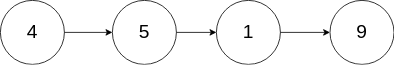
一、编程题 （45 min）

1. 请编写一个函数，使其可以删除某个链表中给定的（非末尾）节点。传入函数的唯一参数为要被删除的节点。

现有一个链表 -- head = [4,5,1,9]，它可以表示为:



示例 1：

输入：head = [4,5,1,9], node = 5

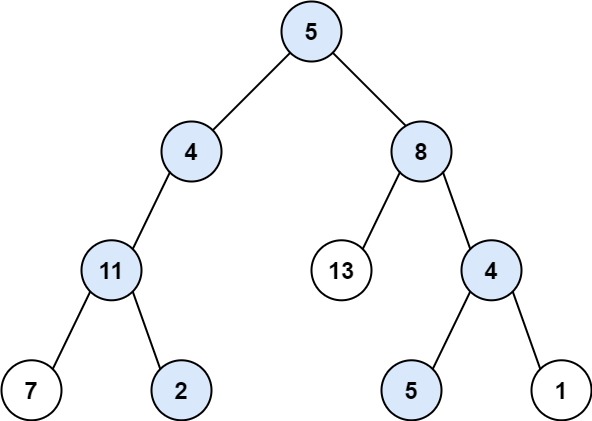
输出：[4,1,9]

解释：给定你链表中值为 5 的第二个节点，那么在调用了你的函数之后，该链表应变为 4 -> 1 -> 9.

2. 给你二叉树的根节点 root 和一个整数目标和 targetSum，找出所有从根节点到叶子节点路径总和等于给定目标和的路径。

叶子节点 是指没有子节点的节点。

示例 1：



输入：root = [5,4,8,11,null,13,4,7,2,null,null,5,1], targetSum = 22

输出：[[5,4,11,2],[5,8,4,5]]

3. 给定正整数 n，找到若干个完全平方数（比如 1, 4, 9, 16, ...）使得它们的和等于 n。你需要让组成和的完全平方数的个数最少。

给你一个整数 n ，返回和为 n 的完全平方数的 最少数量 。

完全平方数 是一个整数，其值等于另一个整数的平方；换句话说，其值等于一个整数自乘的积。例如，1、4、9 和 16 都是完全平方数，而 3 和 11 不是。

示例 1：

输入：n = 12

输出：3

解释：12 = 4 + 4 + 4

提示：

1 <= n <= 104

4. (附加题) 有 n 个气球，编号为0 到 n - 1，每个气球上都标有一个数字，这些数字存在数组 nums 中。

现在要求你戳破所有的气球。戳破第 i 个气球，你可以获得 nums[i - 1] \* nums[i] \* nums[i + 1] 枚硬币。 这里的 i - 1 和 i + 1 代表和 i 相邻的两个气球的序号。如果 i - 1或 i + 1 超出了数组的边界，那么就当它是一个数字为 1 的气球。

求所能获得硬币的最大数量。

示例 1：

输入：nums = [3,1,5,8]

输出：167

解释：

nums = [3,1,5,8] --> [3,5,8] --> [3,8] --> [8] --> []

coins = 3\*1\*5 + 3\*5\*8 + 1\*3\*8 + 1\*8\*1 = 167