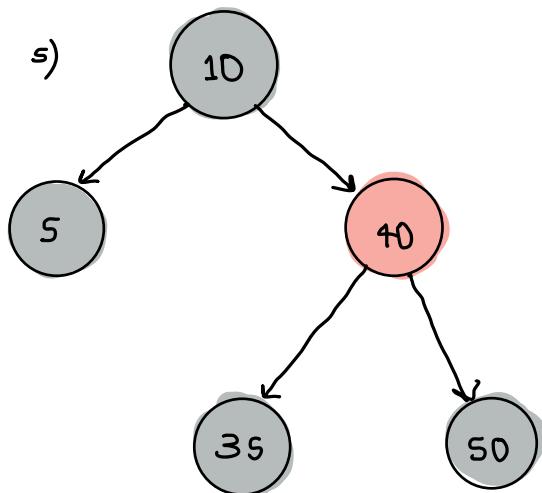
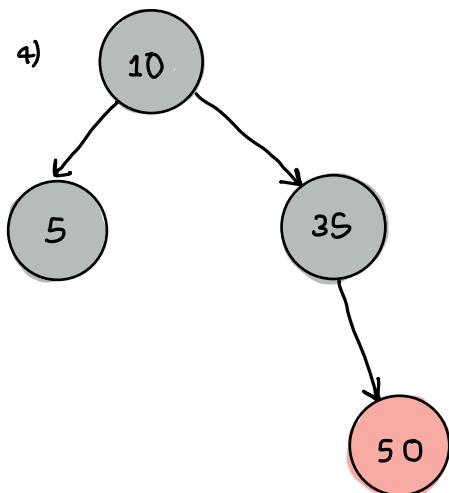
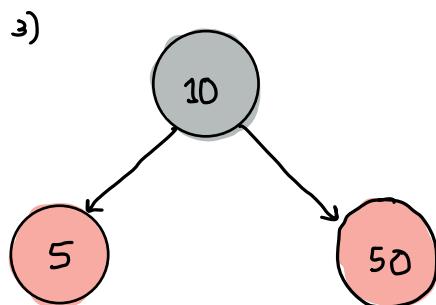
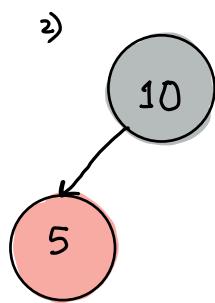
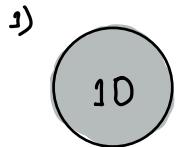


