题目描述：

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 (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.

Example 1:

```
Input: [7, 1, 5, 3, 6, 4]
Output: 5

max. difference = 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

In this case, no transaction is done, i.e. max profit = 0.
```

自己的解题思路：

双层循环嵌套，找出差值最大的，并且满足买入日期小于卖出日期 ，结果 ：时间限制

最终解题思路：应该找到当前最低价，并与当前日期比较价格，这样只需遍历一次 时间复杂度为O(n)

题解代码：

```
class Solution {
    public int maxProfit(int[] prices) {
        int maxDiff = 0;
        int lowest = Integer.MAX_VALUE;
        for(int i:prices){
            lowest = Math.min(lowest,i);
            maxDiff = Math.max(i-lowest,maxDiff);
        }
        return maxDiff;
    }
}
```