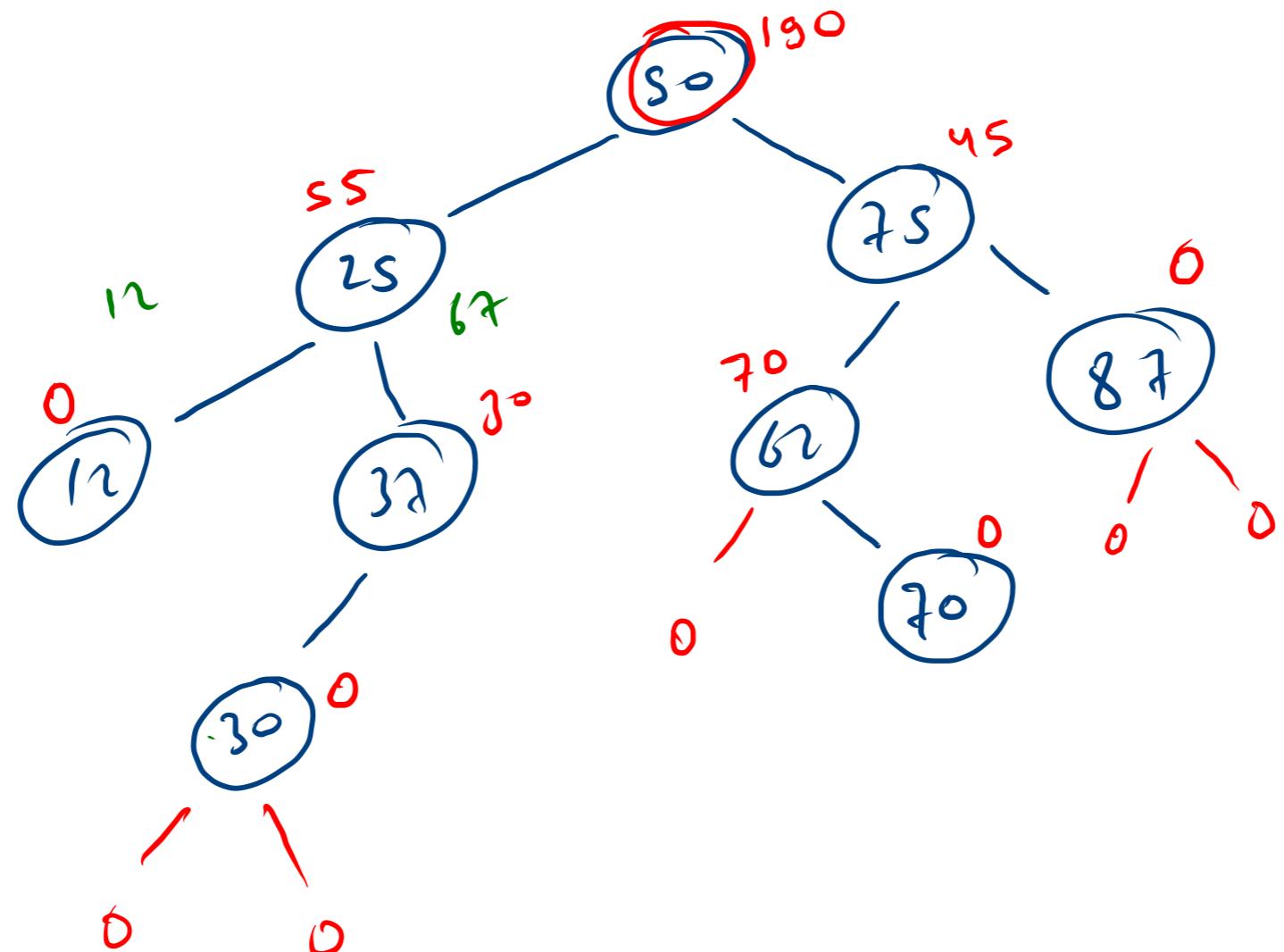


$$\begin{array}{r}
 20 \\
 + 62 \\
 \hline
 82 \\
 - 132 \\
 \hline
 - 50 \\
 \end{array}$$

