494. Target Sum

Medium,
Dynamic Programming, DFS.

You are given a list of non-negative integers, a1, a2, ..., an, and a target, S. Now you have 2 symbols + and -. For each integer, you should choose one from + and - as its new symbol.

Find out how many ways to assign symbols to make sum of integers equal to target S.

Example 1:

```
Input: nums is [1, 1, 1, 1, 1], S is 3.
Output: 5
Explanation:

-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

There are 5 ways to assign symbols to make the sum of nums be target 3.
```

Note:
The length of the given array is positive and will not exceed 20.
The sum of elements in the given array will not exceed 1000.
Your output answer is guaranteed to be fitted in a 32-bit integer.

解法

该题是典型的dfs算法题。
用递归的方式把所有的可能性都遍历一遍。
最重要的地方是return的判断标准，
这里是当cur已经到头，恰好sum等于S时加一，

或者cur已经到头了，但sum不等一S时加0。
最后把所有累加就是答案。

Java

```
class Solution {
    public int findTargetSumWays(int[] nums, int S) {
        if(nums.length==0) return 0;
        return dfs(nums,0,0,S);
    }

    public int dfs(int[] nums,int cur, int sum, int S){
        int res = 0;
        int len = nums.length;
        if(cur==len && sum==S) return 1;
        else if(cur==len) return 0;
        res += dfs(nums,cur+1,sum+nums[cur],S);
        res += dfs(nums,cur+1,sum-nums[cur],S);
        return res;
    }
}
```

Scala

```
object Solution {
    def findTargetSumWays(nums: Array[Int], S: Int): Int = {
        if(nums.length==0) return 0
        dfs(nums,S,0,0)
    }

    def dfs(nums:Array[Int],S:Int,cur:Int,sum:Int):Int={
        var len = nums.length
        var res = 0
        if(cur==len && sum==S) return 1
        else if(cur==len) return 0
        res += dfs(nums,S,cur+1,sum+nums(cur))
        res += dfs(nums,S,cur+1,sum-nums(cur))
```

```
        res
    }
}
```