

0297. 二叉树的序列化与反序列化

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：树、深度优先搜索、广度优先搜索、设计、字符串、二叉树
- 难度：困难

题目链接

- [0297. 二叉树的序列化与反序列化 - 力扣](#)

题目大意

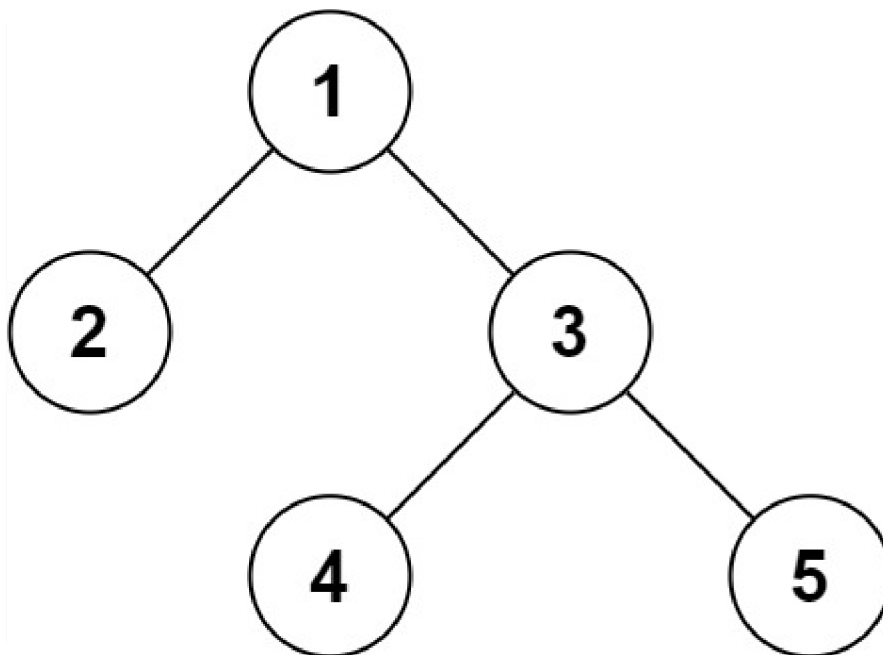
要求：设计一个算法，来实现二叉树的序列化与反序列化。

说明：

- 不限定序列化 / 反序列化算法执行逻辑，只需要保证一个二叉树可以被序列化为一个字符串并且将这个字符串反序列化为原始的树结构。
- 树中结点数在范围 $[0, 10^4]$ 内。
- $-1000 \leq Node.val \leq 1000$ 。

示例：

- 示例 1：



输入: root = [1,2,3,null,null,4,5]

输出: [1,2,3,null,null,4,5]

py

- 示例 2:

输入: root = [1,2]

输出: [1,2]

py

解题思路

思路 1：深度优先搜索

1. 序列化：将二叉树转为字符串数据表示

1. 按照前序顺序递归遍历二叉树，并将根节点跟左右子树的值链接起来（中间用 `,` 隔开）。

注意：如果遇到空节点，则将其标记为 `None`，这样在反序列化时才能唯一确定一棵二叉树。

2. 反序列化：将字符串数据转为二叉树结构

1. 先将字符串按 `,` 分割成数组。然后递归处理每一个元素。
2. 从数组左侧取出一个元素。
 1. 如果当前元素为 `None`，则返回 `None`。
 2. 如果当前元素不为空，则新建一个二叉树节点作为根节点，保存值为当前元素值。并递归遍历左右子树，不断重复从数组中取出元素，进行判断。
3. 最后返回当前根节点。

思路 1：代码

```
class Codec:

    def serialize(self, root):
        """Encodes a tree to a single string.

        :type root: TreeNode
        :rtype: str
        """
        if not root:
            return 'None'
        return str(root.val) + ',' + str(self.serialize(root.left)) + ',' + str(self.serialize(root.right))

    def deserialize(self, data):
        """Decodes your encoded data to tree.

        :type data: str
        :rtype: TreeNode
        """
        def dfs(datalist):
            val = datalist.pop(0)
            if val == 'None':
                return None
            root = TreeNode(int(val))
            root.left = dfs(datalist)
        return dfs(data.split(','))
```

py

```
        root.right = dfs(datalist)
    return root

datalist = data.split(',')
return dfs(datalist)
```

思路 1：复杂度分析

- 时间复杂度： $O(n)$ ，其中 n 为二叉树的节点数。
- 空间复杂度： $O(n)$ 。