2898. Maximum Linear Stock Score

Description

Given a **1-indexed** integer array `prices`, where `prices[i]` is the price of a particular stock on the `i th` day, your task is to select some of the elements of `prices` such that your selection is **linear**.

A selection `indexes`, where `indexes` is a **1-indexed** integer array of length `k` which is a subsequence of the array `[1, 2, ..., n]`, is **linear** if:

- For every `1 < j <= k`, `prices[indexes[j]] - prices[indexes[j - 1]] == indexes[j] - indexes[j - 1]`.

A **subsequence** is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

The **score** of a selection `indexes`, is equal to the sum of the following array: `[prices[indexes[1]], prices[indexes[2]], ..., prices[indexes[k]]]`.

Return *the **maximum** **score** that a linear selection can have*.

Example 1:

```
Input: prices = [1,5,3,7,8]
Output: 20
Explanation: We can select the indexes [2,4,5]. We show that our selection is linear:
For j = 2, we have:
indexes[2] - indexes[1] = 4 - 2 = 2.
prices[4] - prices[2] = 7 - 5 = 2.
For j = 3, we have:
indexes[3] - indexes[2] = 5 - 4 = 1.
prices[5] - prices[4] = 8 - 7 = 1.
The sum of the elements is: prices[2] + prices[4] + prices[5] = 20.
It can be shown that the maximum sum a linear selection can have is 20.
```

Example 2:

```
Input: prices = [5,6,7,8,9]
Output: 35
Explanation: We can select all of the indexes [1,2,3,4,5]. Since each element has a difference of exactly 1 from its previous element, our selection is linear.
The sum of all the elements is 35 which is the maximum possible some out of every selection.
```

Constraints:

- `1 <= prices.length <= 10 5`
- `1 <= prices[i] <= 10 9`