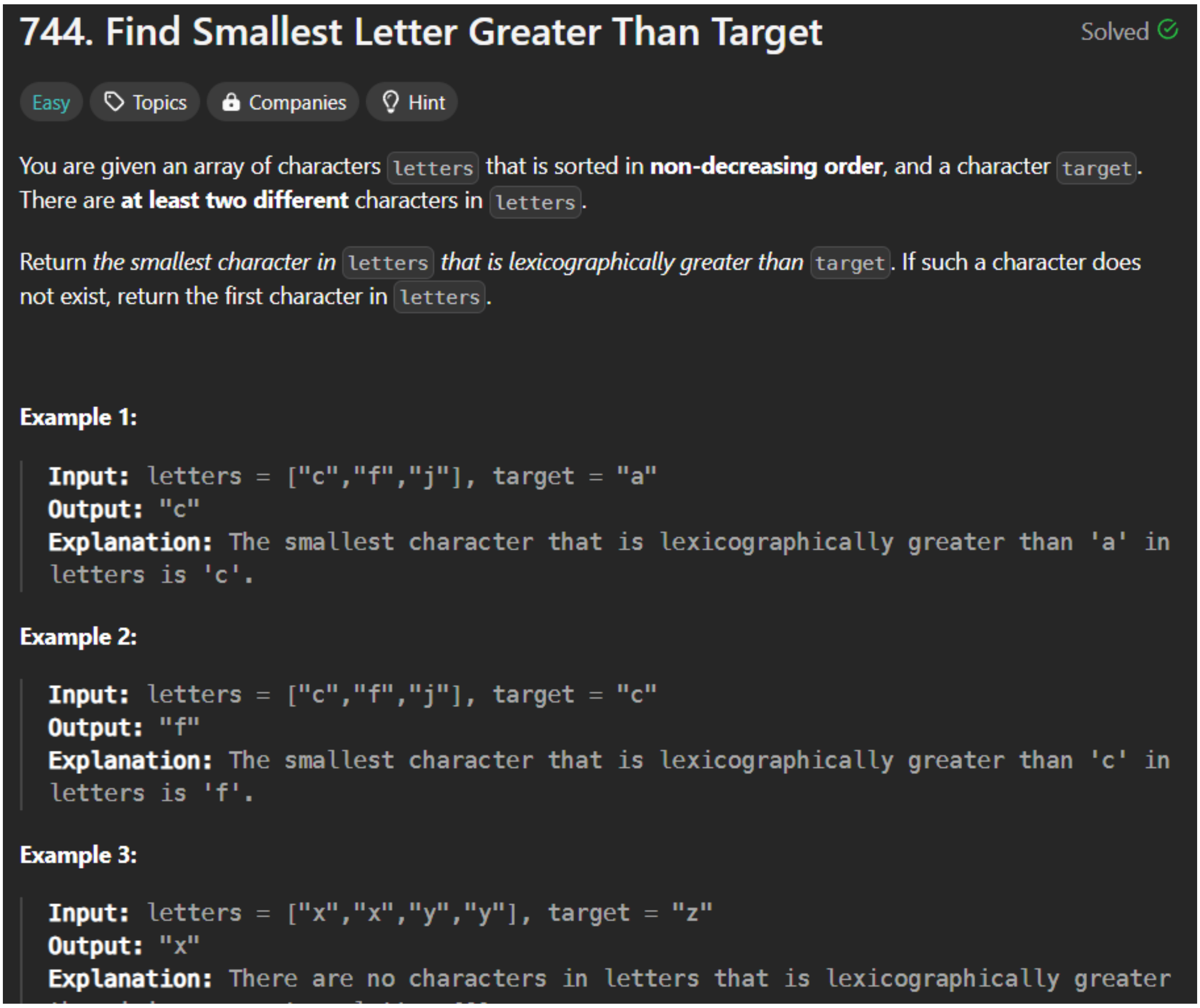


744. Find Smallest Letter Greater Than Target

Solved ✓

You are given an array of characters `letters` that is sorted in **non-decreasing order**, and a character `target`. There are **at least two different** characters in `letters`.

Return *the smallest character in* `letters` *that is lexicographically greater than* `target`. If such a character does not exist, return the first character in `letters`.

