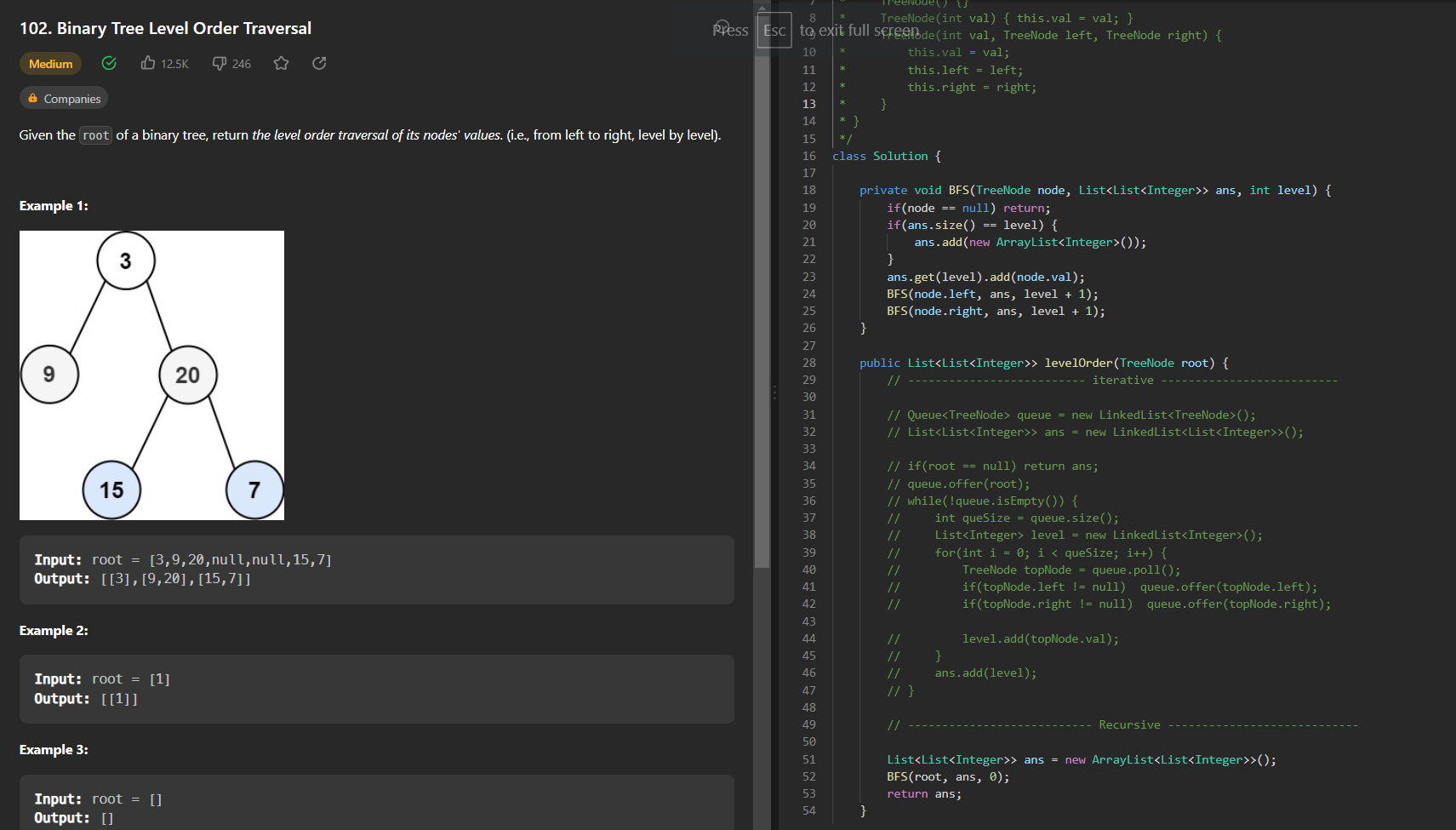
|  |  |
| --- | --- |
| **Trees** | |
| BFS tree traversal | 1 |
| DFS Pre-order traversal | 1 |
| DFS In-Order traversal | 2 |
| DFS Post-Order traversal | 2 |
| Maximum depth of a binary tree | 3 |
| Balanced binary tree | 3 |
| Diameter of a binary tree | 4 |
| Maximum sum path in a binary tree | 4 |
| Binary tree ZigZag level order traversal | 5 |
| Binary tree left side view | 5 |
| Symmetric tree | 6 |
| Convert sorted linked list to a BST | 6 |
|  |  |
|  |  |

**Tree BFS traversal**

Iterative TC: O(N) recursive TC: O(N)

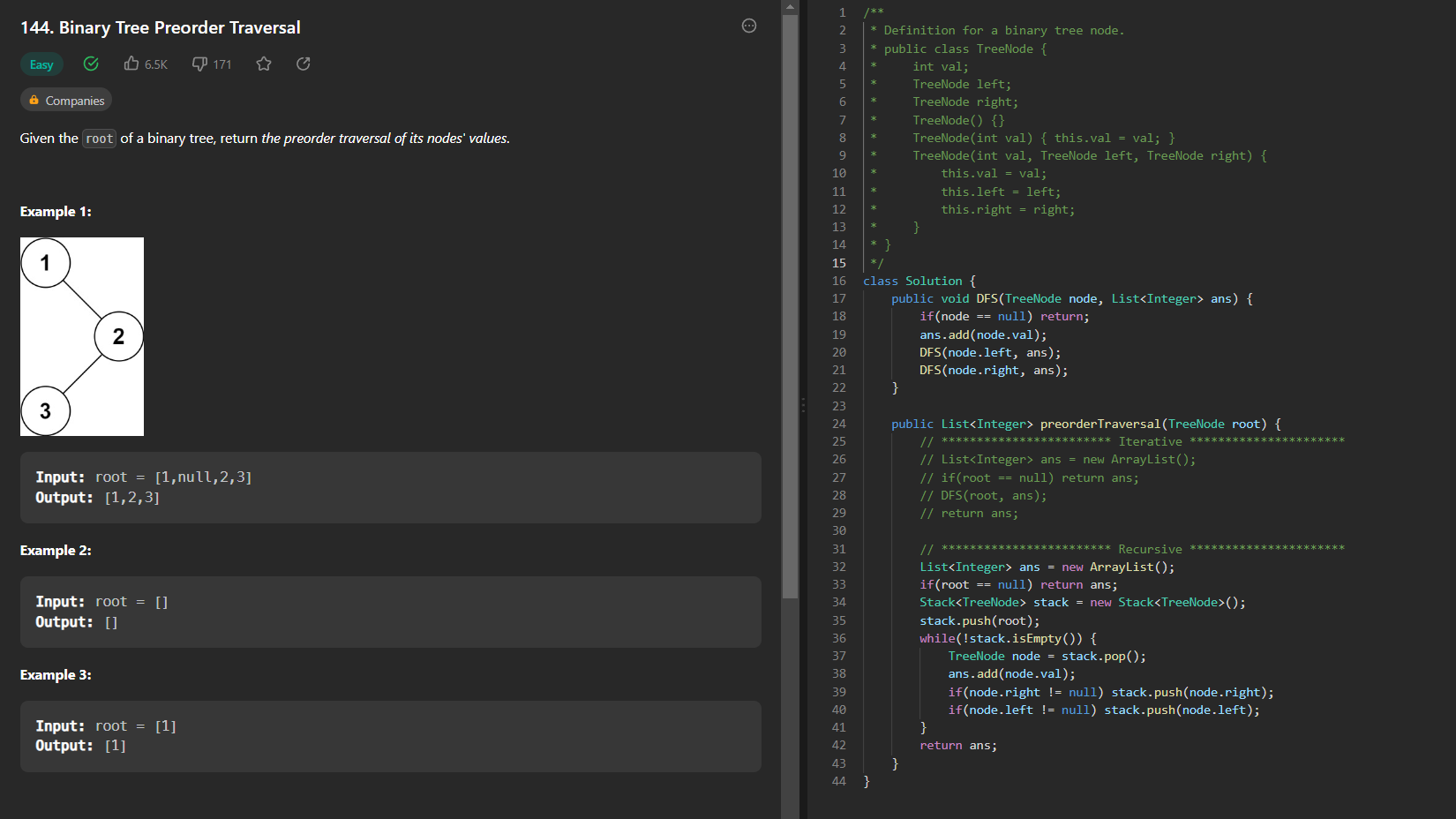
SC: O(L) Max no.of nodes in a level SC: O(H)



