

## 53. Maximum Subarray

### 53. Maximum Subarray

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

Ex:- input:  $nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$

Output = 6

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

$nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$

Brute Approach: iterate through nested l-r find all subarrays sum. then return maximum of them.

Better Approach: we'll use Kadane's Algorithm

$nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$ , output = 6

- (1) we assume our current sum =  $nums[0]$  and max-sum is also  $nums[0]$
- (2) iterate through array and update both values.

```
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;
```

Dry Run:-

$nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$

ind	curr-S	max-S
0	-2	-2
1	$\max(1, -2+1) = 1$	$\max(1, -2) = 1$
2	$\max(-3, 1+(-3)) = -2$	$\max(1, -2) = 1$
3	$\max(4, -2+4) = 4$	$\max(4, 1) = 4$
4	$\max(-1, 4+(-1)) = 3$	$\max(3, 4) = 4$
5	$\max(2, 3+2) = 5$	$\max(5, 4) = 5$
6	$\max(1, 5+1) = 6$	$\max(6, 5) = 6$
7	$\max(-5, 6+(-5)) = 1$	$\max(6, 1) = 6$
8	$\max(4, 1+4) = 5$	$\max(6, 5) = 6 \rightarrow \underline{\underline{6}}$

Time complexity =  $O(n)$

Space complexity =  $O(1)$