题目地址(104. 二叉树的最大深度)

https://leetcode-cn.com/problems/maximum-depth-of-binary-tree

标签

- DFS
- 树

难度

- 简单

入选理由

- 这是一个难度为 easy 的题目，适合作为第一题。
- 此题适合练习递归。
- 这是一个非常常见的考点，只不过有的时候是作为题目的一部分出现，而不是单独考察而已。

题目描述

给定一个二叉树，找出其最大深度。

二叉树的深度为根节点到最远叶子节点的最长路径上的节点数。

说明: 叶子节点是指没有子节点的节点。

示例:
给定二叉树 [3,9,20,null,null,15,7],

```
    3
   / \
  9  20
    /  \
   15   7
```

返回它的最大深度 3 。

前置知识

- 递归

思路

树的题目很适合用来递归来做。 基本上和树的搜索有关的，都可以用递归来做，为什么?

因为树是一种递归的数据结构。而穷举搜索一棵树必然需要遍历其所有节点，而搜索的逻辑对所有的子树都是一样的。因此这就很适合用递归来解决了。

这里给大家介绍一种写递归的小方法 **产品经理法**。

1. 定义函数功能，不用管其具体实现。

从高层次的角度来定义函数功能。 你可以把自己想象成**产品经理**。只需要知道要做什么事情就行了，而怎么实现我不管，那是码农的事情。

具体来说，我需要的功能是**给定一个二叉树的节点，返回以这个节点为根节点的子树的最大深度**。假设这个函数为 f。那么问题转化为 f(root)。

2. 确定大问题和小问题的关系。

要解决 f(root) 这个问题。可以先解决 f(root.right) 和 f(root.left)，当然我们仍然不关心 f 怎么实现。

f(root) 与 f(root.right) 和 f(root.left) 有什么关系呢? 不难看出 `1 + max(f(root.right), f(root.left))` 。

到这里我们还不知道 f 怎么实现的，但是我们已经完成了产品经理的需求。

> 实际上我们知道了，我们怎么知道的?

3. 补充递归终止条件。

如果递归到叶子节点的时候，返回 0 即可。

代码

代码支持：Python3，Java，JS，CPP

Python Code：

```
# Definition for a binary tree node.
class Solution:
    def maxDepth(self, root: TreeNode) -> int:
        if not root: return 0
        return 1 + max(self.maxDepth(root.left), self.maxDepth(root.right))
```

Java Code:

```java
class Solution {
    public int maxDepth(TreeNode root) {
        if(root == null){
            return 0;
        }
        return Math.max(maxDepth(root.left), maxDepth(root.right)) + 1;

    }
}
```

JS Code:

```javascript
/**
 * @param {TreeNode} root
 * @return {number}
 */
var maxDepth = function (root) {
  if (root === null) {
    return 0;
  }
  return Math.max(maxDepth(root.left), maxDepth(root.right)) + 1;
};
```

CPP Code:

```cpp
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if(!root) return 0;
        int left = maxDepth(root->left);
        int right = maxDepth(root->right);
        return max(left, right) + 1;
    }
};
```

复杂度分析

- 时间复杂度：$O(N)$，其中 N 为节点数。
- 空间复杂度：$O(h)$，其中 h 为树的深度，最坏的情况 h 等于 N，其中 N 为节点数，此时树退化到链表。

扩展

大家也可以使用层次遍历的方式来解决，具体来说可以使用队列做 BFS，直接使用带层信息的 BFS 模板即可。关于 BFS 我们会在专题篇的《搜索》进行更详细的介绍。

上一页 下一页