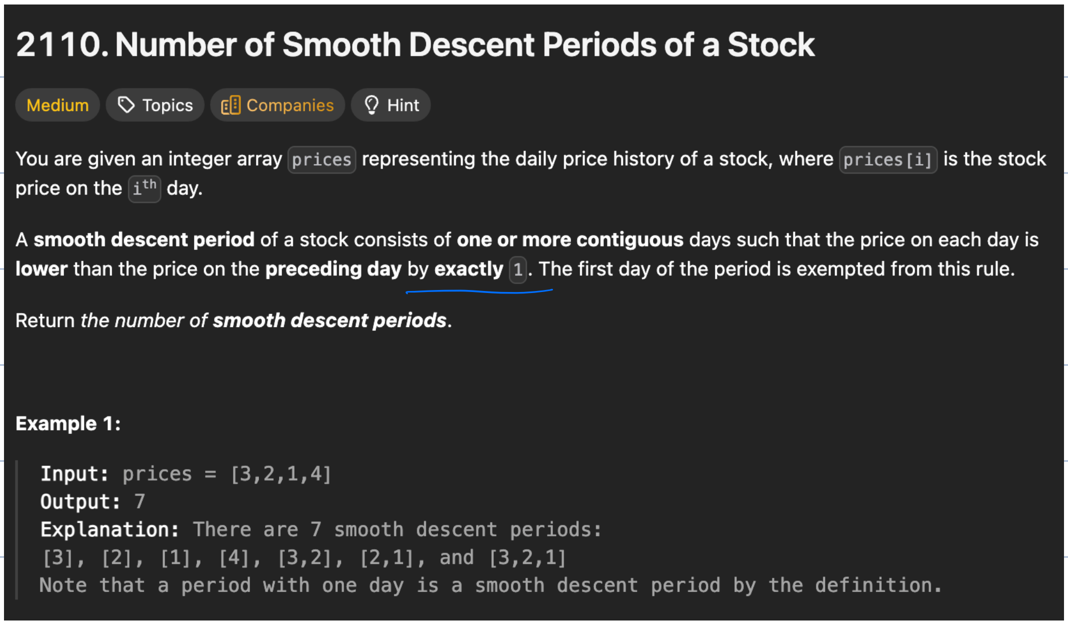


2110. Number of Smooth Descent Periods of a Stock

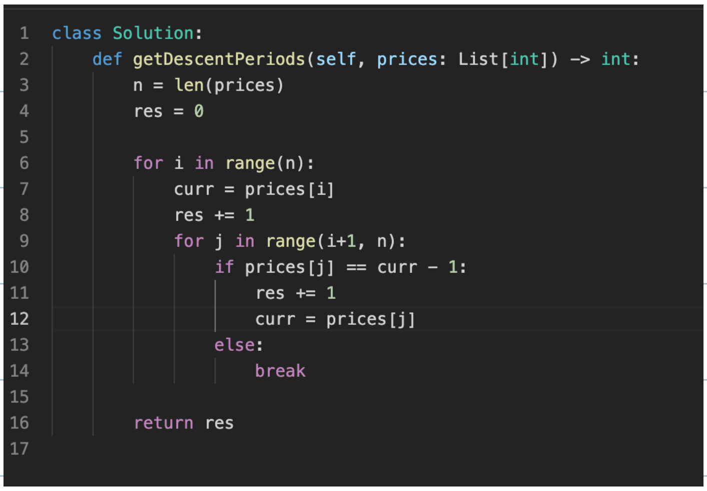
You are given an integer array `prices` representing the daily price history of a stock, where `prices[i]` is the stock price on the i^{th} day.

A **smooth descent period** of a stock consists of **one or more contiguous** days such that the price on each day is **lower** than the price on the **preceding day** by **exactly** `1`. The first day of the period is exempted from this rule.

Return *the number of* ***smooth descent periods***.

Example 1:

```
Input: prices = [3,2,1,4]
Output: 7
Explanation: There are 7 smooth descent periods:
[3], [2], [1], [4], [3,2], [2,1], and [3,2,1]
Note that a period with one day is a smooth descent period by the definition.
```



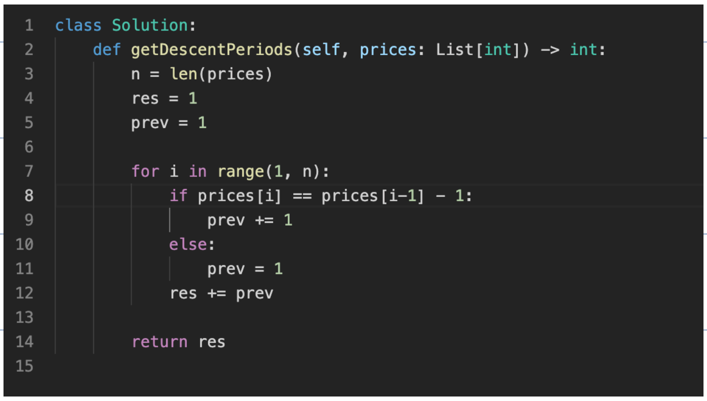
```python
class Solution:
    def getDescentPeriods(self, prices: List[int]) -> int:
        n = len(prices)
        res = 0

        for i in range(n):
            curr = prices[i]
            res += 1
            for j in range(i+1, n):
                if prices[j] == curr - 1:
                    res += 1
                    curr = prices[j]
                else:
                    break

        return res

```

成功，但超时.



```python
class Solution:
    def getDescentPeriods(self, prices: List[int]) -> int:
        n = len(prices)
        res = 1
        prev = 1

        for i in range(1, n):
            if prices[i] == prices[i-1] - 1:
                prev += 1
            else:
                prev = 1
            res += prev

        return res

```

curr 具体val 没必要知道.

只需要知道是否 descending