Find maximum product subarray in a given array

Given an array of integers, find sub-array in it that has maximum product of its elements.

For example,

Input: { -6, 4, -5, 8, -10, 0, 8 }

Output: The maximum product sub-array is {4, -5, 8, -10} having product 1600

Input: { 40, 0, -20, -10 }

Browse

Adobe Algorithm Amazon BFS Binary Search Bit Hacks Collections DFS FIFO Generics Google Greedy Guava Hashing Intro Java 8 Java 9 JSON

Output: The maximum product sub-array is {-20, -10} having product 200

Naive solution would be to consider every sub-array and find product of their elements. Finally, we return the maximum product found among all sub-arrays. The implementation can be seen here. The time complexity of this solution is $O(n^2)$.

A better solution will be to maintain two variables to store the maximum and minimum product ending at current position. Then we traverse the array once and for every index i in the array, we update maximum and minimum product ending at A[i]. We update the result if maximum product ending at any index if more than maximum product found so far.

C++ implementation –

```
#include <iostream>
using namespace std;

// Function to return maximum product of a sub-array of given a
int maxProduct(int arr[], int n)
{
    // maintain two variables to store maximum and minimum prod
    // ending at current index
    int max_ending = 0, min_ending = 0;

    // to store maximum product sub-array found so far
```

```
    int max_so_far = 0;

    // traverse the given array
    for (int i = 0; i < n; i++)
    {
        int temp = max_ending;

        // update maximum product ending at current index
        max_ending = max(arr[i], max(arr[i] * max_ending, arr[i

        // update minimum product ending at current index
        min_ending = min(arr[i], min(arr[i] * temp, arr[i] * mi

        max_so_far = max(max_so_far, max_ending);
    }

    // return maximum product
    return max_so_far;
}

// main function
int main()
{
    int arr[] = { -6, 4, -5, 8, -10, 0, 8 };
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "The maximum product of a sub-array is " <<
            maxProduct(arr, n);

    return 0;
}
```

Download Run Code

Output:

The maximum product of a sub-array is 1600

The time complexity of above solution is O(n) and auxiliary space used by the program is O(1).

Thanks for reading.

Please use ideone or C++ Shell or any other online compiler link to post code in comments.
Like us? Please spread the word and help us grow. Happy coding 🙂

Sharing is caring:

Tweet Share 19

Array
Amazon

← Maximum profit earned by buying and selling shares any number of times

C++ Program to Print Binary Tree Structure →

Leave a Reply

Notify of new follow-up comments Email ›

b i link b-quote u ul ol li code spoiler

Join the discussion

Sort by: newest | oldest | most voted

Andrew

Guest

Hi. This solution works for array+subarray. I think we should consider the case of unordered set+subset as well.
Best.

0 REPLY 7 months 4 days ago

Techie Delight

Author

Andrew, we have published this problem as well – http://www.techiedelight.com/maximum-product-subset-problem/

0 REPLY 6 months 30 days ago