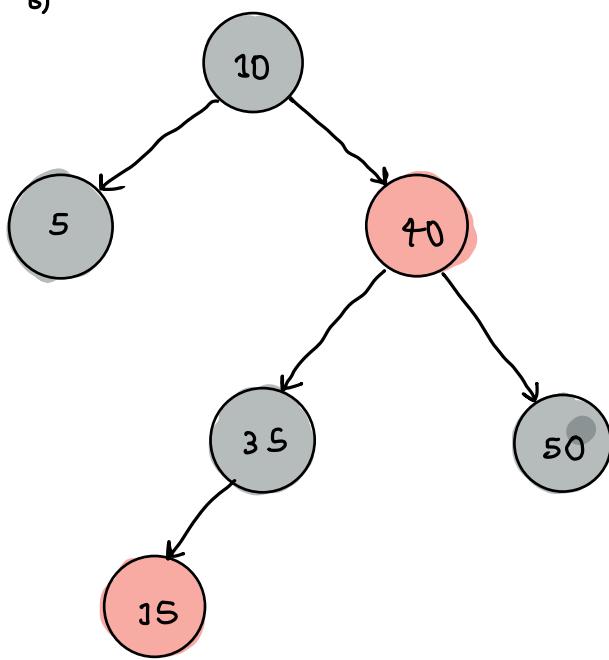
# Red Black Tree

Date = 9/04/22

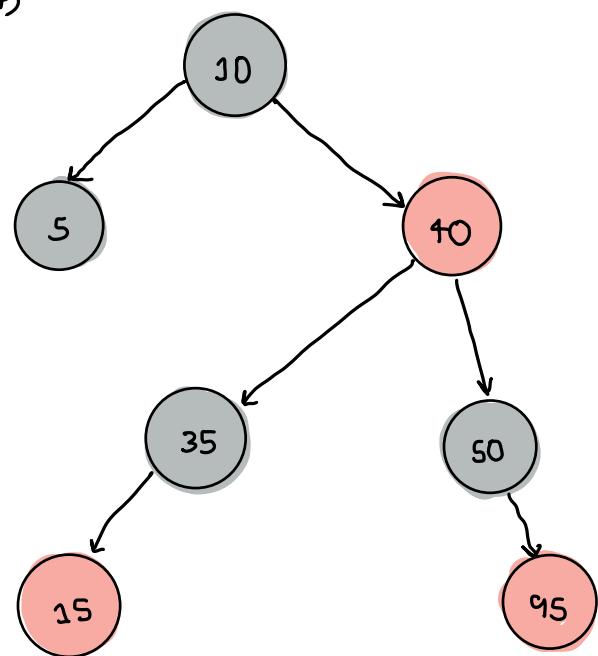
Day = Saturday



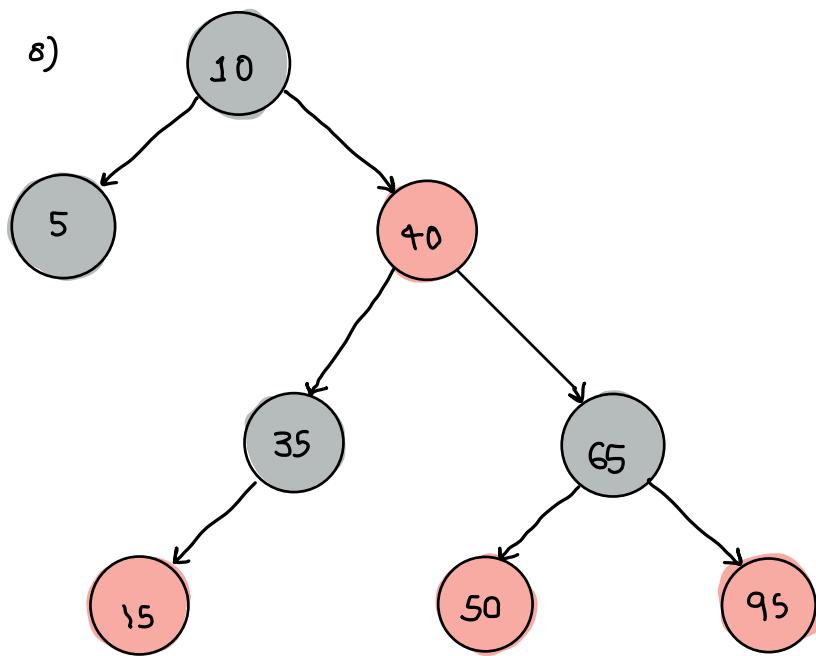
6)

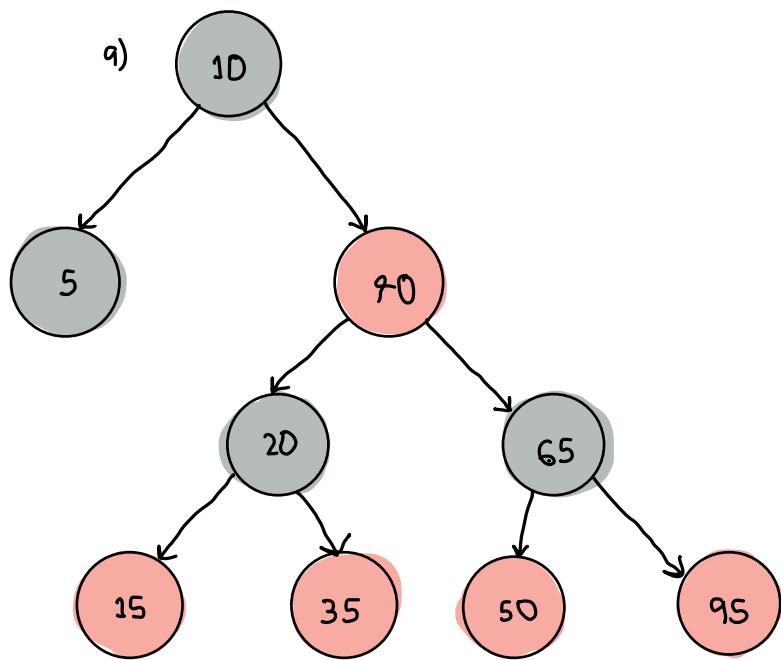


7)

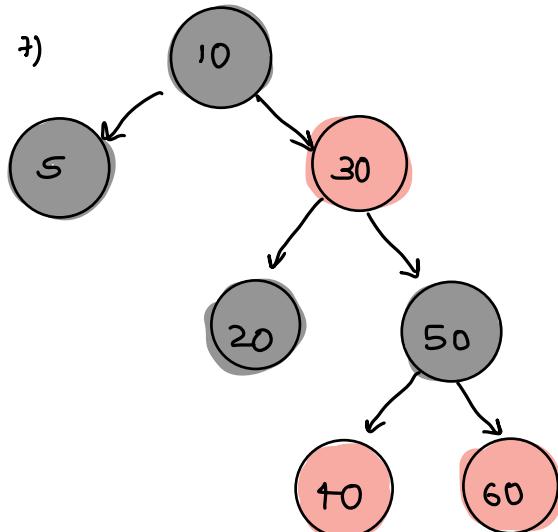
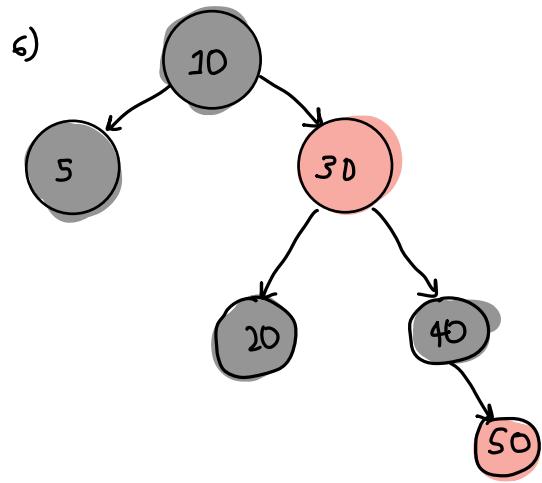
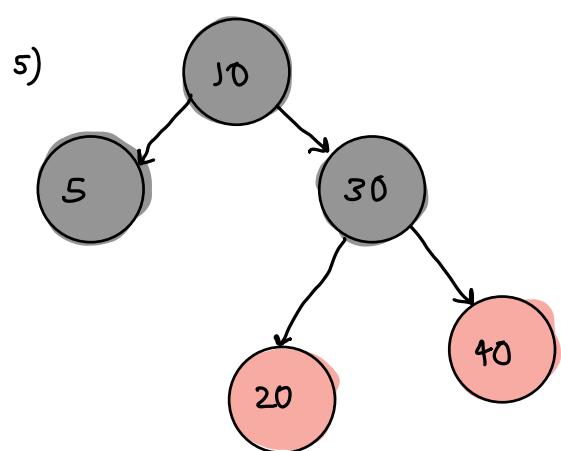
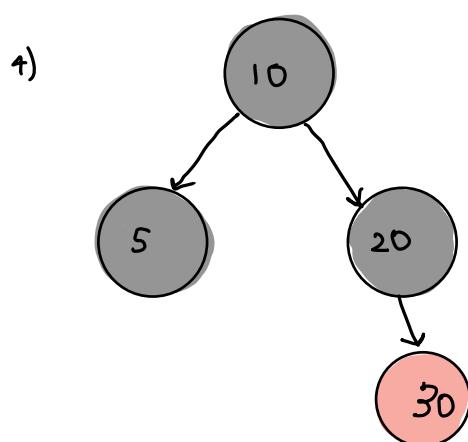
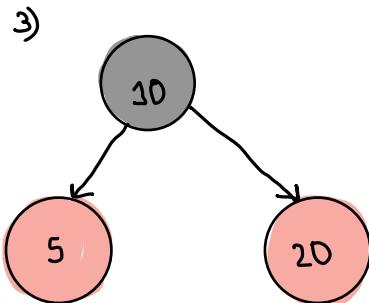
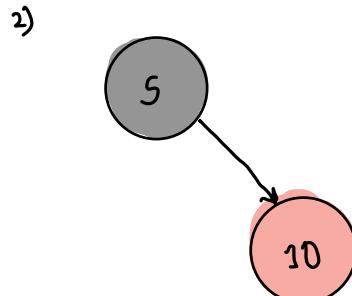
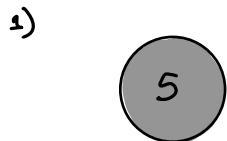


