointers. It has data field.

