8 July 2023

NS 8 July 2023

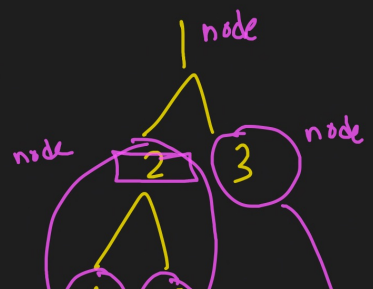
Main.java

Main.java > Main > BinaryTree > sumOfNodes(Node)

```
43              if (node != null) {
44                  postorderTraversal(node.left);      // L
45                  postorderTraversal(node.right);     // R
46                  System.out.print(node.data + " ");  // V
47
48              }
49          }
50
51          int sumOfNodes(Node node)
52          {
53              if(node == null)
54                  return 0;
55              return node.data + sumOfNodes(node.left) + sumOfNodes(node.right);
56          }
57      }
58
        Run | Debug
59      public static void main(String[] args) {
60          BinaryTree binaryTree = new BinaryTree();
61          binaryTree.inorderTraversal(binaryTree.root);
62          System.out.println();
63          binaryTree.preorderTraversal(binaryTree.root);
64          System.out.println();
65          binaryTree.postorderTraversal(binaryTree.root);
66          System.out.println("\nSum of Nodes: "+binaryTree.sumOfNodes(binaryTree.root));
67      }
```

Ln 55, Col 50 Spaces: 4 UTF-8 LF Java Live Share

binary tree



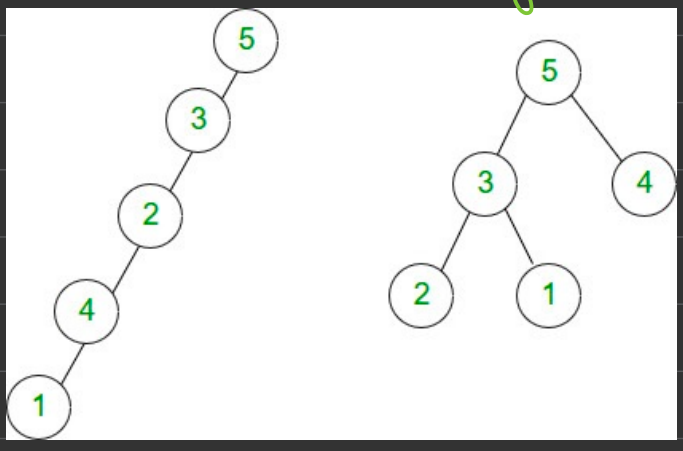
1 + sum(left) + sum(right)

= 1 + 1 + sum(left) + sum(right) + 3 + sum(null) + sum(null)

