Problem 3098: Find the Sum of Subsequence Powers

Problem Information

Difficulty: Hard
Acceptance Rate: 24.33%
Paid Only: No
Tags: Array, Dynamic Programming, Sorting

Problem Description

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

The **power** of a subsequence is defined as the **minimum** absolute difference between **any** two elements in the subsequence.

Return _the**sum** of **powers** of **all** subsequences of _`nums` _which have length_ **_equal to_** `k`.

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

Example 1:

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

Output: 4

Explanation:

There are 4 subsequences in `nums` which have length 3: `[1,2,3]`, `[1,3,4]`, `[1,2,4]`, and `[2,3,4]`. The sum of powers is `|2 - 3| + |3 - 4| + |2 - 1| + |3 - 4| = 4`.

Example 2:

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

Output: 0

Explanation:

The only subsequence in `nums` which has length 2 is `[2,2]`. The sum of powers is `|2 - 2| = 0`.

Example 3:

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

Output: 10

Explanation:

There are 3 subsequences in `nums` which have length 2: `[4,3]`, `[4,-1]`, and `[3,-1]`. The sum of powers is `|4 - 3| + |4 - (-1)| + |3 - (-1)| = 10`.

Constraints:

* `2 <= n == nums.length <= 50` * `-108 <= nums[i] <= 108 ` * `2 <= k <= n`

Code Snippets

C++:

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

}
};
```

Java:

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

}
```

```
}
```

Python3:

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