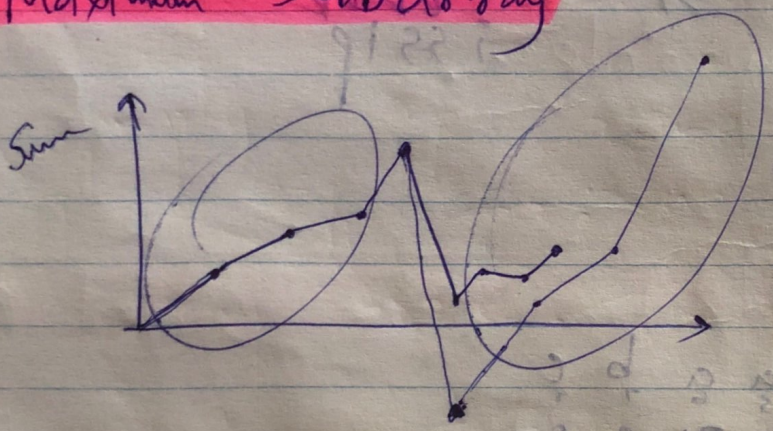
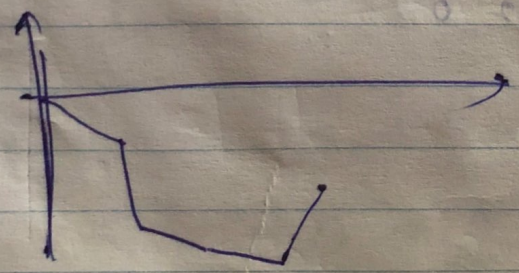
* Maximum Subarray





-2 -1 -9

-3 -1 -4

7 -6 -6

Can use naive approach

5, 4, -1, 7, 8

⇓

5, 9, 8, 15, 23

then subtract each & get max.

5, ⇒ 9, 9-5. ⇒ 8, 8-5, 8-9 ⇒ ----
4 3 -1

DP approach

-2, -1, 4

item = max (preSum + item, item)

-2, -1, 4

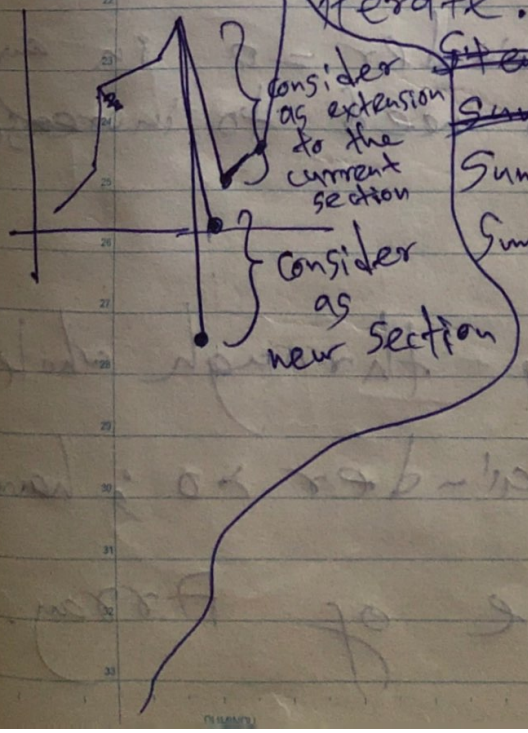
↕ Same like

or, item = [(prev > 0) ? prev : 0] + item

- Kadane's algorithm

-2, 1, -3, 4, -1, 2, 1, -5, 4

iterate:



~~item + curr~~

~~sum + item~~

Sum += item

Sum < 0 ? Sum = item

if preMax < Sum ; preMax = Sum