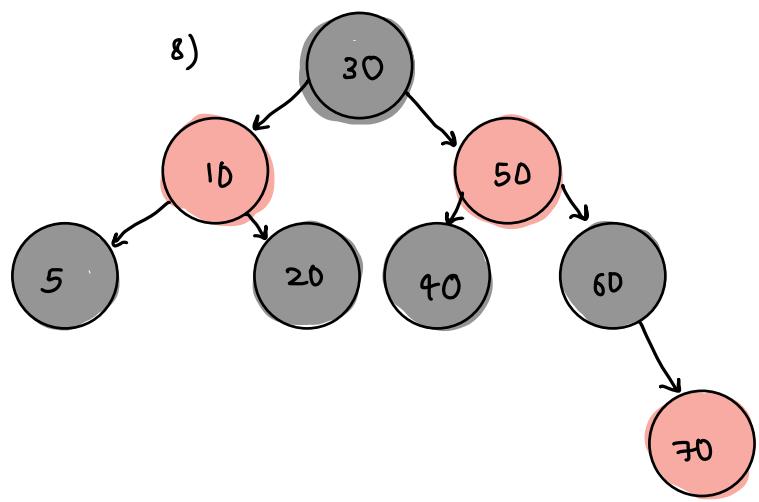
8)



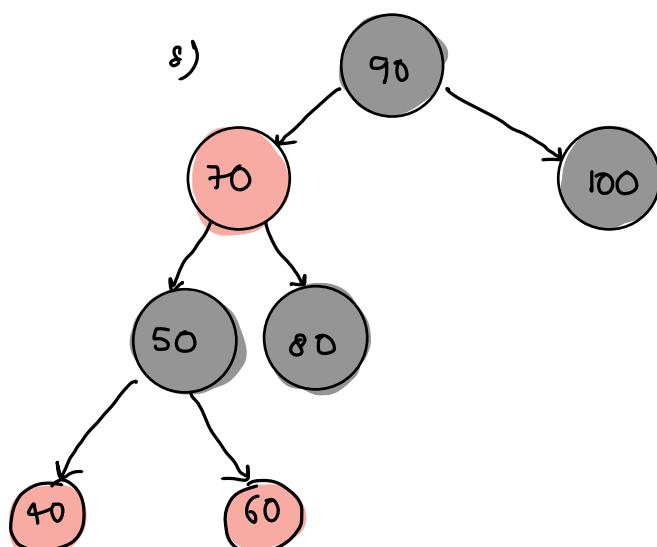
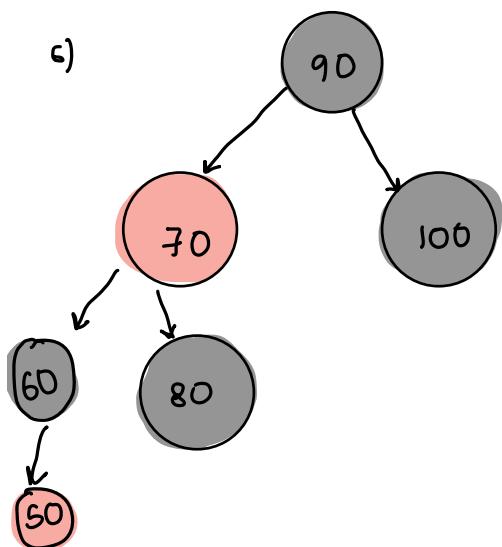
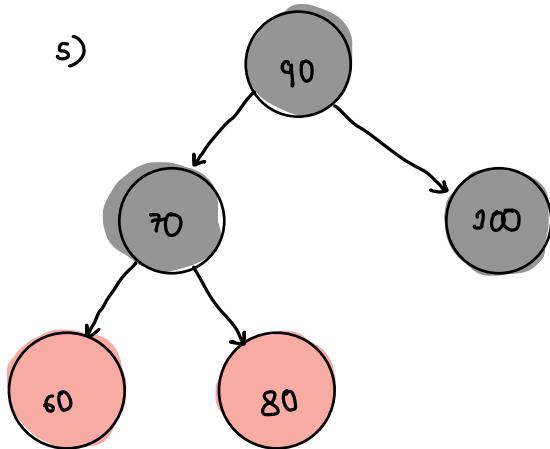
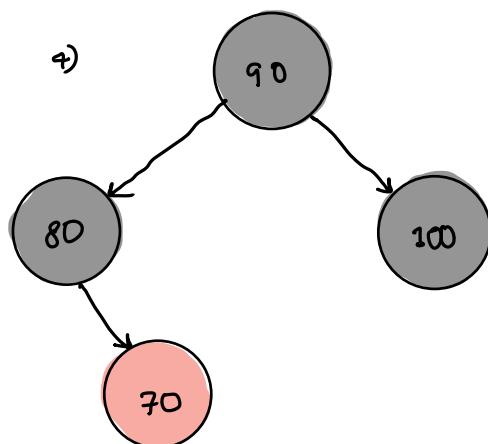
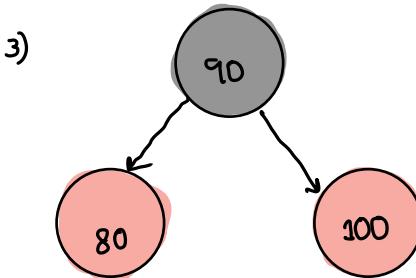
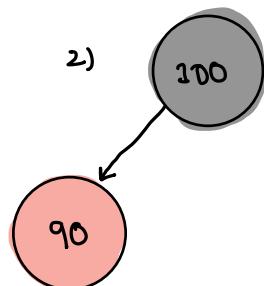


