HACETTEPE UNIVERSITY

DEPARTMENT OF
COMPUTER ENGINEERING

BBM204 PROGRAMMING LAB.
QUIZ 2

Author
Kayla AKYÜZ
21726914
b21726914@cs.hacettepe.edu.tr

Advisor
Dr. Ahmet Selman BOZKIR
selman@cs.hacettepe.edu.tr

April 13,2020

Question 1

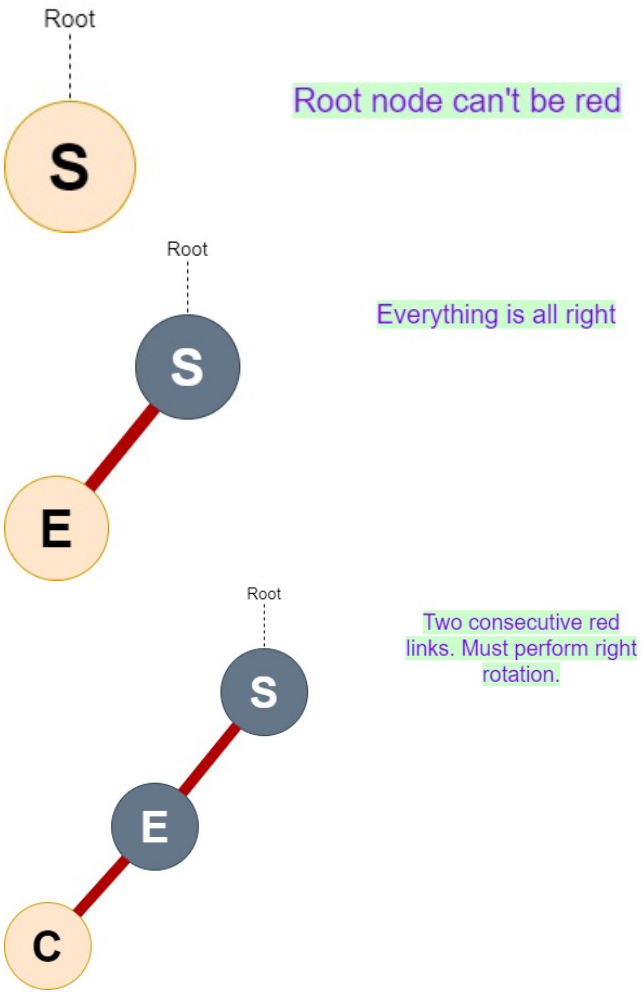
a) What is the level-order traversal of the BST drawn below?

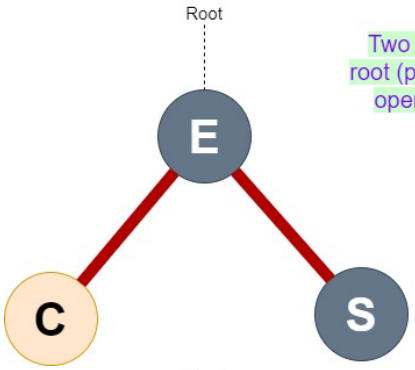
R->E->X->C->M->S->A->H->P

b) What mechanism should you employ to achieve the question above?

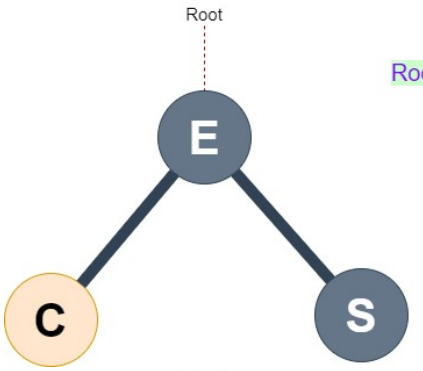
```
for level = 1 to depth(root):
   printNode(root, level)

printNode(root, level):
   if root is NULL then return;
   if level is 1, then
      print(node->value);
   else if level greater than 1, then
      printNode(node->left, level-1);
      printNode(node->right, level-1);
```

