152. Maximum Product Subarray

Medium

177282FavoriteShare

Given an integer array nums, find the contiguous subarray within an array (containing at least one number) which has the largest product.

Example 1:

Input: [2,3,-2,4]  
Output: 6  
Explanation: [2,3] has the largest product 6.

Example 2:

Input: [-2,0,-1]  
Output: 0  
Explanation: The result cannot be 2, because [-2,-1] is not a subarray.

class Solution {

public:

int maxProduct(vector<int>& nums) {

int dp=nums[0];

int min,max;

min=max=nums[0];

int newmin,newmax;

for(int i=1;i<nums.size();i++){

if(nums[i]<0){

newmin=max\*nums[i];

newmax=min\*nums[i];

min=newmin<nums[i]?newmin:nums[i];

max=nums[i]<newmax?newmax:nums[i];

}else if(nums[i]>0){

newmin=min\*nums[i];

newmax=max\*nums[i];

min=newmin<nums[i]?newmin:nums[i];

max=nums[i]<newmax?newmax:nums[i];

}else{

min=max=0;

}

if(dp<max) dp=max;

}

return dp;

}

};

Success

[Details](https://leetcode.com/submissions/detail/211404296/)

Runtime: 8 ms, faster than 100.00% of C++ online submissions for Maximum Product Subarray.

Memory Usage: 8.9 MB, less than 90.18% of C++ online submissions for Maximum Product Subarray.