Part 2 :

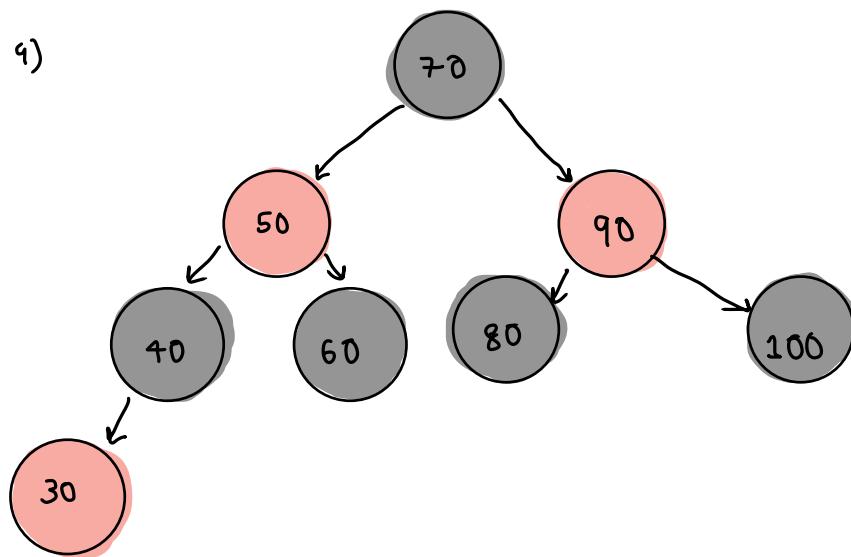




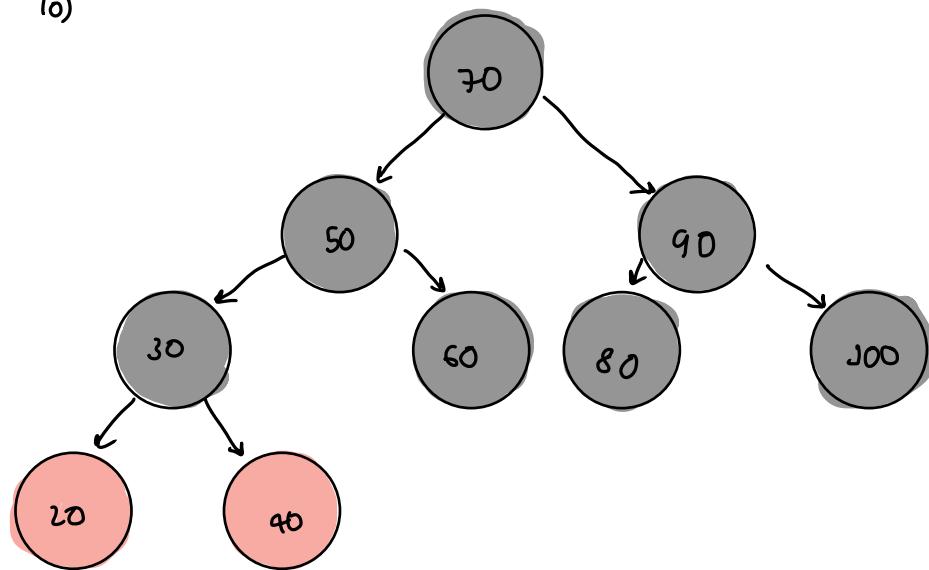
Part 3 :

