

241. Different Ways to Add Parentheses

题目描述： <https://leetcode.com/problems/different-ways-to-add-parentheses/>

给定一个不加括号的式子，算这个式子的加括号可能性，以及每种可能性的结果。

例如: "2 * 3 - 4 * 5"

```
(2*(3-(4*5))) = -34
(2*3)-(4*5) = -14
((2*(3-4))*5) = -10
(2*((3-4)*5)) = -10
(((2*3)-4)*5) = 10
```

解题思路：

递归!!!! 可以记录过程也可不计。就是时间短

代码：

```
class Solution {
public:
    vector<int> findRes(string input, int begin, int end, vector<vector<vector<int>>> &matrix) {
        vector<int> v;
        if(begin > end) {
            return v;
        }
        if(matrix[begin][end].size()) return matrix[begin][end];
        for(int i = begin; i <= end; i++) {
            if(!isdigit(input.at(i))) {
                vector<int> lv = findRes(input, begin, i-1, matrix);
                vector<int> rv = findRes(input, i+1, end, matrix);
                for(auto iteml:lv) {
                    for(auto itemr:rv) {
                        if(input.at(i) == '+') {
                            v.push_back(iteml + itemr);
                        }
                        else if(input.at(i) == '-') {
                            v.push_back(iteml - itemr);
                        }
                        else {
                            v.push_back(iteml * itemr);
                        }
                    }
                }
            }
        }
        matrix[begin][end] = v.size()? v : vector<int>{stoi(input.substr(begin, end-begin+1))};
        return matrix[begin][end];
    }
    vector<int> diffWaysToCompute(string input) {
        int len = input.size();
        vector<vector<vector<int>>> > matrix(len, vector<vector<int>>(len));
        vector<int> res = findRes(input, 0, len-1, matrix);
        return res;
    }
};
```