Example 1:

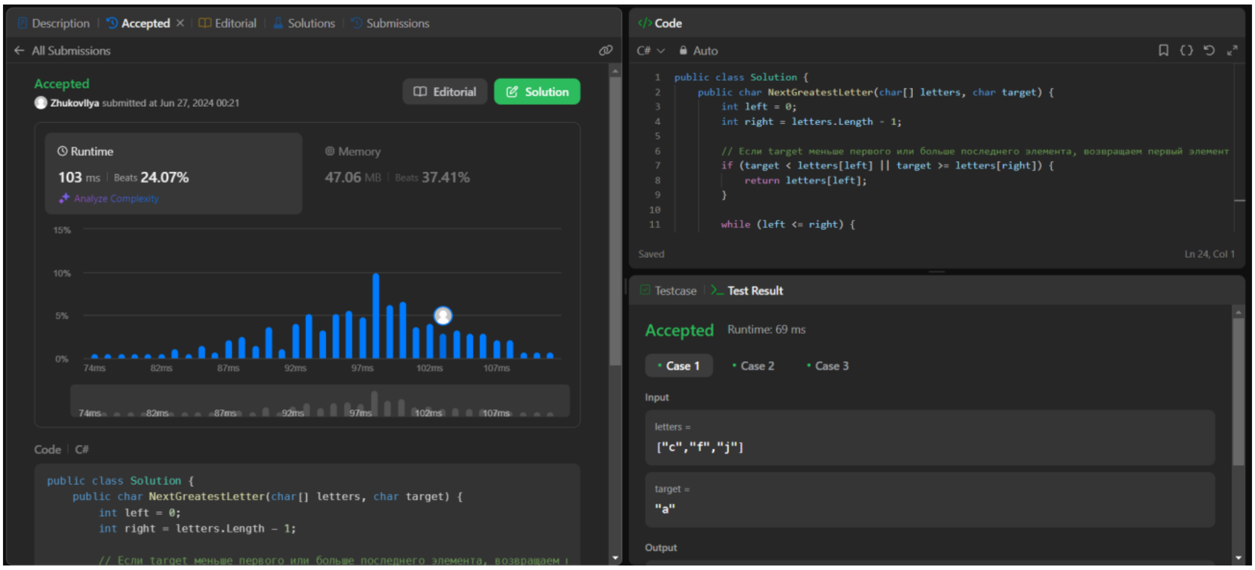
```
Input: letters = ["c","f","j"], target = "a"
Output: "c"
Explanation: The smallest character that is lexicographically greater than 'a' in
letters is 'c'.
```

Example 2:

```
Input: letters = ["c","f","j"], target = "c"
Output: "f"
Explanation: The smallest character that is lexicographically greater than 'c' in
letters is 'f'.
```

Example 3:

```
Input: letters = ["x","x","y","y"], target = "z"
Output: "x"
Explanation: There are no characters in letters that is lexicographically greater
```



Accepted

Zhukovllya submitted at Jun 27, 2024 00:21

Runtime: 103 ms | Beats 24.07%

Memory: 47.06 MB | Beats 37.41%

Code | C#

```
public class Solution {
    public char NextGreatestLetter(char[] letters, char target) {
        int left = 0;
        int right = letters.Length - 1;

        // Если target меньше первого или больше последнего элемента, возвращаем
```

```
public class Solution {
    public char NextGreatestLetter(char[] letters, char target) {
        int left = 0;
        int right = letters.Length - 1;

        // Если target меньше первого или больше последнего элемента, возвращаем первый элемент
        if (target < letters[left] || target >= letters[right]) {
            return letters[left];
        }

        while (left <= right) {
```

Accepted Runtime: 69 ms

Input

letters =
["c","f","j"]

target =
"a"

Output

Код:

```
public class Solution
{
    public char NextGreatestLetter(char[] letters, char target)
    {
        int left = 0;
        int right = letters.Length - 1;
        // Если target меньше первого или больше последнего элемента, возвращаем
первый элемент
        if (target < letters[left] || target >= letters[right])
        {
            return letters[left];
        }

        while (left <= right)
        {
            int mid = left + (right - left) / 2;
            if (letters[mid] > target)
            {
                right = mid - 1;
            }
            else
            {
                left = mid + 1;
            }
        }

        // left будет указывать на наименьший элемент, который больше target
        return letters[left];
    }
}
```