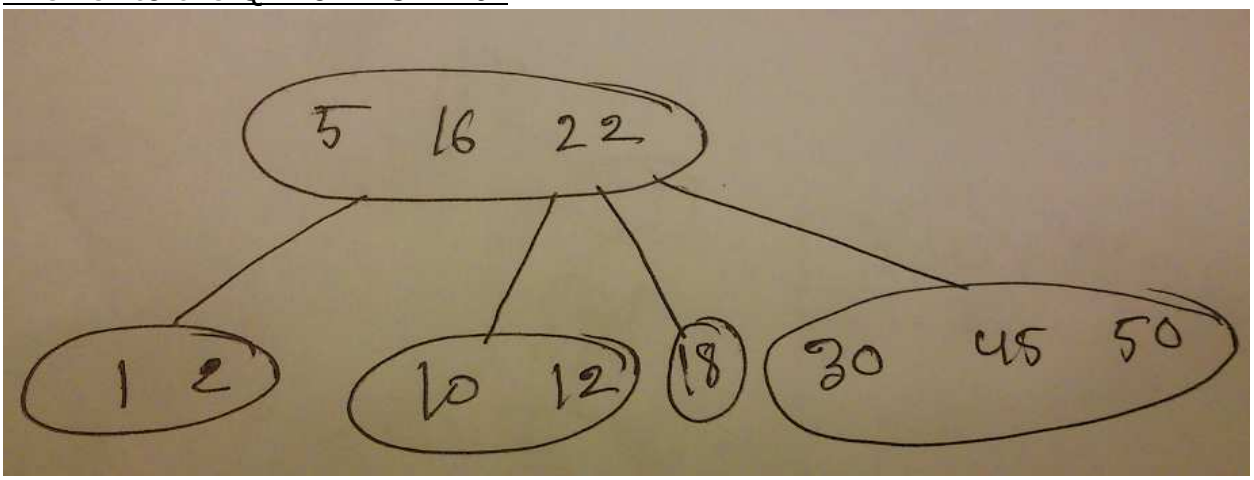
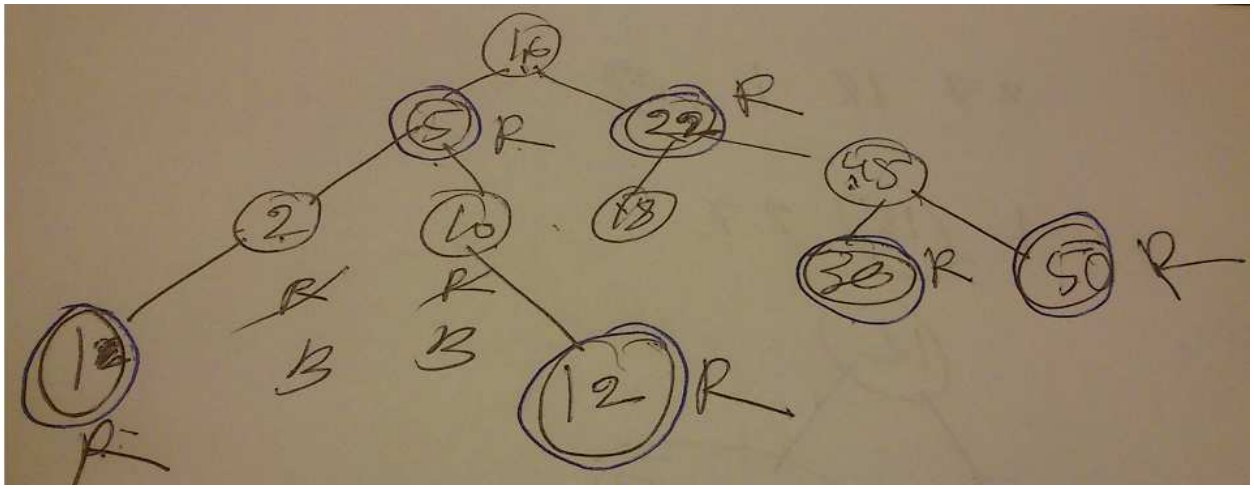
Name: Md Habibur Rony
Student ID: 984582
Weekday: Week 2- Day 9

Answer to the Q. No. R-3.11.a:



Answer to the Q. No. R-3.11.b:



Answer to the Q. No. R-3.14:

a.>> False, Because the root of the red-black tree must be the black and there is a chance that the root of the subtree may be red or black. So, we can't say that a subtree of a red-black tree is itself a red-black tree.

b.>> True, According to the property of the red-black tree we know that all external nodes are black. Again when we insert new node, its color is red.

c.>> False, Red-black tree can become transformed to (2, 4) tree. But it is not unique. The restructure of the nodes using split the positions of nodes can be changed.

d.>> False, (2, 4) tree can become transformed to Red-black tree. But it is not unique. The restructure of the nodes using split the positions of nodes can be changed.

Answer to the Q. No. C-3.10:

Algorithm FindAllInRange(D,k1, k2)
Input: D is anordered Dictionary.k1, k2 are range. so that k1<k<k2
Output: All element of dictionaries with range k1<k<k2
list<-newList

FindInRange (D, D.root(),k1,k2, list)
return list.Iterator()

```
Algorithm FindInRange (T, node, k1, k2, list)
    Input: Tree T, node of a tree, key k1, k2. list is the Sequence
    Output: All element of dictionaries with range k1<k<k2
    k <- T.key(node)
    if k1 <= k ^ k <= k2 then
        S.insertLast(D.findElement(k))
        findElements(T,T.leftChild(node),k1,k2)
        findElements(T,T.rightChild(node),k1,k2)
    else if k < k1 then
        return findElements(T,T.leftChild(node),k1,k2)
    return list
```