

/*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.*/

-
- 思想:
- (1) 动态规划问题, 一个变量记录最大差价 (即最终的输出结果), 一个变量记录当前索引以前的最小价格
- (2) 如果当前价格小于最小价格, 则更新最小价格; 如果大于最小价格, 计算当前价格与历史最小价格的差价, 并与最大差价比较和更新结果