Sliding Window

- To be used with subarrays or sublists. When you have to calculate average / sum / max or something from a subarray, sublists of specific size

A subarray is a contiguous part of array

In array [1, 2, 3, 4] → 10 subarrays are (1), (2), (3), (4), (1, 2), (2, 3), (3, 4) (1, 2, 3), (2, 3, 4) and (1, 2, 3, 4).

Example → Given an array find average of all contiguous subarrays of size 'K'.

Intuition → Let us consider array [1, 3, 2, 6, 7] and K=3

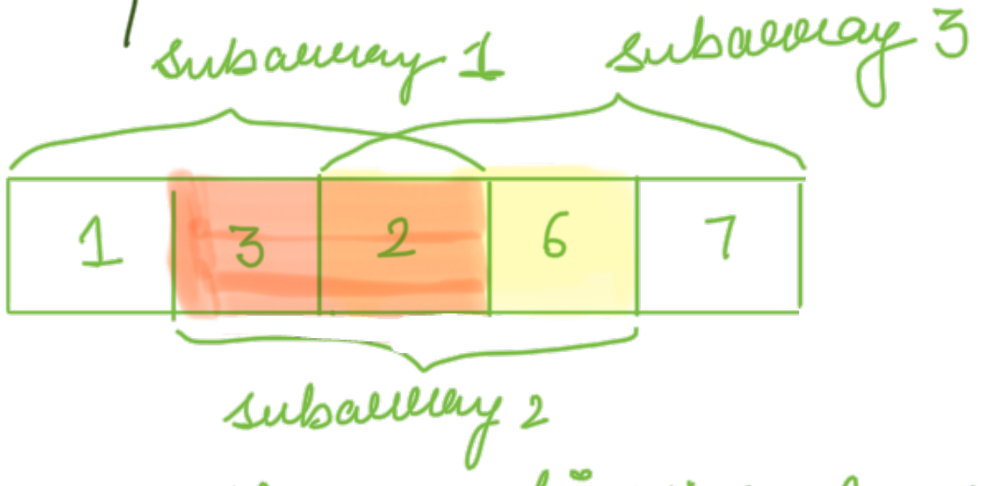
We need to find average of all subarrays of size 3

Average of 1, 3, 2 is (1+3+2)/3 = 2
Average of 3, 2, 6 is (3+2+6)/3 = 3.6
Average of 2, 6, 7 is (2+6+7)/3 = 5

A simple way is to sum all subarrays of size 3 and divide them
Here time complexity will be O(N * K) as we are
calculating sum of 'K' elements 'N' times.

'K' → size of subarray
'N' → size of full array.

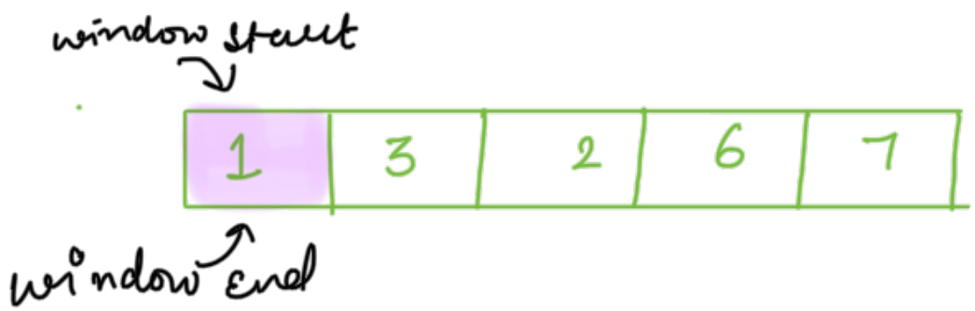
Sliding Window (Optimal Solution)



Here as we can notice, the highlighted elements are common among 2 subarrays.

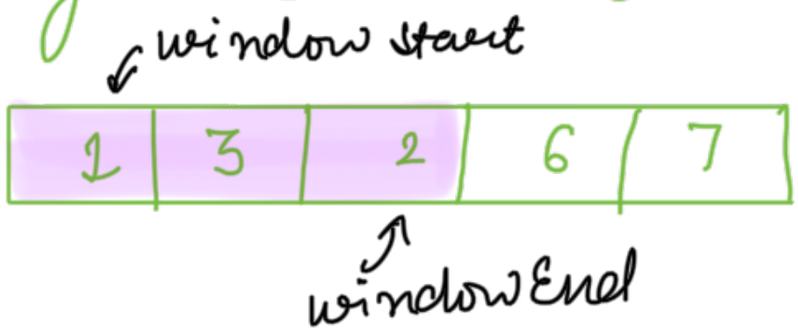
Instead of calculating sum of common elements again and again, we can reuse this sum.

