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 than target.

Since the answer may be too large, return it modulo 10^9 + 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 don't satisfy the condition ([6,7], [7]).

Number of valid subsequences (63 - 2 = 61).

**Example 4:**

**Input:** nums = [5,2,4,1,7,6,8], target = 16

**Output:** 127

**Explanation:** All non-empty subset satisfy the condition (2^7 - 1) = 127

**Constraints:**

* 1 <= nums.length <= 10^5
* 1 <= nums[i] <= 10^6
* 1 <= target <= 10^6