**Pre Order traversal**

Iterative TC: O(N) recursive TC: O(N)

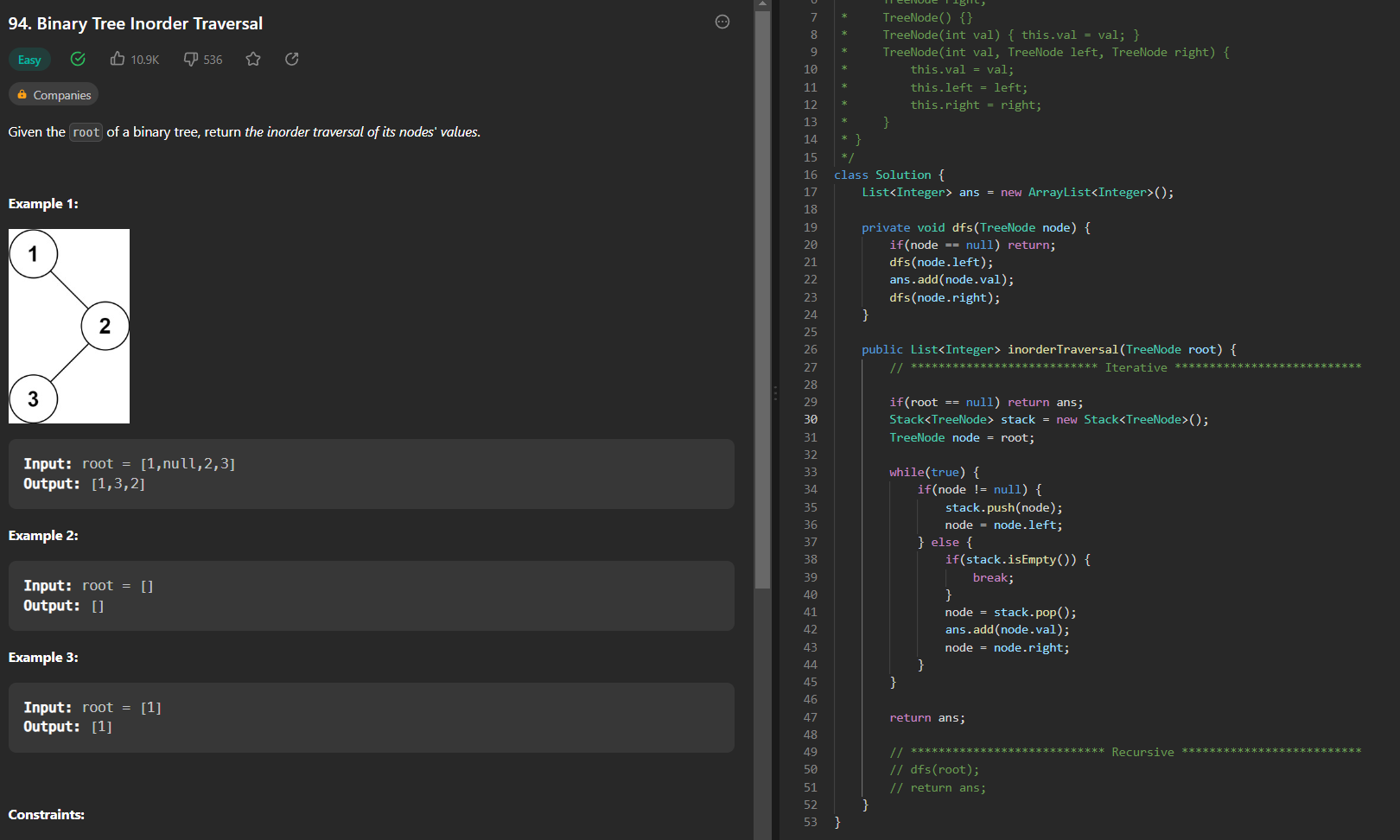
SC: O(H) Height of the tree SC: O(H)



**In Order traversal**

Iterative TC: O(N) recursive TC: O(N)

