Problem | Editorial | Submissions

Maximum Product Subarray

Difficulty: **Medium** Accuracy: **18.09%** Submissions: **416K+** Points: **4**

Given an array **arr[]** that contains positive and negative integers (may contain 0 as well). Find the **maximum** product that we can get in a subarray of **arr[]**.

Note: It is guaranteed that the output fits in a 32-bit integer.

Examples

Input: arr[] = [-2, 6, -3, -10, 0, 2]
Output: 180
Explanation: The subarray with maximum product is {6, -3, -10} with product = 6 * (-3) * (-10) = 180.

Input: arr[] = [-1, -3, -10, 0, 6]
Output: 30
Explanation: The subarray with maximum product is {-3, -10} with product = (-3) * (-10) = 30.

Input: arr[] = [2, 3, 4]
Output: 24

Java (1.8) | Start Timer

```
1  // } Driver Code Ends
23
24
25 class Solution {
26     int maxProduct(int[] arr) {
27         int n = arr.length;
28         int maxProduct = arr[0];
29         int currentMax = arr[0];
30         int currentMin = arr[0];
31         for (int i = 1; i < n; i++) {
32             int tempMax = currentMax;
33             currentMax = Math.max(arr[i], Math.max(currentMax * arr[i], currentMin * arr[i]));
34             currentMin = Math.min(arr[i], Math.min(tempMax * arr[i], currentMin * arr[i]));
35             maxProduct = Math.max(maxProduct, currentMax);
36         }
37         return maxProduct;
38     }
39 }
40
```

Custom Input | Compile & Run | Submit