Link -> (Starts at 12:35pm.)
https://www.desiqna.in/16043/shaw-coding-question-solution-sde1-september-2023-subarrays


Mentor : Kumar K

Program Details:->
https://docs.google.com/document/d/1nFt_jEgy1X-UaKU2bu3OnC365ADCL-ls_cpXUMJrvx0/edit

Understanding : We are given an array B of size "N"; Divide the array into 4 continuous parts such that g = part1 -part2 +part3 -part4 is maximized.

Any part is allowed to be empty as well ; but collectively they should cover the whole array

Example:->

G = p1 + p3 - (p2 + p4)

If you want to make G bigger what should you do ?

According to Tarab's law:-> To maximize G, we should maximize p1 and p3 and minimize p2 and p4

-> We have to find two non-intersecting subarrays whose sum is as minimum as possible. The second subarray should always contain the last element
If you find them ; then p1 and p3 is already fixed so you don't need to check for it

P1 should for sure contain the first element of the array or it should be empty

P4 should contain the last element of the array or it should be empty

=>[1 2 1 -5]

G = p1 + p3 - (p2+p4) = 4 - (-5) = 9.

Structure of p4 : [n,n]
[n-1;n]
[n-2;n]
.
.
.
.
.
.
[1;n]
Or empty = 0

Structure if p2 : [i….j] i<=j ; i→1 t o n ; j→1 to n

Subarray :-> Always analyze the condition or subarray at index "i" ; and fix something to index "i"

Let's take an index "i"; find the p2 ending at index "i" ; then find p4 ; do these for all "i"; min of all of them will be your answer

Algorithm.-> https://ideone.com/9pn4SX

C++ https://ideone.com/KvyaDm

Java. https://ideone.com/tVp8MU

Py https://ideone.com/9mHaFD