Problem 2902: Count of Sub-Multisets With Bounded Sum

Problem Information

Difficulty: Hard
Acceptance Rate: 21.57%
Paid Only: No
Tags: Array, Hash Table, Dynamic Programming, Sliding Window

Problem Description

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 `109 + 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, 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}, {1, 1, 3}, and {1, 2, 2}.

Constraints:

* `1 <= nums.length <= 2 * 104` * `0 <= nums[i] <= 2 * 104` * Sum of `nums` does not exceed `2 * 104`. * `0 <= l <= r <= 2 * 104`

Code Snippets

C++:

```
class Solution {
public:
int countSubMultisets(vector<int>& nums, int l, int r) {

}
};
```

Java:

```
class Solution {
public int countSubMultisets(List<Integer> nums, int l, int r) {

}
}
```

Python3:

```
class Solution:
def countSubMultisets(self, nums: List[int], l: int, r: int) -> int:
```