Problem 3299: Sum of Consecutive Subsequences

Problem Information

Difficulty: Hard
Acceptance Rate: 41.93%
Paid Only: Yes
Tags: Array, Hash Table, Dynamic Programming

Problem Description

We call an array `arr` of length `n` **consecutive** if one of the following holds:

* `arr[i] - arr[i - 1] == 1` for _all_ `1 <= i < n`. * `arr[i] - arr[i - 1] == -1` for _all_ `1 <= i < n`.

The **value** of an array is the sum of its elements.

For example, `[3, 4, 5]` is a consecutive array of value 12 and `[9, 8]` is another of value 17. While `[3, 4, 3]` and `[8, 6]` are not consecutive.

Given an array of integers `nums`, return the _sum_ of the **values** of all **consecutive** _non-empty_ subsequences.

Since the answer may be very large, return it **modulo** `109 + 7.`

Note that an array of length 1 is also considered consecutive.

Example 1:

Input: nums = [1,2]

Output: 6

Explanation:

The consecutive subsequences are: `[1]`, `[2]`, `[1, 2]`.

Example 2:

Input: nums = [1,4,2,3]

Output: 31

Explanation:

The consecutive subsequences are: `[1]`, `[4]`, `[2]`, `[3]`, `[1, 2]`, `[2, 3]`, `[4, 3]`, `[1, 2, 3]`.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int getSum(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int getSum(int[] nums) {

}
}
```

Python3:

```
class Solution:
def getSum(self, nums: List[int]) -> int:
```