

Diameter of Binary Tree [LeetCode](#)

Example:

```
    13
   /  \
  15   1
 /  \
7    8
```

Output: The Diameter of Binary Tree: 4

Approach 1: Function to calculate the diameter of a binary tree

- In the **diameterOfBinaryTree** function, we calculate the diameter of the binary tree using a recursive approach.
- We find the diameter of the left subtree and the diameter of the right subtree recursively.
- The diameter of the current tree is the maximum of the diameters of the left and right subtrees or the sum of heights of the left and right subtrees.
- The height of a tree is calculated by the **height** function, which calculates the height of the left and right subtrees recursively and returns the maximum height plus 1 for the current level.

Time Complexity: $O(N^2)$ in the worst case, where N is the number of nodes in the binary tree. This is because for each node, we may calculate the heights of its left and right subtrees.

Space Complexity: $O(H)$, where H is the height of the binary tree due to the function call stack.

Approach 2: Optimized function to calculate the diameter of a binary tree

- In the **diameterHelper** function, we calculate both the diameter and height of the binary tree simultaneously using a recursive approach.
- The function returns a pair with the first element representing the diameter and the second element representing the height.
- At each node, we calculate the diameter and height of the left and right subtrees recursively.

- The diameter of the current tree is the maximum of the diameters of the left and right subtrees or the sum of heights of the left and right subtrees.
- The height of the current tree is the maximum height of the left and right subtrees plus 1.
- The **diameterOfBinaryTreeOptimized** function calls **diameterHelper** and returns the diameter part of the pair.

Time Complexity: $O(N)$, where N is the number of nodes in the binary tree. We visit each node once.

Space Complexity: $O(H)$, where H is the height of the binary tree due to the function call stack.