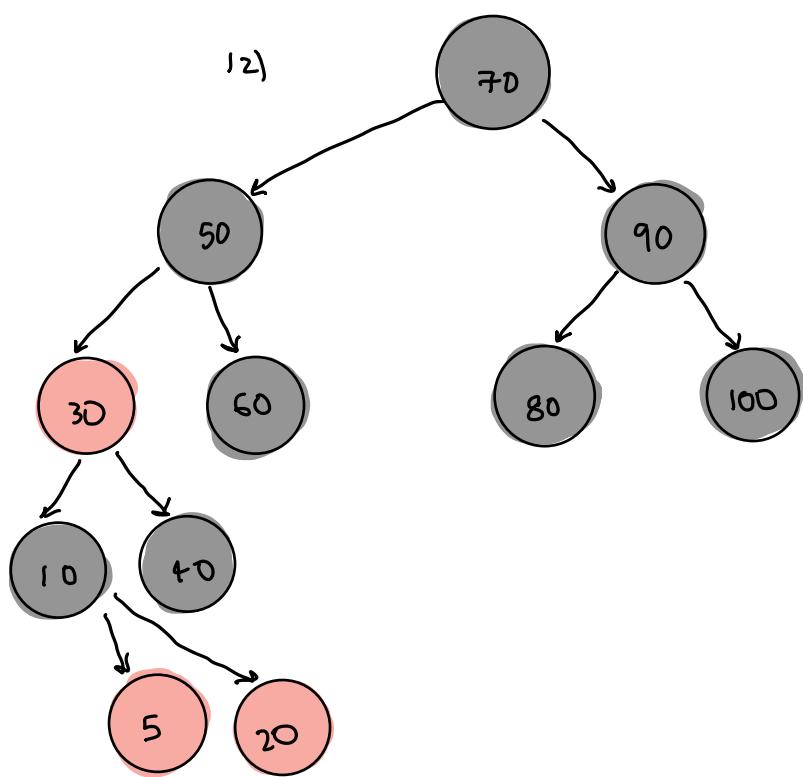
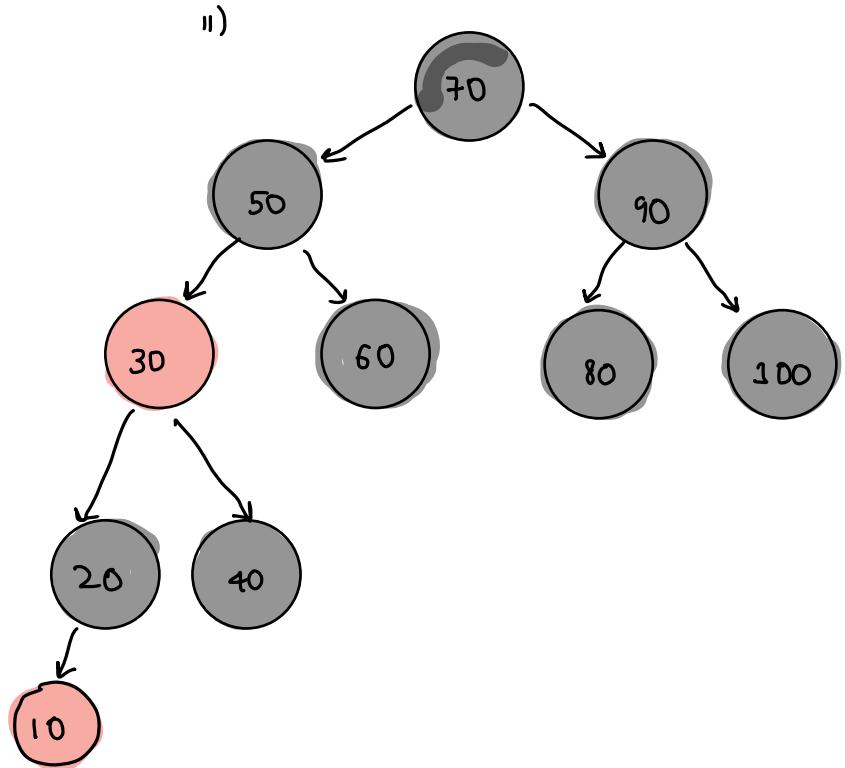


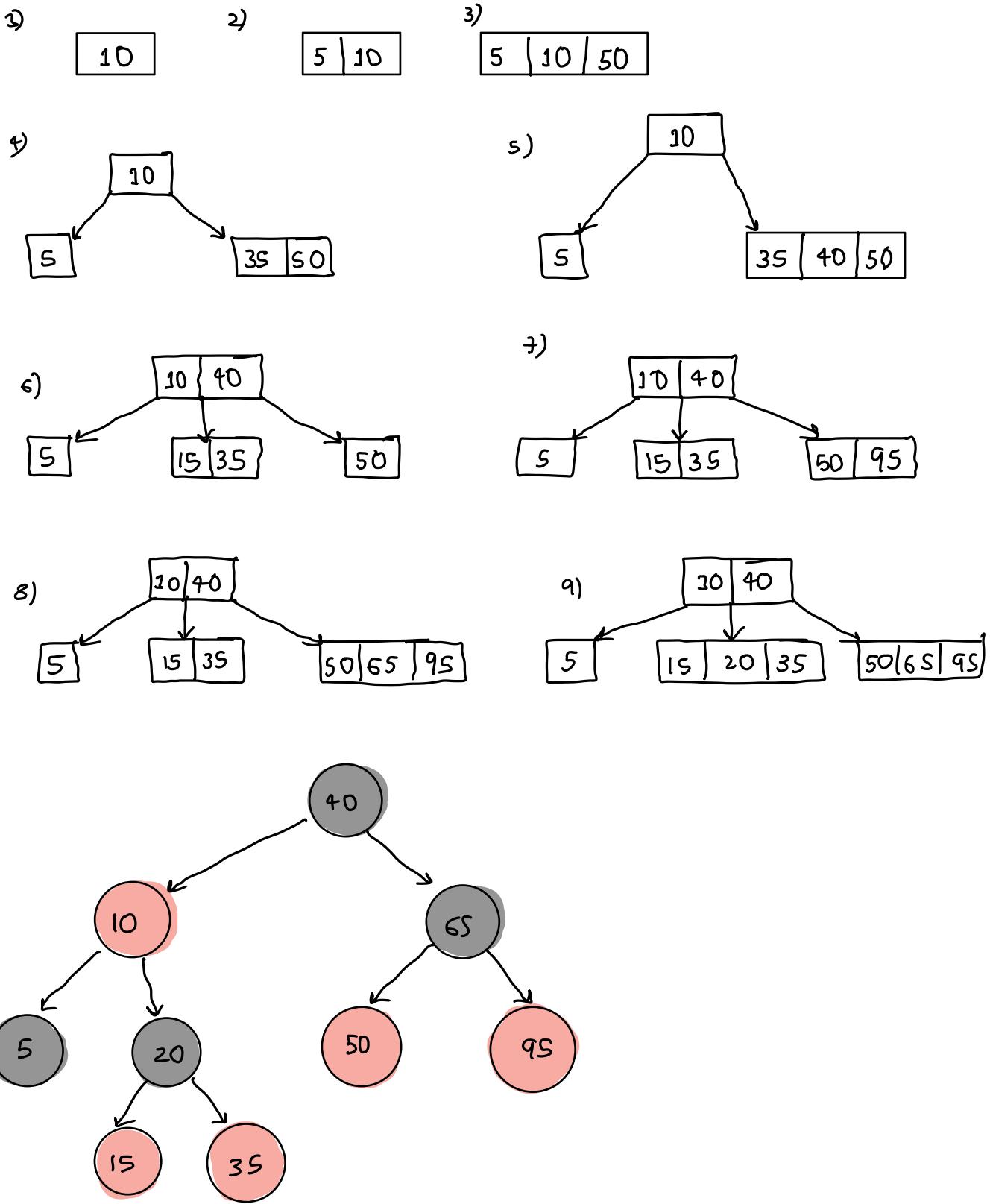
9)

