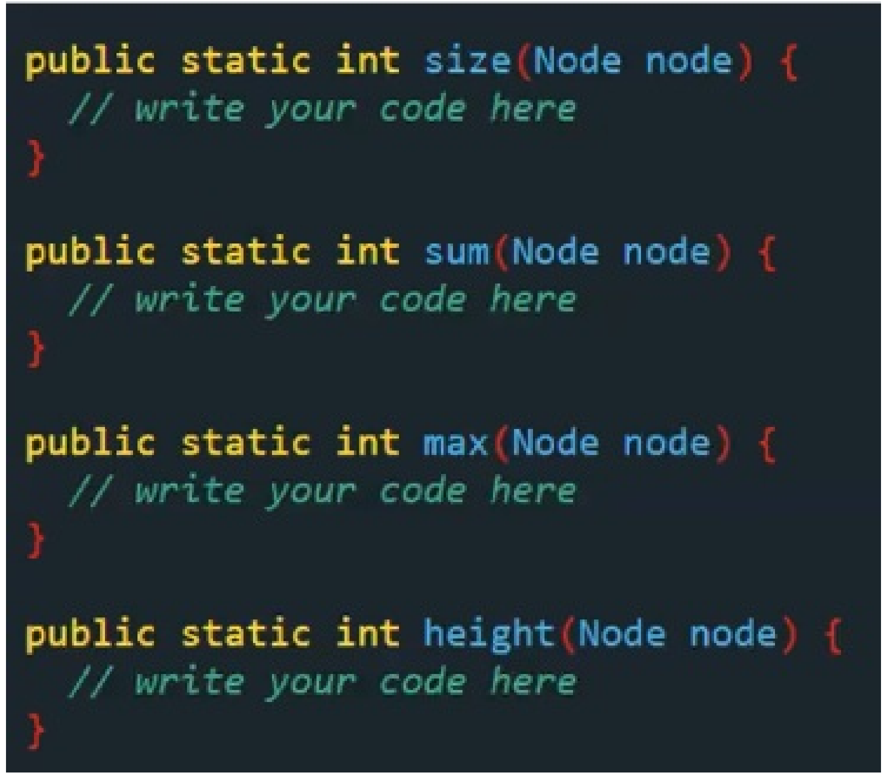
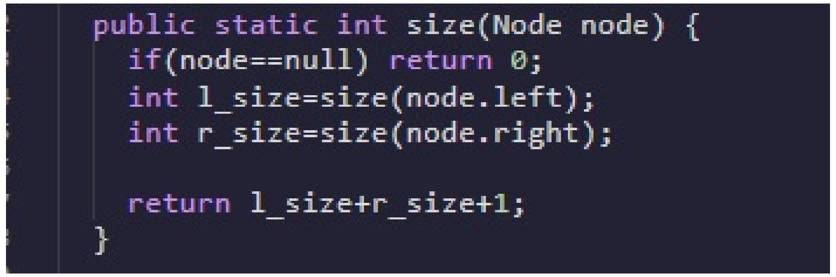
Complete these 4 function for a BT

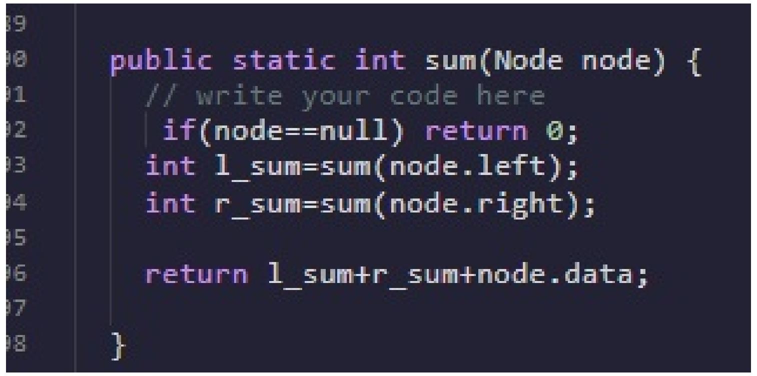


```java
public static int size(Node node) {
  // write your code here
}

public static int sum(Node node) {
  // write your code here
}

public static int max(Node node) {
  // write your code here
}

public static int height(Node node) {
  // write your code here
}
```

