Target Sum

Problem Statement:

Problem Statement Suggest Edit

You are given an array 'ARR' of 'N' integers and a target number, 'TARGET'. Your task is to build an expression out of an array by adding one of the symbols '+' and '-' before each integer in an array, and then by concatenating all the integers, you want to achieve a target. You have to return the number of ways the target can be achieved.

For Example :

You are given the array 'ARR' = [1, 1, 1, 1, 1], 'TARGET' = 3. The number of ways this target can be achieved is:

1. -1 + 1 + 1 + 1 + 1 = 3
2. +1 - 1 + 1 + 1 + 1 = 3
3. +1 + 1 - 1 + 1 + 1 = 3
4. +1 + 1 + 1 - 1 + 1 = 3
5. +1 + 1 + 1 + 1 - 1 = 3

These are the 5 ways to make. Hence the answer is 5.

There are 5 ways I can assign +, -

arr = { 1, 2, 3, 1 }

target = 3

arr = {1, 2, 3, 1}

Basically

we have assign +, or -

So, we have 2 operators + or −, so, we can assign + to some element and − to some and sum should be our target.

+1 +1 −1 −1 −1

(1 +1) − (1 + 1 +1) = Target

S1 − S2 = Target

S1 − S2 = Target

↓

Partition Question

S1 + S2 = Total Sum

S2 = TS − S1

$S_1 - S_2 =$ Target

$TS - S_2 - S_2 =$ Target

$TS - 2S_2 =$ Target

$2S_2 = TS - Target$

$S_2 = \frac{TS - Target}{2}$ → Target Sum

let say,

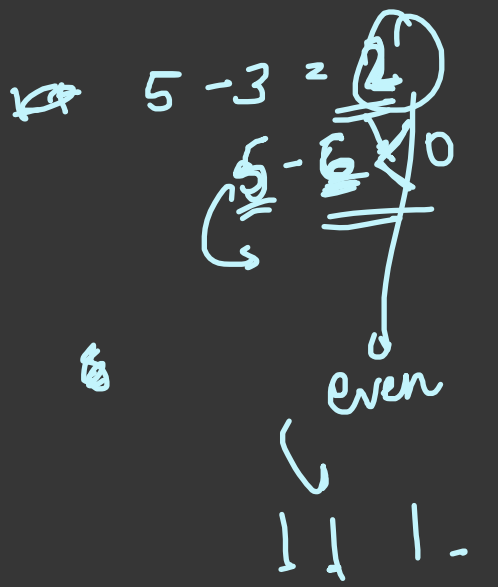
Total Sum - Target >=0 && even

f( )

return 1,

1 1 1 1

to



1 1 1 1

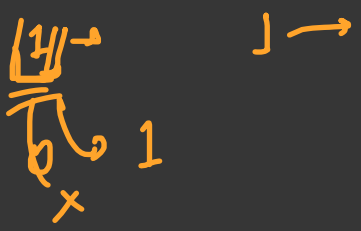
Recursion Relation: We have to count the no. of ways

if(target==0) return 1;

if(i==0) return arr[i]==target;

Base case

if



notpick = f(i-1, target);

pick = 0;

if(arr[i] <= k) pick = f(i-1, target - a[i]);

return pick + notpick

if Constraint allow 0.

0 1+1 = 2

```
if (i==0)
    if( k==0 && arr[0]==0)   return 02,
    if( k==0 || arr[0]==k]   return 1,
    else
    return 0;
```