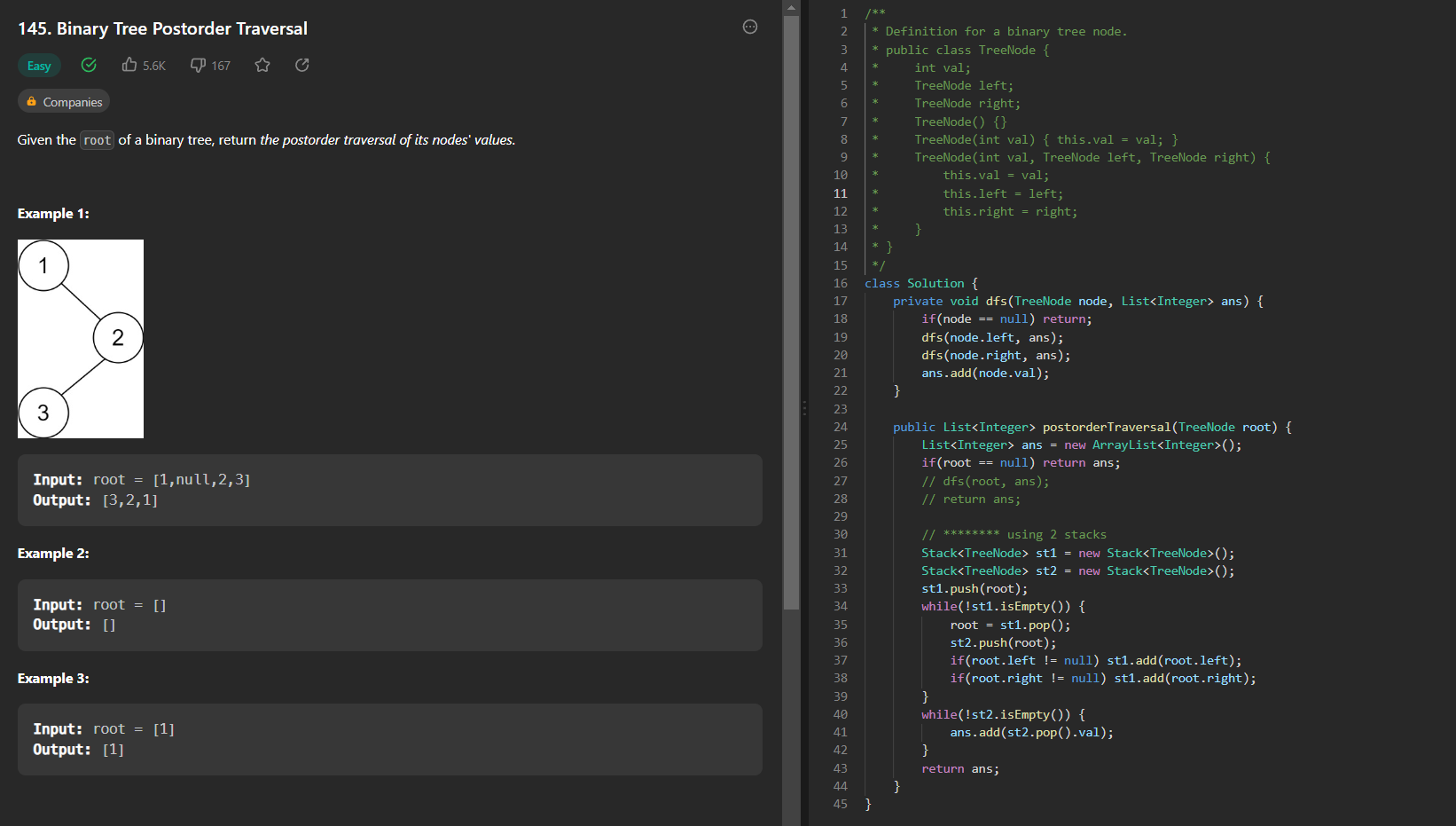
SC: O(H) uses stack SC: O(H)



**Post Order traversal**

Iterative TC: O(N) recursive TC: O(N)

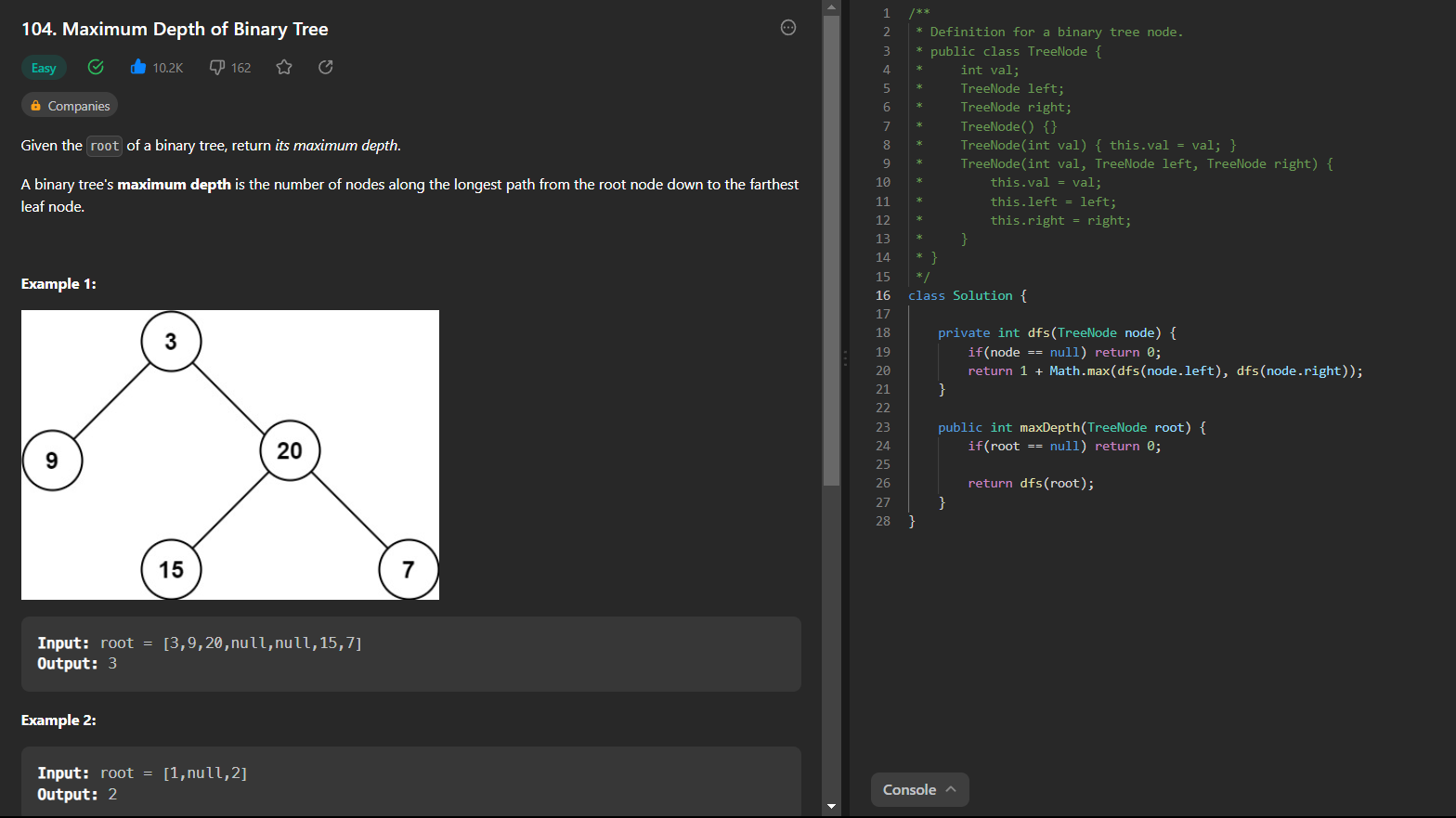
SC: O(N + H) uses 2 stacks SC: O(H)



**Maximum depth of the binary tree**

TC: O(N)

