Problem 2638: Count the Number of K-Free Subsets

Problem Information

Difficulty: Medium
Acceptance Rate: 46.97%
Paid Only: Yes
Tags: Array, Math, Dynamic Programming, Sorting, Combinatorics

Problem Description

You are given an integer array `nums`, which contains **distinct** elements and an integer `k`.

A subset is called a **k-Free** subset if it contains **no** two elements with an absolute difference equal to `k`. Notice that the empty set is a **k-Free** subset.

Return _the number of**k-Free** subsets of _`nums`.

A **subset** of an array is a selection of elements (possibly none) of the array.

Example 1:

Input: nums = [5,4,6], k = 1 **Output:** 5 **Explanation:** There are 5 valid subsets: {}, {5}, {4}, {6} and {4, 6}.

Example 2:

Input: nums = [2,3,5,8], k = 5 **Output:** 12 **Explanation:** There are 12 valid subsets: {}, {2}, {3}, {5}, {8}, {2, 3}, {2, 3, 5}, {2, 5}, {2, 5, 8}, {2, 8}, {3, 5} and {5, 8}.

Example 3:

Input: nums = [10,5,9,11], k = 20 **Output:** 16 **Explanation:** All subsets are valid. Since the total count of subsets is 24 = 16, so the answer is 16.

Constraints:

* `1 <= nums.length <= 50` * `1 <= nums[i] <= 1000` * `1 <= k <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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