链滴

[每日 LeetCode] 682. Baseball Game

作者：Hanseltu
原文链接：https://ld246.com/article/1556892981666
来源网站：链滴

原文链接 [每日LeetCode] 682. Baseball Game

Description:

You're now a baseball game point recorder.

Given a list of strings, each string can be one of the 4 following types:

1. Integer (one round's score): Directly represents the number of points you get in this round.

2. "+" (one round's score): Represents that the points you get in this round are the sum of the last two valid round's points.

3. "D" (one round's score): Represents that the points you get in this round are the doubled d ta of the last valid round's points.

4. "C" (an operation, which isn't a round's score): Represents the last valid round's points you et were invalid and should be removed.

Each round's operation is permanent and could have an impact on the round before and the ound after.

You need to return the sum of the points you could get in all the rounds.

Example 1:

Input: ["5","2","C","D","+"]
Output: 30
Explanation:
Round 1: You could get 5 points. The sum is: 5.
Round 2: You could get 2 points. The sum is: 7.
Operation 1: The round 2's data was invalid. The sum is: 5.
Round 3: You could get 10 points (the round 2's data has been removed). The sum is: 15.
Round 4: You could get 5 + 10 = 15 points. The sum is: 30.

Example 2:

Input: ["5","-2","4","C","D","9","+","+"]
Output: 27
Explanation:
Round 1: You could get 5 points. The sum is: 5.
Round 2: You could get -2 points. The sum is: 3.
Round 3: You could get 4 points. The sum is: 7.
Operation 1: The round 3's data is invalid. The sum is: 3.
Round 4: You could get -4 points (the round 3's data has been removed). The sum is: -1.
Round 5: You could get 9 points. The sum is: 8.
Round 6: You could get -4 + 9 = 5 points. The sum is 13.
Round 7: You could get 9 + 5 = 14 points. The sum is 27.

思路：本题模拟棒球比赛计分，根据不同的字符串表示不同的分数，统计最后得分。按照题意分别判即可，可以使用数组也可用栈来实现。

C++代码（使用数组）

```
class Solution {
public:
    int calPoints(vector<string>& ops) {
        vector<int> v;
        for (string op : ops) {
            if (op == "+") {
                v.push_back(v.back() + v[v.size() - 2]);
            }
            else if (op == "D") {
                v.push_back(2 * v.back());
            }
            else if (op == "C") {
                v.pop_back();
            }
            else {
                v.push_back(stoi(op));
            }
        }
        return accumulate(v.begin(), v.end(), 0);
    }
};
```

运行时间：8ms

运行内存：8.6M

C++代码（使用栈）

```
class Solution {
public:
    int calPoints(vector<string>& ops) {
        stack<int> points;
    for (int i = 0; i < ops.size(); i++) {
        if (ops[i] == "+") {
        int temp = points.top();
        points.pop();
        int newvalue = temp + points.top();
        points.push(temp);
        points.push(newvalue);
        }
        else if (ops[i] == "C") points.pop();
        else if (ops[i] == "D") points.push(points.top() * 2);

        else points.push(stoi(ops[i]));
    }

    int sum = 0;
    while (!points.empty()) {
        sum += points.top();
        points.pop();
    }
```

```
    return sum;
    }
};
```

运行时间：8ms

运行内存：8.6M