Question 2

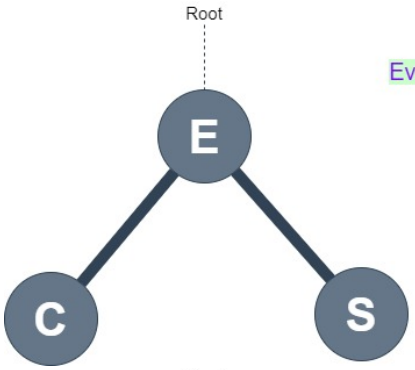




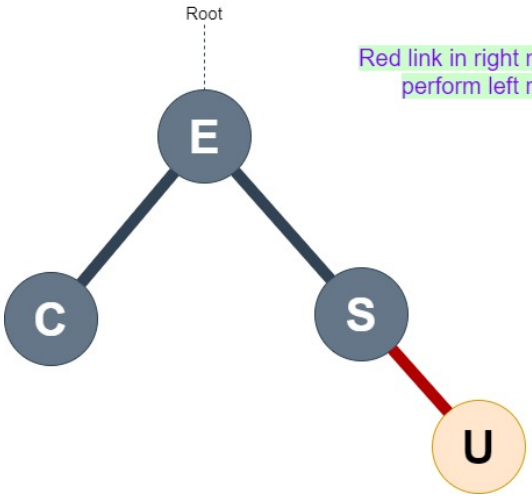
Two red links attached to root (parent) node. Flip color operation must be done



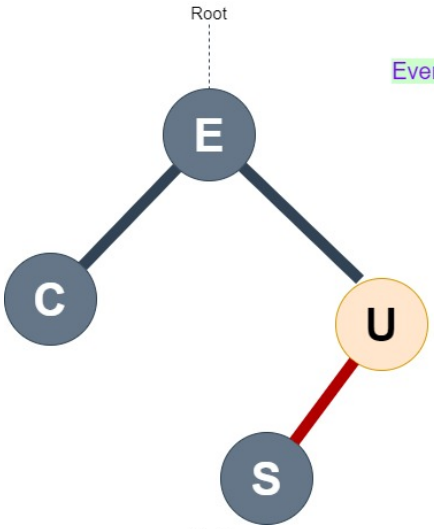
Root can not be red.



