Problem 494: Target Sum

Problem Information

Difficulty: Medium
Acceptance Rate: 51.48%
Paid Only: No
Tags: Array, Dynamic Programming, Backtracking

Problem Description

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

You want to build an **expression** out of nums by adding one of the symbols `'+'` and `'-'` before each integer in nums and then concatenate all the integers.

* For example, if `nums = [2, 1]`, you can add a `'+'` before `2` and a `'-'` before `1` and concatenate them to build the expression `"+2-1"`.

Return the number of different **expressions** that you can build, which evaluates to `target`.

Example 1:

Input: nums = [1,1,1,1,1], target = 3 **Output:** 5 **Explanation:** There are 5 ways to assign symbols to make the sum of nums be target 3. -1 + 1 + 1 + 1 + 1 = 3 +1 - 1 + 1 + 1 + 1 = 3 +1 + 1 - 1 + 1 + 1 = 3 +1 + 1 + 1 - 1 + 1 = 3 +1 + 1 + 1 + 1 - 1 = 3

Example 2:

Input: nums = [1], target = 1 **Output:** 1

Constraints:

* `1 <= nums.length <= 20` * `0 <= nums[i] <= 1000` * `0 <= sum(nums[i]) <= 1000` * `-1000 <= target <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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