

# 剑指 Offer 63. 股票的最大利润

Friday, June 24, 2022 8:58 AM

<https://leetcode.cn/problems/gu-piao-de-zui-da-li-run-lcof/>

中等

假设把某股票的价格按照时间先后顺序存储在数组中，

请问**买卖该股票一次**可能获得的最大利润是多少？

输入: [7,1,5,3,6,4]

输出: 5

解释: 在第 2 天（股票价格 = 1）的时候买入，在第 5 天（股票价格 = 6）的时候卖出，  
最大利润 = 6-1 = 5。

注意利润不能是 7-1 = 6, 因为卖出价格需要大于买入价格。

思路：

开始现金为0，买入：现金数额为  $-\text{prices}[i]$ ，卖出：持有的现金状态 $+\text{prices}[i]$

这么多天只能买卖一次！

$\text{dp}[i][0]$ :当天**持有**（**之前没买今天买入或之前买了今天不卖，延续之前的买入状态**）股票的所得最多现金，

$\text{dp}[i][1]$ :当天**不持有**（**之前买了今天卖了或延续前一天已经卖过的状态，今天不操作**），所得最多的现金

递推公式：

当天持有 最大所得 = **前一天（也就是之前）持有最大所得** 与 **之前没持有，今天买入** 中较大的那个

$\text{dp}[i][0] = \max(\text{dp}[i-1][0], -\text{prices}[i]);$

当天不持有 最大所得 = **前一天不持有最大所得** 与 **之前持有今天卖出** 中较大的那个

$\text{dp}[i][1] = \max(\text{dp}[i-1][1], \text{prices}[i] + \text{dp}[i-1][0]);$

code:

```
class Solution {
    public int maxProfit(int[] prices) {
        // 判空
        if(prices.length==0||prices==null){
            return 0;
        }
        // dp[0][i] 表示第一天的操作结果
        int[][] dp=new int[prices.length][2];
        dp[0][0]=-prices[0]; // 第一天买入，则所得现金为 -prices[0]
        dp[0][1]=0; // 第一天没买，则所得现金为 0;
        for(int i=1;i<prices.length;i++){
            // 今天持有的情况：前一天持有，今天不操作 与 之前没买过，今天买 的大的那个值
            dp[i][0]=Math.max(dp[i-1][0], -prices[i]);
            dp[i][1]=Math.max(dp[i-1][1], prices[i]+dp[i-1][0]);
        }
        return dp[prices.length-1][1];
    }
}
```

```
dp[i][0]=Math.max(dp[i-1][0],-prices[i]);
    // 今天不持有的情况: 前一天卖过, 今天不操作 与 前一天持有, 今天卖出 的大
    的那个值
    dp[i][1]=Math.max(dp[i-1][1],prices[i]+dp[i-1][0]);
}
// 最后一天不持有就是累计的最大值
return dp[prices.length-1][1];
}
}
```