

859. Buddy Strings

Solved

Easy

Topics

Companies

Given two strings `s` and `goal`, return `true` if you can swap two letters in `s` so the result is equal to `goal`, otherwise, return `false`.

Swapping letters is defined as taking two indices `i` and `j` (0-indexed) such that `i != j` and swapping the characters at `s[i]` and `s[j]`.

- For example, swapping at indices `0` and `2` in `"abcd"` results in `"cbad"`.

Example 1:

Input: `s = "ab", goal = "ba"`

Output: `true`

Explanation: You can swap `s[0] = 'a'` and `s[1] = 'b'` to get `"ba"`, which is equal to `goal`.

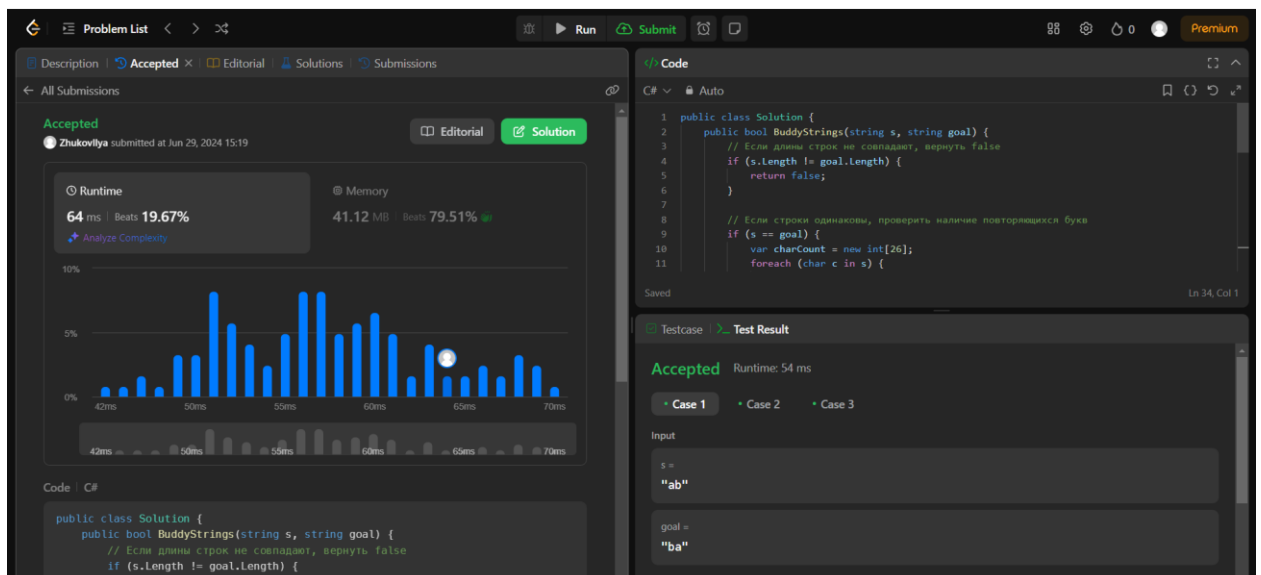
Example 2:

Input: `s = "ab", goal = "ab"`

Output: `false`

Explanation: The only letters you can swap are `s[0] = 'a'` and `s[1] = 'b'`, which results in `"ba" != goal`.

Example 3:



Код:

```

using System.Collections.Generic;

public class Solution
{
    public bool BuddyStrings(string s, string goal)

```

```

{
    // Если длины строк не совпадают, вернуть false
    if (s.Length != goal.Length)
    {
        return false;
    }

    // Если строки одинаковы, проверить наличие повторяющихся букв
    if (s == goal)
    {
        var charCount = new int[26];
        foreach (char c in s)
        {
            charCount[c - 'a']++;
            if (charCount[c - 'a'] > 1)
            {
                return true; // Найдена повторяющаяся буква
            }
        }
        return false; // Повторяющихся букв нет
    }

    // Найти индексы, где строки отличаются
    List<int> diff = new List<int>();
    for (int i = 0; i < s.Length; i++)
    {
        if (s[i] != goal[i])
        {
            diff.Add(i);
        }
    }

    // Проверить, что количество различий ровно два и можно ли их поменять
    return diff.Count == 2 &&
        s[diff[0]] == goal[diff[1]] &&
        s[diff[1]] == goal[diff[0]];
}
}

```