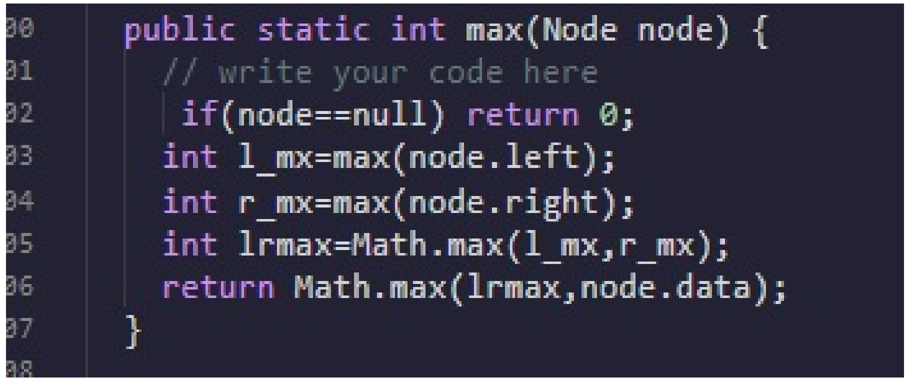
Solution



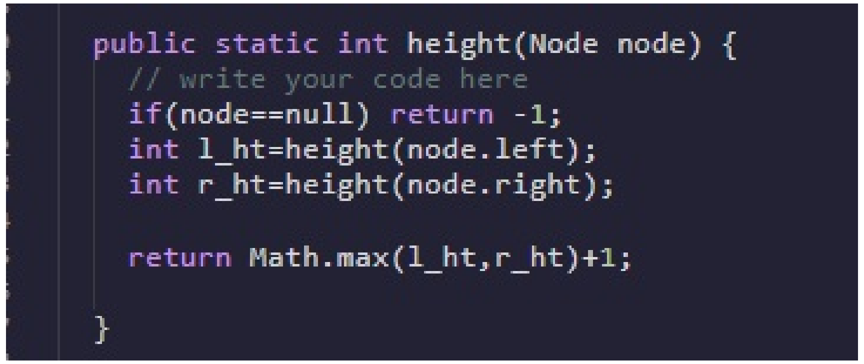
```java
public static int size(Node node) {
    if(node==null) return 0;
    int l_size=size(node.left);
    int r_size=size(node.right);

    return l_size+r_size+1;
}
```



```java
public static int sum(Node node) {
    // write your code here
     if(node==null) return 0;
    int l_sum=sum(node.left);
    int r_sum=sum(node.right);

    return l_sum+r_sum+node.data;

}
```



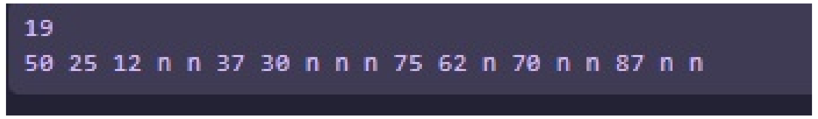
```java
public static int max(Node node) {
    // write your code here
     if(node==null) return 0;
    int l_mx=max(node.left);
    int r_mx=max(node.right);
    int lrmax=Math.max(l_mx,r_mx);
    return Math.max(lrmax,node.data);
}
```



```java
public static int height(Node node) {
    // write your code here
    if(node==null) return -1;
    int l_ht=height(node.left);
    int r_ht=height(node.right);

    return Math.max(l_ht,r_ht)+1;

}
```

