


## 643. Maximum Average Subarray I

Solved 

Easy

Topics

Companies

You are given an integer array `nums` consisting of `n` elements, and an integer `k`.

Find a contiguous subarray whose **length is equal to** `k` that has the maximum average value and return *this value*. Any answer with a calculation error less than  $10^{-5}$  will be accepted.

### Example 1:

**Input:** `nums = [1,12,-5,-6,50,3]`, `k = 4`

**Output:** `12.75000`

**Explanation:** Maximum average is  $(12 - 5 - 6 + 50) / 4 = 51 / 4 = 12.75$

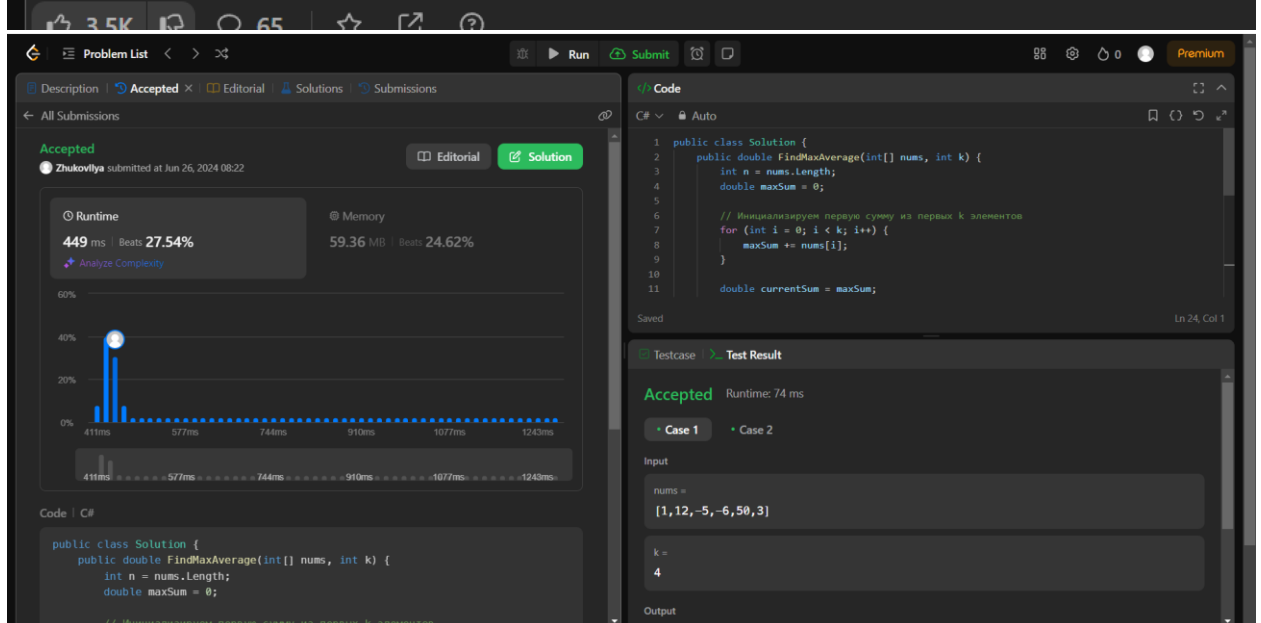
### Example 2:

**Input:** `nums = [5]`, `k = 1`

**Output:** `5.00000`

### Constraints:

- `n == nums.length`
- `1 <= k <= n <= 10^5`
- `-10^4 <= nums[i] <= 10^4`



The screenshot displays a code editor interface for the problem "Maximum Average Subarray I". The code is written in C# and implements a sliding window algorithm to find the maximum average subarray of length `k`. The code is as follows:

```
public class Solution {
    public double FindMaxAverage(int[] nums, int k) {
        int n = nums.Length;
        double maxSum = 0;

        // Инициализируем первую сумму из первых k элементов
        for (int i = 0; i < k; i++) {
            maxSum += nums[i];
        }

        double currentSum = maxSum;
```

The editor also shows runtime statistics: 449 ms (Beats 27.54%) and 59.36 MB (Beats 24.62%). A test case result is shown as "Accepted" with a runtime of 74 ms. The test case input is `nums = [1,12,-5,-6,50,3]` and `k = 4`, with the output being `12.75000`.

Код:

```
public class Solution
{
    public double FindMaxAverage(int[] nums, int k)
    {
        int n = nums.Length;
        double maxSum = 0;
```

```

// Инициализируем первую сумму из первых k элементов
for (int i = 0; i < k; i++)
{
    maxSum += nums[i];
}

double currentSum = maxSum;

// Применяем метод скользящего окна
for (int i = k; i < n; i++)
{
    currentSum += nums[i] - nums[i - k];
    if (currentSum > maxSum)
    {
        maxSum = currentSum;
    }
}

return maxSum / k;
}
}

```