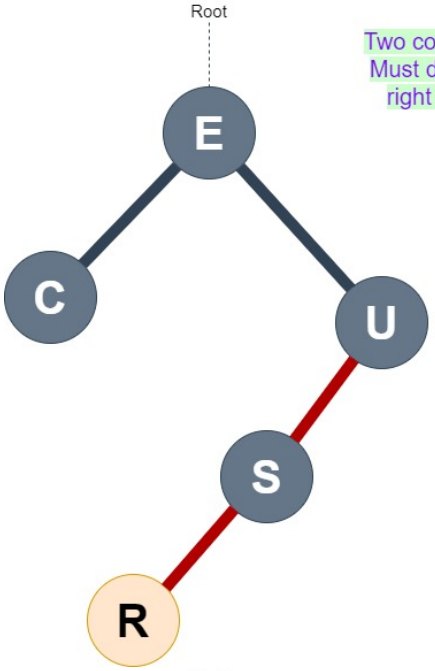
Everything is all right



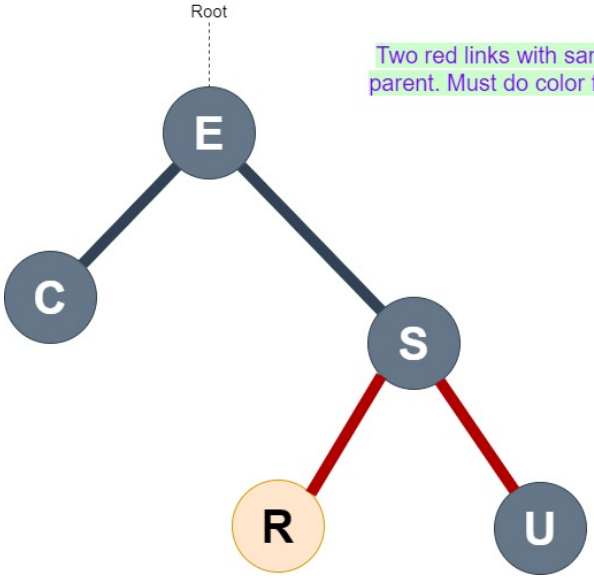
Red link in right node must perform left rotate.



