Problem 1498: Number of Subsequences That Satisfy the Given Sum Condition

Problem Information

Difficulty: Medium
Acceptance Rate: 49.39%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Sorting

Problem Description

You are given an array of integers `nums` and an integer `target`.

Return _the number of**non-empty** subsequences of _`nums` _such that the sum of the minimum and maximum element on it is less or equal to_`target`. Since the answer may be too large, return it **modulo** `109 + 7`.

Example 1:

Input: nums = [3,5,6,7], target = 9 **Output:** 4 **Explanation:** There are 4 subsequences that satisfy the condition. [3] -> Min value + max value <= target (3 + 3 <= 9) [3,5] -> (3 + 5 <= 9) [3,5,6] -> (3 + 6 <= 9) [3,6] -> (3 + 6 <= 9)

Example 2:

Input: nums = [3,3,6,8], target = 10 **Output:** 6 **Explanation:** There are 6 subsequences that satisfy the condition. (nums can have repeated numbers). [3] , [3] , [3,3], [3,6] , [3,6] , [3,3,6]

Example 3:

Input: nums = [2,3,3,4,6,7], target = 12 **Output:** 61 **Explanation:** There are 63 non-empty subsequences, two of them do not satisfy the condition ([6,7], [7]). Number of valid subsequences (63 - 2 = 61).

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 106` * `1 <= target <= 106`

Code Snippets

C++:

```
class Solution {
public:
int numSubseq(vector<int>& nums, int target) {

}
};
```

Java:

```
class Solution {
public int numSubseq(int[] nums, int target) {

}
}
```

Python3:

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