

5月5日报

本日学习内容

1. 每日算法题
2. 复习第五章前三节
3. 完成周报

今日算法题


题目1:[买卖股票的最佳时机三](#)

123. 买卖股票的最佳时机 III

已解答 

困难

 相关标签

 相关企业

Ax

给定一个数组，它的第 i 个元素是一支给定的股票在第 i 天的价格。

设计一个算法来计算你所能获取的最大利润。你最多可以完成 **两笔** 交易。

注意：你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）。

示例 1:

输入: `prices = [3,3,5,0,0,3,1,4]`

输出: 6

解释: 在第 4 天（股票价格 = 0）的时候买入，在第 6 天（股票价格 = 3）的时候卖出，这笔交易所能获得利润 = $3 - 0 = 3$ 。

随后，在第 7 天（股票价格 = 1）的时候买入，在第 8 天（股票价格 = 4）的时候卖出，这笔交易所能获得利润 = $4 - 1 = 3$ 。

示例 2:

输入: `prices = [1,2,3,4,5]`

输出: 4

解释: 在第 1 天（股票价格 = 1）的时候买入，在第 5 天（股票价格 = 5）的时候卖出，这笔交易所能获得利润 = $5 - 1 = 4$ 。

注意你不能在第 1 天和第 2 天接连购买股票，之后再将它们卖出。

```
#define MAX(a, b) ((a) > (b) ? (a) : (b))
#define MIN(a, b) ((a) > (b) ? (b) : (a))

int maxProfit(int* prices, int pricesSize) {
    int buy1 = prices[0];
```

```
int buy2 = prices[0];
int result1 = 0, result2 = 0;
for (int i = 0; i < pricesSize; i++) {
    buy1 = MIN(buy1, prices[i]);
    result1 = MAX(result1, prices[i] - buy1);
    buy2 = MIN(buy2, prices[i] - result1);
    result2 = MAX(result2, prices[i] - buy2);
}
return result2;
}
```

本日遇到的问题

1. 算法及数据结构进度缓慢
2. oc中前面知识的具体应用练习较少

明日学习计划

1. 每日两道题
2. 继续回顾复习第五章后面的内容
3. 开始第六章的学习