75 Days of Code

Day14 Problem no: 643. Maximum Average Subarray I (leetcode)

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

For this problem , we are gonna use sliding window approach

What is a sliding window Technique and when it is used , key terms included in the question which is solved by sliding window ?

- Sliding window technique is a common algorithm used for solving problem that includes array , strings , sequence or other data structure with a defined order
- Its main purpose or it aims to reduce the nested loop to single loop and reduce time complexity
- **Sliding window technique pseudo code**
    1. **Initialize :** defining the size and initialize the beginning sequence
    2. **Process initial window :** perform some initial calculation within the window

3. **Slide the window :** Move the window by one step (element) to the right. This means removing the leftmost element from the window and adding the next element on the right side.

4. **Update Computation :**update the ongoing computation or analysis based on the change in the window content **.**

5. **Repeat :** repeat until the end of the loop

Key terms to know problem can be solved by sliding window :

Question includes : **Array ,SubArray , SubString , Largest , Smallest ,Maximum and Minimum**

Solution of the above problem using sliding window

1. Calculate the sum up to k for initial window (Initialization nad process with initial calculation)
2. Initialize a variable as maxsum before starting sliding through the loop
3. Slide the sum var (intial window) starting with k upto total number.length -k (Sliding the window , update and repeat)
4. Return the ans as maxnum/k

```javascript
function findMaxAverage(nums: number[], k: number): number {
  let numsSize = nums.length;
  if (numsSize < k) return -1;
  let sum: number = Number.MIN_VALUE;
  for (let numIndex = 0; numIndex < k; numIndex++) {
    sum += nums[numIndex];
  }

  let maxSum = sum;
  let startIndex = 0;
  for (let numIndex = k; numIndex < numsSize; numIndex++) {
    sum = sum - nums[startIndex];
    sum = sum + nums[numIndex];
    maxSum = Math.max(maxSum, sum);
    startIndex++;
  }

  let ans = maxSum / k;
  return ans;
}

let answer = findMaxAverage([1, 12, -5, -6, 50, 3], 4);
console.log("Answer :", answer);
```

```
[Running] node "c:\Users\Shubham\Desktop\75daysOfCode\75DaysOfCode\day14.js"
Answer : 12.75
```