Problem 3539: Find Sum of Array Product of Magical Sequences

Problem Information

Difficulty: Hard
Acceptance Rate: 62.03%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Bit Manipulation, Combinatorics, Bitmask

Problem Description

You are given two integers, `m` and `k`, and an integer array `nums`.

A sequence of integers `seq` is called **magical** if:

* `seq` has a size of `m`. * `0 <= seq[i] < nums.length` * The **binary representation** of `2seq[0] + 2seq[1] + ... + 2seq[m - 1]` has `k` **set bits**.

The **array product** of this sequence is defined as `prod(seq) = (nums[seq[0]] * nums[seq[1]] * ... * nums[seq[m - 1]])`.

Return the **sum** of the **array products** for all valid **magical** sequences.

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

A **set bit** refers to a bit in the binary representation of a number that has a value of 1.

Example 1:

Input: m = 5, k = 5, nums = [1,10,100,10000,1000000]

Output: 991600007

Explanation:

All permutations of `[0, 1, 2, 3, 4]` are magical sequences, each with an array product of 1013.

Example 2:

Input: m = 2, k = 2, nums = [5,4,3,2,1]

Output: 170

Explanation:

The magical sequences are `[0, 1]`, `[0, 2]`, `[0, 3]`, `[0, 4]`, `[1, 0]`, `[1, 2]`, `[1, 3]`, `[1, 4]`, `[2, 0]`, `[2, 1]`, `[2, 3]`, `[2, 4]`, `[3, 0]`, `[3, 1]`, `[3, 2]`, `[3, 4]`, `[4, 0]`, `[4, 1]`, `[4, 2]`, and `[4, 3]`.

Example 3:

Input: m = 1, k = 1, nums = [28]

Output: 28

Explanation:

The only magical sequence is `[0]`.

Constraints:

* `1 <= k <= m <= 30` * `1 <= nums.length <= 50` * `1 <= nums[i] <= 108`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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