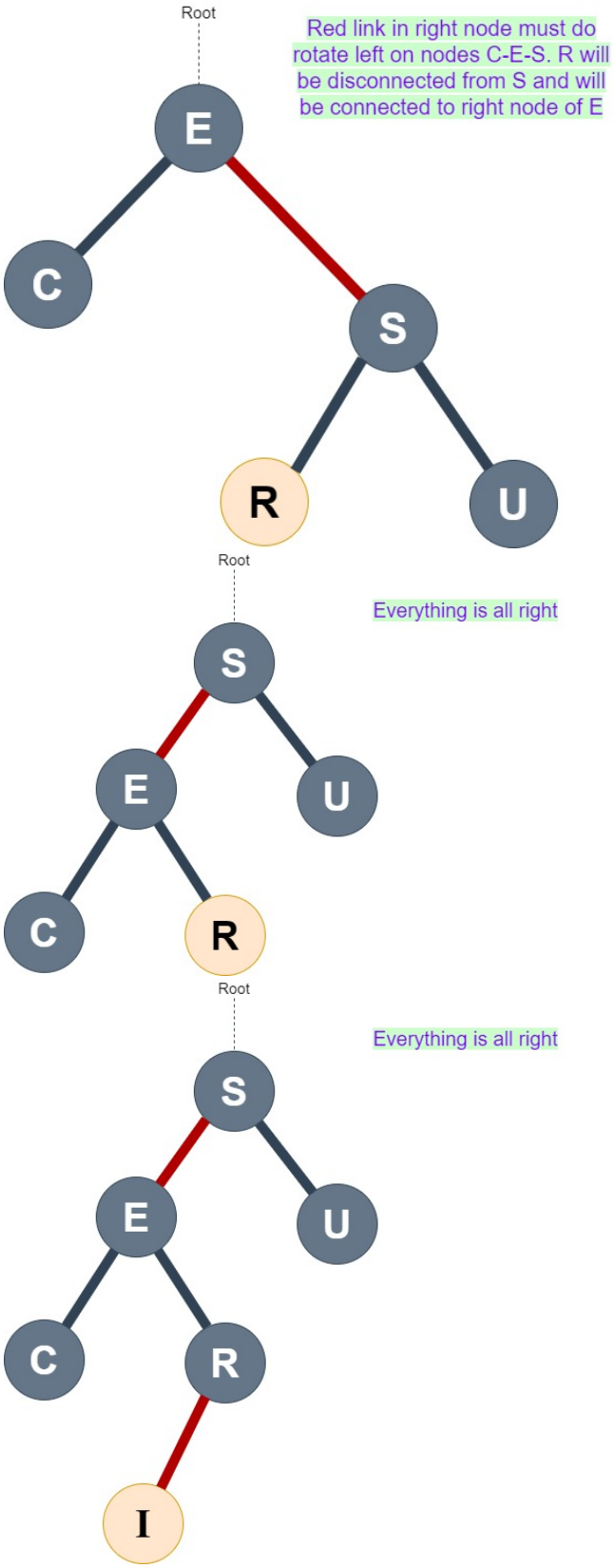
Everything is all right

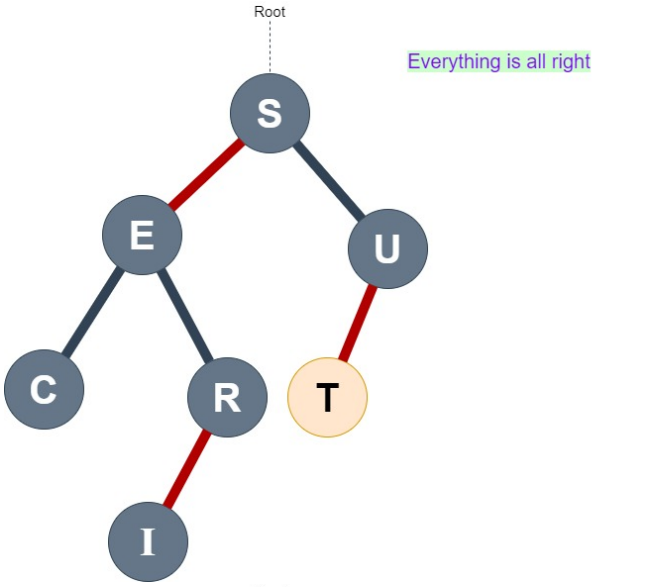


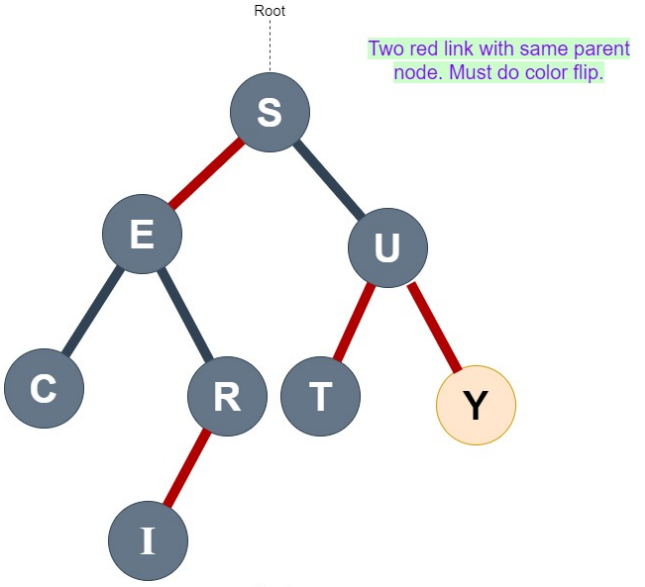
Two consecutive red links. Must do temporary rotate right on nodes R-S-U.