$$\begin{array}{r}
 67 \\
 + 12 \\
 \hline
 55
 \end{array}$$

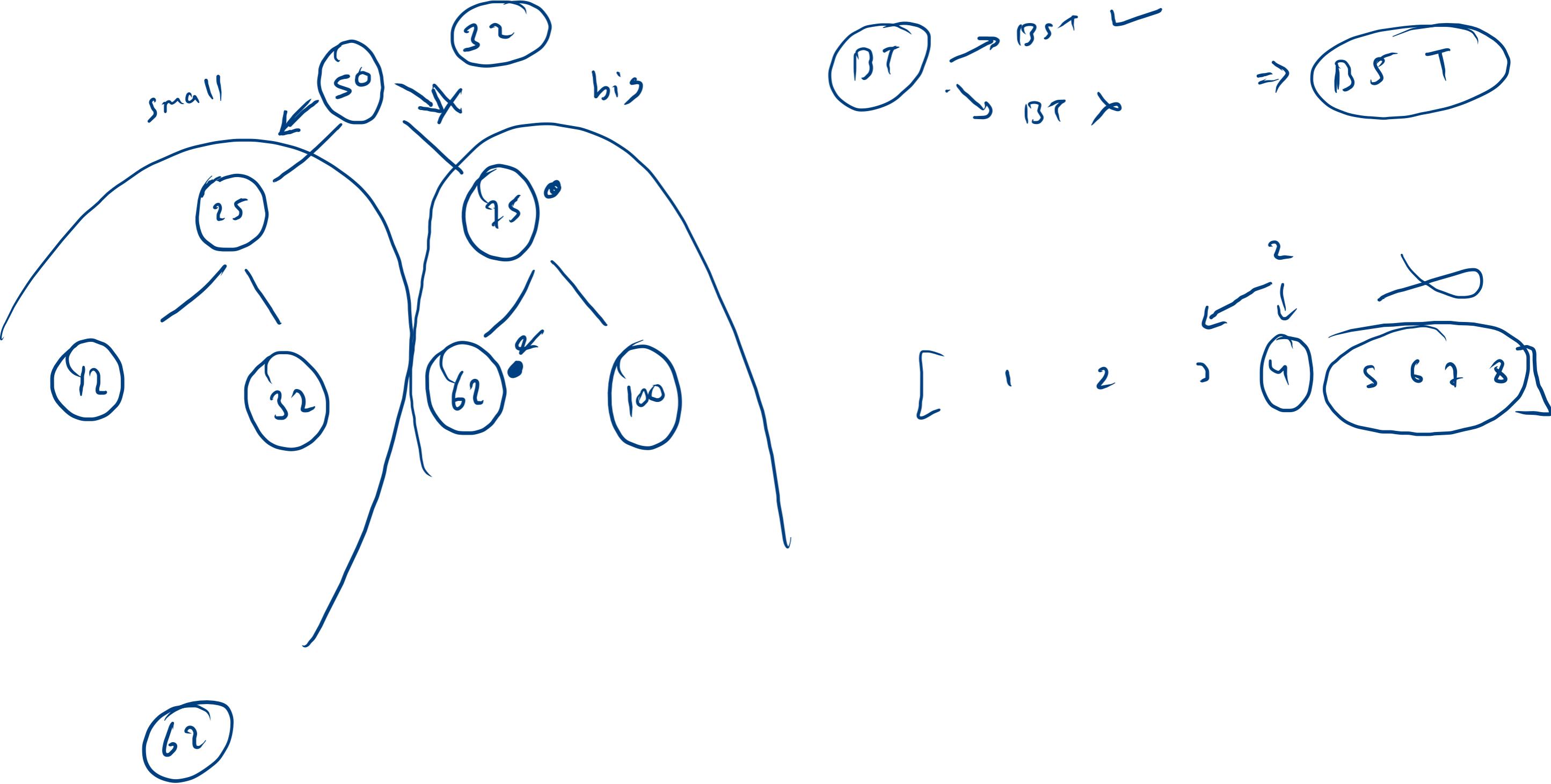
$$(12 + 67 + 25)$$

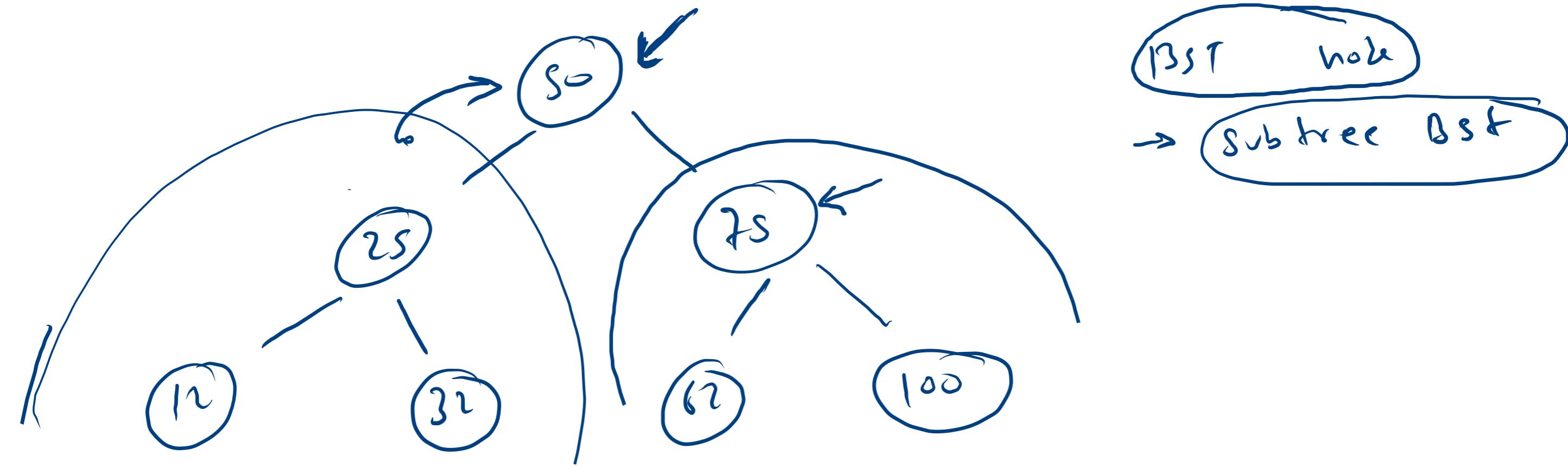
$$(132 + 87 + 75)$$



list \rightarrow node
 list \rightarrow tree
 390

leftsum - rightsum



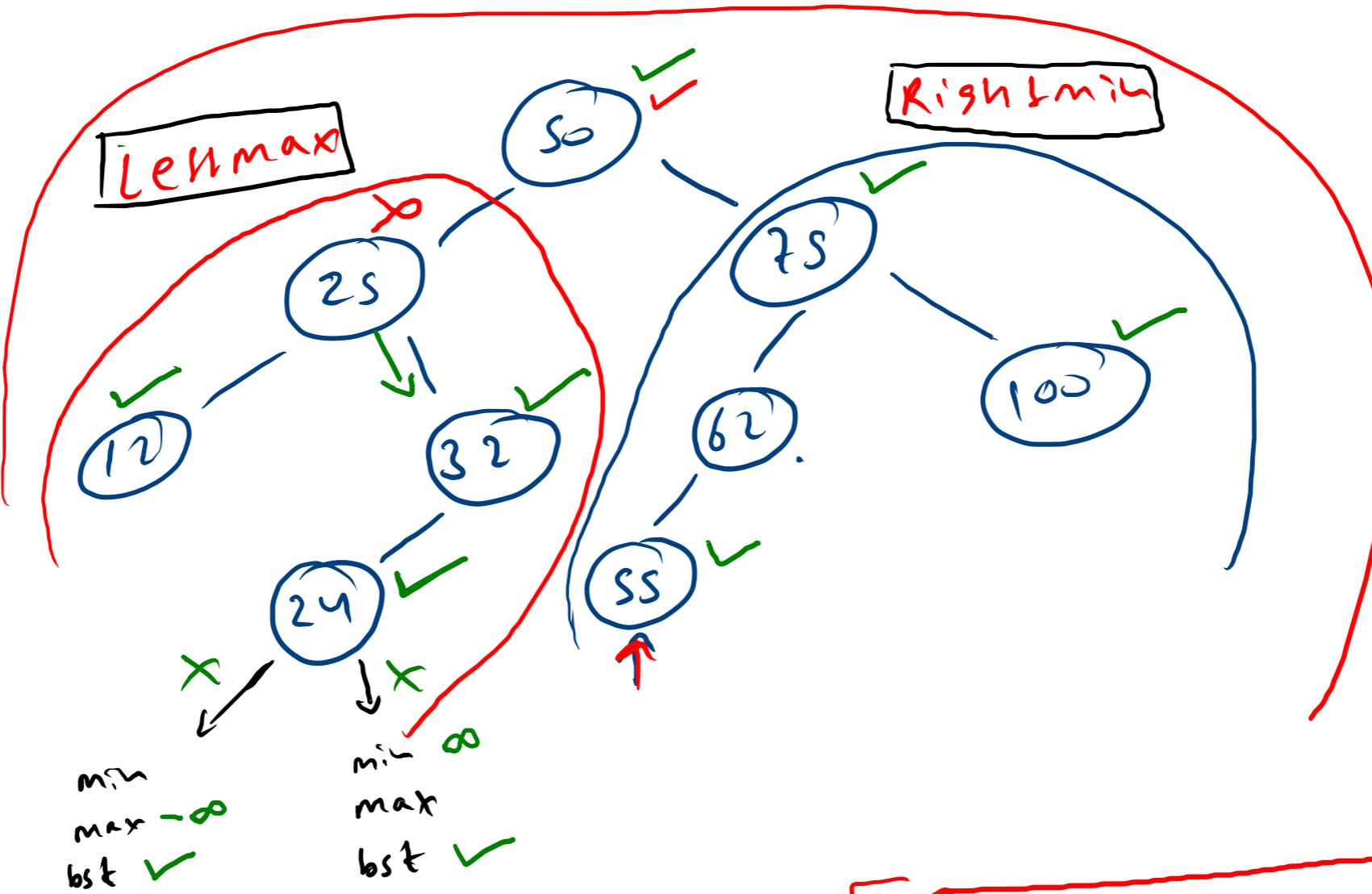


BST node
→ Subtree BST

left < node < right

node ✓
left BST ✓
right BST ✓

BST ✓



$-\infty < \text{node}$
Leftmax < node < Rightmin



```

public static BstPair isBst(Node node){
    if(node == null){
        BstPair p = new BstPair();
        p.isBst = true;
        p.min = Integer.MAX_VALUE;
        p.max = Integer.MIN_VALUE;
        return p;
    }

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

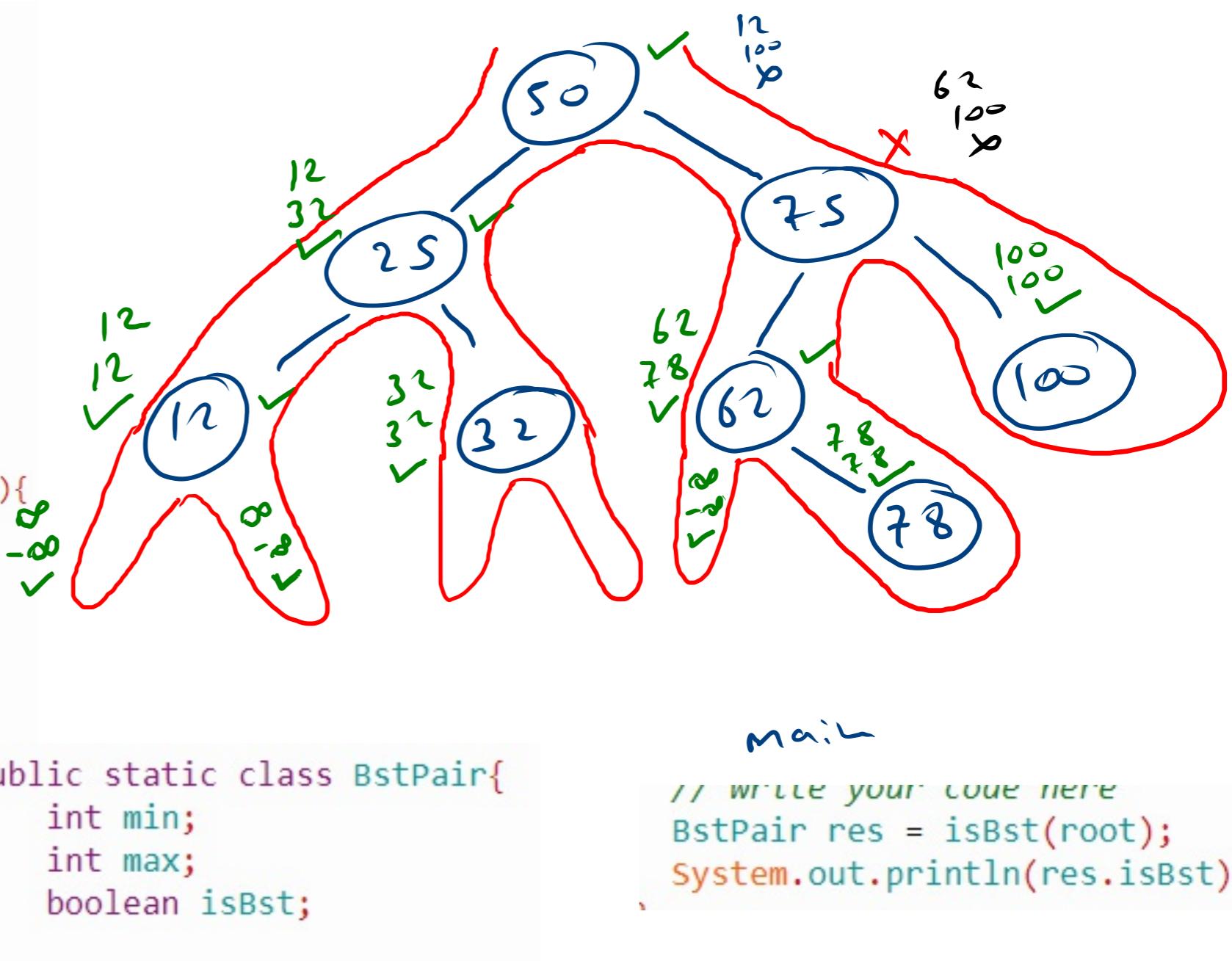
    boolean nodeok; 32 < 50 < 82
    if(lbst.max < node.data && node.data < rbst.min){
        nodeok = true;
    }else{
        nodeok = false;
    }

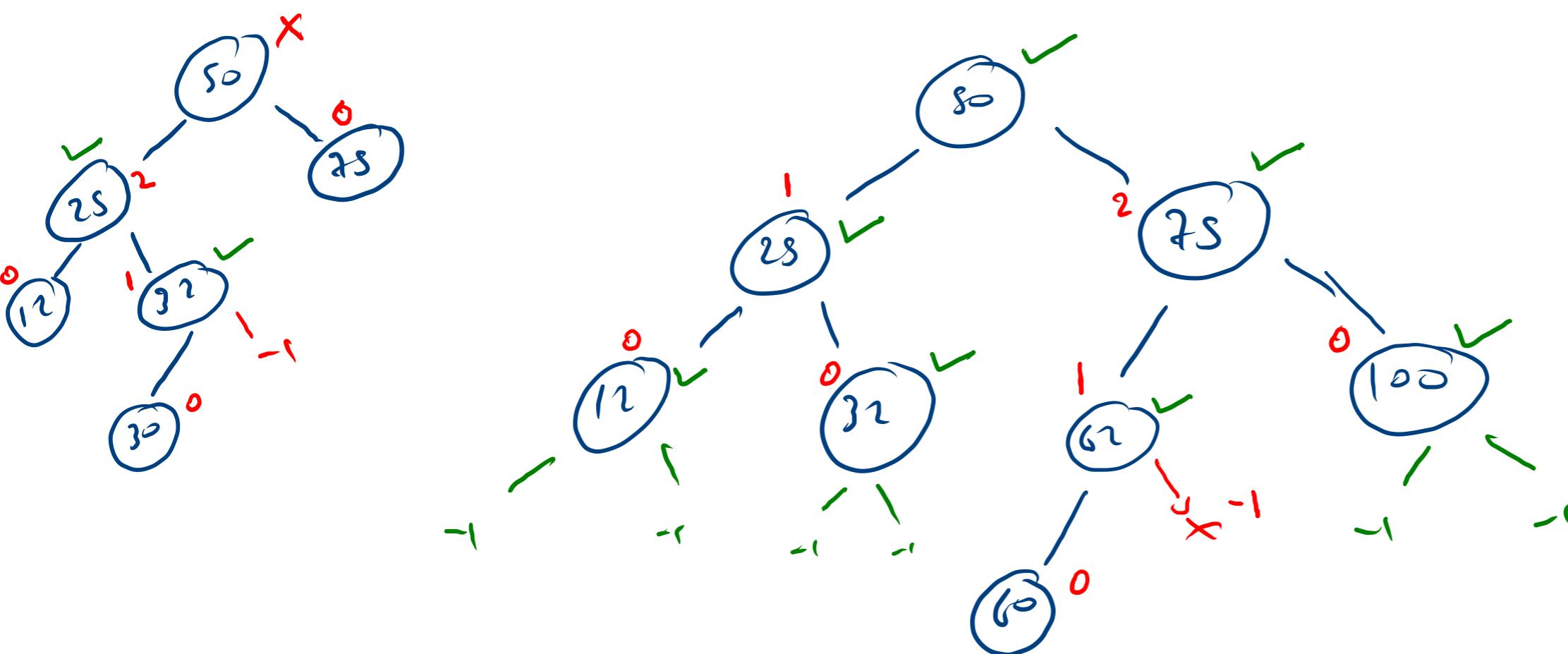
    boolean meBst; ✓ ✗
    if(nodeok && lbst.isBst && rbst.isBst){
