224. Basic Calculator

QuestionEditorial Solution

My Submissions

- Total Accepted: **30737**
- Total Submissions:**133262**
- Difficulty: **Hard**

Implement a basic calculator to evaluate a simple expression string.

The expression string may contain open (and closing parentheses), the plus + or minus sign -, **non-negative** integers and empty spaces .

You may assume that the given expression is always valid.

Some examples:

"1 + 1" = 2 " 2-1 + 2 " = 3 "(1+(4+5+2)-3)+(6+8)" = 23

```
//C++
//Author : ZZW
class Solution {
public:
    bool level(char s2)
    {
        if(s2 == '(' || s2 == ')')
            return false;
        else return true;
    }
    int calculate(int a,int b,char op)
    {
        if(op == '+')
            return a+b;
        else
            return b-a;
    }
    int calculate(string s) {
        s.erase(remove(s.begin(),s.end(),' '),s.end());
        if(s.length()==0) return 0;
        if(s.length()==1) return s[0] - '0';
        stack<char> op;
        stack<int> vals;
        for(int i=0;i<s.size();i++)
        {
            if(s[i] >= '0' && s[i] <= '9')
            {
                string value;
                while(i<s.size() && s[i]>='0' && s[i] <='9')
                    value+=s[i++];
                vals.push(stoi(value));
            }
            if(s[i]=='(')
                op.push(s[i]);
```

```
            if(s[i]==')')
            {
                while(op.top()!='(')
                {
                    int a = vals.top();vals.pop();
                    int b = vals.top();vals.pop();
                    int c = calculate(a,b,op.top());
                    vals.push(c);
                    op.pop();
                }
                // pop '('
                op.pop();
            }
            if(s[i]=='-' || s[i] == '+')
            {
                while(!op.empty() && level(op.top()))
                {
                    int a = vals.top();vals.pop();
                    int b = vals.top();vals.pop();
                    int c = calculate(a,b,op.top());
                    vals.push(c);
                    op.pop();
                }
                op.push(s[i]);
            }
        }
        while(!op.empty() && level(op.top()))
        {
            int a = vals.top();vals.pop();
            int b = vals.top();vals.pop();
            int c = calculate(a,b,op.top());
            vals.push(c);
            op.pop();
        }
        return vals.top();
    }
};
```