# 题目

给定一个数组，它的第 i 个元素是一支给定股票第 i 天的价格。

如果你最多只允许完成一笔交易（即买入和卖出一支股票一次），设计一个算法来计算你所能获取的最大利润。

注意：你不能在买入股票前卖出股票。

**示例 1:**

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；同时，你不能在买入前卖出股票。

**示例 2:**

输入: [7,6,4,3,1]

输出: 0

解释: 在这种情况下, 没有交易完成, 所以最大利润为 0。

类似题目：剑指offer 63

# 分析

## 方法一：动态规划

class Solution {

public:

int maxProfit(vector<int>& prices) {

int profit = 0;

int price = INT\_MAX; //先设定价格为最大值，收益为0

for(int i=0;i<prices.size();i++)

{

price = min(price,prices[i]);

profit = max(profit,prices[i]-price);

}

return profit;

}

};