```
Question 5
a)
COUNT(x.key)
      y = x \\ y points to the tree

      while y is not null
            if y.key == x.key
                  break
            else if y.key < x,key
                  y = y.right
            else
                  y = y.left

      y = y.right

      if y == null
            return

      else
            return y.size

\\ this should be O(log n) same time as RB-Traverals

b) you would have to set x.size = to size.x.right + size.x.left + 1,
after an insertion operation, this operation is O(1)

LEFT-ROTATE(T, x)
      y = x.right
      x.right = y.left

      if y.left is not T.nil
            y.left.p = x

      y.p = x.p

      if x.p == T.nil
            T.root = y

      elseif x == x.p.left
            x.p.left = y

      else x.p.right = y
            y.left = x
            x.p = y

RB-INSERT(T, z.key)
      y = T.nil
      x = T.root

      while x is not T.nil
            y = x

            if z.key < x.key
                  x = x.left
```

```
                else
                    x = x.right

        z.p = y

        if y == T.nil
            T.root = z

        else
            y.right = z

        z.left = T.nil
        z.right = T.nil
        z.color = RED
        z.size += x.right.size + x.left.size + 1

RB-INSERT-FIXUP(T, z)
    while z.p.color == RED
        if z.p == z.p.p.left
            y = z.p.p.right

            if y.color == RED
                z.p.color = BLACK
                y.color = BLACK
                z.p.p.color = RED
                z = z.p.p
            else if z == z.p.right
                z = z.p
                LEFT-ROTATE(T, z)
                z.p.color = BLACK
                z.p.p.color = RED
                RIGHT-ROTATE(T, z.p.p)
        else
            z = z.p
            RIGHT-ROTATE(T, z)
            z.p.color = BLACK
            z.p.p.color = RED
            LEFT-ROTATE(T, z.p.p)

    z.size += x.right.size + x.left.size + 1

    T.root.color = BLACK
```