subarray with largest sum.

Kadane's Algorithm: [Maximum Subarray]

- Iterative DP algo
- Calculates max sum subarray ending at particular position ⇒ using max. sum subarray ending at previous position.

1) currSum ⇒ max. sum ending here
   maxSum ⇒ max. sum so far
2) currSum = 0   maxSum = INT-MIN (initialize)
3) for i in 0 to n-1

```
3) for i in 0 to n-1
       ~~if~~ currSum = currSum + arr[i]
       if currSum > maxSum
           maxSum = currSum
       if currSum < 0
           currSum = 0
4) print maxSum.
```

eg:

arr

0	1	2	3
-2	3	-1	2

currSum=0

maxSum = INT_MIN

i=0   arr[0] = -2   ∴ currSum = 0 + -2 = -2

∴ currSum = 0   [∵ currSum (-2) < 0] maxSum=-2

i=1   arr[1] = 3   ∴ currSum = 0 + 3 = 3

∴ maxSum = 3   [∵ currSum > maxSum]

i=2   arr[2] = -1   currSum = 3 - 1 = 2

maxSum = 3.

i=3   arr[3] = 2   currSum = 2 + 2 = 4

∴ maxSum = 4