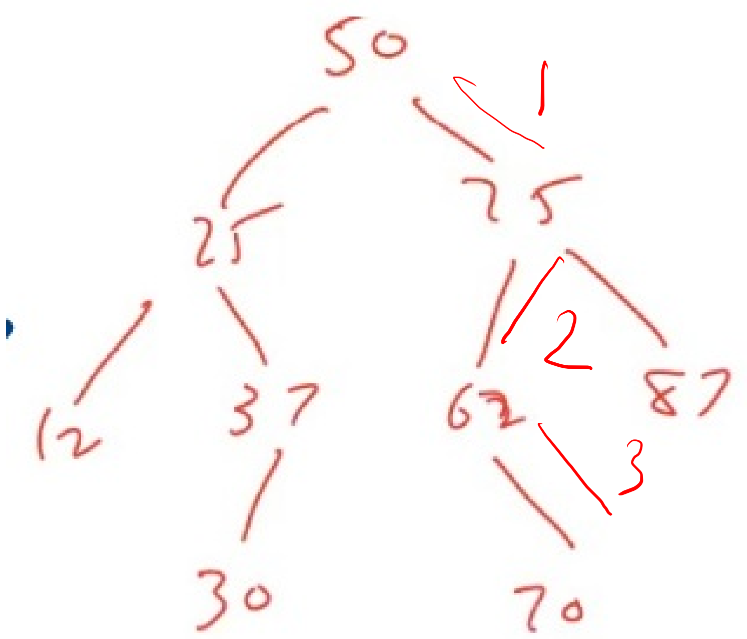
Height mai BC -1 isliye lia kyoki Edges ke hisab se height hai for height

See on next page



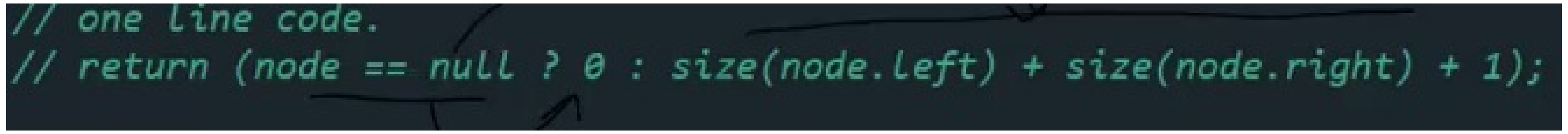
```
19
50 25 12 n n 37 30 n n n 75 62 n 70 n n 87 n n
```

height acc to Q is 3
So Edges ki height li hai



if you put if(node==null) return 0; it returns height as 4

So height agr Edges ke hisab se chaie toh BC mei return −1; if nodes ke hisab se chaie return 0;



```
// one line code.
// return (node == null ? 0 : size(node.left) + size(node.right) + 1);
```

In tree we can have such one line code of these basic functions

for sum we can write

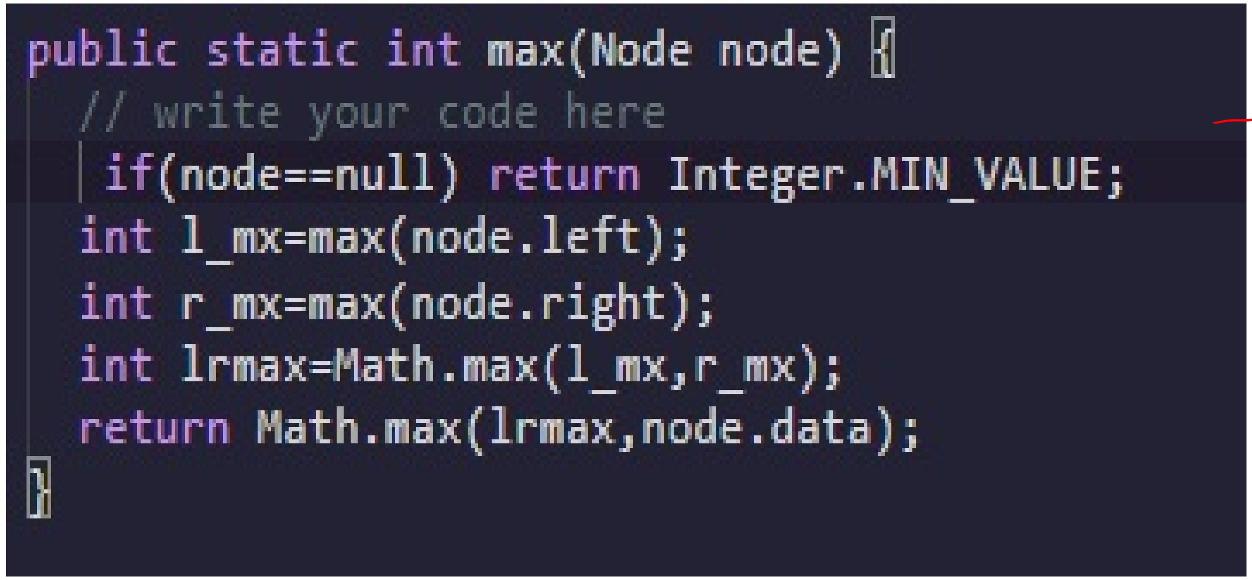
```
return (node == null) ? 0 : sum(node.left) + sum(node.right)
                     + node.data;
```