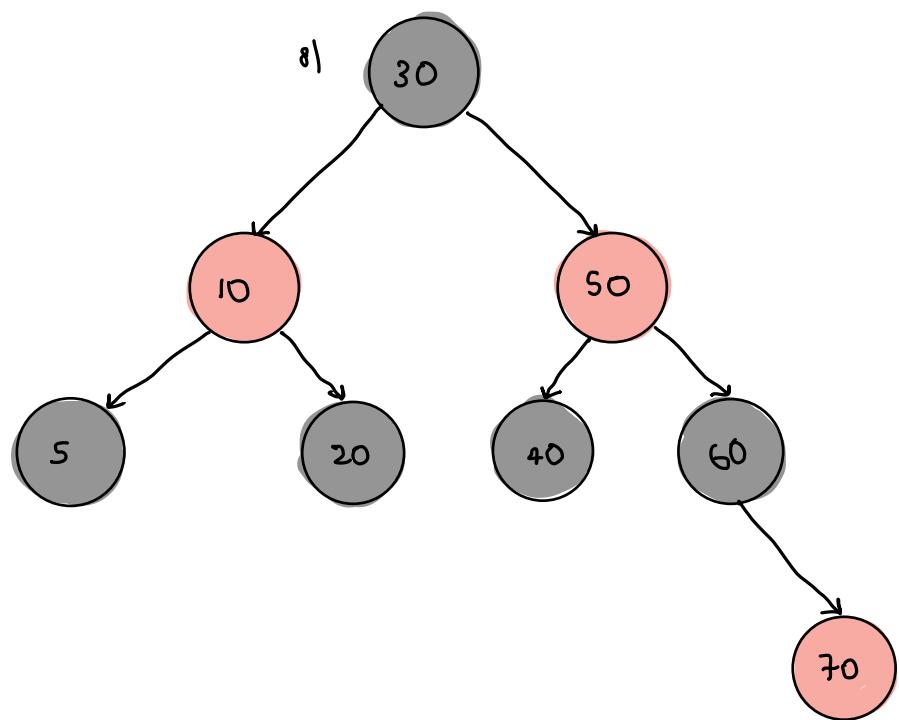
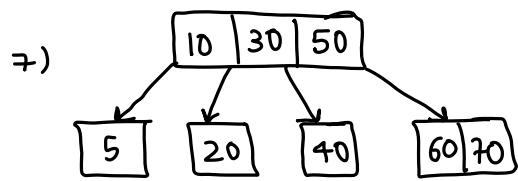
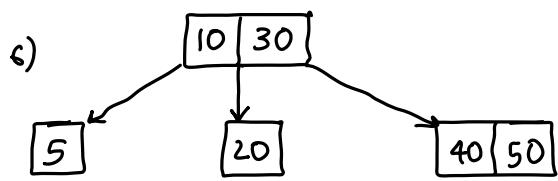
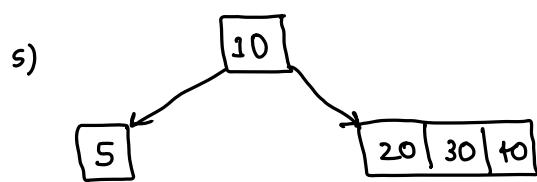
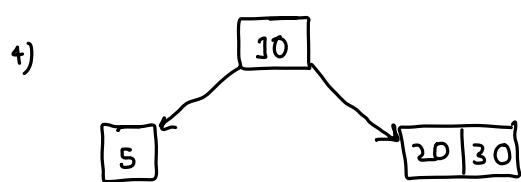


