

[每日 LeetCode] 121. Best Time to Buy and Sell Stock

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1553523815765>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Description:

Say you have an array for which the i thelement 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 sha 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.

思路：这题挺有意思，要求数组中最大差值，隐含的限制是该最大差值中的最大值的下标应该大于最值的下标。所以找整个数组的最大最小值之差的方法未必有效，因为很可能最大值出现在最小值之前但是可以利用类似思路，在扫描数组的同时来更新一个当前最小值minPrice。这样能保证当扫到 i 时，inPrices必然是 i 之前的最小值。即，遍历数组时

- 如果 $prices[i] < minPrice$ ，则更新 $minPrice = prices[i]$ 。并且该天不应该卖出。
- 如果 $prices[i] \geq minPrice$ ，则该天可能是最好的卖出时间，计算 $prices[i] - minPrice$ ，并与当前单笔最大利润比较更新。

C++代码

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        if(prices.empty()) return 0;
        int ret = 0, minPrice = prices[0];
        for(int i=1; i<prices.size(); i++) {
            if(prices[i]<minPrice)
                minPrice = prices[i];
            else
                ret = max(prices[i]-minPrice,ret);
        }
        return ret;
    }
};
```

运行时间: 8ms

运行内存: 9.5M