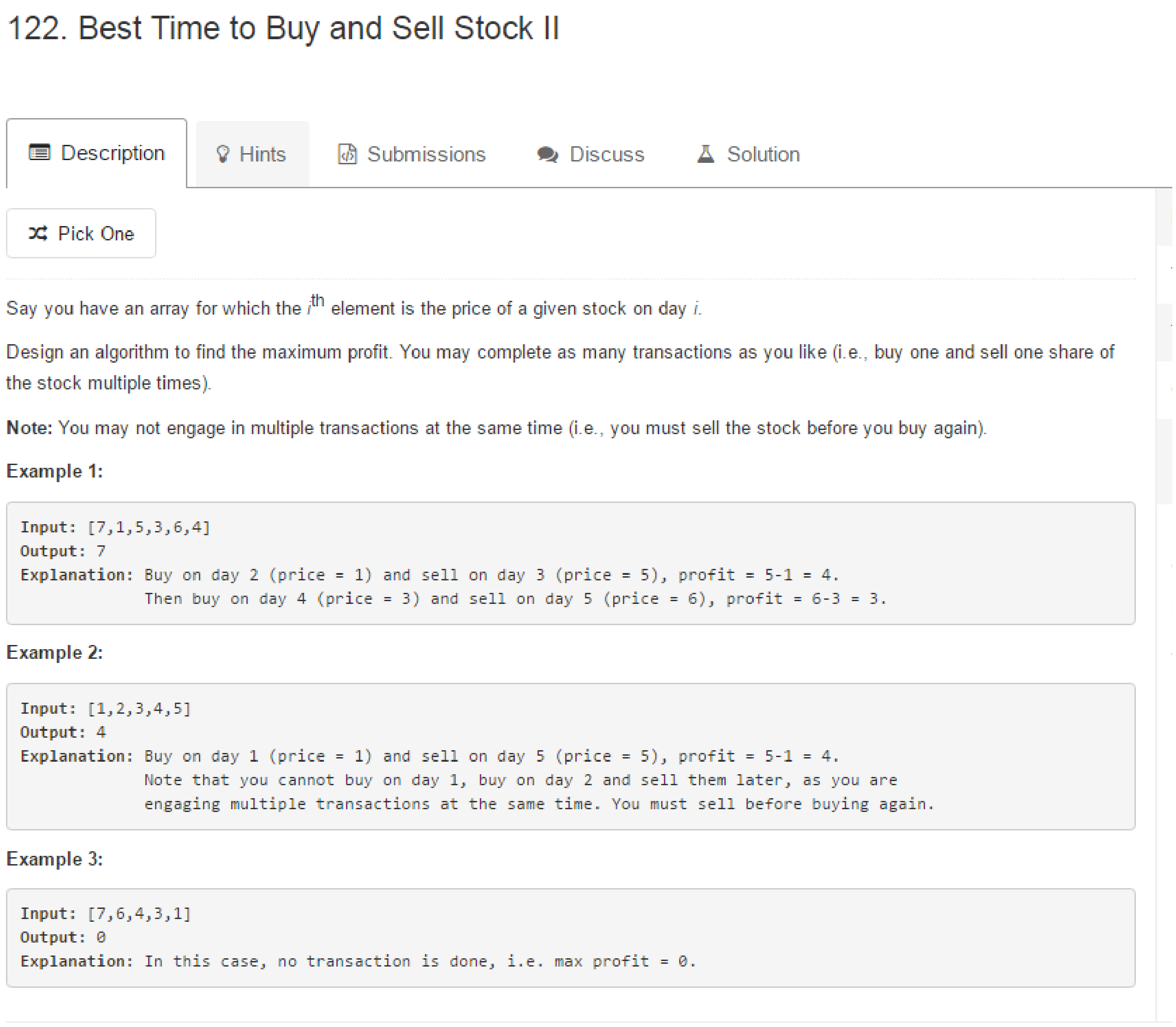


122. Best Time to Buy and Sell Stock II

Say you have an array for which the i^{th} element is the price of a given stock on day *i*.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (i.e., buy one and sell one share of the stock multiple times).

Note: You may not engage in multiple transactions at the same time (i.e., you must sell the stock before you buy again).

Example 1:

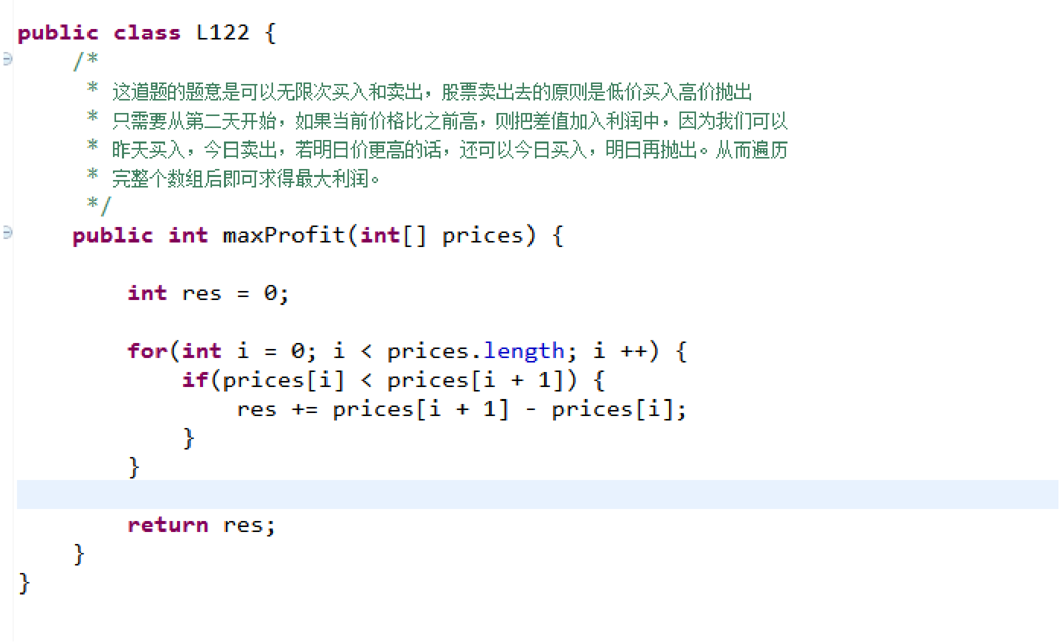
```
Input: [7,1,5,3,6,4]
Output: 7
Explanation: Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4.
             Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3.
```

Example 2:

```
Input: [1,2,3,4,5]
Output: 4
Explanation: Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4.
             Note that you cannot buy on day 1, buy on day 2 and sell them later, as you are
             engaging multiple transactions at the same time. You must sell before buying again.
```

Example 3:

```
Input: [7,6,4,3,1]
Output: 0
Explanation: In this case, no transaction is done, i.e. max profit = 0.
```



```java
public class L122 {
    /*
     * 这道题的题意是可以无限次买入和卖出，股票卖出去的原则是低价买入高价抛出
     * 只需要从第二天开始，如果当前价格比之前高，则把差值加入利润中，因为我们可以
     * 昨天买入，今日卖出，若明日价更高的话，还可以今日买入，明日再抛出。从而遍历
     * 完整个数组后即可求得最大利润。
     */
    public int maxProfit(int[] prices) {

        int res = 0;

        for(int i = 0; i < prices.length; i ++) {
            if(prices[i] < prices[i + 1]) {
                res += prices[i + 1] - prices[i];
            }
        }

        return res;
    }
}
```