

给你一个整数数组 `nums` 和一个整数 `target` 。

向数组中的每个整数前添加 '+' 或 '-' ，然后串联起所有整数，可以构造一个 表达式 ：

例如，`nums = [2, 1]` ，可以在 2 之前添加 '+' ，在 1 之前添加 '-' ，然后串联起来得到表达式 `" +2-1 "` 。

返回可以通过上述方法构造的、运算结果等于 `target` 的不同 表达式 的数目。

494

回溯法解

```
class Solution {
public:
    int findTargetSumWays(std::vector<int>& nums, int target) {
        int res = 0;
        findTargetSumWays(nums, target, 0, res);
        return res;
    }

private:
    void findTargetSumWays(std::vector<int>& nums, int target, int i, int& res) {
        // 终止循环
        if (i == nums.size()) {
            if (target == 0) { // 找到一个组合
                res++;
            }
            return;
        }

        // 做一个选择
        target -= nums[i];
        findTargetSumWays(nums, target, i + 1, res);
        // 撤销选择
        target += nums[i];

        // 做一个选择
        target += nums[i];
        findTargetSumWays(nums, target, i + 1, res);
        // 撤销选择
        target -= nums[i];
    }
};
```