

Problem 377: Combination Sum IV

Problem Information

Difficulty: Medium

Acceptance Rate: 54.85%

Paid Only: No

Tags: Array, Dynamic Programming

Problem Description

Given an array of **distinct** integers `nums` and a target integer `target`, return _the number of possible combinations that add up to_ `target`.

The test cases are generated so that the answer can fit in a **32-bit** integer.

Example 1:

Input: nums = [1,2,3], target = 4 **Output:** 7 **Explanation:** The possible combination ways are: (1, 1, 1, 1) (1, 1, 2) (1, 2, 1) (1, 3) (2, 1, 1) (2, 2) (3, 1) Note that different sequences are counted as different combinations.

Example 2:

Input: nums = [9], target = 3 **Output:** 0

Constraints:

* `1 <= nums.length <= 200` * `1 <= nums[i] <= 1000` * All the elements of `nums` are **unique**. * `1 <= target <= 1000`

Follow up: What if negative numbers are allowed in the given array? How does it change the problem? What limitation we need to add to the question to allow negative numbers?

Code Snippets

C++:

```
class Solution {  
public:  
    int combinationSum4(vector<int>& nums, int target) {  
  
    }  
};
```

Java:

```
class Solution {  
public int combinationSum4(int[] nums, int target) {  
  
}  
}
```

Python3:

```
class Solution:  
    def combinationSum4(self, nums: List[int], target: int) -> int:
```