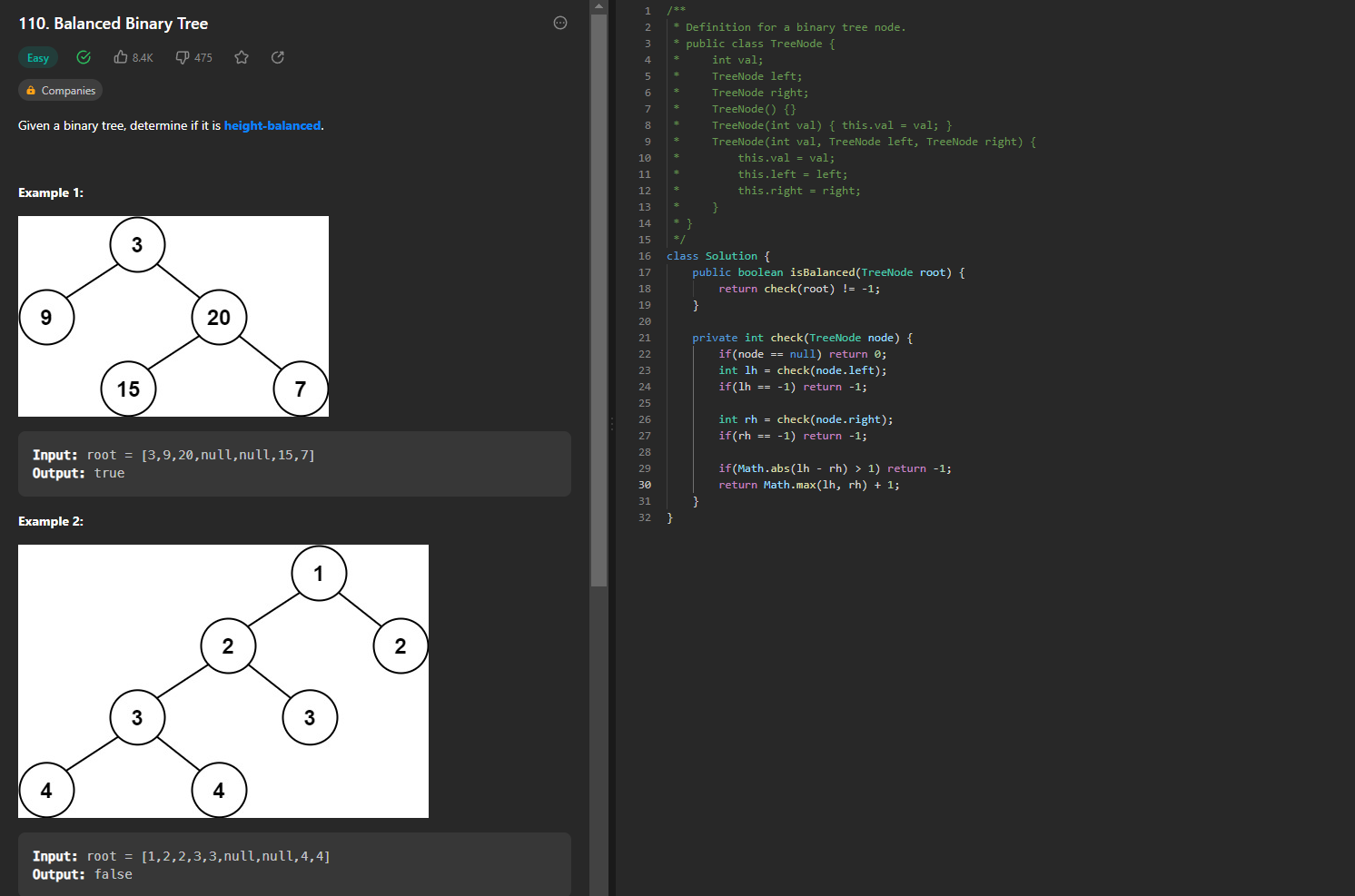
SC: O(H)



**Balanced binary tree**

TC: O(N)

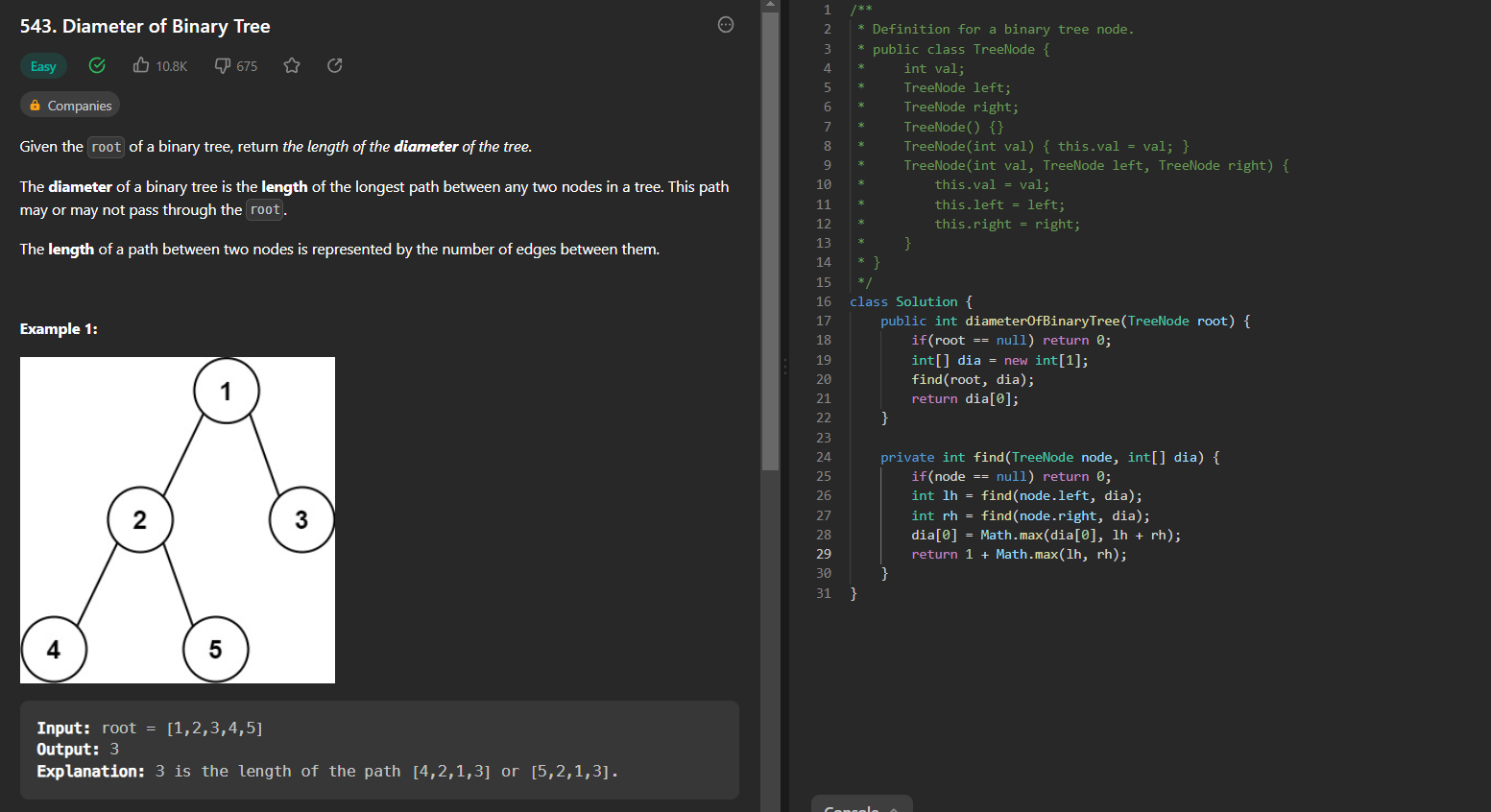
SC: O(H)