        meBst = true;
    }else{
        meBst = false;
    }

    BstPair p = new BstPair();
    p.min = Math.min(node.data, lbst.min);
    p.max = Math.max(node.data, rbst.max);
    p.isBst = meBst;
    return p;
}

```

*min
max
isbst*





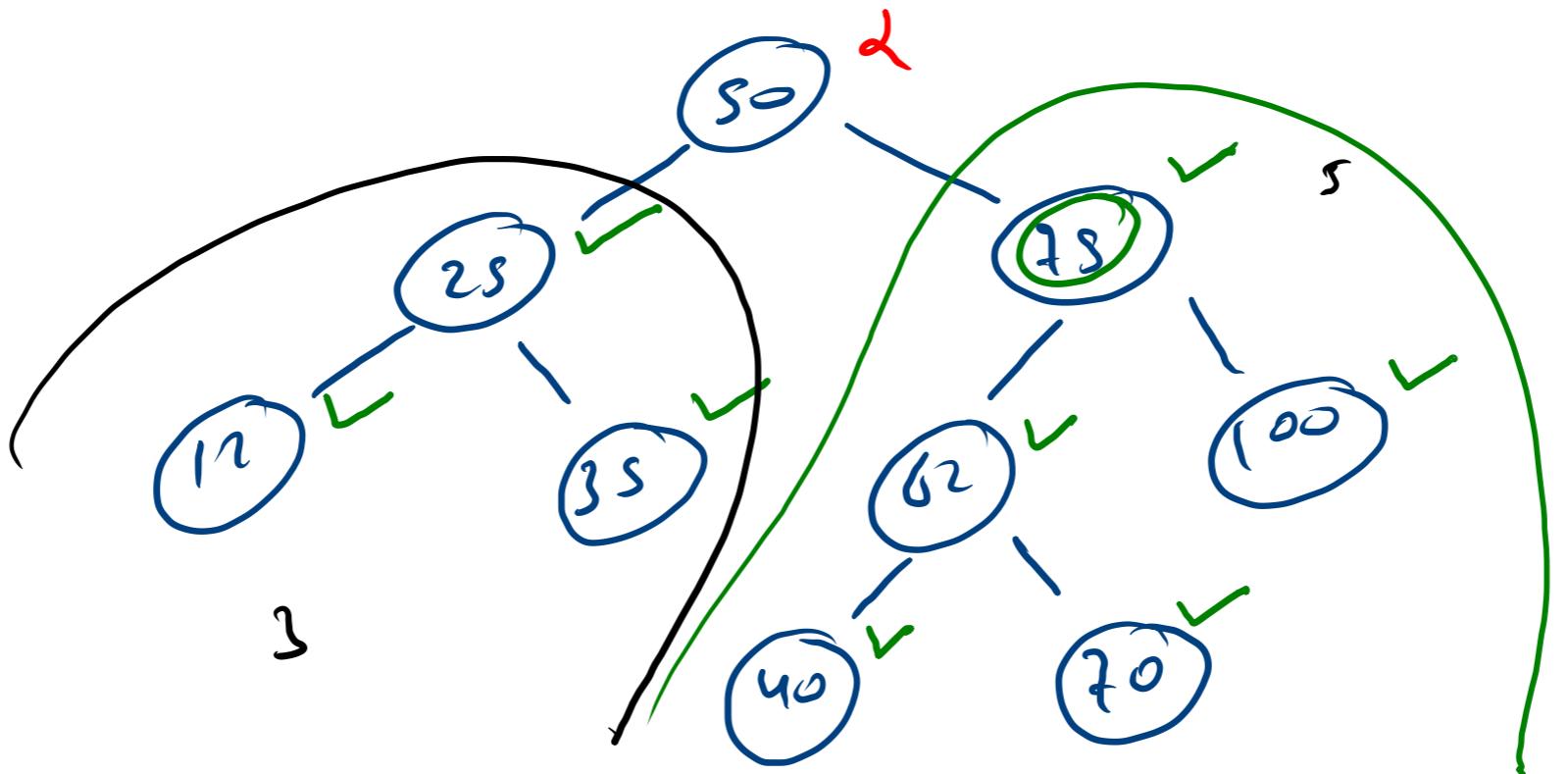
$$0 - (-1) = 1$$

$\cancel{X} \quad 1 < \left| \begin{array}{l} \text{left.height} - \text{right.height} \end{array} \right|$

$\checkmark \quad 1 > = \left| \begin{array}{l} \text{left.height} - \text{right.height} \end{array} \right|$

balanced
node
Tree
all nodes balanced

node ✓
left ✓
right ✓



75 @ 5
↓ Data
size

pair {
int min
max } is BST
bst
int size ;
}

static data
size

```
BstPair lbst = isBst(node.left);
BstPair rbst = isBst(node.right);
```

```
boolean nodeok;
if(lbst.max < node.data && node.data < rbst.min){
    nodeok = true;
} else{
    nodeok = false;
}
```

```
boolean meBst;
if(nodeok && lbst.isBst && rbst.isBst){
    meBst = true;
} else{
    meBst = false;
}
```