Similarly for height

```
return (node == null) ? -1 : height(node.left) + height(node.
                     right) + 1;
```

Dont try to write in one line for all of the code of Binary Tree

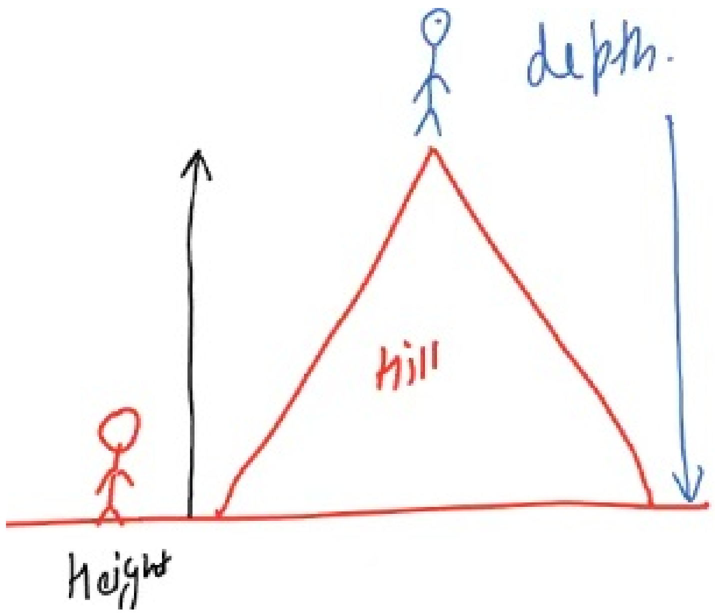
Our Max f^n BC was wrong as if we have all negative values then it is going to return 0; So our BC must be if(node == null) return Integer. MIN_VALUE;



```
public static int max(Node node) {
  // write your code here
   if(node==null) return Integer.MIN_VALUE;
  int l_mx=max(node.left);
  int r_mx=max(node.right);
  int lrmax=Math.max(l_mx,r_mx);
  return Math.max(lrmax,node.data);
}
```

→ This is correct code

Height vs Depth



Person from below says its height

Person from above the hill says its depth.

$|height| = |depth|$

So height & depth are one & same thing

Height is from deepest leaf to root in BT

Depth is from root to deepest leaf of BT

$height|_{node} = height|_{Edge} + 1$ → Point to Remember ** if (root == null)

$$\text{height}\big|_{Edge} = \,?$$

height By default Edge ke hisab se nikalte hai.

we know if (root == null) $\text{height}\big|_{node} = 0;$

Also we know $$\underbrace{h\big|_{node}}_{0} = h\big|_{Edge} + 1;$$

$$0 = h\big|_{edge} + 1;$$

$$\boxed{h\big|_{Edge} = -1} \longrightarrow$$ Thatshow Edge wale BC aaya