# 题目

给你一个下标从 0 开始的字符串 s ，以及一个下标从 0 开始的整数数组 spaces 。

数组 spaces 描述原字符串中需要添加空格的下标。每个空格都应该插入到给定索引处的字符值 之前 。

例如，s = "EnjoyYourCoffee" 且 spaces = [5, 9] ，那么我们需要在 'Y' 和 'C' 之前添加空格，这两个字符分别位于下标 5 和下标 9 。因此，最终得到 "Enjoy Your Coffee" 。

请你添加空格，并返回修改后的字符串。

示例 1：

输入：s = "LeetcodeHelpsMeLearn", spaces = [8,13,15]

输出："Leetcode Helps Me Learn"

解释：

下标 8、13 和 15 对应 "LeetcodeHelpsMeLearn" 中加粗斜体字符。

接着在这些字符前添加空格。

示例 2：

输入：s = "icodeinpython", spaces = [1,5,7,9]

输出："i code in py thon"

解释：

下标 1、5、7 和 9 对应 "icodeinpython" 中加粗斜体字符。

接着在这些字符前添加空格。

示例 3：

输入：s = "spacing", spaces = [0,1,2,3,4,5,6]

输出：" s p a c i n g"

解释：

字符串的第一个字符前可以添加空格。

提示：

1 <= s.length <= 3 \* 105

s 仅由大小写英文字母组成

1 <= spaces.length <= 3 \* 105

0 <= spaces[i] <= s.length - 1

spaces 中的所有值 严格递增

# 分析

## 方法一：模拟

思路：

1、遍历数组 spaces，对于每个空格位置 idx，在字符串 s 中的位置为 idx，则在 s 的 idx 位置前插入一个空格。

2、插入完所有空格后，返回修改后的字符串。

代码：

class Solution {

public:

string addSpaces(string s, vector<int>& spaces) {

int offset = 0; // 记录已插入空格数目对字符串的偏移量

for (int idx : spaces) {

s.insert(idx + offset, " "); // 在idx+offset位置插入空格

offset++; // 更新偏移量

}

return s;

}

};