121. Best Time to Buy and Sell Stock

You are given an array `prices` where `prices[i]` is the price of a given stock on the `ith` day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, return `0`.

Example 1:

```
Input: prices = [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.
Note that buying on day 2 and selling on day 1 is not allowed because you
must buy before you sell.
```

Example 2:

```
Input: prices = [7,6,4,3,1]
Output: 0
Explanation: In this case, no transactions are done and the max profit =
0.
```

Constraints:

- `1 <= prices.length <= 105`
- `0 <= prices[i] <= 104`

```
import sys
class Solution:
    def maxProfit(self, prices: List[int]) -> int:
        nge = [0]*len(prices)
        maxRight = -sys.maxsize
        for i in range(len(prices)-1,-1,-1):
            maxRight = max(prices[i],maxRight)
            if maxRight==prices[i]:
                nge[i] = -1
            else:
```

```
                nge[i]=maxRight
        profit = 0
        for i in range(len(prices)):
            if nge[i]!=-1:
                profit = max(profit,nge[i]-prices[i])
        return profit
```