****

**Diameter of Binary tree:**

TC: O(N)

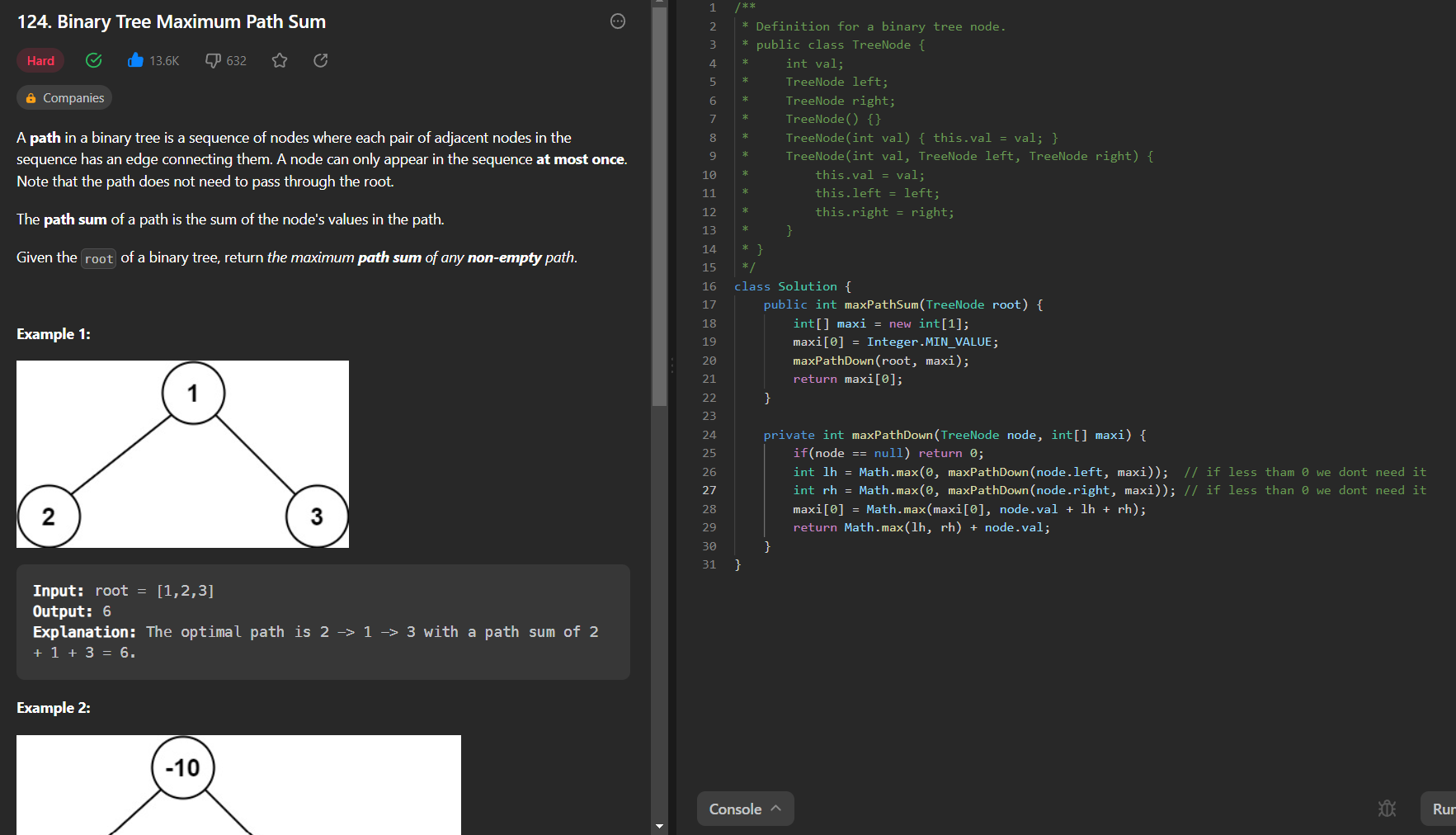
SC: O(H)

****

**Maximum Path sum in a binary tree:**

TC: O(N)

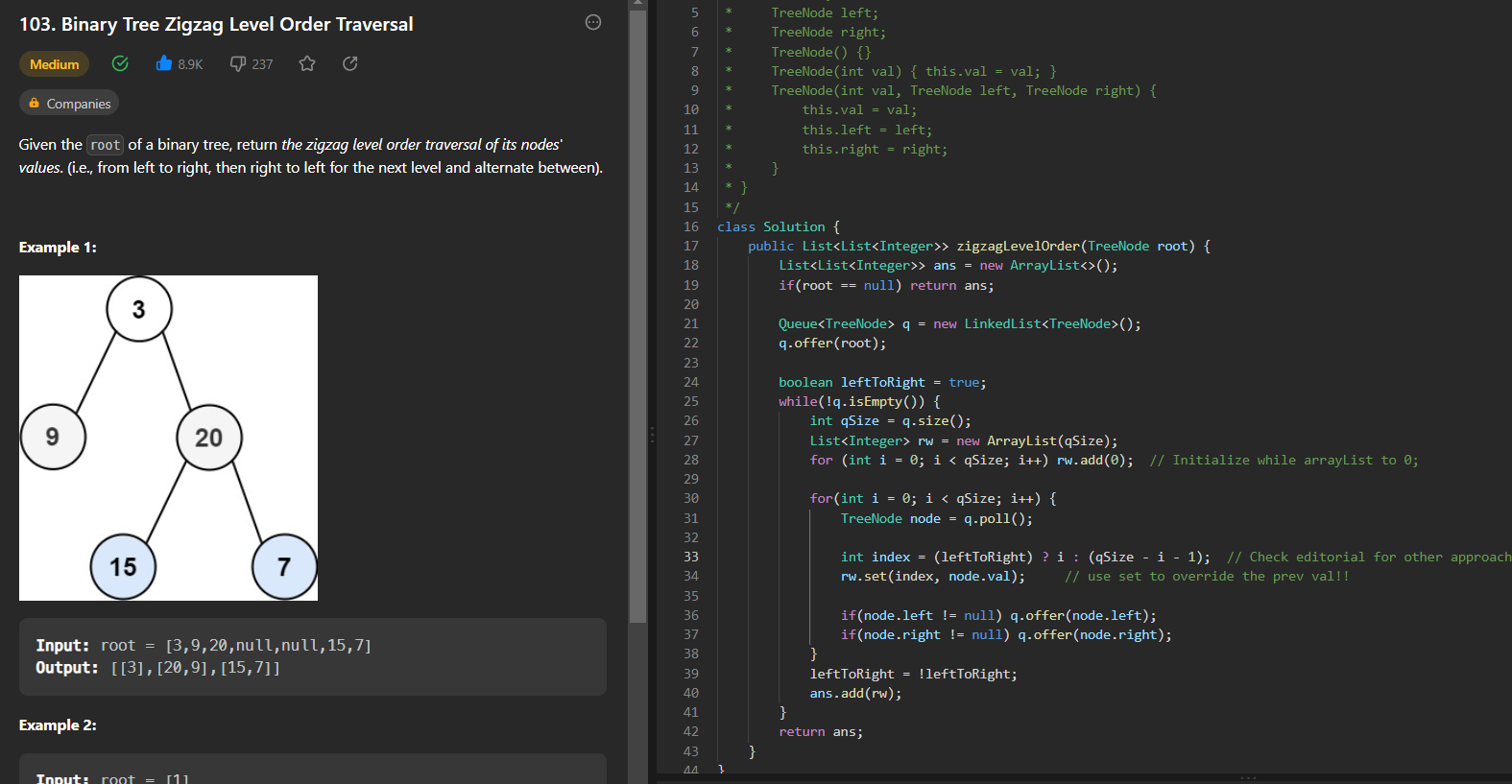
SC: O(H)

