Problem 2597: The Number of Beautiful Subsets

Problem Information

Difficulty: Medium
Acceptance Rate: 50.87%
Paid Only: No
Tags: Array, Hash Table, Math, Dynamic Programming, Backtracking, Sorting, Combinatorics

Problem Description

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

A subset of `nums` is **beautiful** if it does not contain two integers with an absolute difference equal to `k`.

Return _the number of**non-empty beautiful** subsets of the array_ `nums`.

A **subset** of `nums` is an array that can be obtained by deleting some (possibly none) elements from `nums`. Two subsets are different if and only if the chosen indices to delete are different.

Example 1:

Input: nums = [2,4,6], k = 2 **Output:** 4 **Explanation:** The beautiful subsets of the array nums are: [2], [4], [6], [2, 6]. It can be proved that there are only 4 beautiful subsets in the array [2,4,6].

Example 2:

Input: nums = [1], k = 1 **Output:** 1 **Explanation:** The beautiful subset of the array nums is [1]. It can be proved that there is only 1 beautiful subset in the array [1].

Constraints:

* `1 <= nums.length <= 18` * `1 <= nums[i], k <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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