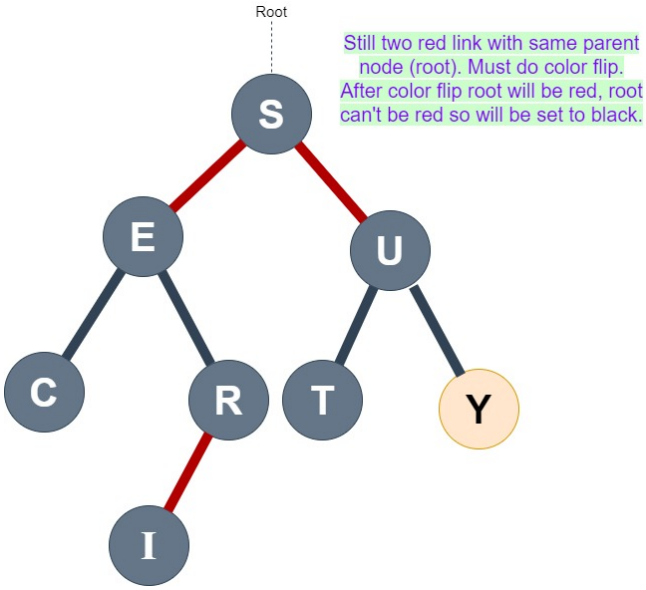


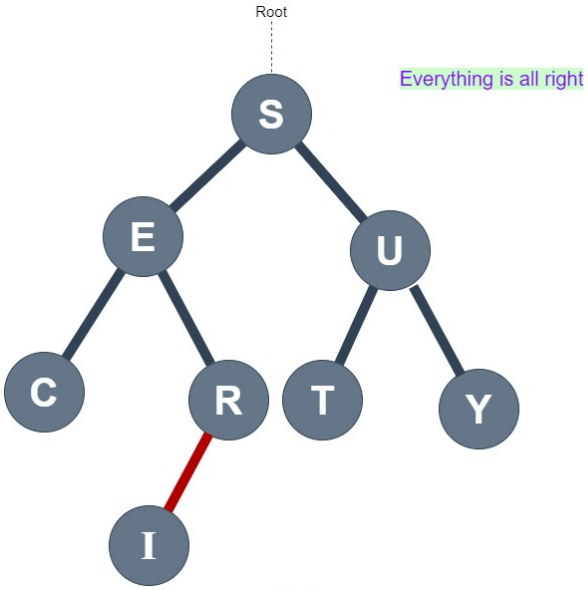
Two red links with same parent. Must do color flip.

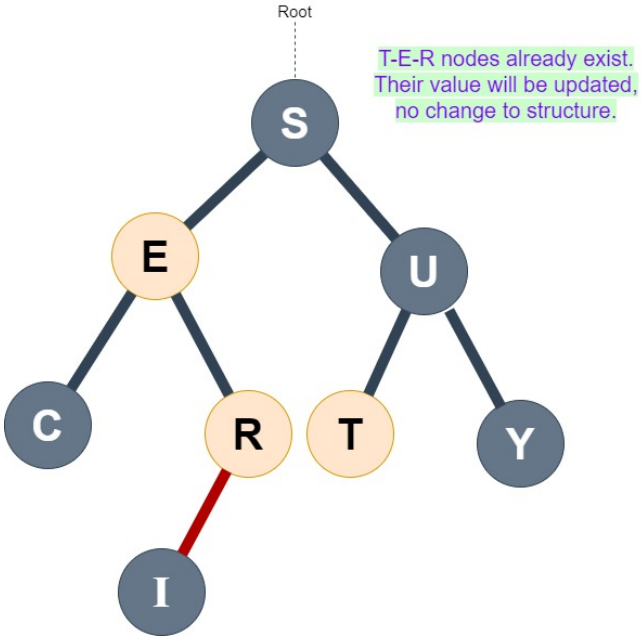

Root
Red link in right node must do rotate left on nodes C-E-S. R will be disconnected from S and will be connected to right node of E
E
C
S
R
U
Root
Everything is all right
S
E
U
C
R
Root
Everything is all right
S
E
U
C
R
I