******

**Binary Tree Zigzag Level Order Traversal:**

TC: O(N)

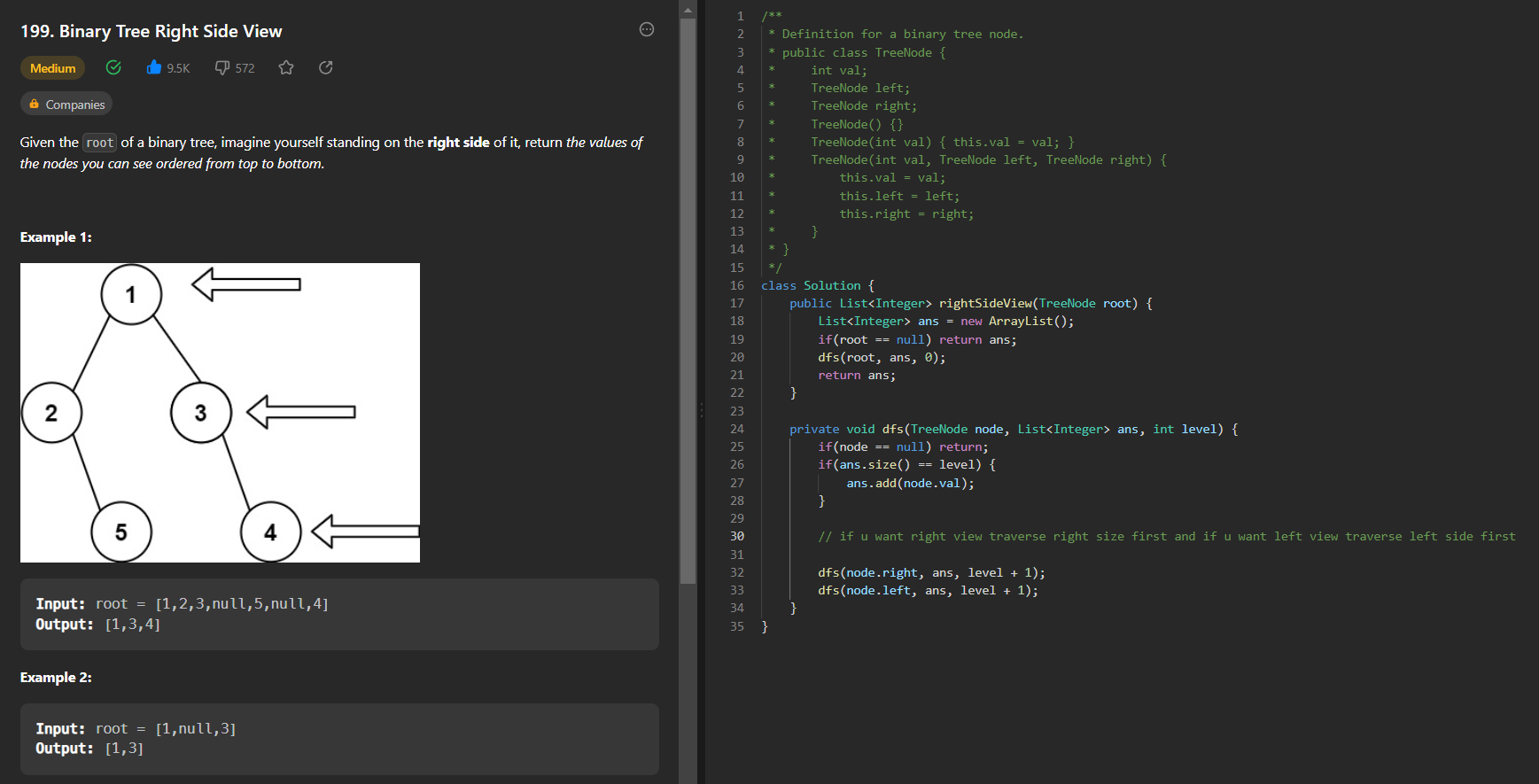
SC: O(L)

****

**Binary Tree Right Side View**

TC: O(N)