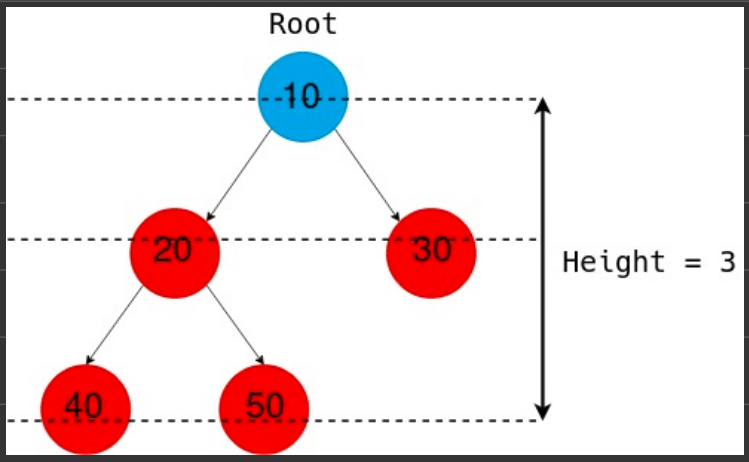
0 0

Height of Tree

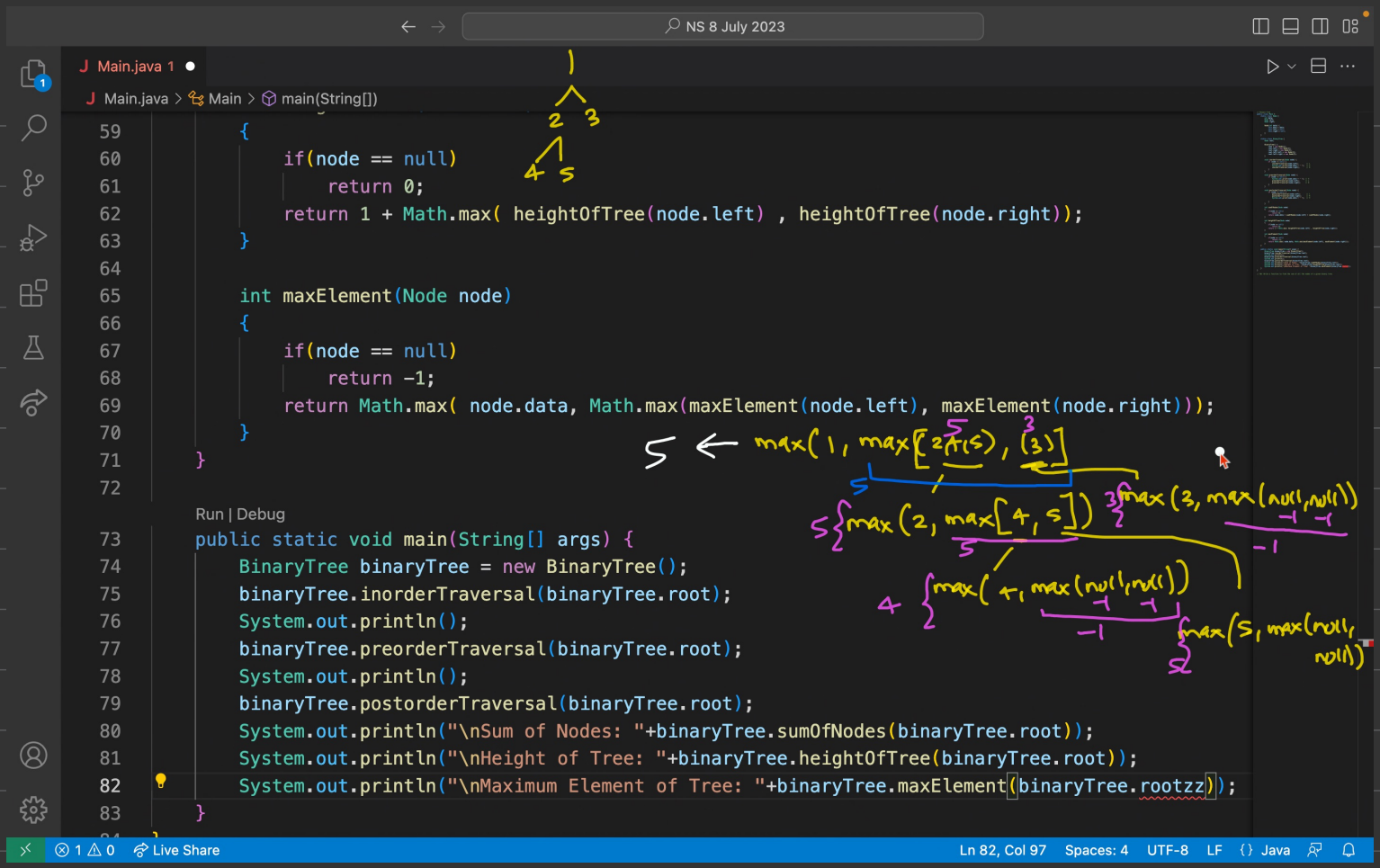
Height = 5



Height = 3



return 1 + Math.max(hOT(root.left), hOT(root.right))



Full | Strict Binary Tree: It has 0 | 2 children

Types of Binary Trees