Problem | Editorial | Submissions

Kadane's Algorithm

Difficulty: **Medium** Accuracy: **36.28%** Submissions: **1M** Points: **4**

Given an integer array **arr[].** You need to find the **maximum** sum of a subarray.

Examples:

Input: arr[] = [2, 3, -8, 7, -1, 2, 3]
Output: 11
Explanation: The subarray {7, -1, 2, 3} has the largest sum 11.

Input: arr[] = [-2, -4]
Output: -2
Explanation: The subarray {-2} has the largest sum -2.

Input: arr[] = [5, 4, 1, 7, 8]
Output: 25
Explanation: The subarray {5, 4, 1, 7, 8} has the largest sum 25.

Constraints:

$1 \leq arr.size() \leq 10^5$

$-10^9 \leq arr[] \leq 10^4$

Java (1.8) | Average Time: 20m | Start Timer

```
1  // } Driver Code Ends
37
38
39 // User function Template for Java
40 class Solution {
41     int maxSubarraySum(int[] arr) {
42         int maxSum = Integer.MIN_VALUE;
43         int currentSum = 0;
44         for (int i = 0; i < arr.length; i++) {
45             currentSum = Math.max(arr[i], currentSum + arr[i]);
46             maxSum = Math.max(maxSum, currentSum);
47         }
48         return maxSum;
49     }
50 }
51
```

Custom Input | Compile & Run | Submit