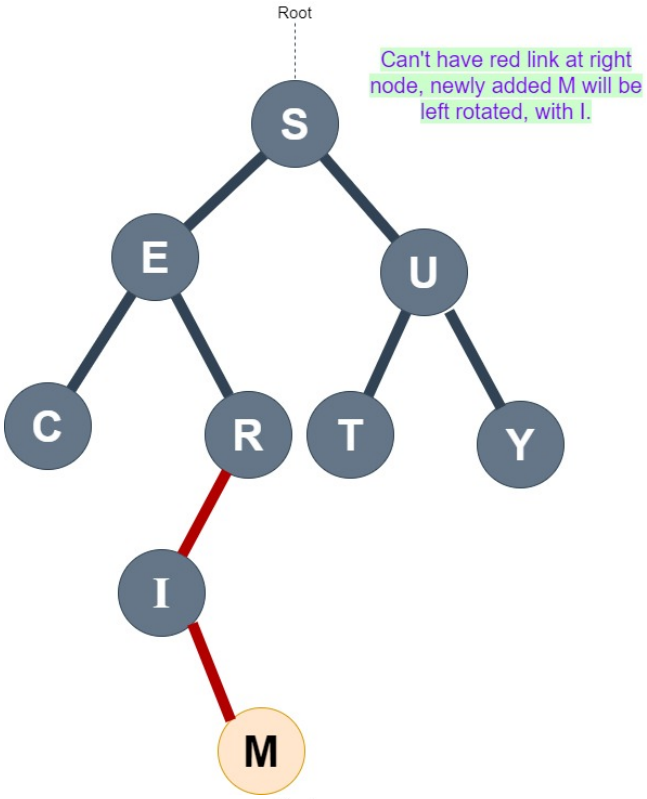

Root
Everything is all right
S
E
U
C
R
T
I

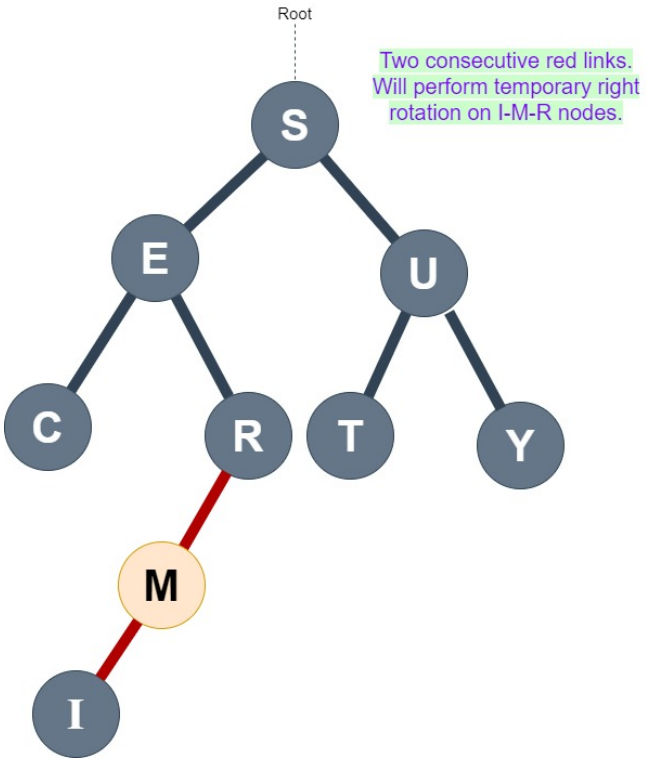

Root
Two red link with same parent node. Must do color flip.
S
E
U
C
R
T
Y
I

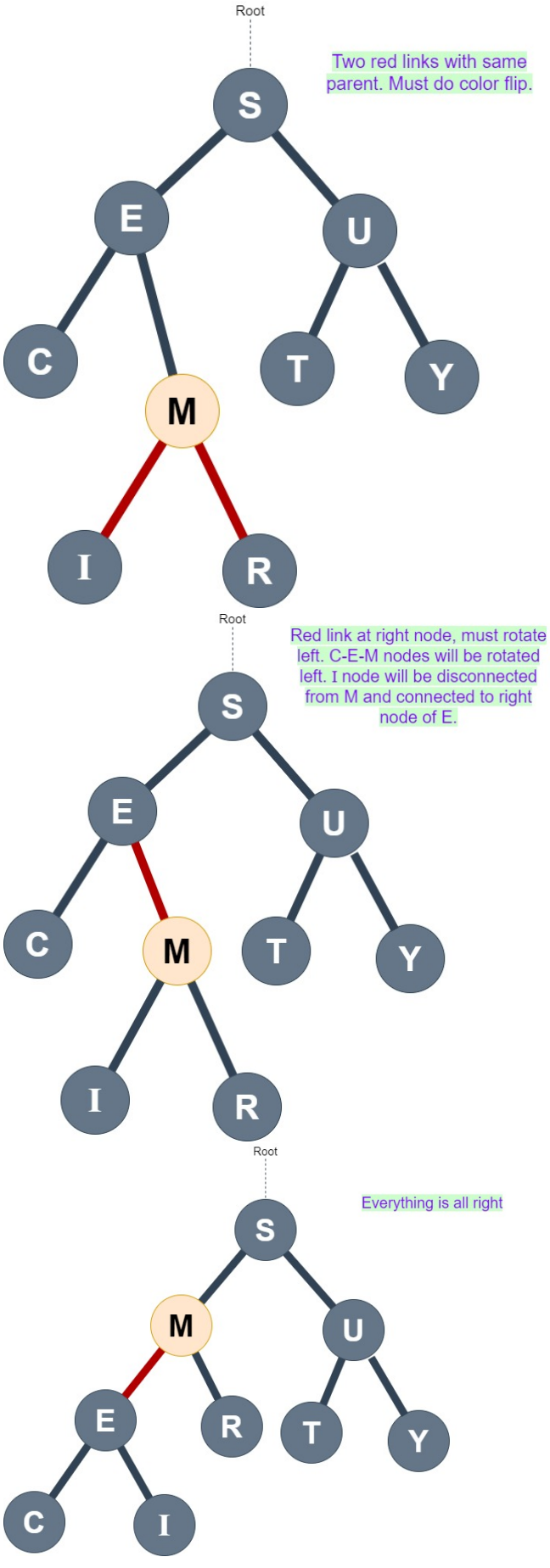

Root
Still two red link with same parent node (root). Must