Problem 1223: Dice Roll Simulation

Problem Information

Difficulty: Hard
Acceptance Rate: 50.46%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

A die simulator generates a random number from `1` to `6` for each roll. You introduced a constraint to the generator such that it cannot roll the number `i` more than `rollMax[i]` (**1-indexed**) consecutive times.

Given an array of integers `rollMax` and an integer `n`, return _the number of distinct sequences that can be obtained with exact_`n` _rolls_. Since the answer may be too large, return it **modulo** `109 + 7`.

Two sequences are considered different if at least one element differs from each other.

Example 1:

Input: n = 2, rollMax = [1,1,2,2,2,3] **Output:** 34 **Explanation:** There will be 2 rolls of die, if there are no constraints on the die, there are 6 * 6 = 36 possible combinations. In this case, looking at rollMax array, the numbers 1 and 2 appear at most once consecutively, therefore sequences (1,1) and (2,2) cannot occur, so the final answer is 36-2 = 34.

Example 2:

Input: n = 2, rollMax = [1,1,1,1,1,1] **Output:** 30

Example 3:

Input: n = 3, rollMax = [1,1,1,2,2,3] **Output:** 181

Constraints:

* `1 <= n <= 5000` * `rollMax.length == 6` * `1 <= rollMax[i] <= 15`

Code Snippets

C++:

```
class Solution {
public:
int dieSimulator(int n, vector<int>& rollMax) {

}
};
```

Java:

```
class Solution {
public int dieSimulator(int n, int[] rollMax) {

}
}
```

Python3:

```
class Solution:
def dieSimulator(self, n: int, rollMax: List[int]) -> int:
```