

Tilt of node
 $|L\ tree\ sum - r\ tree\ sum|$

n Tilt of Tree
 \leq tilt of a node

expectation tree sum
 tilt

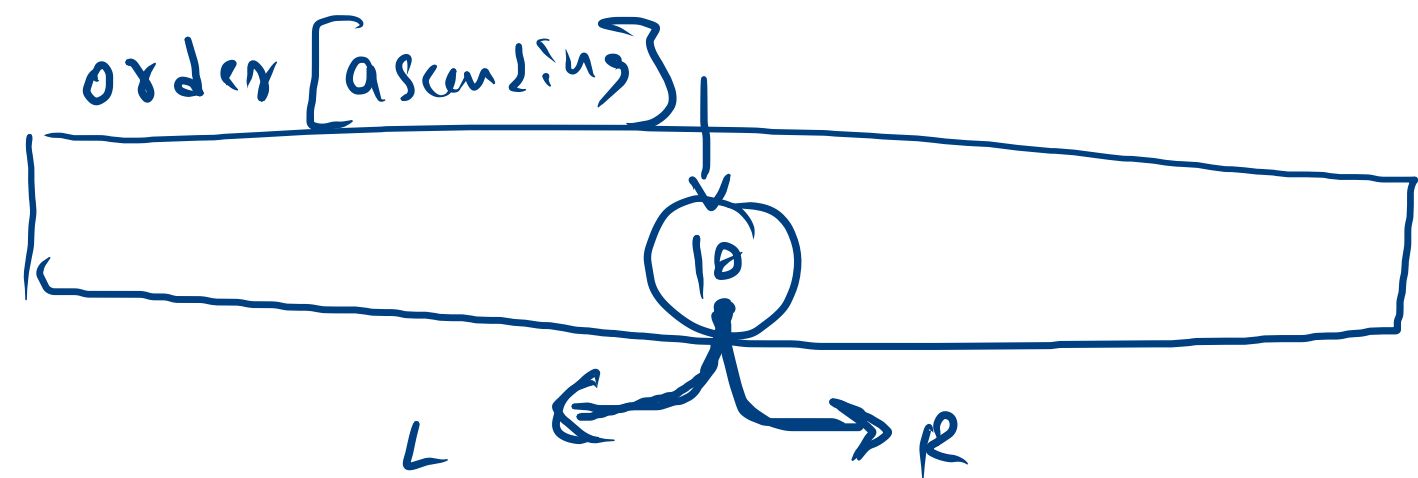
$10 + 6 + 14$

$16 + 14$
 30

Tree

= Binary Search

Binary Search Tree



find n

is bst \rightarrow false
min \rightarrow 12, max \rightarrow 100

isbst T/F
max 37

50

isbst T
min 49
max 100

isbst T
min 12
max 37

isbst T
min 12
max 12

12

isbst T
min 32
max 37

37

49

100

75

25

32

bst

isbst T
min
max $-\infty$

isbst T
min $+\infty$
max

isbst
min
max

bst

Logic

1 node

return

bst

isbst - t/f
min
max

if (left.data < node.data & right.data < node.data)

