

53. Maximum Subarray

53. Maximum Subarray

Given array, find subarray with largest sum, and return its sum.

Ex:- Input: $\text{nums} = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$

Output = 6

→ The subarray $[4, -1, 2, 1]$ has the largest sum 6.

$$\text{nums} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ -2 & 1 & -3 & 4 & -1 & 2 & 1 & -5 & 4 \end{bmatrix}$$

Brute Approach: iterate through nested loop find all subarrays sum, then return maximum of them.

Better Approach: we'll use Kadane's Algorithm

$$\text{nums} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ -2 & 1 & -3 & 4 & -1 & 2 & 1 & -5 & 4 \end{bmatrix}, \text{output} = 6$$

(i) we assume our current sum = $\text{nums}[0]$
and max-sum is also $\text{nums}[0]$

(ii) iterate through array and update both values.

Dry Run :-

$$\text{nums} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ -2 & 1 & -3 & 4 & -1 & 2 & 1 & -5 & 4 \end{bmatrix}$$

Ind	curr-S	max-S	curr-S	max-S
0	-2	-2	$\max(-2, -2+1)$	-2
1	= 1	= 1	$\max(1, -2+1)$	$\max(1, -2)$
2	$\max(-3, 1+(-3))$	$\max(1, -2)$	$\max(-3, 1+(-3))$	$\max(1, -2)$
3	$\max(-3, 1+(-3)+4)$	$\max(1, -2)$	$\max(-3, 1+(-3)+4)$	$\max(1, -2)$
4	$\max(-1, 1+(-3)+4)$	$\max(3, 4)$	$\max(-1, 1+(-3)+4)$	$\max(3, 4)$
5	$\max(2, 2+3)$	$\max(5, 4)$	$\max(2, 2+3)$	$\max(5, 4)$
6	$\max(1, 6)$	$\max(6, 5)$	$\max(1, 6)$	$\max(6, 5)$
7	$\max(-5, 6+(-5))$	$\max(6, 1)$	$\max(-5, 6+(-5))$	$\max(6, 1)$
8	$\max(4, 1+4)$	$\max(6, 5)$	$\max(4, 1+4)$	$\max(6, 5)$

```
int curr-Sum = nums[0];
int max-Sum = nums[0];
for(int i=1; i<n; i++) {
    curr-Sum = max(nums[i], curr-Sum + nums[i]);
    max-Sum = max(max-Sum, curr-Sum);
}
return max-Sum;
```

Time Complexity = $O(n)$

Space Complexity = $O(1)$