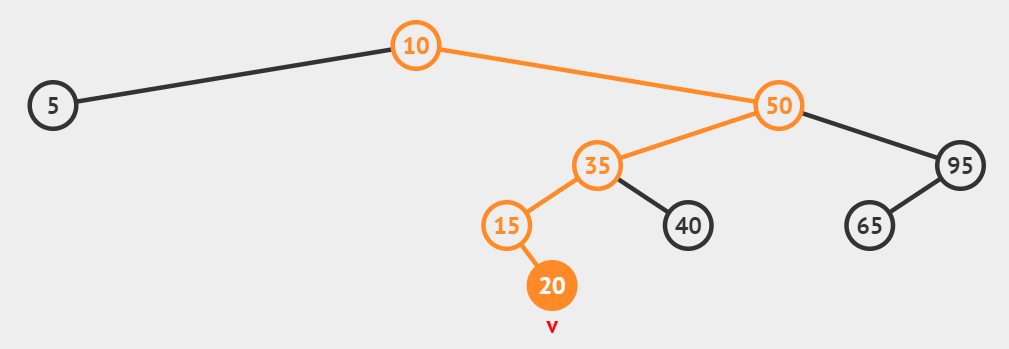
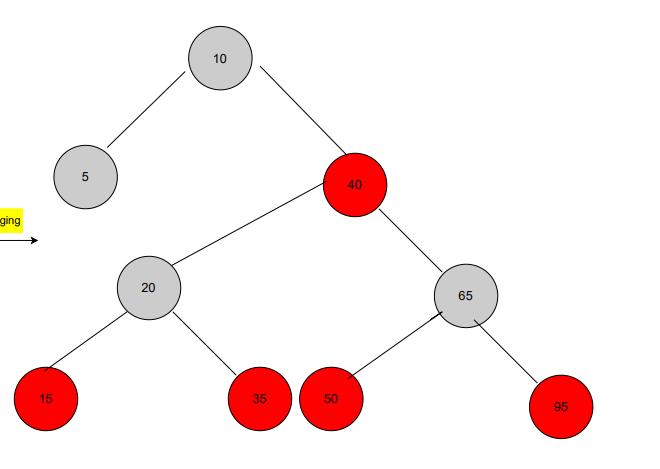
**Question 4(Reflection)**

1. **Compare the heights of the resultant trees - how do they compare with a Binary Search Tree for the same input values?**

All advanced tree structures drawn in the activity are of a lesser height, when comparing with their corresponding BST which is due to self-balancing feature of advanced trees,

Since, 2-3-4 and B-Trees as they store multiple data points in each node, it is obvious that they are of a lesser height.

Ex. BTS corresponding to first data set



**4 levels**

BRed-Black Tree of it is :

**3 levels**

1. **Compare the complexity of the algorithms, how much work would be required for**

**the main operations: insert | find | delete? Compare this to BST.**

All three advanced trees are more complex for insertion and deletion due to the re-arrangement feature used for self-balancing. All operations on advanced trees are similar to BST but slightly better as they are always balanced (O(logN)), where a normal BST can degrade to O(N).

1. **Compare the understandability of these algorithms, which would be easier to implement?**

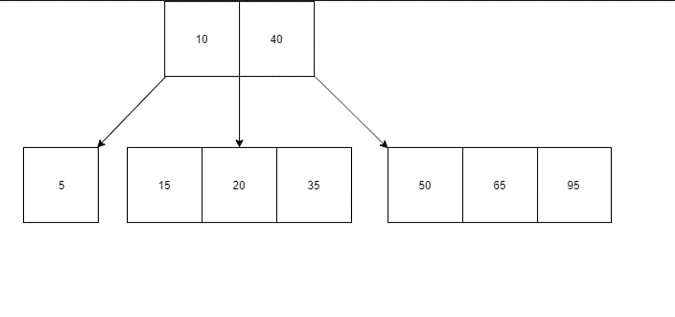
All these trees would be harder to implement than a normal BST. Because 2-3-4 and Red-Black Trees are essentially the same, they would be about the same difficulty, I think that B-Trees would be slightly harder to implement than the others due to the more complicated splitting mechanism.

1. **Describe how an in-order traversal would work on each type of tree.**

**Red-Black tree**

Traverse from the left subtree to the root then to the right subtree just as in binary search tree.

**2-3-4 Tree**

Same as in Binary search trees it will start the traversal from the leftmost leaf node and traverse through all of the values it contains and then the corresponding parent of it followed by its right child. Then it will visit the parent of that sub tree followed by the same procedure in the previous sub tree finally visiting the right sub tree in the same manner.

It would be clearer with the example for the traversal of 1st 2-3-4 tree in the examples shown below

Visiting all the values contained in the node

In-order traversal path = {5},{10},{15},{20},{35},{40},{50},{65},{95}

Visiting the root of leftmost node

Visiting the leftmost node

Visiting all the values contained in the node

**B-Trees**

This is similar to the traversal in 2-3-4 trees since both contain a similar structure even though B trees try to minimize the splitting by containing them to the maximum within the node.