643. Maximum Average Subarray I

Description HintsSubmissionsDiscussSolution

DiscussPick One

Given an array consisting of `n` integers, find the contiguous subarray of given length `k` that has the maximum average value. And you need to output the maximum average value.

Example 1:

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

Output: 12.75

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

Note:

1. 1 <= `k` <= `n` <= 30,000.
2. Elements of the given array will be in the range [-10,000, 10,000].

```
class Solution {
public:
    double findMaxAverage(vector<int>& nums, int k) {
        long long sum = 0;
        for(int i=0;i<k;i++) sum += nums[i];
        long long max_val = sum;
        for(int kk=k;kk<nums.size();++kk)
        {
            sum+=nums[kk]-nums[kk-k];
            max_val = max(max_val,sum);
        }
        return max_val/1.0/k;
    }
};
```

644. Maximum Average Subarray II

Description HintsSubmissionsDiscussSolution

DiscussPick One

Given an array consisting of `n` integers, find the contiguous subarray whose **length is greater than or equal to** `k` that has the maximum average value. And you need to output the maximum average value.

Example 1:

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

Output: 12.75

Explanation:

when length is 5, maximum average value is 10.8,

when length is 6, maximum average value is 9.16667.

Thus return 12.75.
```

Note:

1. 1 <= `k` <= `n` <= 10,000.
2. Elements of the given array will be in range [-10,000, 10,000].
3. The answer with the calculation error less than 10^{-5} will be accepted.

Seen this question in a real interview before?

Yes

```
 class Solution {
public:


    bool check(vector<int>& nums, double mid, int k)
    {
        int n = nums.size();
        vector<double> a(n,0);
        for(int i=0;i<n;i++) a[i] = nums[i]-mid;
        double now=0,last=0;
        for(int i=0;i<k;i++) now +=a[i];
```

```
    if(now>=0) return true;
    double max_val = now;
    for(int i=k;i<n;i++)
    {
        //now+=a[i]-a[i-k];
        //max_val = max(max_val,now);
        now+=a[i];
        last+=a[i-k];
        if(last<0)
        {
            now-=last;
            last=0;
        }
        if(now>=0) return true;
    }
    //if(max_val>now) return true;
    return false;
}

double findMaxAverage(vector<int>& nums, int k) {
    double left = INT_MAX, right = INT_MIN, mid;

    for(int num:nums){
        right = max(right, double(num));
        left = min(left, double(num));
    }
    while(left+0.000004<right)
    {
        double mid = (left+right)/2;
        bool flag = check(nums,mid,k);
        if(flag)
        {
            left = mid;
```

```
            }else
            {
                right = mid;
            }
        }
        return right;
    }
};
```