


## 821. Shortest Distance to a Character

Solved 

Easy

Topics

Companies

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"`

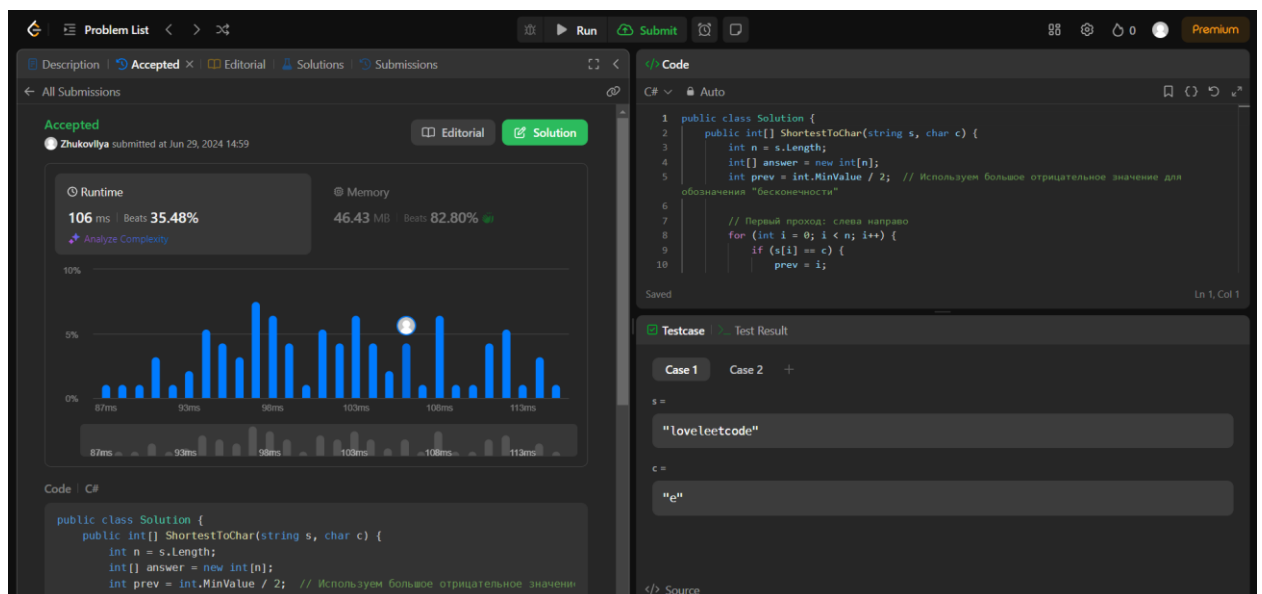
**Output:** `[3,2,1,0,1,0,0,1,2,2,1,0]`

**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]`



The screenshot displays a code editor interface for a C# solution. On the left, a sidebar shows the problem description and a submission by 'Zhukovilya' with a runtime of 106 ms and memory of 46.43 MB. The main editor shows the C# code, and the right sidebar shows the test case input: `s = "loveleetcode"` and `c = "e"`.

```
public class Solution {
    public int[] ShortestToChar(string s, char c) {
        int n = s.Length;
        int[] answer = new int[n];
        int prev = int.MinValue / 2; // Используем большое отрицательное значение для обозначения "бесконечности"

        // Первый проход: слева направо
        for (int i = 0; i < n; i++) {
            if (s[i] == c) {
                prev = i;
            }
        }

        // Второй проход: справа налево
        for (int i = n - 1; i >= 0; i--) {
            if (s[i] == c) {
                prev = i;
            }
            answer[i] = Math.Min(i - prev, prev - i);
        }

        return answer;
    }
}
```

Код:

```
using System;
```

```
public class Solution
```

```
{
```

```
    public int[] ShortestToChar(string s, char c)
```

```
    {
```

```
        int n = s.Length;
```

```

        int[] answer = new int[n];
        int prev = int.MinValue / 2; // Используем большое отрицательное значение
        для обозначения "бесконечности"

        // Первый проход: слева направо
        for (int i = 0; i < n; i++)
        {
            if (s[i] == c)
            {
                prev = i;
            }
            answer[i] = i - prev;
        }

        prev = int.MaxValue / 2; // Используем большое положительное значение для
        обозначения "бесконечности"

        // Второй проход: справа налево
        for (int i = n - 1; i >= 0; i--)
        {
            if (s[i] == c)
            {
                prev = i;
            }
            answer[i] = Math.Min(answer[i], prev - i);
        }

        return answer;
    }
}

```