Problem 821: Shortest Distance to a Character

Problem Information

Difficulty: Easy
Acceptance Rate: 72.53%
Paid Only: No
Tags: Array, Two Pointers, String

Problem Description

Given a string `s` and a character `c` that occurs in `s`, return _an array of integers_`answer` _where_`answer.length == s.length` _and_`answer[i]`_is the**distance** from index _`i` _to the**closest** occurrence of character _`c` _in_`s`.

The **distance** between two indices `i` and `j` is `abs(i - j)`, where `abs` is the absolute value function.

Example 1:

Input: s = "loveleetcode", c = "e" **Output:** [3,2,1,0,1,0,0,1,2,2,1,0] **Explanation:** The character 'e' appears at indices 3, 5, 6, and 11 (0-indexed). The closest occurrence of 'e' for index 0 is at index 3, so the distance is abs(0 - 3) = 3. The closest occurrence of 'e' for index 1 is at index 3, so the distance is abs(1 - 3) = 2. For index 4, there is a tie between the 'e' at index 3 and the 'e' at index 5, but the distance is still the same: abs(4 - 3) == abs(4 - 5) = 1. The closest occurrence of 'e' for index 8 is at index 6, so the distance is abs(8 - 6) = 2.

Example 2:

Input: s = "aaab", c = "b" **Output:** [3,2,1,0]

Constraints:

* `1 <= s.length <= 104` * `s[i]` and `c` are lowercase English letters. * It is guaranteed that `c` occurs at least once in `s`.

Code Snippets

C++:

```
class Solution {
public:
vector<int> shortestToChar(string s, char c) {

}
};
```

Java:

```
class Solution {
public int[] shortestToChar(String s, char c) {

}
}
```

Python3:

```
class Solution:
def shortestToChar(self, s: str, c: str) -> List[int]:
```