Problem 3428: Maximum and Minimum Sums of at Most Size K Subsequences

Problem Information

Difficulty: Medium
Acceptance Rate: 21.06%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Sorting, Combinatorics

Problem Description

You are given an integer array `nums` and a positive integer `k`. Return the sum of the **maximum** and **minimum** elements of all **subsequences** of `nums` with **at most** `k` elements.

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

Example 1:

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

Output: 24

Explanation:

The subsequences of `nums` with at most 2 elements are:

Subsequence | Minimum | Maximum | Sum ---|---|---|--- `[1]` | 1 | 1 | 2 `[2]` | 2 | 2 | 4 `[3]` | 3 | 3 | 6 `[1, 2]` | 1 | 2 | 3 `[1, 3]` | 1 | 3 | 4 `[2, 3]` | 2 | 3 | 5 **Final Total** | | | 24 The output would be 24.

Example 2:

Input: nums = [5,0,6], k = 1

Output: 22

Explanation:

For subsequences with exactly 1 element, the minimum and maximum values are the element itself. Therefore, the total is `5 + 5 + 0 + 0 + 6 + 6 = 22`.

Example 3:

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

Output: 12

Explanation:

The subsequences `[1, 1]` and `[1]` each appear 3 times. For all of them, the minimum and maximum are both 1. Thus, the total is 12.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `1 <= k <= min(70, nums.length)`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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