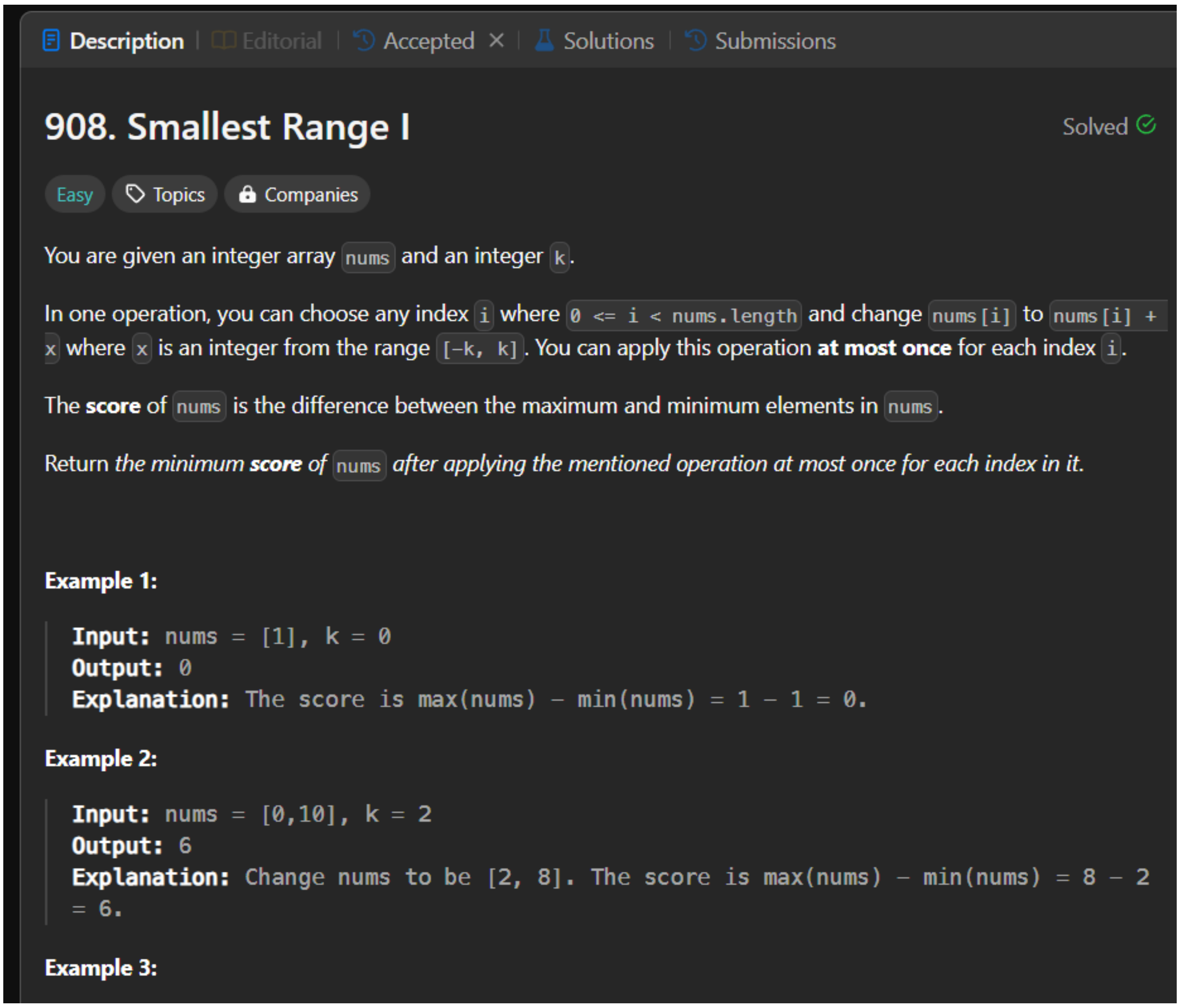

Description | Editorial | Accepted × | Solutions | Submissions

908. Smallest Range I

Solved ✓

You are given an integer array `nums` and an integer `k`.

In one operation, you can choose any index `i` where `0 <= i < nums.length` and change `nums[i]` to `nums[i] + x` where `x` is an integer from the range `[-k, k]`. You can apply this operation **at most once** for each index `i`.