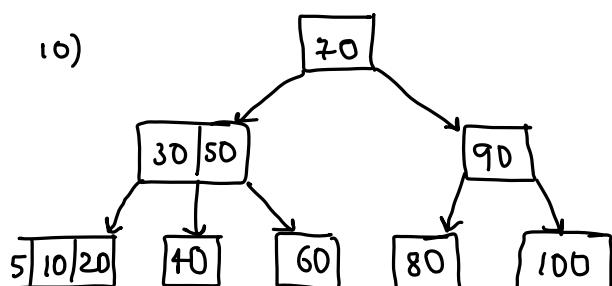
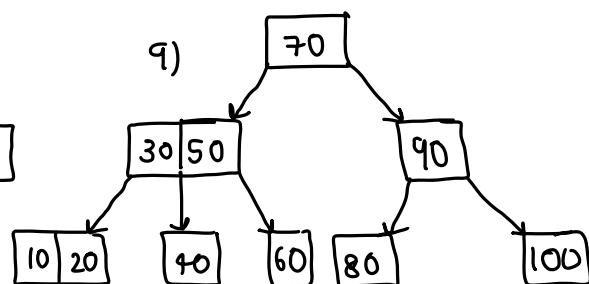
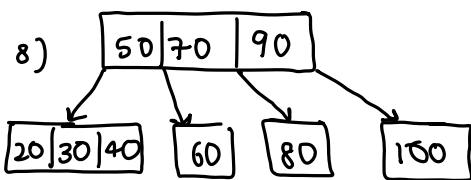
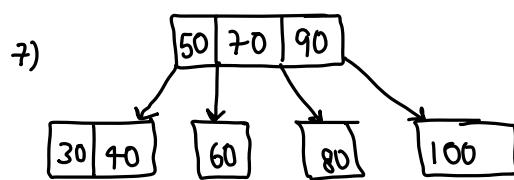
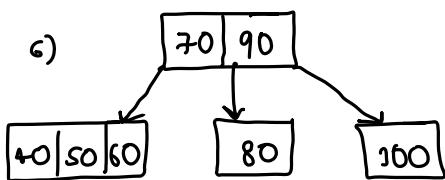
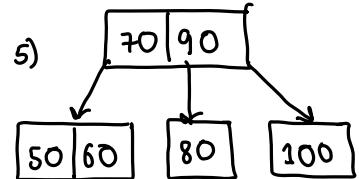
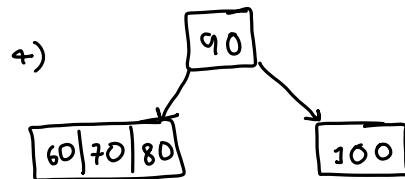
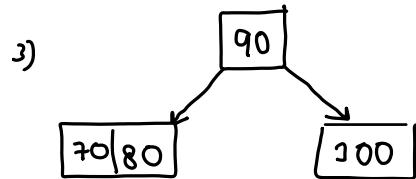
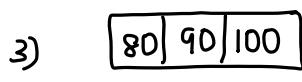
10)

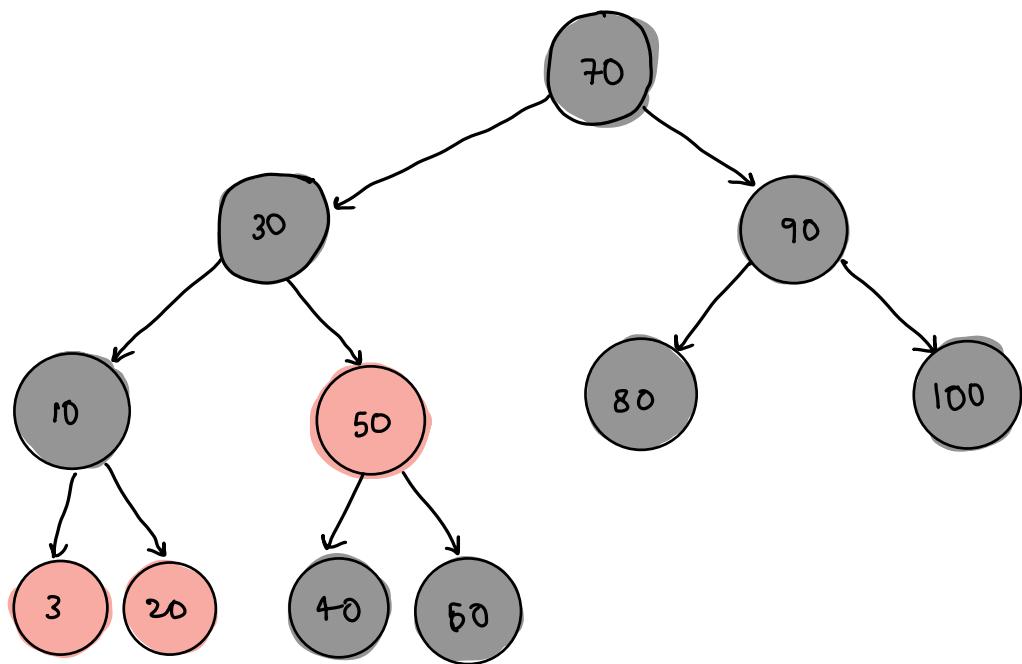












# B-Trees

1) 

10
----

2) 

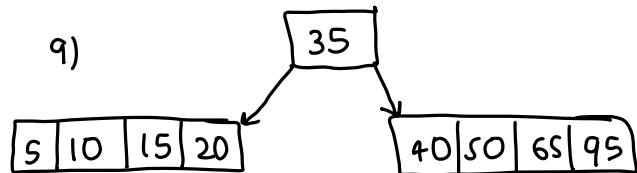
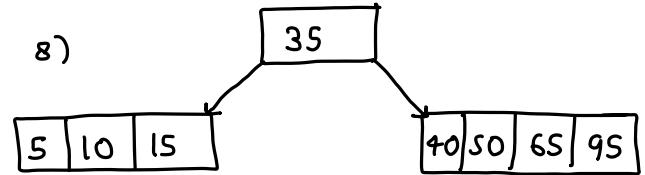
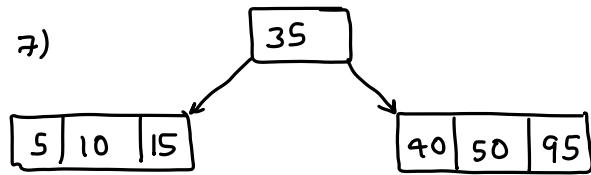
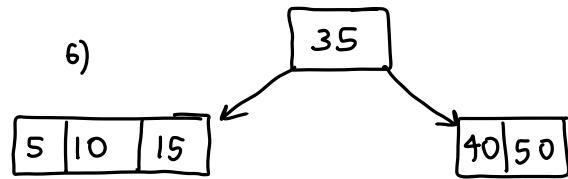
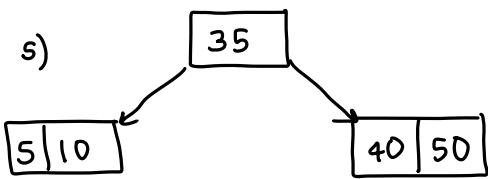
5	10
---	----

