```
□ | 6 BROSEPC X | 6 BROSEPC X | 6 Find the Winner of the Cir. X 6 Maximum Subarray - Lett. X 6 SS 最大开始组织 - 力拉 □ X | 6 SS 最大开始组织 - 力拉 □ X | 6 BROSEPC X | 6 BROSEPC X | 6 BROSEPC X | 6 BROSEPCC X | 6 B
                                           力和「中文社区」現己上线。全新「个人主西」更优殊验,節刻加入先人一步撰职分)

新功能無幹。簡全新吃り商场机品 上万社区面解 | 企业部年 | 国北規以 | 夏多克森 | 社会政策
                                                                                                                                                                                                                  Premium (5) (5) 0
   C LeetCode
                                                                                                             〈 ≡ Problem List 〉
                                                                                                      × Close
                                                                                                                             723319809
Mar 16, 2023 22:23
                                                                                                                                                                                                                   O Details + Solution

    121. Best Time to Buy and Sell Stock
    152. Maximum Product Subarray
    697. Degree of an Array

源代码如下:
#include<iostream>
#include<vector>
using namespace std;
struct subSums {
             int 1Sum, rSum, mSum, iSum;
};
class Solution {
private:
             subSums update(subSums left,subSums right) {
                          subSums res;
                         res.iSum = left.iSum + right.iSum;
                         res.lSum = max(left.lSum, left.iSum + right.lSum);
                         res.rSum = max(right.rSum, right.iSum + left.rSum);
                         res.mSum = max(max(left.mSum, right.mSum), left.rSum +
right.1Sum);
                         return res;
             subSums get(vector<int>& arr, const int left, const int right) {
                          subSums res;
                         if (left == right) {
                                      res.lSum = arr[left];
                                      res.rSum = arr[left];
                                      res.mSum = arr[left];
                                      res.iSum = arr[left];
                                      return res;
                          int mid = (left + right) / 2;
                          subSums leftSub=get(arr, left, mid);
                         subSums rightSub=get(arr, mid + 1, right);
                         return update(leftSub,rightSub);
             }
```

```
public:
    int maxSubArray(vector<int>& nums) {
        int s = nums.size() - 1;
        subSums ans = get(nums, 0, s);
        return ans.mSum;
    }
};
int main() {
        Solution test;
        vector<int>arr = { -2,1,-3,4,-1,2,1,-5,4 };
        cout << "The max sum of sub string is: " << test.maxSubArray(arr) << endl;
        return 0;
}</pre>
```