6038. Minimize Result by Adding Parentheses to Expression

My Submissions (/contest/weekly-contest-288/problems/minimize-result-by-adding-parentheses-to-expression/submissions/)

Back to Contest (/contest/weekly-contest-288/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

You are given a **0-indexed** string `expression` of the form `"<num1>+<num2>"` where `<num1>` and `<num2>` represent positive integers.

Add a pair of parentheses to `expression` such that after the addition of parentheses, `expression` is a **valid** mathematical expression and evaluates to the **smallest** possible value. The left parenthesis **must** be added to the left of `'+'` and the right parenthesis **must** be added to the right of `'+'`.

Return `expression` *after adding a pair of parentheses such that* `expression` *evaluates to the* ***smallest*** *possible value.* If there are multiple answers that yield the same result, return any of them.

The input has been generated such that the original value of `expression`, and the value of `expression` after adding any pair of parentheses that meets the requirements fits within a signed 32-bit integer.

Example 1:

```
Input: expression = "247+38"
Output: "2(47+38)"
Explanation: The expression evaluates to 2 * (47 + 38) = 2 * 85 = 170.
Note that "2(4)7+38" is invalid because the right parenthesis must be to the right of the '+'.
It can be shown that 170 is the smallest possible value.
```

Example 2:

```
Input: expression = "12+34"
Output: "1(2+3)4"
Explanation: The expression evaluates to 1 * (2 + 3) * 4 = 1 * 5 * 4 = 20.
```

Example 3:

```
Input: expression = "999+999"
Output: "(999+999)"
Explanation: The expression evaluates to 999 + 999 = 1998.
```

Constraints:

- `3 <= expression.length <= 10`
- `expression` consists of digits from `'1'` to `'9'` and `'+'`.
- `expression` starts and ends with digits.
- `expression` contains exactly one `'+'`.
- The original value of `expression`, and the value of `expression` after adding any pair of parentheses that meets the requirements fits within a signed 32-bit integer.

JavaScript

```javascript
const minimizeResult = (s) => {
    let [a, b] = s.split('+'), d = [];
    for (let i = 0; i <= a.length; i++) {
        let al = a.slice(0, i), ar = a.slice(i);
        // pr("\nleft", al, ar);
        for (let j = 0; j <= b.length; j++) {
            let bl = b.slice(0, j), br = b.slice(j);
            let t = al + '(' + ar + '+' + bl + ')' + br;
            // pr("right", bl, br, "t", t, valid(t));
            if (valid(t)) {
                // pr("t", t);
                let v = cal(t);
                d.push([t, v]);
            }
        }
    }
    d.sort((x, y) => x[1] - y[1]);
    // pr(d);
    return d[0][0];
};

const cal = (s) => {
    let l = s.indexOf('('), m = s.indexOf('+'), r = s.indexOf(')');
    let a = s.slice(0, l), b = s.slice(l + 1, m), c = s.slice(m + 1, r), d = s.slice(r + 1);
    let xa = a.length == 0 ? 1 : a - '0';
    let xb = b - '0';
    let xc = c - '0';
    let xd = d.length == 0 ? 1 : d - '0';
    let v = xa * (xb + xc) * xd;
    // pr("each", 'a', a, 'b', b, 'c', c, 'd', d, v);
    return v;
};

const valid = (s) => {
    if (s.indexOf('+)') != -1 || s.indexOf('(+') != -1) return false;
    return true;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/677353611/) More Details (/submissions/detail/677353611/)

Share your acceptance!

1

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)