100029. Count of Sub-Multisets With Bounded Sum

My Submissions (/contest/biweekly-contest-115/problems/count-of-sub-multisets-with-bounded-sum/submissions/)

Back to Contest (/contest/biweekly-contest-115/)

User Accepted:	2
User Tried:	29
Total Accepted:	2
Total Submissions:	53
Difficulty:	Hard

You are given a **0-indexed** array `nums` of non-negative integers, and two integers `l` and `r`.

Return *the **count of sub-multisets** within* `nums` *where the sum of elements in each subset falls within the inclusive range of* `[l, r]`.

Since the answer may be large, return it modulo $10^9 + 7$.

A **sub-multiset** is an **unordered** collection of elements of the array in which a given value `x` can occur `0, 1, ..., occ[x]` times, where `occ[x]` is the number of occurrences of `x` in the array.

Note that:

- Two **sub-multisets** are the same if sorting both sub-multisets results in identical multisets.
- The sum of an **empty** multiset is `0`.

Example 1:

```
Input: nums = [1,2,2,3], l = 6, r = 6
Output: 1
Explanation: The only subset of nums that has a sum of 6 is {1, 2, 3}.
```

Example 2:

```
Input: nums = [2,1,4,2,7], l = 1, r = 5
Output: 7
Explanation: The subsets of nums that have a sum within the range [1, 5] are {1}, {2}, {4}, {2, 2}, {1, 2}, {1, 4}, and {1, 2,
```

Example 3:

```
Input: nums = [1,2,1,3,5,2], l = 3, r = 5
Output: 9
Explanation: The subsets of nums that have a sum within the range [3, 5] are {3}, {5}, {1, 2}, {1, 3}, {2, 2}, {2, 3}, {1, 1, 2
```

Constraints:

- $1 <= \text{nums.length} <= 2 * 10^4$
- $0 <= \text{nums[i]} <= 2 * 10^4$
- Sum of `nums` does not exceed $2 * 10^4$.
- $0 <= l <= r <= 2 * 10^4$

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 mod = 1e9 + 7;
const countSubMultisets = (a, l, r) => {
    let f = Array(r + 1).fill(0), m = counter(a), res = 0;
    f[0] = 1;
    for (const [x, occ] of m) {
        if (x == 0) {
            f = f.map(e => e * (occ + 1));
        } else {
            for (let i = x; i <= r; i++) {
                f[i] += f[i - x];
                f[i] %= mod;
            }
            for (let i = r; i >= (occ + 1) * x; i--) {
                f[i] -= f[i - (occ + 1) * x];
                f[i] %= mod;
            }
        }
    }
    for (let i = l; i <= r; i++) {
        res += f[i];
        res %= mod;
    }
    return (res + mod) % mod;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted **(/submissions/detail/1075186441/)** More Details (/submissions/detail/1075186441/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)