Baseball Game

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 data 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 get were invalid and should be removed.

Each round's operation is permanent and could have an impact on the round before and the round 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: 1
5.
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.
```

Note:

- The size of the input list will be between 1 and 1000.
- Every integer represented in the list will be between -30000 and 30000.

Solution 1

```java
class Solution {
    public int calPoints(String[] ops) {
        int sum = 0;
        LinkedList<Integer> list = new LinkedList<>();
        for (String op : ops) {
            if (op.equals("C")) {
                sum -= list.removeLast();
            }
            else if (op.equals("D")) {
                list.add(list.peekLast() * 2);
                sum += list.peekLast();
            }
            else if (op.equals("+")) {
                list.add(list.peekLast() + list.get(list.size() - 2));
                sum += list.peekLast();
            }
            else {
                list.add(Integer.parseInt(op));
                sum += list.peekLast();
            }
        }
        return sum;
    }
}
```

written by shawngao original link here

Solution 2

It not stated in the question but you can assume there is not invalid operations, such as `C` when the points history is empty, it still passed.

```python
class Solution(object):
    def calPoints(self, ops):
        # Time: O(n)
        # Space: O(n)
        history = []
        for op in ops:
            if op == 'C':
                history.pop()
            elif op == 'D':
                history.append(history[-1] * 2)
            elif op == '+':
                history.append(history[-1] + history[-2])
            else:
                history.append(int(op))
        return sum(history)
```

written by wanling original link here

Solution 3

I am confused!!!

written by linzihan original link here