

# Problem 644: Maximum Average Subarray II

## Problem Information

**Difficulty:** Hard

**Acceptance Rate:** 37.61%

**Paid Only:** Yes

**Tags:** Array, Binary Search, Prefix Sum

## Problem Description

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

Find a contiguous subarray whose \*\*length is greater than or 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:** - When the length is 4, averages are [0.5, 12.75, 10.5] and the maximum average is 12.75 - When the length is 5, averages are [10.4, 10.8] and the maximum average is 10.8 - When the length is 6, averages are [9.166667] and the maximum average is 9.166667 The maximum average is when we choose a subarray of length 4 (i.e., the sub array [12, -5, -6, 50]) which has the max average 12.75, so we return 12.75 Note that we do not consider the subarrays of length < 4.

**Example 2:**

**Input:** nums = [5], k = 1 **Output:** 5.00000

**Constraints:**

\* `n == nums.length` \* `1 <= k <= n <= 104` \* `-104 <= nums[i] <= 104`

## Code Snippets

**C++:**

```
class Solution {  
public:  
double findMaxAverage(vector<int>& nums, int k) {  
  
}  
};
```

**Java:**

```
class Solution {  
public double findMaxAverage(int[] nums, int k) {  
  
}  
}
```

**Python3:**

```
class Solution:  
def findMaxAverage(self, nums: List[int], k: int) -> float:
```