LAB ASSIGNMENT 9
DATA STRUCTURES AND ALGORITHMS– II
(SEM – IV)

CASE STUDY: **Rohaan Advani - 111903151**

1. Explain Red Black Trees (RB Trees)

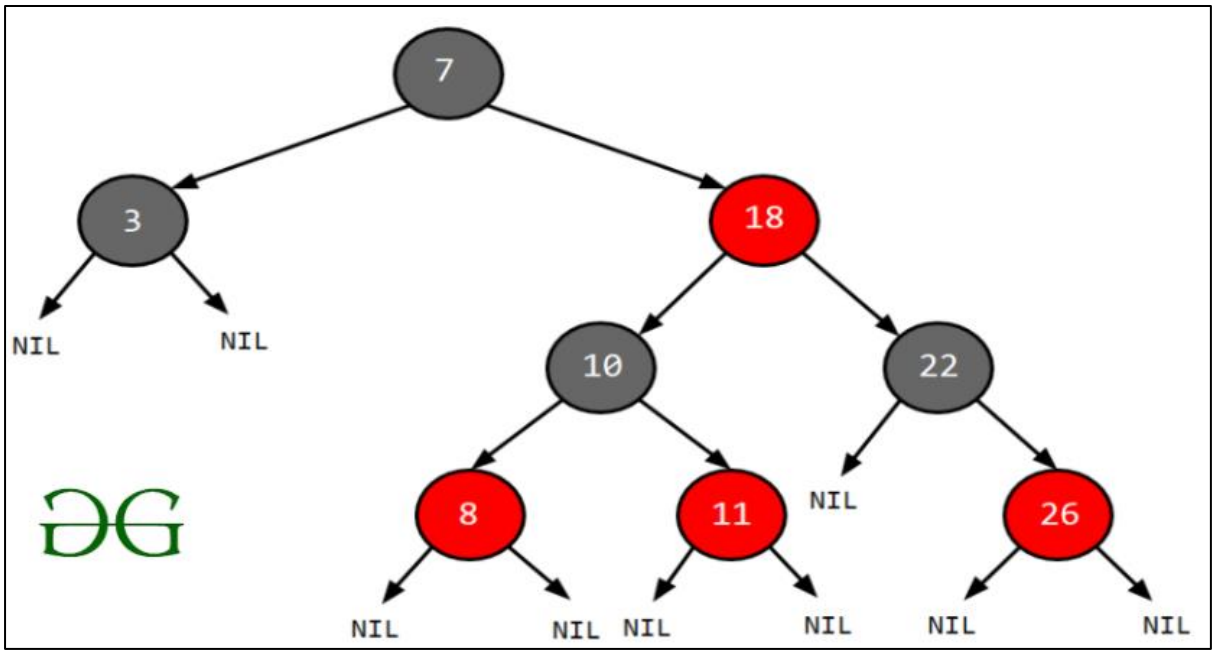
Ans. A red-black tree is a kind of self-balancing binary search tree where each node has an extra bit, and that bit is often interpreted as the colour (red or black). These colours are used to ensure that the tree remains balanced during insertions and deletions. Although the balance of the tree is not perfect, it is good enough to reduce the searching time and maintain it around O(log n) time, where n is the total number of elements in the tree.

Rules Every Red-Black Tree Follows:

1. Every node has a colour either red or black.
2. The root of the tree is always black.
3. There are no two adjacent red nodes (A red node cannot have a red parent or red child).
4. Every path from a node (including root) to any of its descendants NULL nodes has the same number of black nodes.

2. Draw the representation of RB Tree Structure and write the basic operations that can be performed on RB Trees

Ans.



Basic Operations we perform of Red-Black Trees: Insert, Search & Delete.

3. Explain the rules for balancing RB Trees

Ans. In the Red-Black tree, we use two tools to do the balancing.

1. Recoloring
2. Rotation

Recolouring is the change in colour of the node i.e. if it is red then change it to black and vice versa. It must be noted that the colour of the NULL node is always black. Moreover, we always try recolouring first, if recolouring doesn't work, then we go for rotation.
The algorithm has mainly two cases depending upon the colour of the uncle. If the uncle is red, we do recolour. If the uncle is black, we do rotations and/or recolouring.

Algorithm:
Let x be the newly inserted node.

1. Perform standard BST insertion and make the colour of newly inserted nodes as RED.
2. If x is the root, change the colour of x as BLACK (Black height of complete tree increases by 1).
3. Do the following if the color of x's parent is not BLACK and x is not the root.
   a) If x's uncle is RED (Grandparent must have been black)
      (i) Change the colour of parent and uncle as BLACK.
      (ii) Colour of a grandparent as RED.
      (iii) Change x = x's grandparent, repeat steps 2 and 3 for new x.
   b) If x's uncle is BLACK, then there can be four configurations for x, x's parent (p) and x's grandparent (g)
      (i) Left Left Case (p is left child of g and x is left child of p)
      (ii) Left Right Case (p is left child of g and x is the right child of p)
      (iii) Right Right Case (Mirror of case i)
      (iv) Right Left Case (Mirror of case ii)

4. Mention the Complexity Analysis of insert, delete and search operations in RB Trees.

Ans.

Sr. No.	Algorithm	Time Complexity
1.	Search	O(log n)
2.	Insert	O(log n)
3.	Delete	O(log n)

5. Perform a comparative study of RB tree with Binary Search Tree

Ans. Comparison of BST & RBT is as follows:

1. Most of the BST operations (e.g., search, max, min, insert, delete, etc) take O(h) time where h is the height of the BST. The cost of these operations may become O(n) for a skewed Binary tree. If we make sure that the height of the tree remains O(log n) after every insertion and deletion, then we can guarantee an upper bound of O(log n) for all these operations. The height of a Red-Black tree is always O(log n) where n is the number of nodes in the tree.
2. BST stores only key value while RBT stores key and colour.
3. Rotation Functions are not used in BST as there is no requirement to Balance BST but are used in RBT as it's a Balanced Tree.