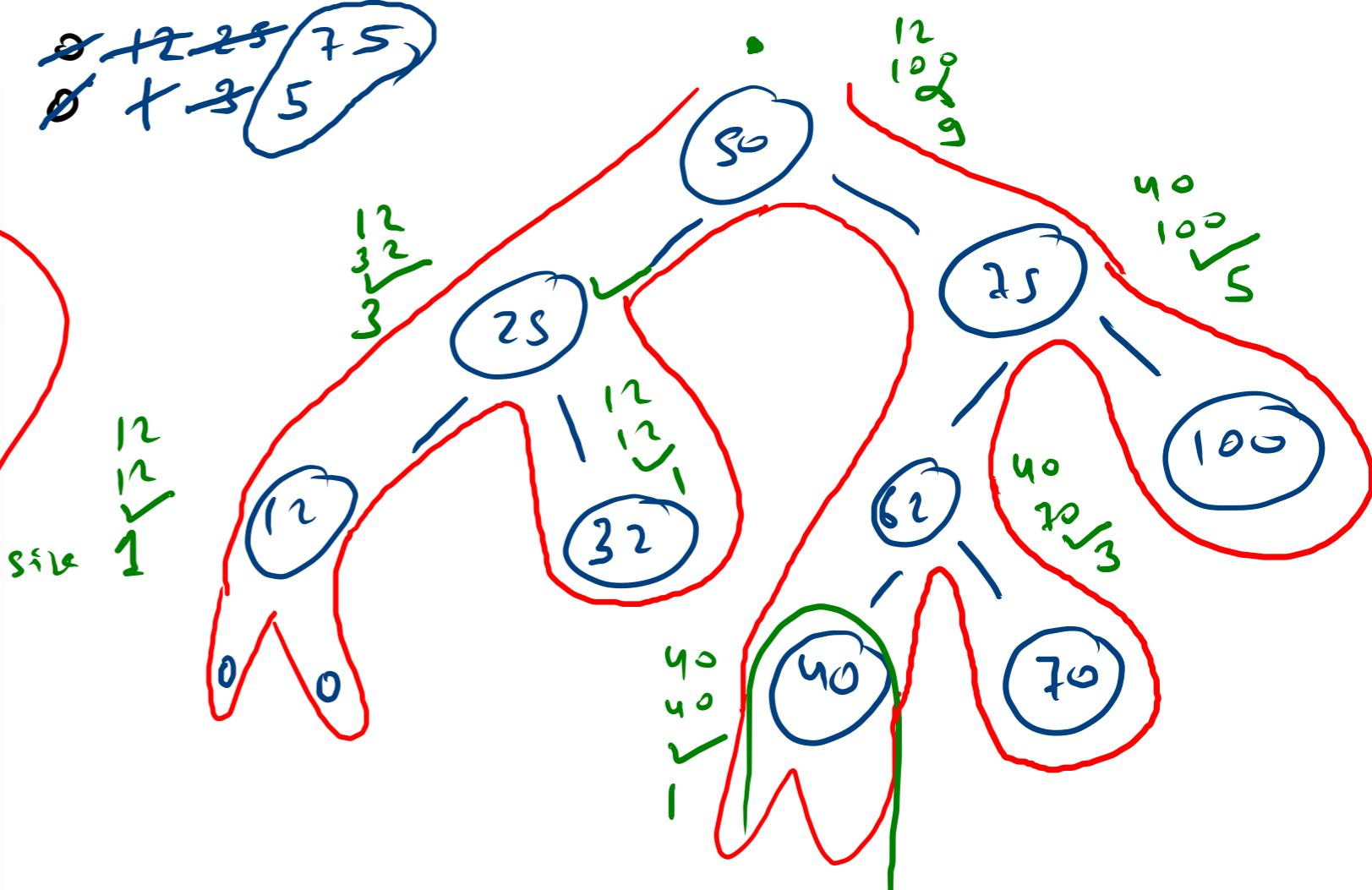
```
BstPair p = new BstPair();
p.min = Math.min(node.data, lbst.min);
p.max = Math.max(node.data, rbst.max);
p.isBst = meBst;
p.size = lbst.size + rbst.size + 1;
```

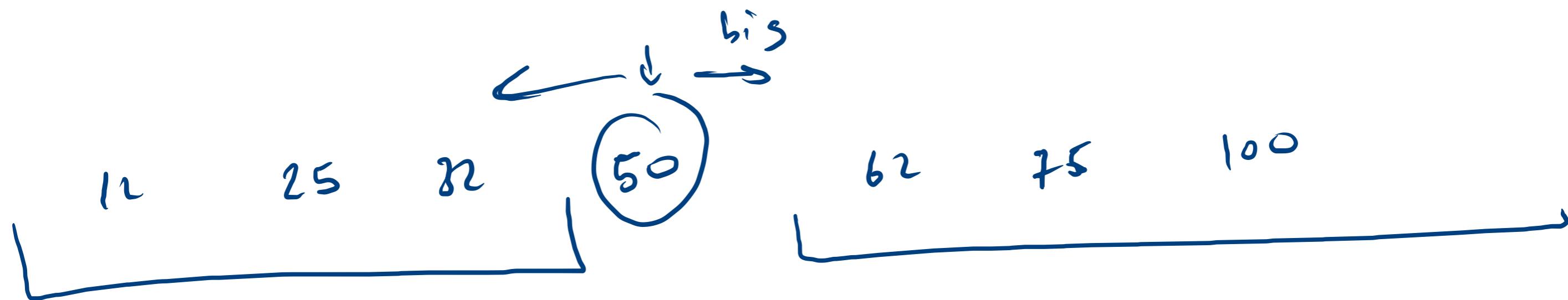
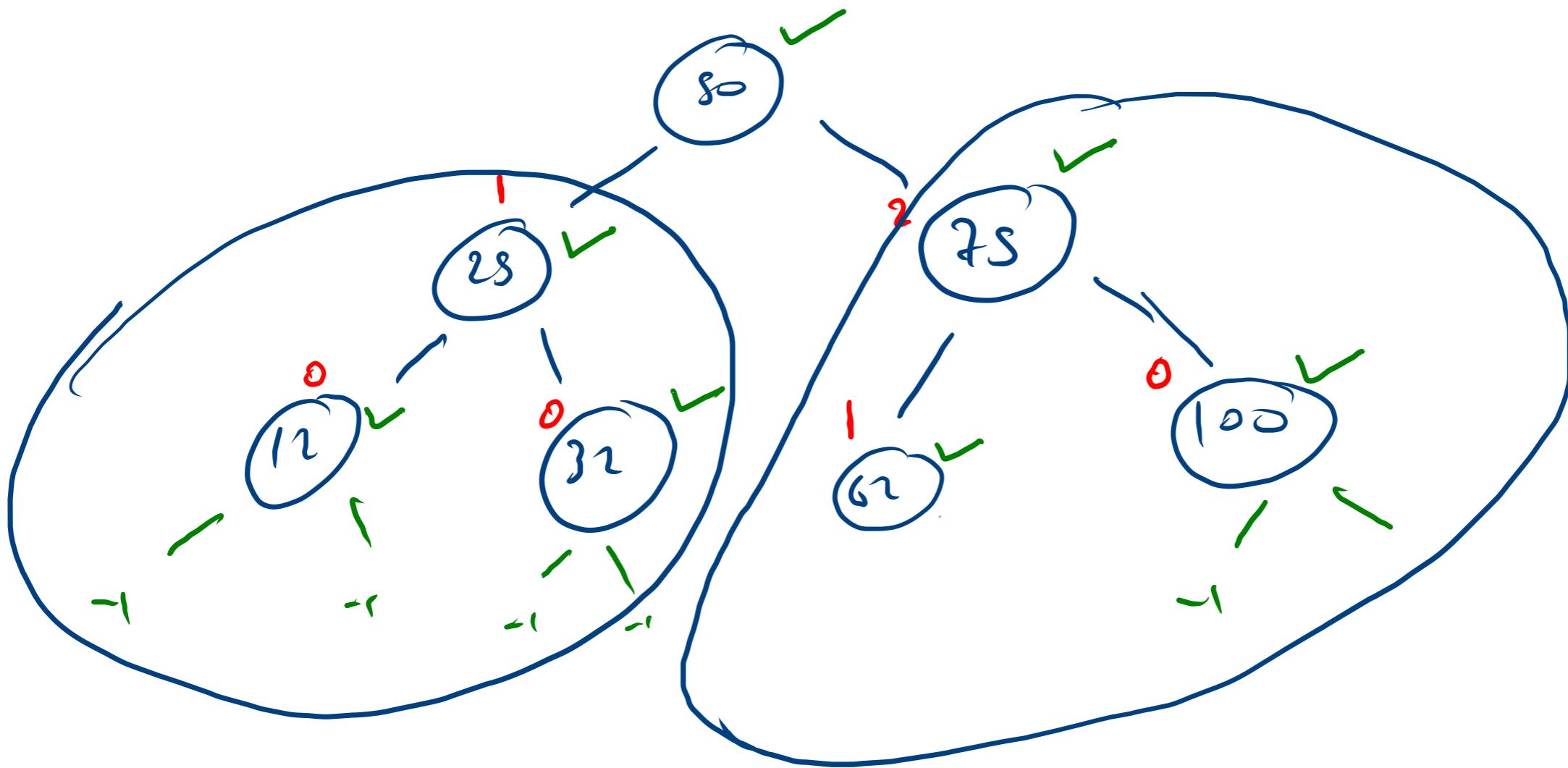
```
if(meBst && p.size > bstsize){
    bstdata = node.data;
    bstsize = p.size;
}
```

```
return p;
```