Left tree bst

Right tree bst

bst
min
max

```
public static BSTPair isBst(Node node){
    if(node == null){
        BSTPair nbst = new BSTPair();
        nbst.isbst = true;
        nbst.min = Integer.MAX_VALUE;
        nbst.max = Integer.MIN_VALUE;
        return nbst;
    }

    BSTPair lbst = isBst(node.left);
    BSTPair rbst = isBst(node.right);

    BSTPair mbst = new BSTPair();

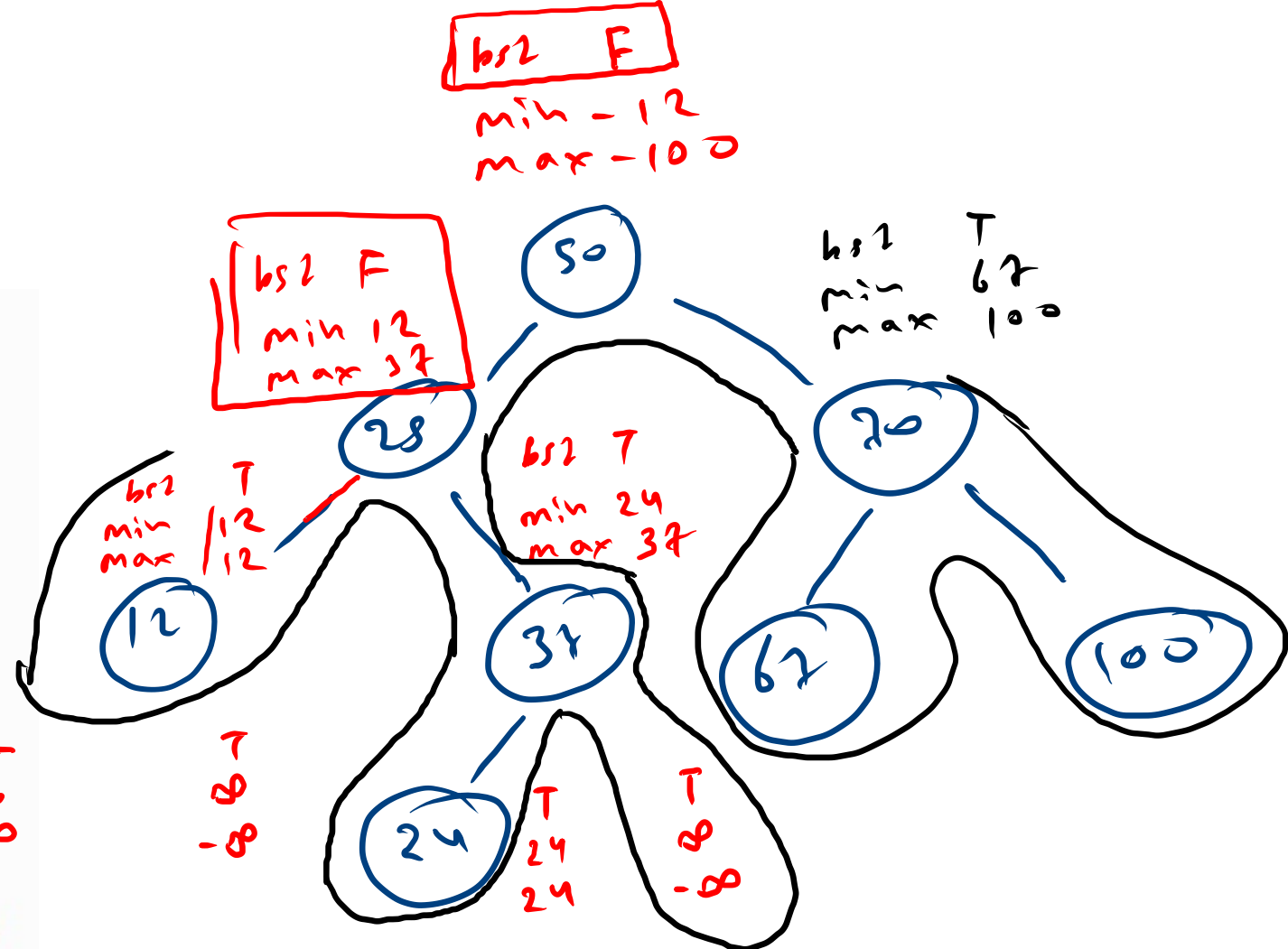
    mbst.min = Math.min(node.data, Math.min(lbst.min, rbst.min));
    mbst.max = Math.max(node.data, Math.max(lbst.max, rbst.max));

    if(node.data > lbst.max && node.data < rbst.min && lbst.isbst && rbst.isbst){
        mbst.isbst = true;
    }else{
        mbst.isbst = false;
    }

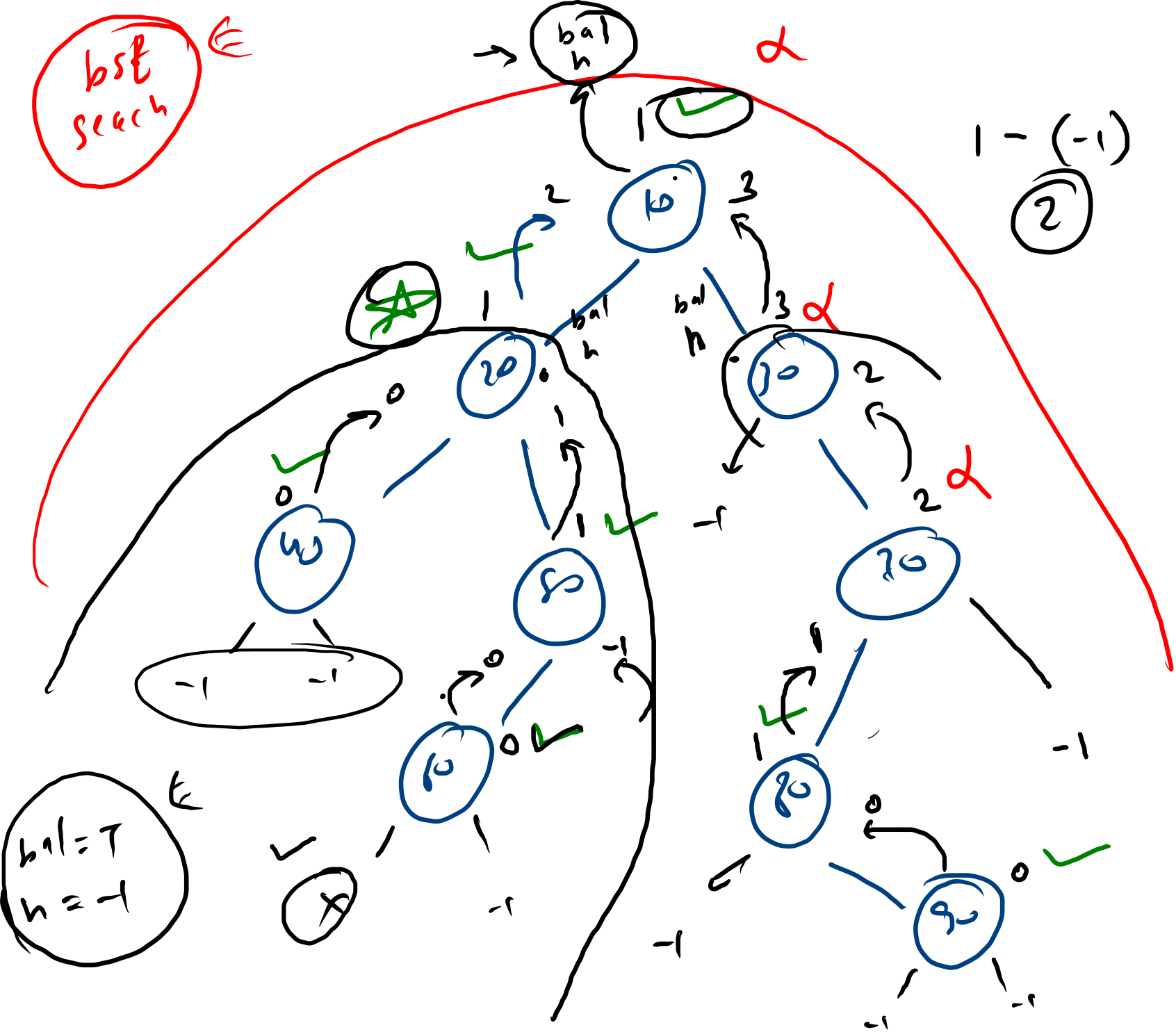
    return mbst;
}
```

