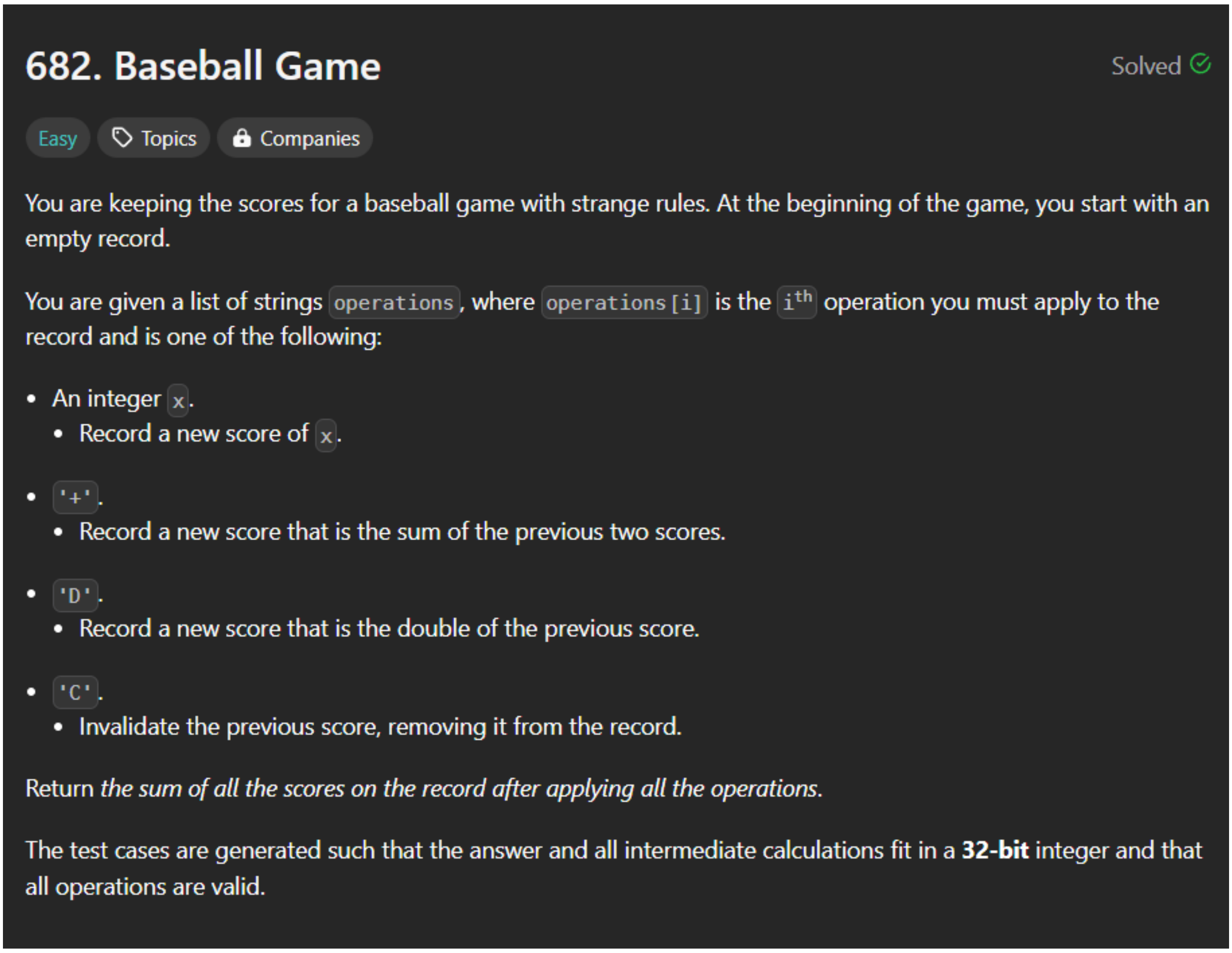


682. Baseball Game

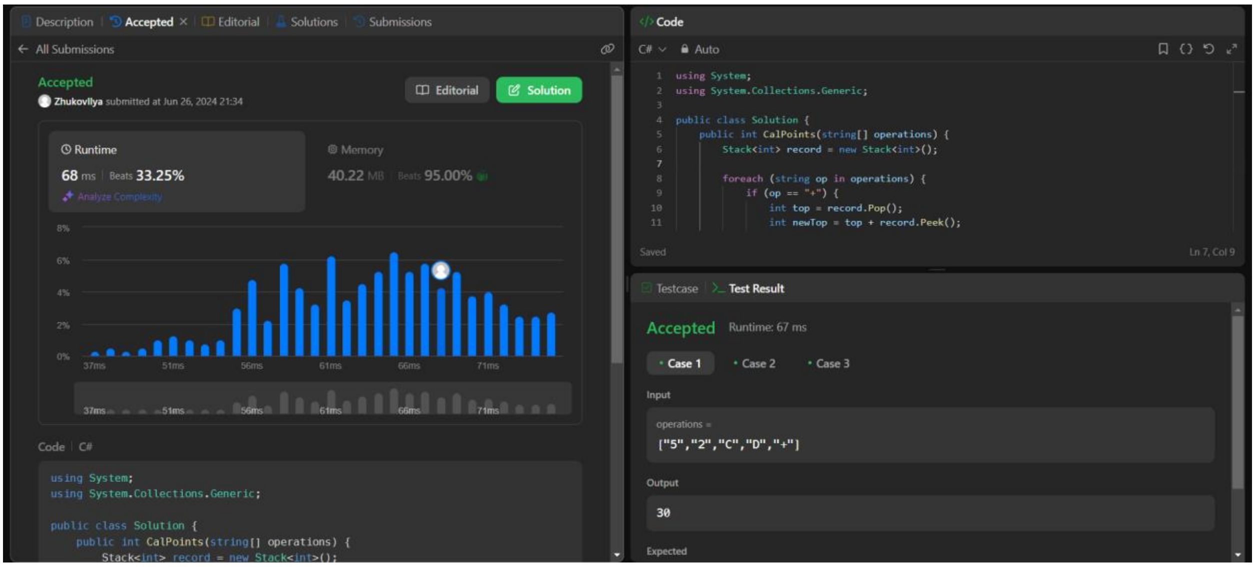
You are keeping the scores for a baseball game with strange rules. At the beginning of the game, you start with an empty record.

You are given a list of strings `operations`, where `operations[i]` is the i^{th} operation you must apply to the record and is one of the following:

- An integer `x`.
  - Record a new score of `x`.
- `'+'`.
  - Record a new score that is the sum of the previous two scores.
- `'D'`.
  - Record a new score that is the double of the previous score.
- `'C'`.
  - Invalidate the previous score, removing it from the record.

Return *the sum of all the scores on the record after applying all the operations.*

The test cases are generated such that the answer and all intermediate calculations fit in a **32-bit** integer and that all operations are valid.



Accepted
Zhukovllya submitted at Jun 26, 2024 21:34

Runtime
68 ms | Beats 33.25%

Memory
40.22 MB | Beats 95.00%

Code | C#

```
using System;
using System.Collections.Generic;

public class Solution {
    public int CalPoints(string[] operations) {
        Stack<int> record = new Stack<int>();
```

```
using System;
using System.Collections.Generic;

public class Solution {
    public int CalPoints(string[] operations) {
        Stack<int> record = new Stack<int>();

        foreach (string op in operations) {
            if (op == "+") {
                int top = record.Pop();
                int newTop = top + record.Peek();
```

Saved
Ln 7, Col 9

Accepted Runtime: 67 ms

Input

operations =
["5","2","C","D","+"]

Output

30

Expected

Код:

```
using System;
using System.Collections.Generic;

public class Solution
{
    public int CalPoints(string[] operations)
    {
        Stack<int> record = new Stack<int>();

        foreach (string op in operations)
        {
            if (op == "+")
            {
                int top = record.Pop();
                int newTop = top + record.Peek();
                record.Push(top);
                record.Push(newTop);
            }
            else if (op == "D")
            {
                record.Push(2 * record.Peek());
            }
            else if (op == "C")
            {
                record.Pop();
            }
            else
            {
                record.Push(int.Parse(op));
            }
        }

        int sum = 0;
        foreach (int score in record)
        {
            sum += score;
        }

        return sum;
    }
}
```