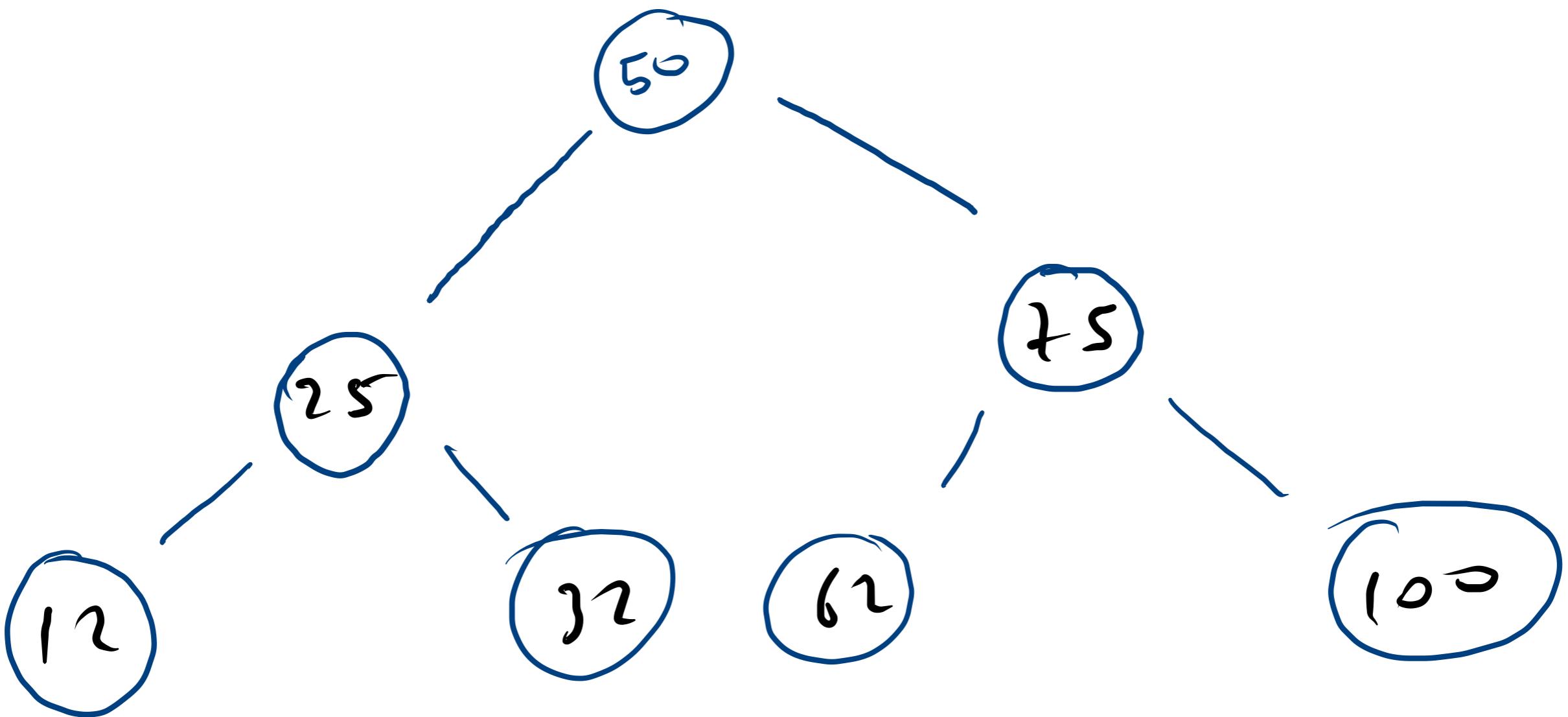
bst data
bst size

0 + 2 25 75
0 + 2 5

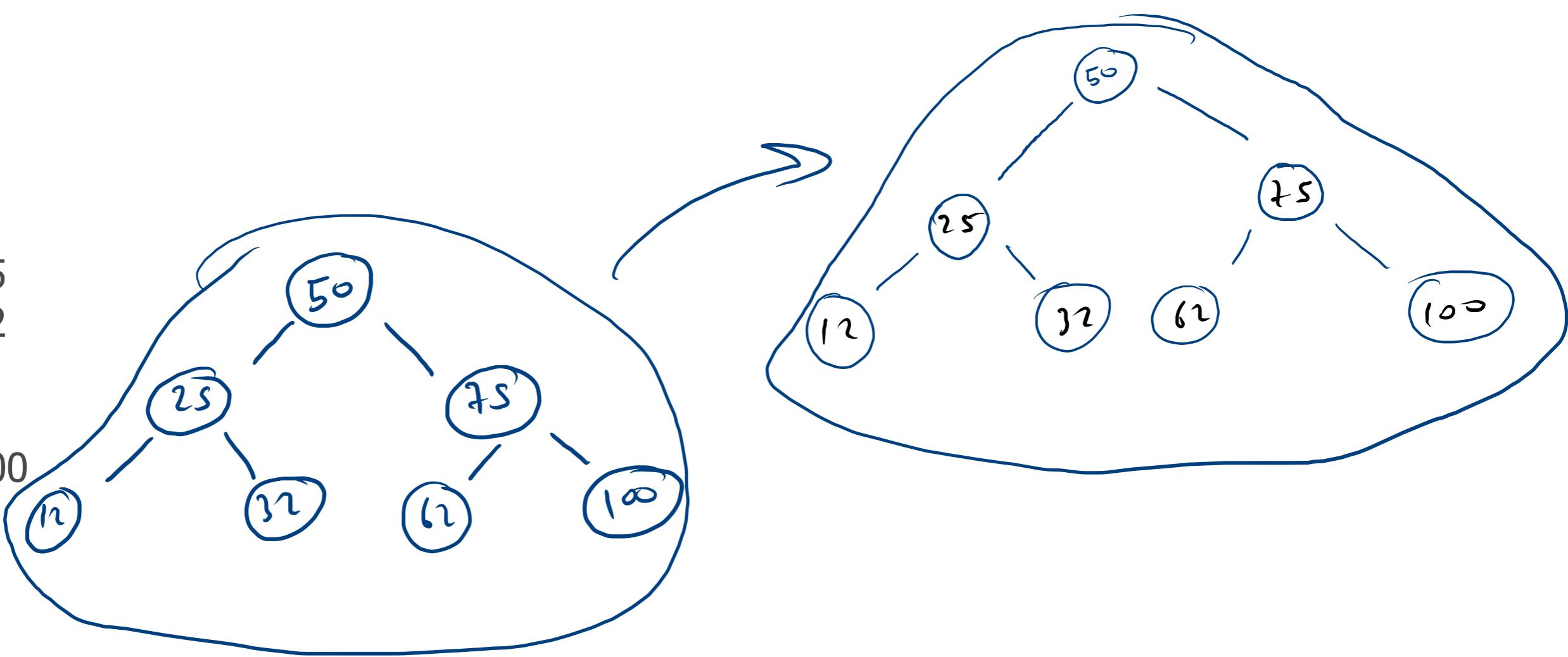


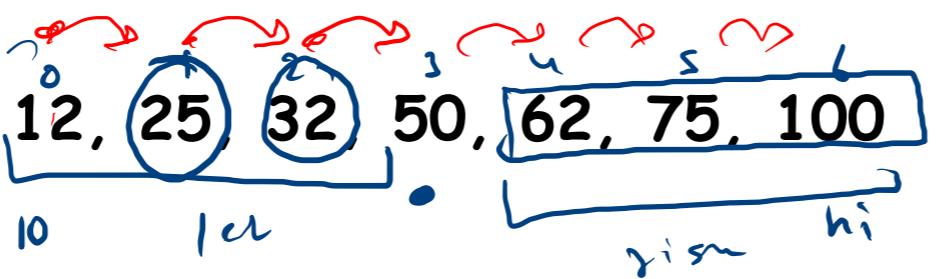


[
12 25 32 50
0 1 2 3
 ↑
 62 75 100
 4 5 6]



$25 <- 50 -> 75$
 $12 <- 25 -> 32$
 $\cdot <- 12 -> \cdot$
 $\cdot <- 32 -> \cdot$
 $62 <- 75 -> 100$
 $\cdot <- 62 -> \cdot$
 $\cdot <- 100 -> \cdot$





