

Maximum Height of Binary Tree [LeetCode](#)

The program for finding the maximum height of a binary tree

Example:



Output: The Maximum Height of Binary Tree: 5

Approach 1: Function to find the maximum height of the binary tree recursively

- In the **maxHeightRecursively** function, we calculate the maximum height of the binary tree recursively.
- If the tree is empty (root is null), its height is 0.
- We calculate the height of the left and right subtrees recursively and return the maximum of these heights, adding 1 for the current level.

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.

Approach 2: Function to find the maximum height of the binary tree iteratively

- In the **maxHeightIteratively** function, we calculate the maximum height of the binary tree iteratively using a queue.
- We start with the root node and process levels.
- We enqueue all nodes at the current level and increment the height for each level.

- The final height is the maximum height of the tree.

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 creation of Queue.