6352. The Number of Beautiful Subsets

My Submissions (/contest/weekly-contest-337/problems/the-number-of-beautiful-subsets/submissions/) Back to Contest (/contest/weekly-contest-337/)

User Accepted:	65
User Tried:	318
Total Accepted:	65
Total Submissions:	448
Difficulty:	Medium

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 <= 20`
- `1 <= nums[i], k <= 1000`

JavaScript

```javascript
const counter = (a_or_s) => { let m = new Map(); for (const x of a_or_s) m.set(x, m.get(x) + 1 || 1); return m; };

const beautifulSubsets = (a, k) => {
    let n = a.length, res = 0;
    for (let i = 0; i < 1 << n; i++) {
        let sub = [];
        for (let j = 0; j < n; j++) {
            if (i & (1 << j)) {
                sub.push(a[j]);
            }
        }
        if (sub.length > 0 && ok(sub, k)) res++;
    }
    return res;
};

const ok = (a, k) => {
    let m = counter(a);
    for (const x of a) {
        if (m.has(x + k) || m.has(x - k)) return false;
    }
    return true;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/917832531/) More Details > (/submissions/detail/917832531/)