do color flip. After color flip root will be red, root can't be red so will be set to black.
S
E
U
C
R
T
Y
I

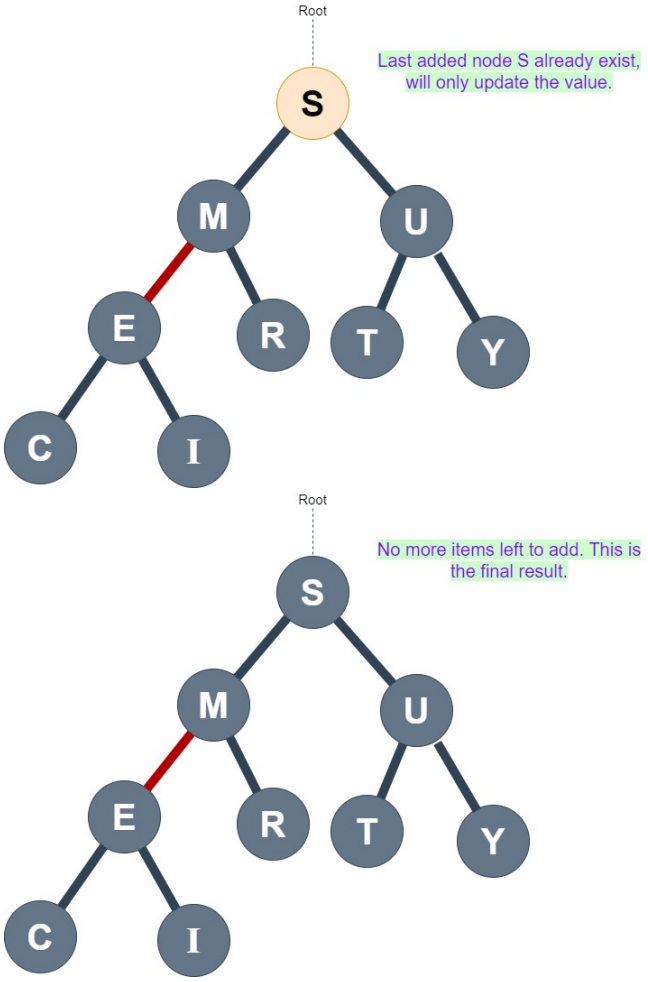

Root
Everything is all right
S
E
U
C
R
T
Y
I


Root
T-E-R nodes already exist.
Their value will be updated,
no change to structure.
S
E
U
C
R
T
Y
I

