

Maximum Subarray

[Solution](#)

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

Example:

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

Follow up:

If you have figured out the $O(n)$ solution, try coding another solution using the divide and conquer approach, which is more subtle.

Java



```

1 class Solution {
2     public int maxSubArray(int[] nums) {
3         int sum = 0;
4         int result = Integer.MIN_VALUE;
5         for(int i = 0; i < nums.length; i++){
6             sum += nums[i];
7             result = Math.max(result, sum);
8             sum = Math.max(0, sum);
9         }
10
11         return result;
12     }
13 }
```

☐ Custom Testcase ([Contribute](#))

[Run Code](#)
[Submit](#)