

查阅更多的题解, 请点击

Problem

121. Best Time to Buy and Sell Stock(Easy)

Say you have an array for which the i th element is the price of a given stock on day i .

If you were only permitted to complete at most one transaction (i.e., buy one and sell one share of the stock), design an algorithm to find the maximum profit.

Note that you cannot sell a stock before you buy one.

Example 1:

Input: [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
Not 7-1 = 6, as selling price needs to be larger than buying price.

Example 2:

Input: [7,6,4,3,1]

Output: 0

Explanation: In this case, no transaction is done, i.e. max profit = 0.

Solution

这道题解法也是用了**动态规划**的思想:

定义子问题为在第 k 天sell时, 获得的最高利润($DP[k] < 0 ? 0 : DP[k]$), 则问题解为:

$$\max(DP[2], DP[2], \dots, DP[n])$$

[GitHub传送门](#)

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        if(prices.size()<2)
            return 0;
        int max=0;
        int min=prices[0];
        for(int i=1;i<prices.size();++i){
            auto temp=prices[i]-min;
            if(temp>max)
                max=temp;
            if(min>prices[i])
                min=prices[i];
        }
        return max;
    }
};
```