For an array with non-negative integers(b[i]>=0) ; find the longest subarray such that its sum<=k(at most k)

Link to problem :- <https://www.geeksforgeeks.org/longest-subarray-sum-elements-atmost-k/>

B = [2 5 8 8 5 8]

K = 7

Answer :- [1,2] = [2+5] = 7 -> 2(length is 2)

K = 4

B = [1 2 1 0 1 1 0]

Answer :-

One possibility :- [2 3 4 5] = [2 + 1 + 0 + 1] = 4 —> length is 4 ; is there any better length possible.

Correct Answer : - [3 4 5 6 7] = [1 0 1 1 0] = 3 <=K —> length of subarray is 5

Hence the final answer is 5.

-> Observation : sum[1-1] <= sum[1-2] <=sum[1-3]<=sum[1-4]<=sum[1-5]<=.........sum[1<=n]

Brute Force :->

->Try to find the longest valid subarray starting at index 1

->Try to find the longest valid subarray starting at index 2

.

.

.

.

-> Try to find the longest valid subarray starting at index n

-> How to find the longest valid subarray starting at index i?

Pseudo Code :- <https://ideone.com/Fj0Sgz>

TC :- O(N\*N)

Efficient :->

Let's think..

You are first trying to calculate longest subarray which is valid :- [1….j1]

[2……j2]

-> Say you find out [1…j1] is the longest valid subarray starting at index 1 ; then why are you again checking from start for index 2 ; isn't it common sense.?

Why?

You already know that [2….j1]<=[1….j1] it means ➖

[2….j1] < = [1….j1] < =K ;

It means [2….j1]<=k ; so your j pointer should directly start from j = j1 ; instead of j = i + 1

[3……..j3]

.

.

.

.

.

Pseudo Code <https://ideone.com/pVT36G>

i->[1…N]

j->[1…N]

Total op :- 2N (i moves n times and j moves n times)

O(2N)-->O(N)

P2 :-

Google Coding Interview Problem :- <https://leetcode.com/discuss/interview-question/3549864/Google-or-sliding-window>

-> Given an array of numbers which are sorted find the largest subarray such that the difference between abs(maximum-minimum)<=K

Observation :- abs[1-1] <= abs[1-2] <=abs[1-3] <=...abs[1-N]

TC : - O(N) <https://ideone.com/1STqN9>

P3:-

For an array with non-negative integers ; find the number of subarrays such that its sum<=k

Link to problem :-

<https://www.geeksforgeeks.org/number-subarrays-sum-less-k/>

What do we do in the longest subarray problem ?

-> [1…j1] is the longest subarray starting at 1 ; so will [1…j1+1] be valid? No [1…..j1+1] will not be <=k

Meaning :-> it means that all the subarrays ➖{

[1..1]

[1…2] [1…3] [1…4] [1…j1] were the subarrays whose sum<=k

-> it means there are j1 subarrays whose sum<=k

-> like this you can find the number of subarrays whose sum<=k and starting at index i

}

→ [2…j2]

→ [3….j3]

->[4….j4]

->[5…j5]

Answer = j1 + (j2-1) + (j3-2) + ……

Pseudo-Code :-> <https://ideone.com/YP1T0X>

Similar LeetCode Problem :-

<https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit/>

Similar GFG Problem :- <https://practice.geeksforgeeks.org/problems/count-the-number-of-subarrays/1>

<https://www.geeksforgeeks.org/count-pairs-array-whose-sum-less-x/>

<https://ideone.com/UUQdl6>

<https://ideone.com/lpFBo4>

<https://ideone.com/sL6iRw>

<https://ideone.com/A0ml5w>