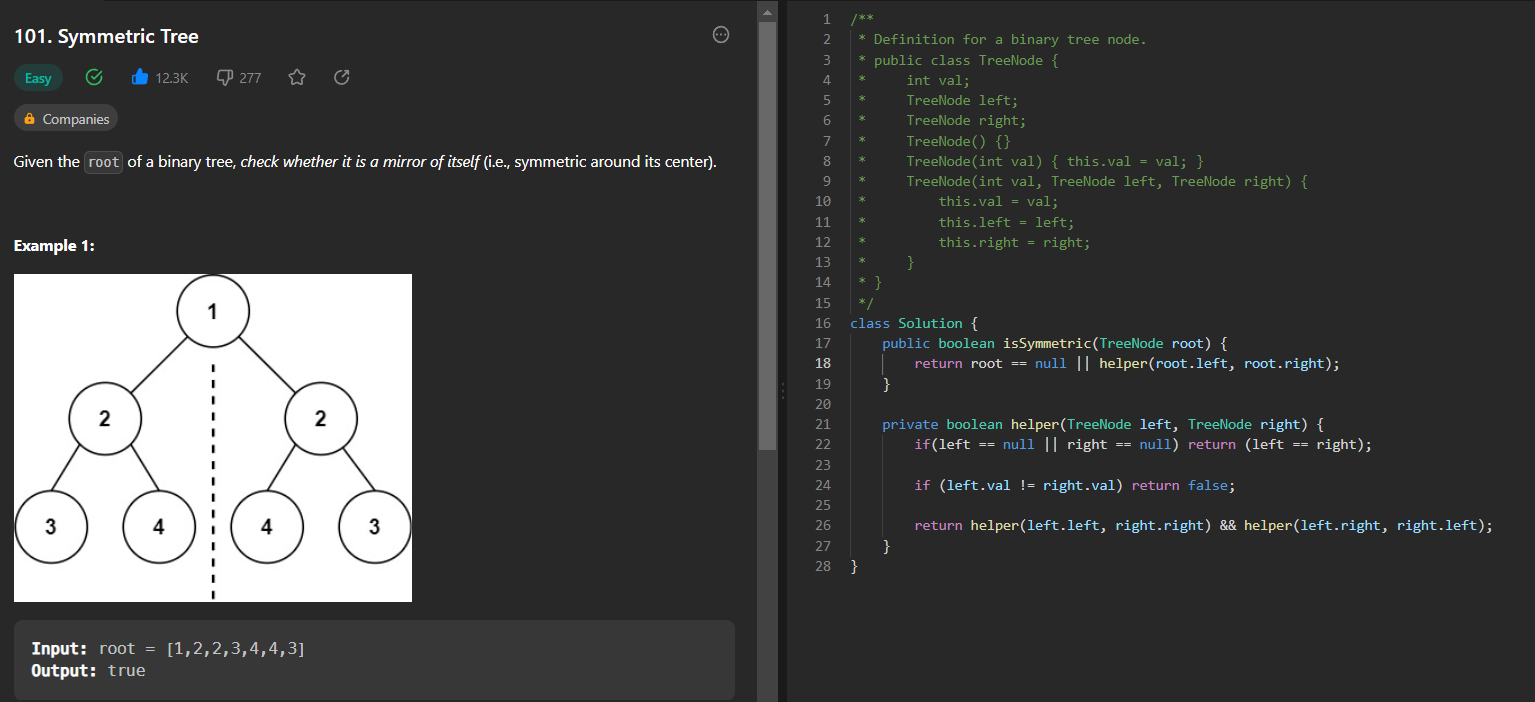
SC: O(H)

****

**Symmetric Tree:**

TC: O(N)

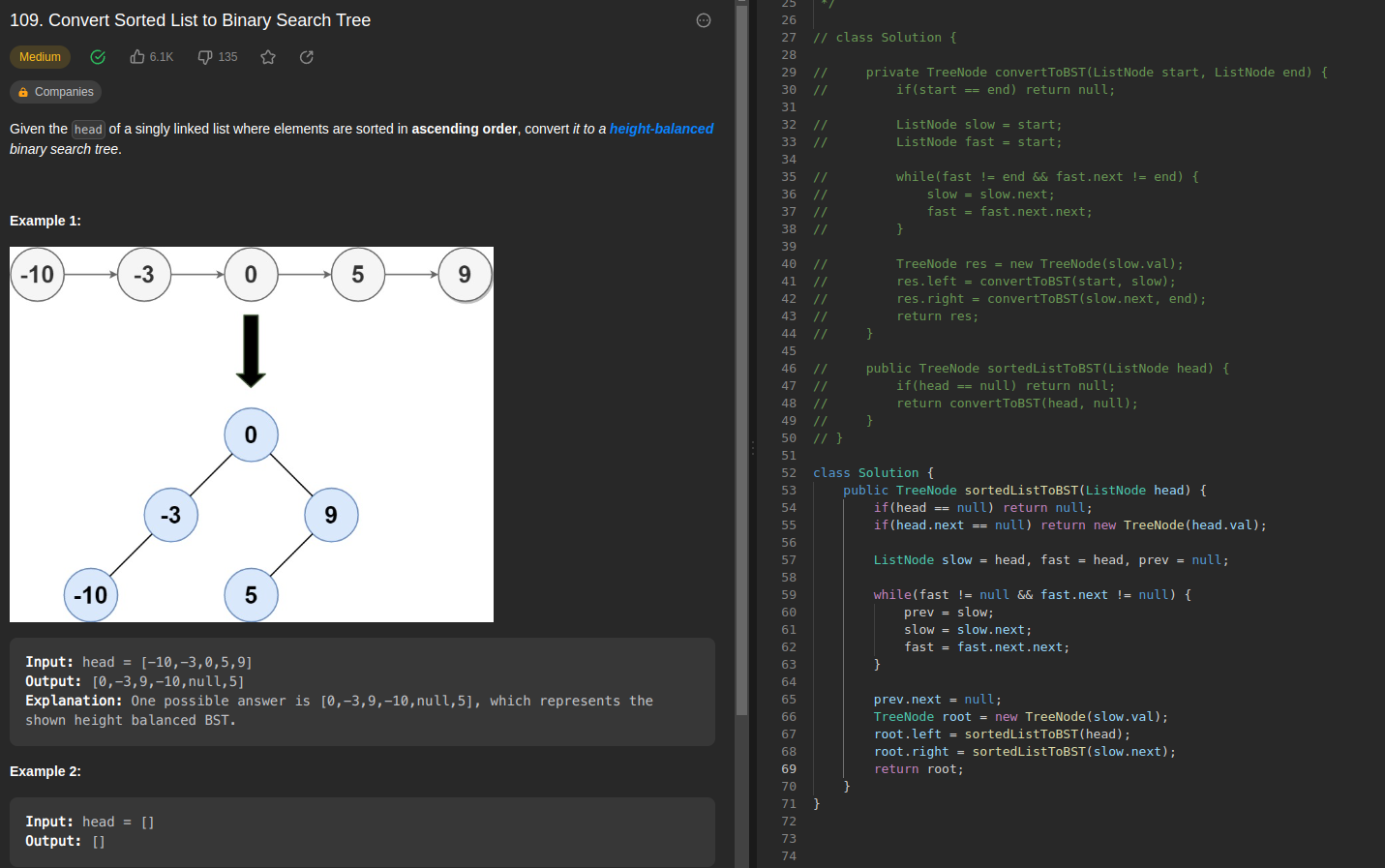
SC: O(H)



**Convert a sorted linked list to BST:**

TC: O(N)

