# 题目

给你一个字符串 s 和一个字符 c ，且 c 是 s 中出现过的字符。

返回一个整数数组 answer ，其中 answer.length == s.length 且 answer[i] 是 s 中从下标 i 到离它 最近 的字符 c 的 距离 。

两个下标 i 和 j 之间的 距离 为 abs(i - j) ，其中 abs 是绝对值函数。

示例 1：

输入：s = "loveleetcode", c = "e"

输出：[3,2,1,0,1,0,0,1,2,2,1,0]

解释：字符 'e' 出现在下标 3、5、6 和 11 处（下标从 0 开始计数）。

距下标0最近的'e'出现在下标3，所以距离为abs(0 - 3) = 3 。

距下标1最近的'e'出现在下标3，所以距离为abs(1 - 3) = 2 。

对于下标4，出现在下标3和下标5处的'e'都离它最近，但距离是一样的abs(4 - 3) == abs(4 - 5) = 1。

距下标8最近的'e'出现在下标6，所以距离为abs(8 - 6) = 2。

示例 2：

输入：s = "aaab", c = "b"

输出：[3,2,1,0]

提示：

1 <= s.length <= 104

s[i] 和 c 均为小写英文字母

题目数据保证 c 在 s 中至少出现一次

# 分析

## 方法一：两次遍历

class Solution {

public:

vector<int> shortestToChar(string s, char c) {

int n = s.length();

vector<int> ans(n);

for (int i = 0, idx = -n; i < n; ++i) {

if (s[i] == c) {

idx = i;

}

ans[i] = i - idx;

}

for (int i = n - 1, idx = 2 \* n; i >= 0; --i) {

if (s[i] == c) {

idx = i;

}

ans[i] = min(ans[i], idx - i);

}

return ans;

}

};