if (1bst.isbst == false)
return

T
-∞



bst search



balance node

$|L.height - R.height| \leq 1$

```
class Node {
    int height;
    boolean balanced;
}
```

true/false

- node balance ≤ 1
- Left balanced == true
- right balance == true

75 @ 5

75, 5

multiple subtree bst ✓

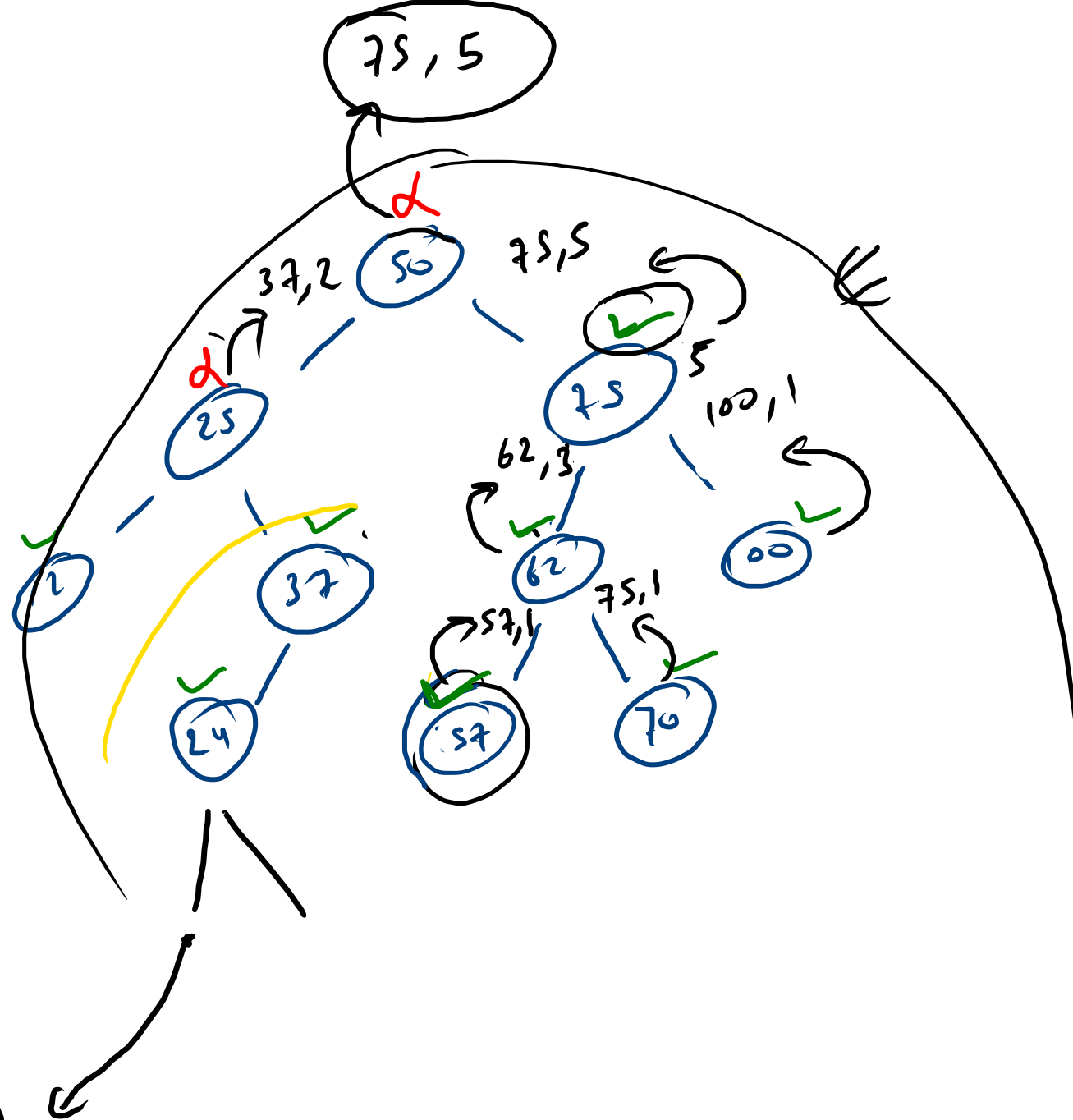
size \rightarrow L.size + r.size + 1

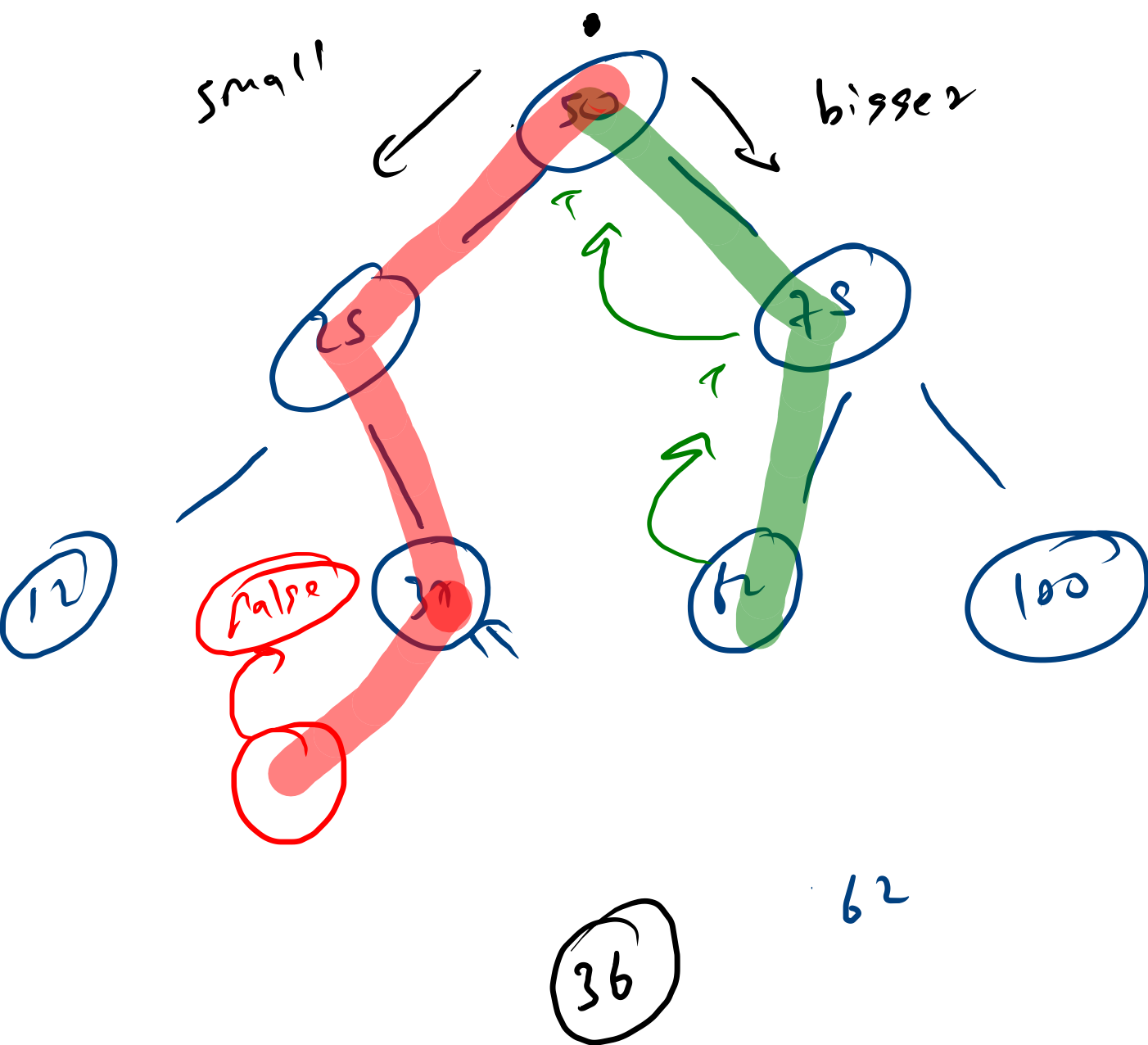
Question ?

