

53. Maximum Subarray

53. Maximum Subarray Solved

Medium Topics Companies

Given an integer array `nums`, find the **subarray** with the largest sum, and return its **sum**.

Example 1:
Input: `nums = [-2,1,-3,4,-1,2,1,-5,4]`
Output: 6
Explanation: The subarray `[4,-1,2,1]` has the largest sum 6.

Example 2:
Input: `nums = [1]`
Output: 1
Explanation: The subarray `[1]` has the largest sum 1.

Example 3:
Input: `nums = [5,4,-1,7,8]`
Output: 23
Explanation: The subarray `[5,4,-1,7,8]` has the largest sum 23.

Submission Interface:
Runtime: 0 ms | Beats 100.00%
Memory: 71.76 MB | Beats 53.26%
2.99% of solutions used 2 ms of runtime

Testcase: Case 1 Case 2 Case 3 +
nums = [-2,1,-3,4,-1,2,1,-5,4]

Code:

```
class Solution {
```

```
public:
```

```
    int maxSubArray(vector<int>& nums) {
```

```
        // kadane's algorithm
```

```
        int maxi = INT_MIN;
```

```
        int currSum = 0;
```

```
        for (int i = 0; i < nums.size(); i++) {
```

```
            currSum += nums[i];
```

```
            if (currSum > maxi) //if current sum is greater than maximum sum then maxi=currSum
```

```
                maxi = currSum;
```

```
            if (currSum < 0) //if currentsum become -ve reset it to 0
```

```
                currSum = 0;
```

```
        }
```

```
        return maxi;
```

```
    }
```

```
};
```