Problem 3082: Find the Sum of the Power of All Subsequences

Problem Information

Difficulty: Hard
Acceptance Rate: 37.39%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums` of length `n` and a **positive** integer `k`.

The **power** of an array of integers is defined as the number of subsequences with their sum **equal** to `k`.

Return _the**sum** of **power** of all subsequences of_ `nums` _._

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

Example 1:

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

Output: 6

Explanation:

There are `5` subsequences of nums with non-zero power:

* The subsequence `[_**1**_ ,_**2**_ ,_**3**_]` has `2` subsequences with `sum == 3`: `[1,2,_3_]` and `[_1_ ,_2_ ,3]`. * The subsequence `[_**1**_ ,2,_**3**_]` has `1` subsequence with `sum == 3`: `[1,2,_3_]`. * The subsequence `[1,_**2**_ ,_**3**_]` has `1` subsequence with `sum == 3`: `[1,2,_3_]`. * The subsequence `[_**1**_ ,_**2**_ ,3]` has `1` subsequence with `sum == 3`: `[_1_ ,_2_ ,3]`. * The subsequence `[1,2,_**3**_]` has `1` subsequence with

`sum == 3`: `[1,2,_3_]`.

Hence the answer is `2 + 1 + 1 + 1 + 1 = 6`.

Example 2:

Input: nums = [2,3,3], k = 5

Output: 4

Explanation:

There are `3` subsequences of nums with non-zero power:

* The subsequence `[_**2**_ ,_**3**_ ,_**3**_]` has 2 subsequences with `sum == 5`: `[_2_ ,3,_3_]` and `[_2_ ,_3_ ,3]`. * The subsequence `[_**2**_ ,3,_**3**_]` has 1 subsequence with `sum == 5`: `[_2_ ,3,_3_]`. * The subsequence `[_**2**_ ,_**3**_ ,3]` has 1 subsequence with `sum == 5`: `[_2_ ,_3_ ,3]`.

Hence the answer is `2 + 1 + 1 = 4`.

Example 3:

Input: nums = [1,2,3], k = 7

Output: 0

**Explanation: **There exists no subsequence with sum `7`. Hence all subsequences of nums have `power = 0`.

Constraints:

* `1 <= n <= 100` * `1 <= nums[i] <= 104` * `1 <= k <= 100`

Code Snippets

C++:

```
class Solution {
public:
int sumOfPower(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int sumOfPower(int[] nums, int k) {

}
}
```

Python3:

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