Largest st
bst ✓

count nodes

75 @ 5





size 7
min → 12
max 100
sum → 2

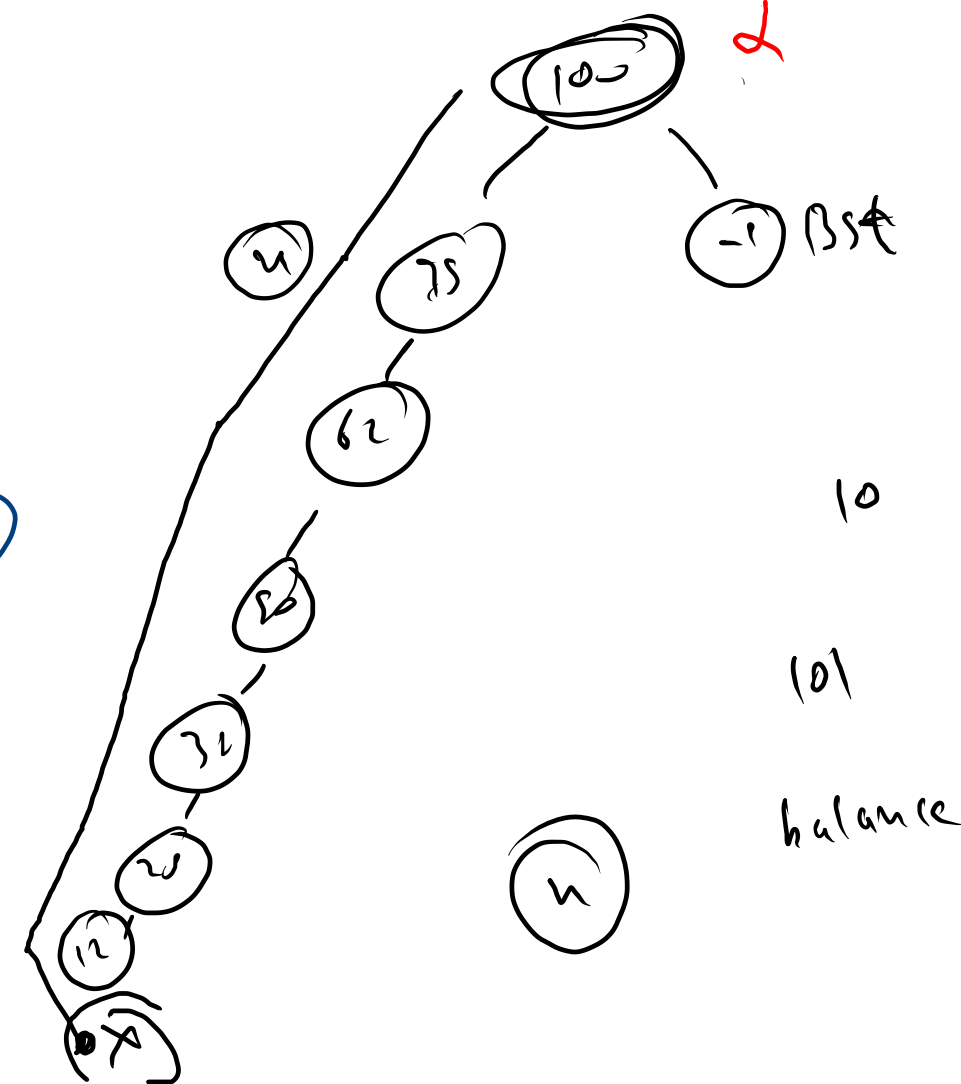
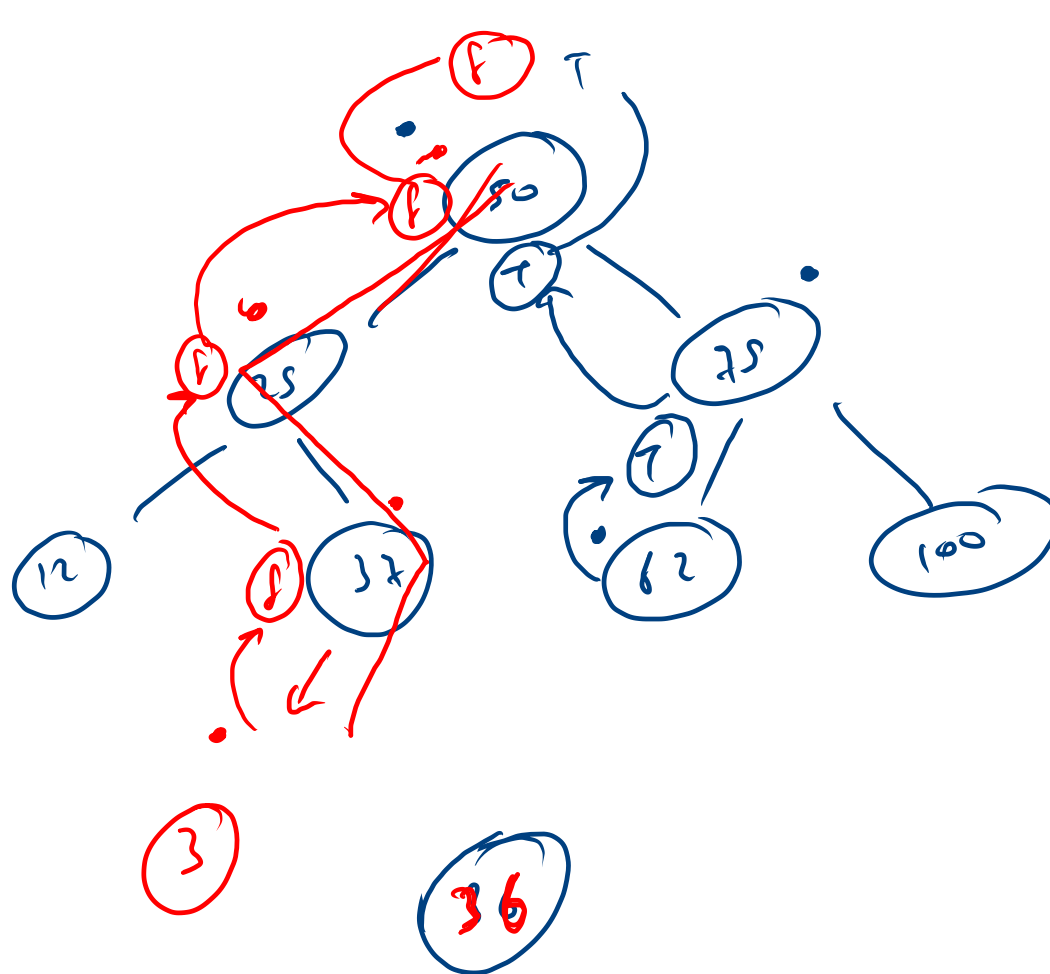
find	37	→	true
	38	→	false


```

public static boolean find(Node node, int data){