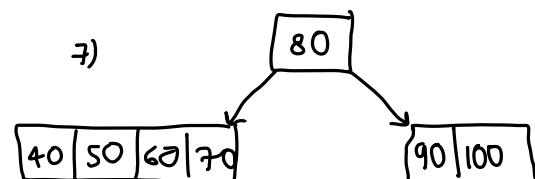
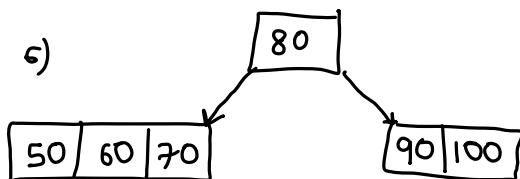
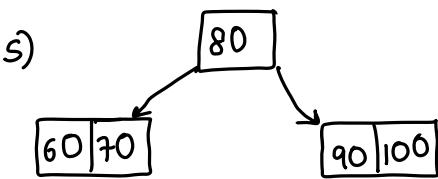
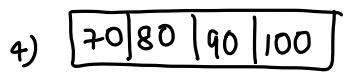
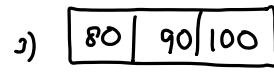
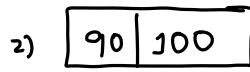
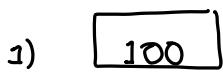
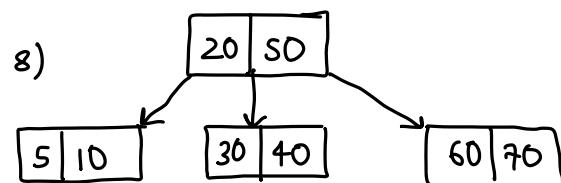
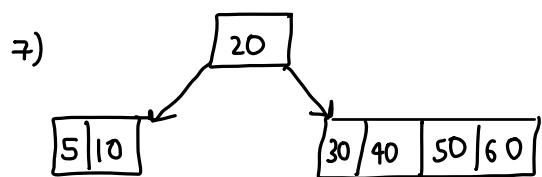
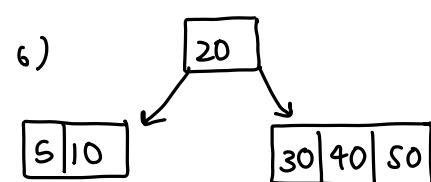
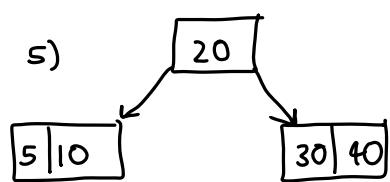
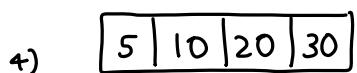
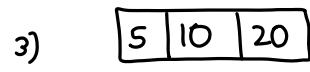
3) 

5	10	50
---	----	----

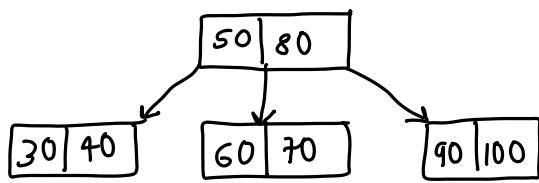
4) 

5	10	35	50
---	----	----	----

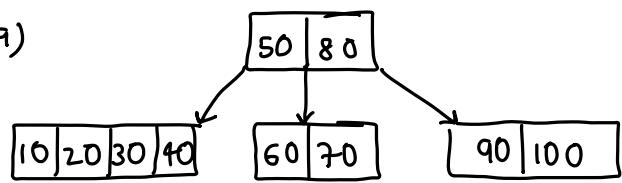




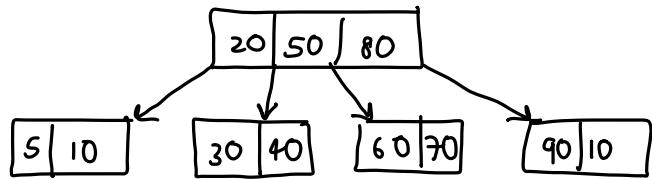
8)



9)



10)



# Reflection

## ↳ Comparison of height & compare with BST?

The trees have less height in general, due to the properties of self balancing especially the 2-3-4 and B-trees as they store a lot of data in each node making the height significantly shorter.

## ↳ compare complexity of algorithm?

The trees are complex in terms of operations like insertion and deletion because they are self balancing. Although the operations are the same they can said to be better because they are always balanced  $O(\log N)$  as for the normal Binary search it can degrade to  $O(N)$ .

## ↳ Comparison of understandability?

Understanding the trees is not a very tough task although all of them are harder to implement in comparison with Binary search tree since they have a certain set of rules that always need to satisfy.

2-3-4 and red black trees are almost of the same difficulty in terms of implementation however the B-trees are a bit more harder than these, both because of the complicated splitting mechanism. hence 2-3-4 and B-trees would be easier to implement.

## ↳ how in-order traversal would work on each tree?

The effect would be the same as on all advanced trees as it is on Binary search tree, all keys would be in ascending order.