# 题目

给你一个正整数 num ，请你将它分割成两个非负整数 num1 和 num2 ，满足：

num1 和 num2 直接连起来，得到 num 各数位的一个排列。

换句话说，num1 和 num2 中所有数字出现的次数之和等于 num 中所有数字出现的次数。

num1 和 num2 可以包含前导 0 。

请你返回 num1 和 num2 可以得到的和的 最小 值。

注意：

num 保证没有前导 0 。

num1 和 num2 中数位顺序可以与 num 中数位顺序不同。

示例 1：

输入：num = 4325

输出：59

解释：我们可以将 4325 分割成 num1 = 24 和 num2 = 35 ，和为 59 ，59 是最小和。

示例 2：

输入：num = 687

输出：75

解释：我们可以将 687 分割成 num1 = 68 和 num2 = 7 ，和为最优值 75 。

提示：

10 <= num <= 10^9

# 分析

## 方法一：排序

class Solution {

public:

int splitNum(int num) {

string s = to\_string(num);

sort(s.begin(), s.end());

int a[2]{};

for (int i = 0; i < s.length(); i++)

a[i % 2] = a[i % 2] \* 10 + s[i] - '0'; // 按照奇偶下标分组

return a[0] + a[1];

}

};