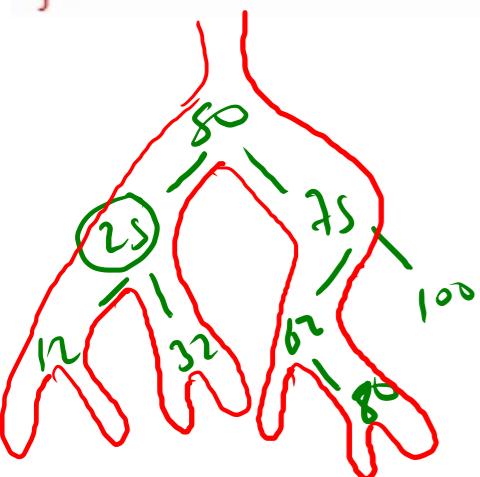
```

public static Node construct(int arr[], int lo, int hi){
    if(lo>hi) return null;

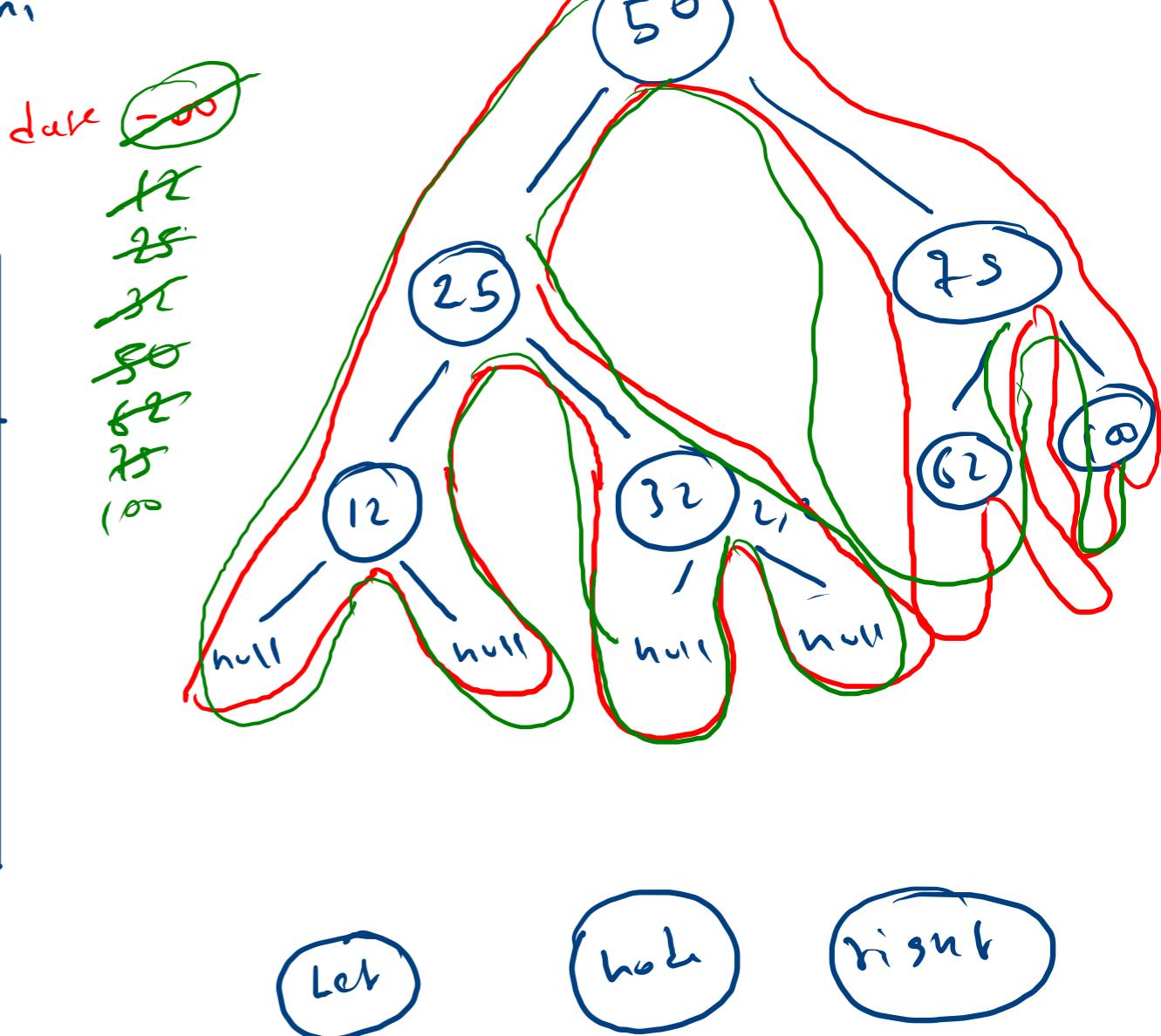
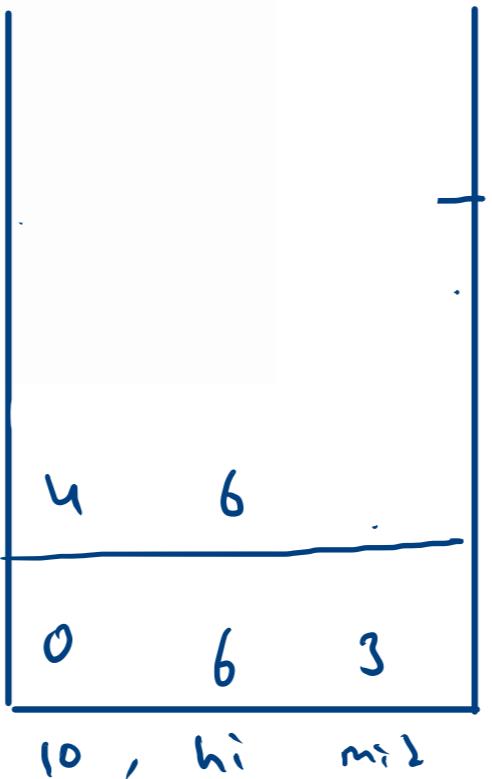
    int mid = (lo+hi)/2;

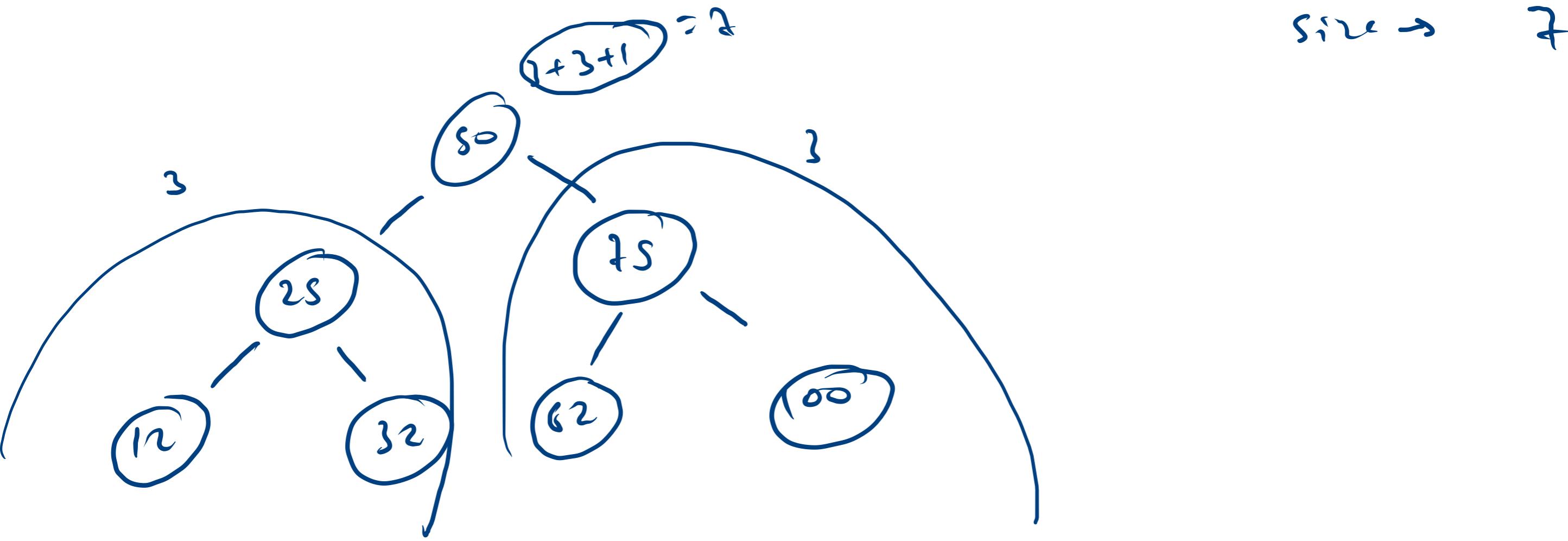
    Node root = new Node(arr[mid]);
    root.left = construct(arr, lo, mid-1);
    root.right = construct(arr, mid+1, hi);
    return root;
}

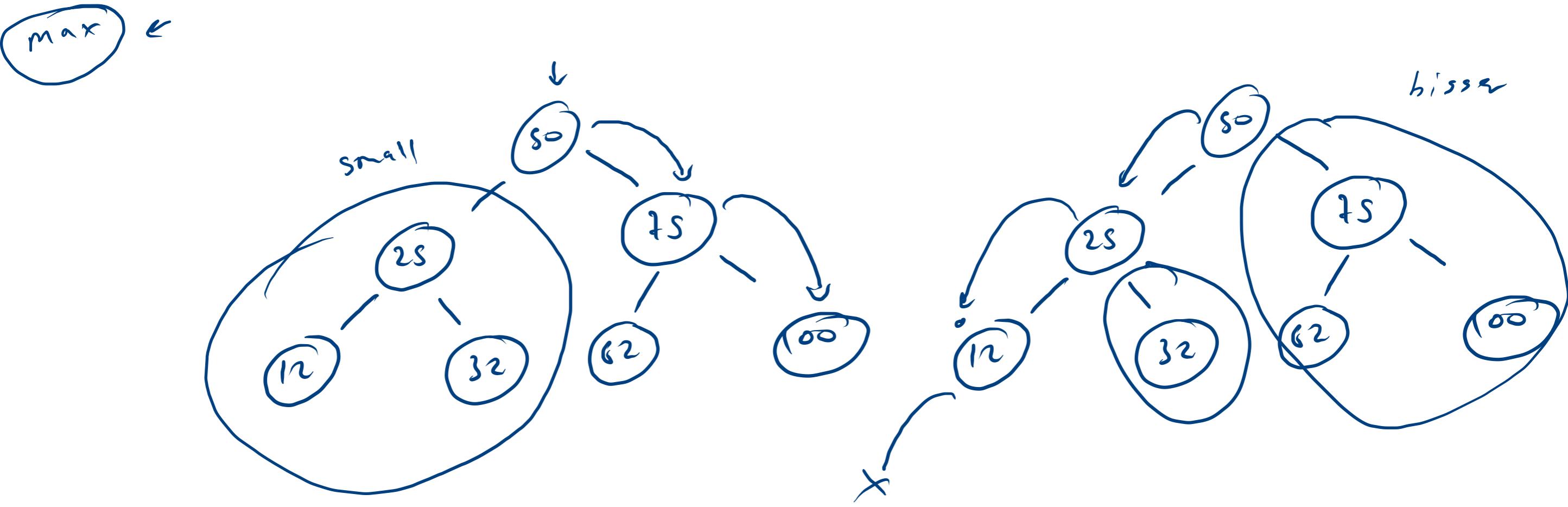
```