    if(node == null){
        return false;
    }
    32 < 36
    if(node.data < data){
        return find(node.right, data);
    }else if(node.data > data){ 32 > 36
        return find(node.left, data);
    }else{
        return true;
    }
}

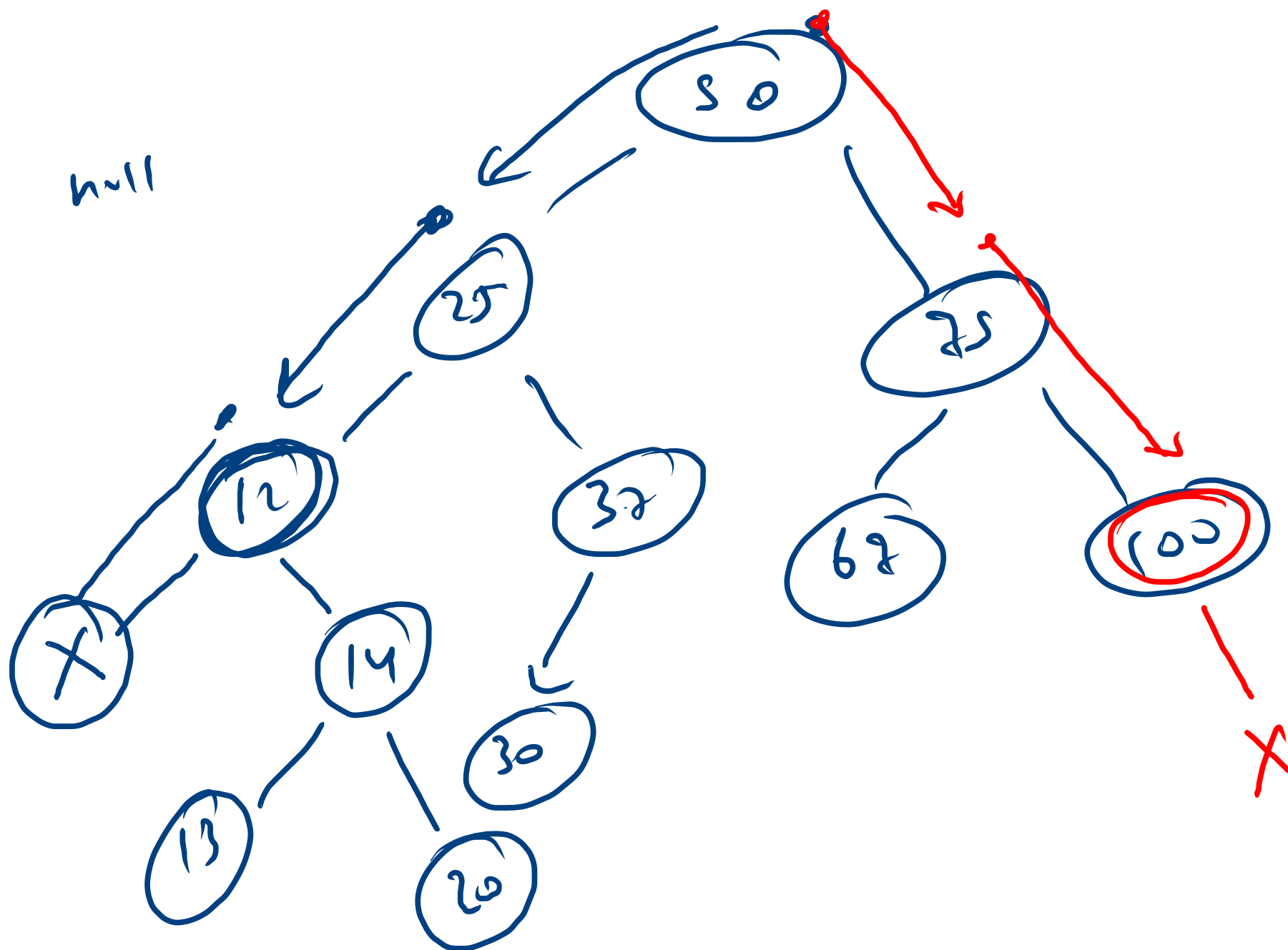
```

height



min
left
min
null

max



structure

value

size

sum

max

min

find

\Rightarrow BST

same

same

α

α

α

BST

BSTs

same

same

✓

✓

✓