Kadane's Algorithm

$\hookrightarrow$ It is used to find the ~~subArray~~ continuous subArray whose sum is maximum.

-2	-3	2	-1	3	-2	8	-9
0	1	2	3	4	5	6	7

arr

$sum = arr[0] = -\cancel{2} \; \cancel{2} \; \cancel{4} \; 10$

$tempSum = \cancel{0} \; -\cancel{2} \; \cancel{0} \; \cancel{3} \; \cancel{8} \; \cancel{2} \; \cancel{X} \; \cancel{Y} \; \cancel{2} \; \cancel{6} \; 1$

$\boxed{sum = 10}$ sub-array

2	-1	3	-2	8
2	3	4	5	6

$\underline{T = O(n)} \quad , \quad \underline{S = O(1)}$

<u>Note</u>: Each time $(tempSum < 0)$ we ~~make~~ set it to 0. Because there is no point of going forward with $\boxed{-ve}$ sum