1. Best Time to Buy and Sell Stock

Say you have an array for which the *i*th element 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 share 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.

**解法1** 暴力求解，计算出

**解法2** one-pass。在全局最低点买入，卖出一定在该点之后，因此一边寻找min\_p，一边计算max\_profit

class Solution {  
public:  
 int maxProfit(vector<int>& prices) {  
 int n = prices.size();  
 int profit = 0;  
 int min\_p = INT\_MAX;  
 for(int i = 0; i < n; ++i){  
 if(prices[i] < min\_p)min\_p = prices[i];  
 profit = max(profit, prices[i] - min\_p);  
 }  
 return profit;  
 }  
};