刷題筆記

Data Structures

Java Concepts

Algorithms

BINARY SEARCH

Binary Search Template

三步翻轉法

4. Median of Two Sorted Arrays

35. Search Insert Position

33. Search in Rotated Sorted Array

81. Search in Rotated Sorted Array II

278. First Bad Version

88. Merge Sorted Array

103. Binary Tree Zigzag Level Order Traversal

153. Find Minimum in Rotated Sorted Array

4. Median of Two Sorted Arrays

124. Binary Tree Maximum Path Sum

Given a **non-empty** binary tree, find the maximum path sum.

For this problem, a path is defined as any sequence of nodes from some starting node to any node in the tree along the parent-child connections. The path must contain **at least one node** and does not need to go through the root.

Example 1:

```
Input: [1,2,3]

       1
      / \
     2   3

Output: 6
```

Example 2:

```
Input: [-10,9,20,null,null,15,7]

   -10
   / \
  9  20
    /  \
   15   7

Output: 42
```

Time: O(n)

Space:O(lgn) n = nodes #

```
class Solution {
    int max;
    public int maxPathSum(TreeNode root) {
        max = Integer.MIN_VALUE;
        helper(root);
        return max;
    }
    private int helper(TreeNode root) {
        if (root == null) return 0;
        int l = Math.max(0, helper(root.left));
        int r = Math.max(0, helper(root.right));
        max = Math.max(max, l + r + root.val);
        return Math.max(l, r) + root.val;
    }
}
```

TREE & DFS & BFS - Previous
Dfs template

Next - TREE & DFS & BFS
110. Balanced Binary Tree