



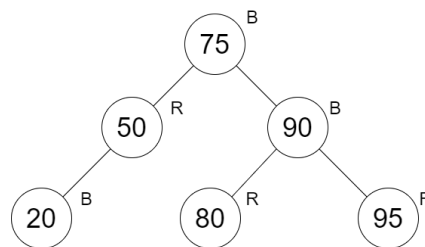
**University of Colombo School of Computing**  
**SCS 1308 - Foundations of Algorithms**  
**Take-Home 13**

**Instructions**

- Try the following questions and upload your answer script as a zip file to the given link in the UGVLE on/before 09th March at 6 pm.
- Note: Rename your zip file with your index number and name. (i.e: indexNo\_Name.zip).

1) Red-black trees are a fundamental data structure in computer science known for their efficient balancing properties, making them suitable for a wide range of applications.

- a) Mention 4 properties of a Red-Black Tree.
- b) Is the following BST a Red-Black Tree? If not, which requirements does it not satisfy?  
(R - Red, B - Black)



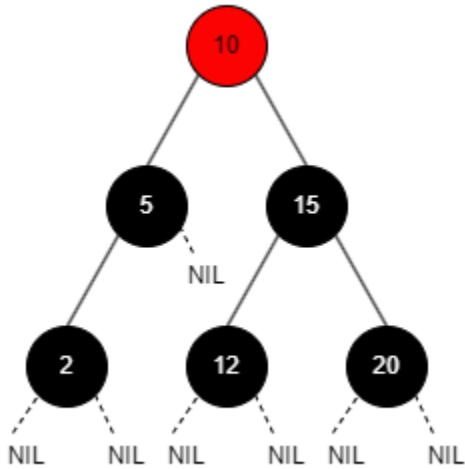
- c) Show the red-black trees after successively inserting the keys 8, 15, 5, 12, 14, 30, 25, 40 into an initially empty red-black tree. Show necessary steps.

2). Discuss the differences between AVL Tree and Red Black Tree.

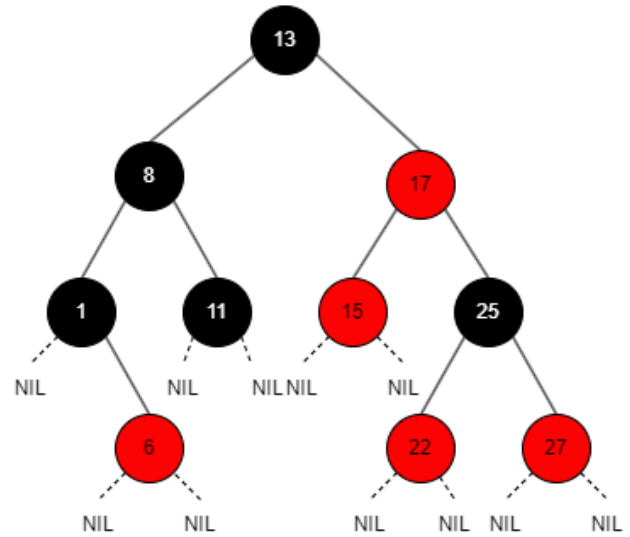
Parameter	Red Black Tree	AVL Tree
Searching		
Insertion and Deletion		
Color of the node		
Balance factor		
Strictly balanced		

- 3). How are red-black trees maintained in a balanced state?
- 4). How do you fix rule violations when a node is inserted or deleted from a red-black tree?
- 5). Identify whether the following trees are Red-black trees or not.

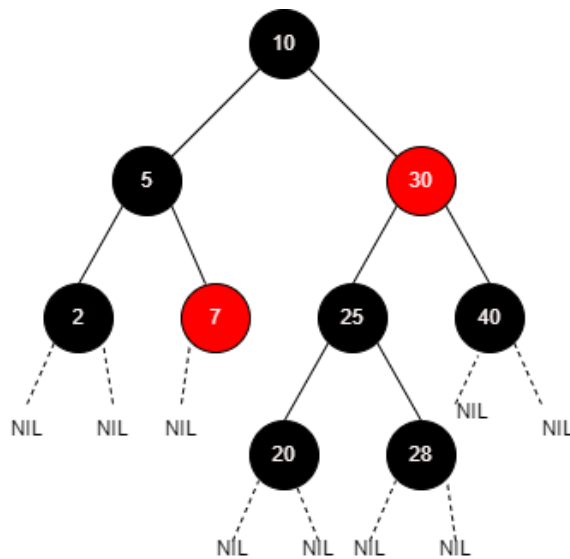
A.



B.



C.



6). Show the red-black trees after successively inserting the following keys into an initially empty red-black tree. Show all the steps.

I. 18, 7, 15, 16, 30, 25, 40, 60, 2, 1, 70

II. 20, 5, 25, 10, 9, 8, 11, 15, 6, 4, 2, 1, 7, 12, 35

7). Given the following red-black tree, show the resulting red-black tree after inserting the key 12 inside. Show each step whenever you change a node's color or make a rotation, write B (Black) or R (Red) next to each node to indicate its color, also write your operations clearly under the arrow. (Note: shaded nodes are BLACK)