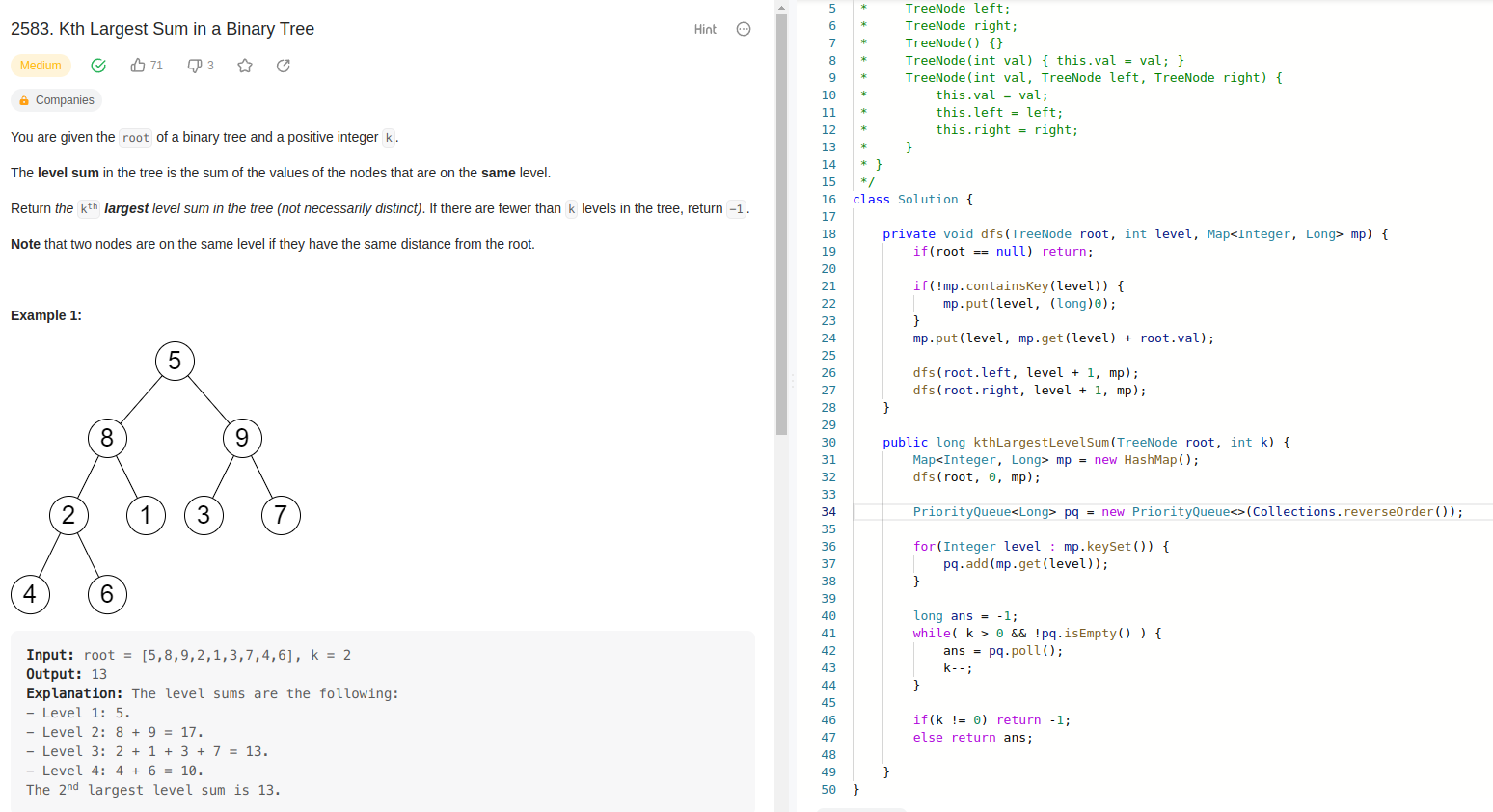
SC: O(H + N) call stack + N treenodes

****

**Kth largest sum in a binary tree using DFS:**

TC: O(N)

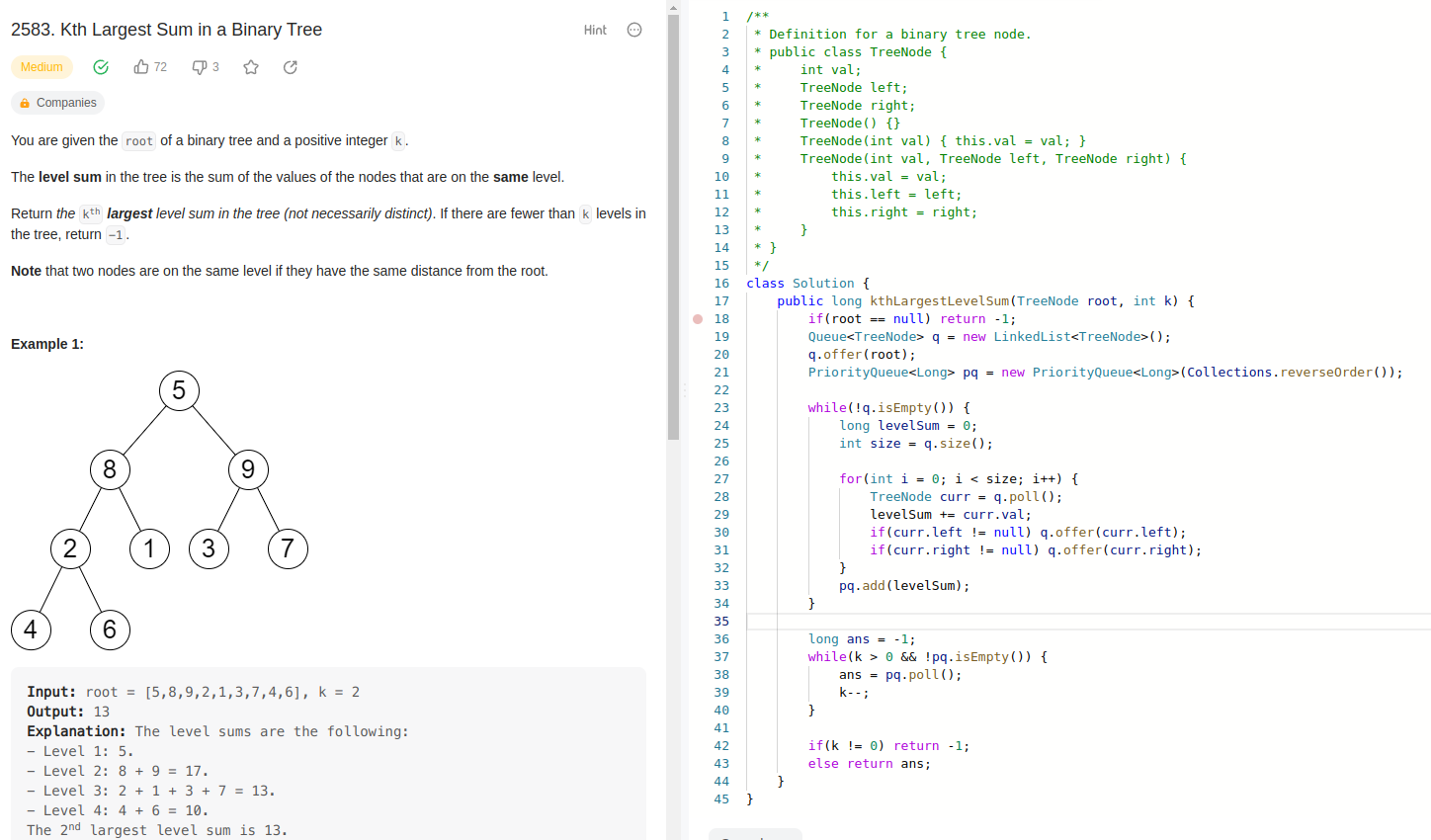
SC: O(H + H) for call stack and priority queue



**Kth largest sum in a binary tree using BFS:**

TC: O(N)

SC: O(H + H) for queue and priority queue

****