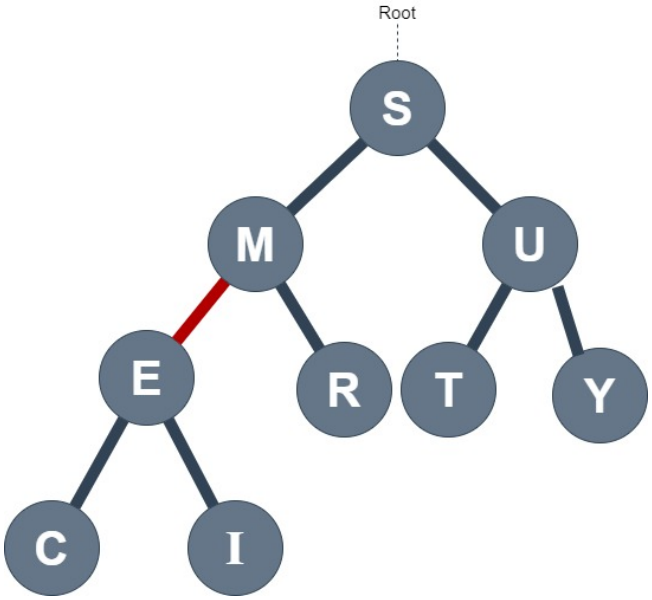

Root
Can't have red link at right node, newly added M will be left rotated, with I.
S
E
U
C
R
T
Y
I
M


Root
Two consecutive red links. Will perform temporary right rotation on I-M-R nodes.
S
E
U
C
R
T
Y
M
I


Root
Two red links with same parent. Must do color flip.
S
E
U
C
M
T
Y
I
R
Root
Red link at right node, must rotate left. C-E-M nodes will be rotated left. I node will be disconnected from M and connected to right node of E.
S
E
U
C
M
T
Y
I
R
Root
Everything is all right
S
M
U
E
R
T
Y
C
I



Final Left-Leaning Red-Black Tree.



Question 3

```
import java.lang.Comparable;

public class BST <Key extends Comparable<Key>, Value>{

    public Node root;

    public class Node{
        public Key key;
        public Value val;
        public Node left, right;
        public Node(Key key, Value val){
            this.key = key;
            this.val = val;
        }
    }

    public Value get(Key key)
    {
        while (root != null) {
            if (key.compareTo(root.key) > 0)
                root = root.right;
            else if (key.compareTo(root.key) < 0)
                root = root.left;
            else
                return root.val;
        }
        return root.val;
    }

    public void put(Key key, Value val) {
        Node curr = root;
        boolean override = false;
        Node parent = null;
        if (root == null) {
            root = new Node(key, val);
            return;
        }

        while (curr != null)
        {
            parent = curr;
            if (key.compareTo(curr.key) < 0) {
                curr = curr.left;
            }
            else if(key.compareTo(curr.key) > 0){
                curr = curr.right;
            }else if(key.compareTo(curr.key) == 0){
                override = true;
                break;
            }
        }
```

```
        if(override){
            curr.val = val;
        }else{
            if (key.compareTo(parent.key) < 0) {
                parent.left = new Node(key, val);
            }
            else {
                parent.right = new Node(key, val);
            }
        }
    }

    public static void main(String[] args) {
        BST program = new BST();
        program.start();
    }

    public void start() {
        BST bst = new BST();
        BST.Node[] nodes = {new BST.Node(2, 0), new BST.Node(7, 1), new BST.Node(4,
            2), new BST.Node(88, 3), new BST.Node(4, 4), new BST.Node(2, 5), new
            BST.Node(7, 6)};
        for (BST.Node node : nodes) {
            bst.put(node.key, node.val);
        }
        Object temp = bst.get(7);
        temp = (Integer) temp;
        System.out.println(temp);
    }
}
```

REFERENCES

Quiz Paper
LaTex Tutorials
Study Papers