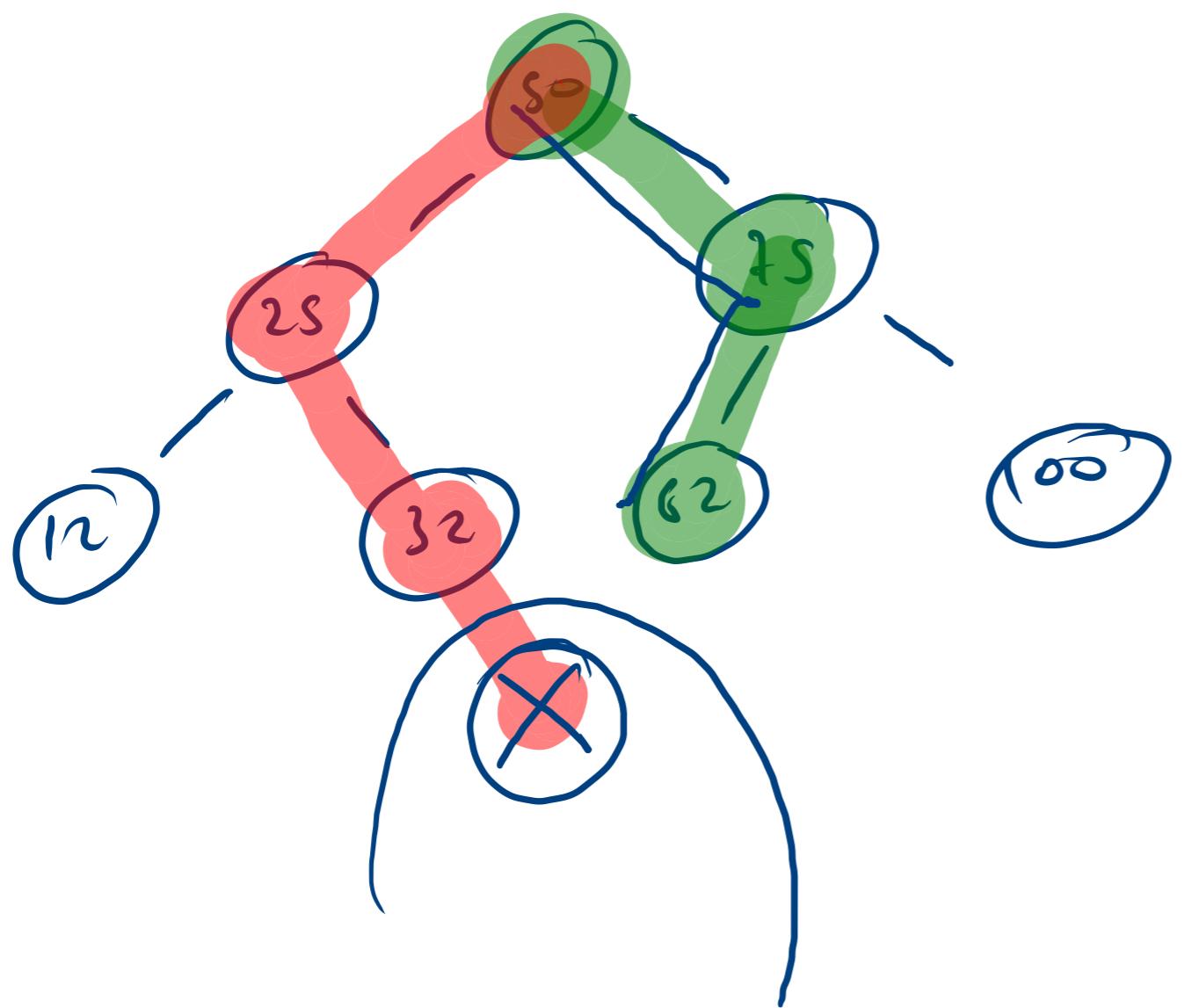


data
 50
 25
 75
 12
 32
 62
 100
 right
 !< as





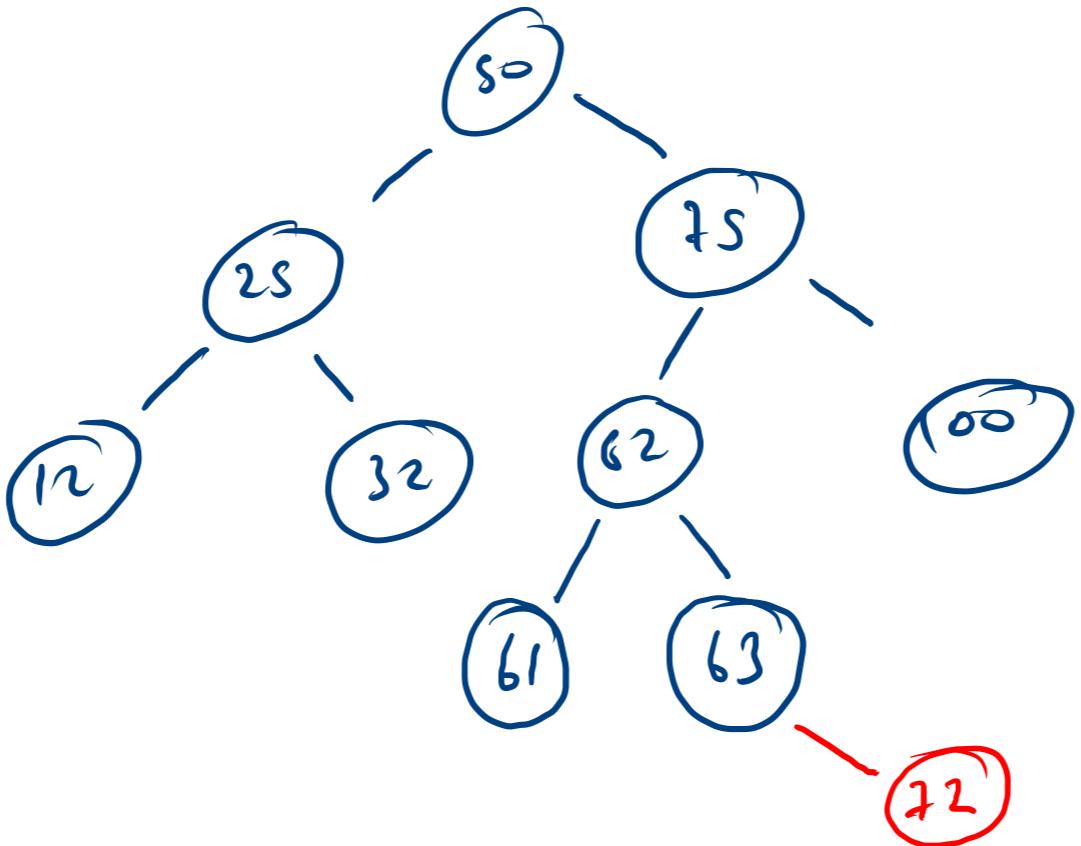




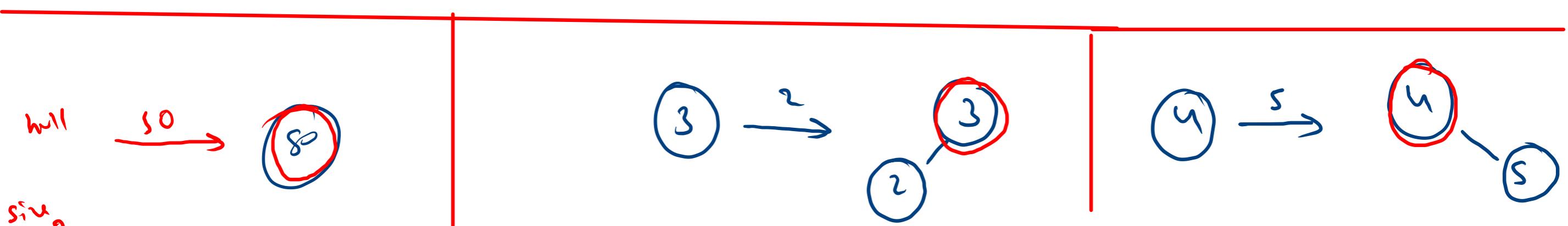
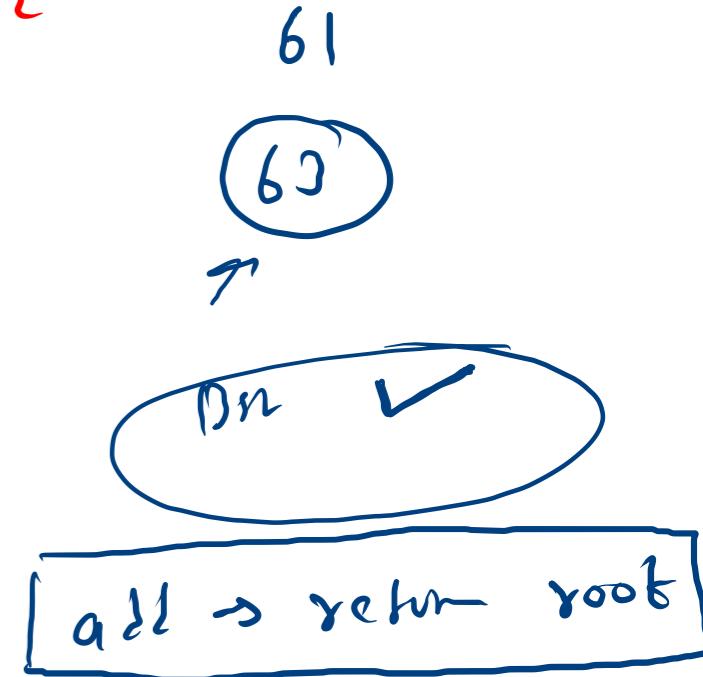
line 62 ✓

line 34

58



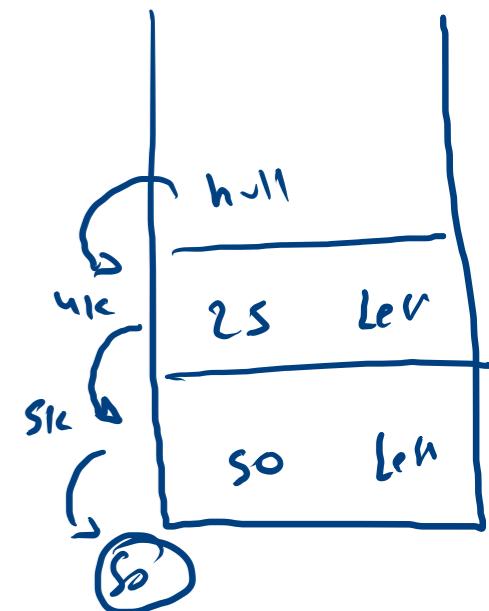
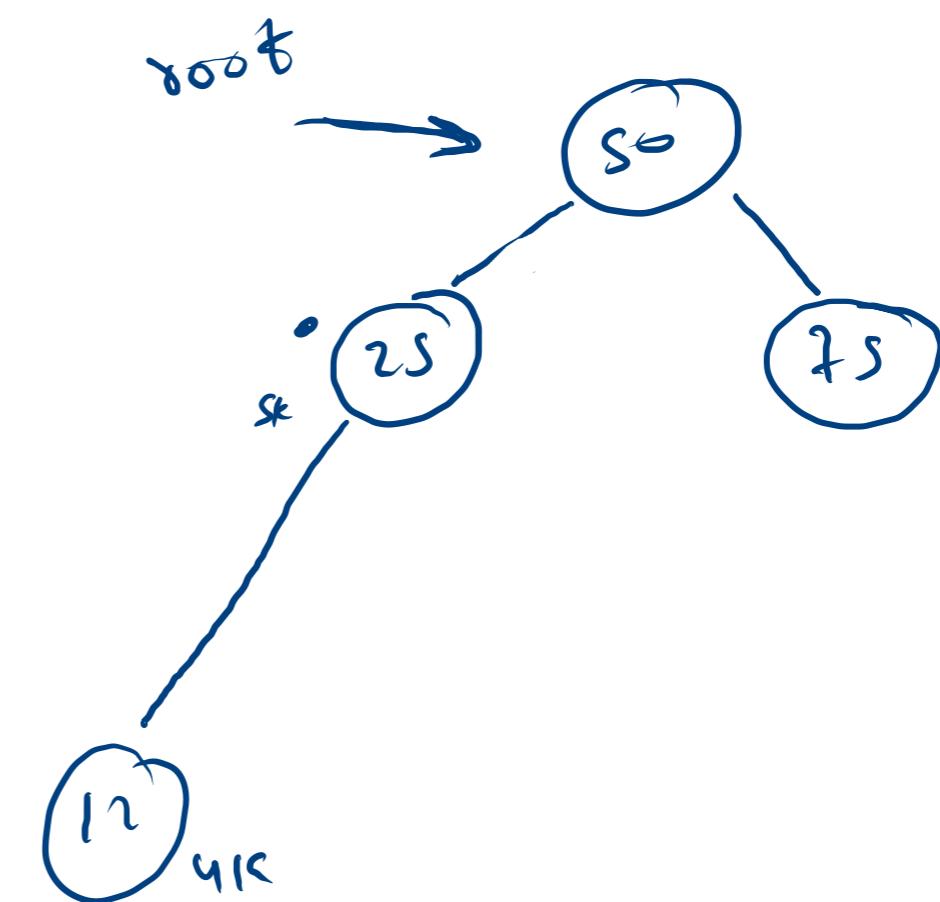
72

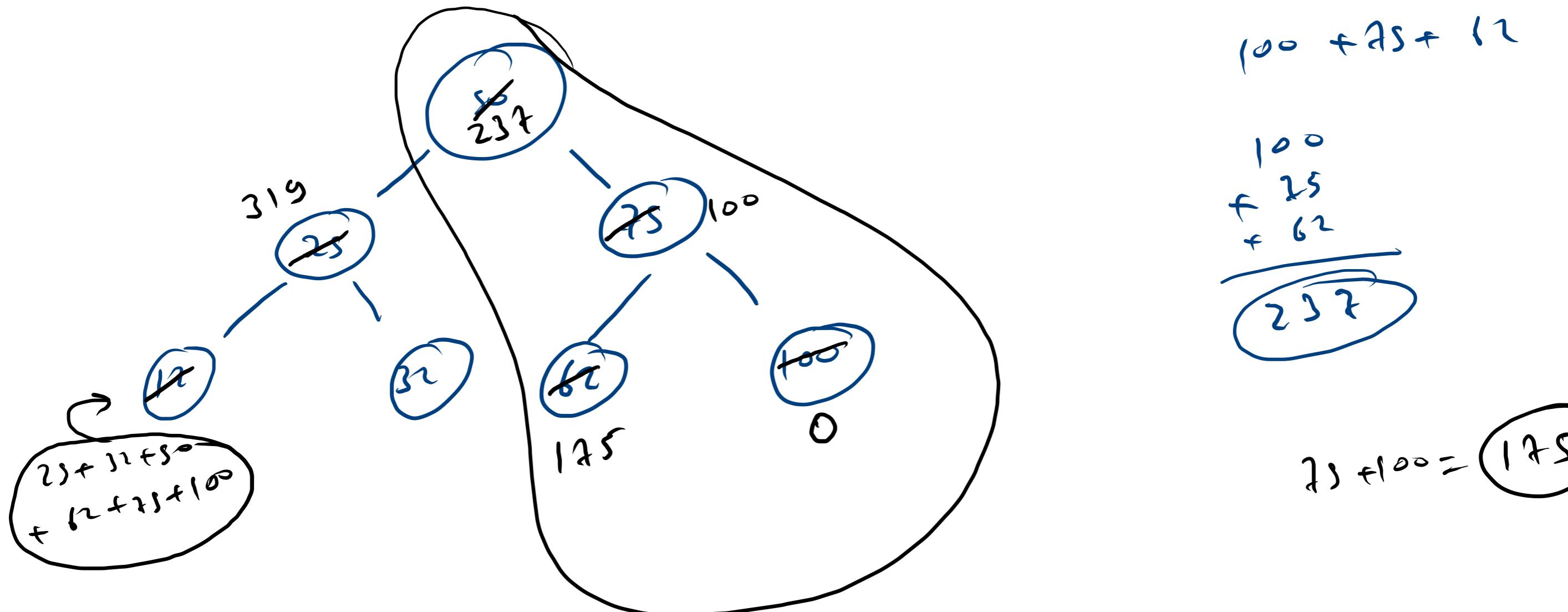


~~root~~ → ~~X~~

```
public static Node add(Node node, int data) {  
    if(node == null){  
        node = new Node(data, null, null);  
        return node;  
    }  
    if(data < node.data){      Sic  
        node.left = add(node.left, data);  
    }  
    if(node.data < data){  
        node.right = add(node.right, data);  
    }  
  
    return node;  
}
```

12





Innovator
 small self biz

$$\text{sum} = \varnothing 100 + 75 + 62 \\ + 80 \\ + 72$$

