#秋日生活打卡季#

题目：

给定一个整数数组 prices，其中 prices[i]表示第 i 天的股票价格 ；整数 fee 代表了交易股票的手续费用。

你可以无限次地完成交易，但是你每笔交易都需要付手续费。如果你已经购买了一个股票，在卖出它之前你就不能再继续购买股票了。

返回获得利润的最大值。

注意：这里的一笔交易指买入持有并卖出股票的整个过程，每笔交易你只需要为支付一次手续费。

示例 1：

输入：prices = [1, 3, 2, 8, 4, 9], fee = 2
输出：8
解释：能够达到的最大利润:
在此处买入 prices[0] = 1
在此处卖出 prices[3] = 8
在此处买入 prices[4] = 4
在此处卖出 prices[5] = 9
总利润: ((8 - 1) - 2) + ((9 - 4) - 2) = 8
示例 2：

输入：prices = [1,3,7,5,10,3], fee = 3
输出：6

提示：

1 <= prices.length <= 5 * 10^4
1 <= prices[i] < 5 * 10^4
0 <= fee < 5 * 10^4

java代码：

```
class Solution {    public int maxProfit(int[] prices, int f
ee) {        int n = prices.length;        int[][] dp = new
int[n][2];        dp[0][0] = 0;        dp[0][1] = -prices[0]
;        for (int i = 1; i < n; ++i) {            dp[i][0] =
 Math.max(dp[i - 1][0], dp[i - 1][1] + prices[i] - fee);
        dp[i][1] = Math.max(dp[i - 1][1], dp[i - 1][0] - pri
ces[i]);        }        return dp[n - 1][0];    }}
```