1. Start with windowstart=0 and windowEnd=0



windowSize = 1
windowSum = 1

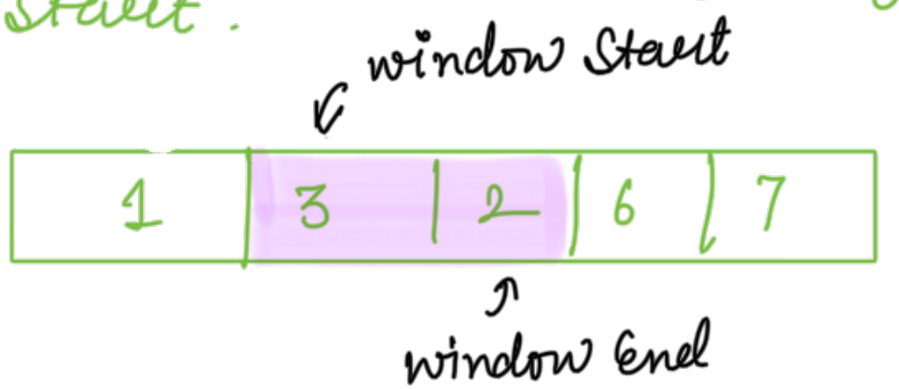
2. Keep incrementing windowEnd till windowsize becomes equal or greater than K. Calculate average when window size=K



windowSize = 3
windowSum = 6
Average = windowSum/K
= 2

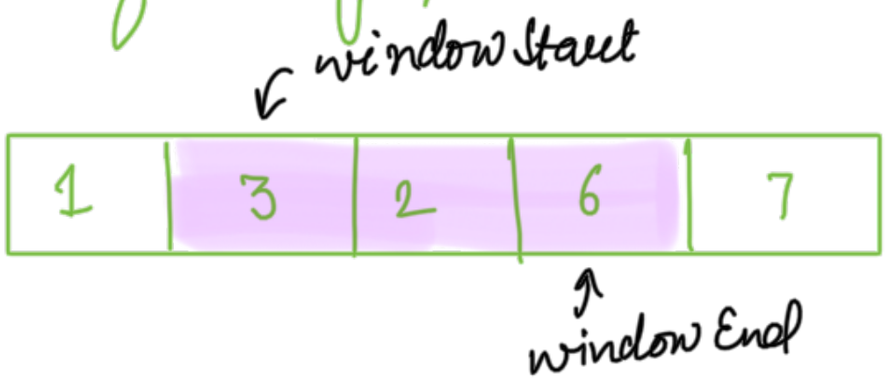
3. Print the average when windowSize is equal to k. And shrink the windowsize by incrementing windowStart.

Decrement windowSum also.

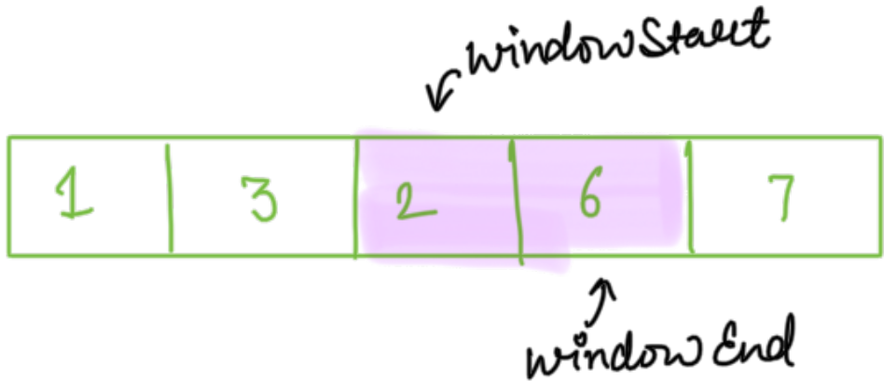


windowSize = 2
windowSum = 5

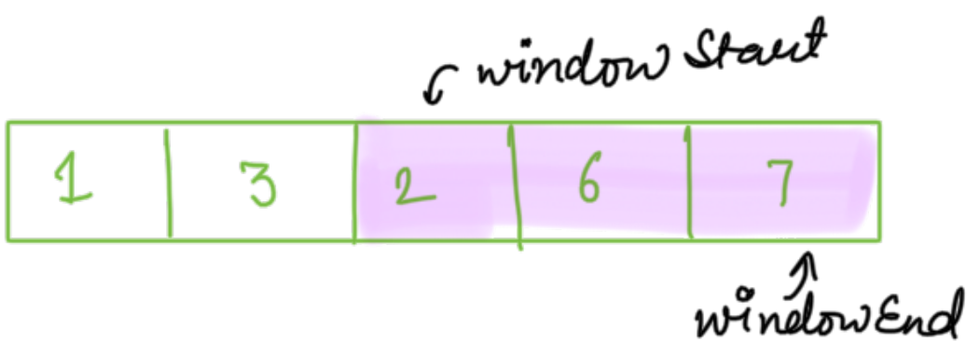
4. Repeat the steps 1, 2, 3 till windowEnd is less than length of array / list.



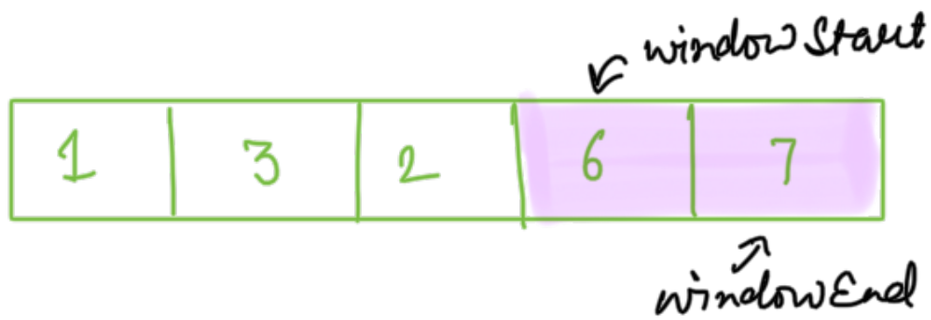
windowSize = 3
windowSum = 11
Average = 3.6



windowSize = 2
windowSum = 8



windowSize = 3
windowSum = 15
Average = 5



windowSize = 2
windowSum = 15

Process ends as we can't increment windowEnd any further.