

高频算法题

刷题地址: <https://leetcode-cn.com/>

刷题vip账号可扫水印二维码加微信索取

1. 旋转数组的最小数字 (剑指 Offer 11 难度: easy)

题目: 把一个数组最开始的若干个元素搬到数组的末尾, 我们称之为数组的旋转。输入一个递增排序的数组的一个旋转, 输出旋转数组的最小元素。例如, 数组 $[3, 4, 5, 1, 2]$ 为 $[1, 2, 3, 4, 5]$ 的一个旋转, 该数组的最小值为1。

示例 1:

输入: $[3, 4, 5, 1, 2]$

输出: 1

示例 2:

输入: $[2, 2, 2, 0, 1]$

输出: 0

2. 合并两个有序链表 (难度: easy)

将两个升序链表合并为一个新的 升序 链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。

示例:

输入: $1 \rightarrow 2 \rightarrow 4, 1 \rightarrow 3 \rightarrow 4$

输出: $1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 4$

2. 数组中的第K个最大元素 (TopK 问题 难度: medium)

在未排序的数组中找到第 k 个最大的元素。请注意, 你需要找的是数组排序后的第 k 个最大的元素, 而不是第 k 个不同的元素。

示例 1:

输入: $[3, 2, 1, 5, 6, 4]$ 和 $k = 2$

输出: 5

示例 2:

输入: $[3, 2, 3, 1, 2, 4, 5, 5, 6]$ 和 $k = 4$

输出: 4

说明: 你可以假设 k 总是有效的, 且 $1 \leq k \leq$ 数组的长度。



3. 二叉树的最近公共祖先 (难度: medium)

给定一个二叉树, 找到该树中两个指定节点的最近公共祖先。

最近公共祖先的定义为: “对于有根树 T 的两个结点 p、q, 最近公共祖先表示为一个结点 x, 满足 x 是 p、q 的祖先且 x 的深度尽可能大 (一个节点也可以是它自己的祖先)。”

例如, 给定如下二叉树: root = [3,5,1,6,2,0,8,null,null,7,4]

示例 1:

输入: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1

输出: 3

解释: 节点 5 和节点 1 的最近公共祖先是节点 3。

示例 2:

输入: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 4

输出: 5

解释: 节点 5 和节点 4 的最近公共祖先是节点 5。因为根据定义最近公共祖先节点可以为节点本身。

说明:

所有节点的值都是唯一的。

p、q 为不同节点且均存在于给定的二叉树中。

二叉树另外一个常考问题:

- 输出二叉树节点数最多的那一层的节点数

4. 二分查找 (要求非递归方式)

二分查找详解: <https://www.cnblogs.com/kyoner/p/11080078.html>

递归与非递归写法: <https://blog.csdn.net/u011969595/article/details/91903090>

找出两个字符串中相同的片段

5. 岛屿数量 (主要考察DFS) 难度: medium

给你一个由 '1' (陆地) 和 '0' (水) 组成的二维网格, 请你计算网格中岛屿的数量。

岛屿总是被水包围, 并且每座岛屿只能由水平方向或竖直方向上相邻的陆地连接形成。

此外, 你可以假设该网格的四条边均被水包围。

示例 1:

输入:

[
['1','1','1','1','0'],



```
['1','1','0','1','0'],
['1','1','0','0','0'],
['0','0','0','0','0']
]
```

输出: 1

示例 2:

输入:

```
[
['1','1','0','0','0'],
['1','1','0','0','0'],
['0','0','1','0','0'],
['0','0','0','1','1']
]
```

输出: 3

解释: 每座岛屿只能由水平和/或竖直方向上相邻的陆地连接而成。

思路:

- 1、线性遍历二维数组的所有数据
- 2、遇到1时，岛屿数目+1，开始深度优先遍历找到整个岛屿，并在遍历过程中把找到的陆地置为0
- 3、深度优先算法 使用

6. 复原IP地址 （考察：DFS 回溯 剪枝 难度：中等）

给定一个只包含数字的字符串，复原它并返回所有可能的 IP 地址格式。

有效的 IP 地址正好由四个整数（每个整数位于 0 到 255 之间组成），整数之间用 '.' 分隔。

示例:

输入: "25525511135"

输出: ["255.255.11.135", "255.255.111.35"]