The **score** of `nums` is the difference between the maximum and minimum elements in `nums`.

Return *the minimum* ***score*** *of* `nums` *after applying the mentioned operation at most once for each index in it.*

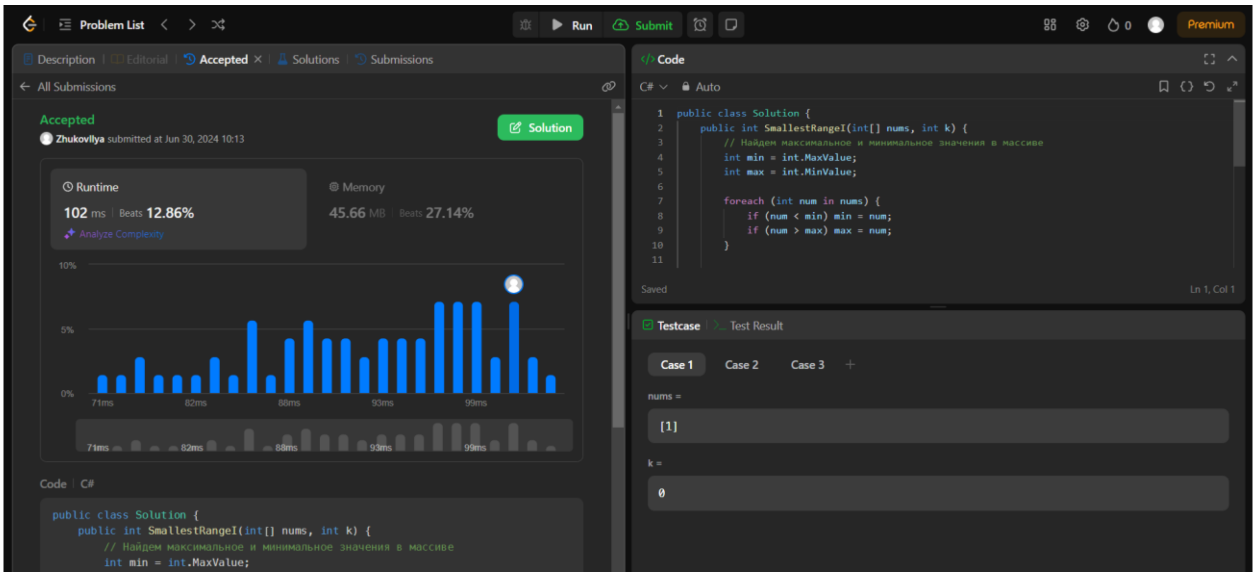
Example 1:

```
Input: nums = [1], k = 0
Output: 0
Explanation: The score is max(nums) - min(nums) = 1 - 1 = 0.
```

Example 2:

```
Input: nums = [0,10], k = 2
Output: 6
Explanation: Change nums to be [2, 8]. The score is max(nums) - min(nums) = 8 - 2
= 6.
```

Example 3:



Accepted
Zhukovllya submitted at Jun 30, 2024 10:13

Runtime
102 ms | Beats 12.86%

Memory
45.66 MB | Beats 27.14%

Code | C#

```
public class Solution {
    public int SmallestRangeI(int[] nums, int k) {
        // Найдем максимальное и минимальное значения в массиве
        int min = int.MaxValue;
```

```
public class Solution {
    public int SmallestRangeI(int[] nums, int k) {
        // Найдем максимальное и минимальное значения в массиве
        int min = int.MaxValue;
        int max = int.MinValue;

        foreach (int num in nums) {
            if (num < min) min = num;
            if (num > max) max = num;
        }

```

nums =

```
[1]
```

k =

```
0
```

Код:

```
public class Solution
{
    public int SmallestRangeI(int[] nums, int k)
    {
        // Найдем максимальное и минимальное значения в массиве
        int min = int.MaxValue;
        int max = int.MinValue;

        foreach (int num in nums)
        {
            if (num < min) min = num;
            if (num > max) max = num;
        }

        // Вычислим новую разницу
        int newMin = min + k;
        int newMax = max - k;

        // Если newMin больше или равен newMax, то разница будет 0
        if (newMin >= newMax)
        {
            return 0;
        }

        // Иначе возвращаем разницу
        return newMax - newMin;
    }
}
```