Problem 241: Different Ways to Add Parentheses

Problem Information

Difficulty: Medium
Acceptance Rate: 72.83%
Paid Only: No
Tags: Math, String, Dynamic Programming, Recursion, Memoization

Problem Description

Given a string `expression` of numbers and operators, return _all possible results from computing all the different possible ways to group numbers and operators_. You may return the answer in **any order**.

The test cases are generated such that the output values fit in a 32-bit integer and the number of different results does not exceed `104`.

Example 1:

Input: expression = "2-1-1" **Output:** [0,2] **Explanation:** ((2-1)-1) = 0 (2-(1-1)) = 2

Example 2:

Input: expression = "2*3-4*5" **Output:** [-34,-14,-10,-10,10] **Explanation:** (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

Constraints:

* `1 <= expression.length <= 20` * `expression` consists of digits and the operator `'+'`, `'-'`, and `'*'`. * All the integer values in the input expression are in the range `[0, 99]`. * The integer values in the input expression do not have a leading `'-'` or `'+'` denoting the sign.

Code Snippets

C++:

```
class Solution {
public:
vector<int> diffWaysToCompute(string expression) {

}
};
```

Java:

```
class Solution {
public List<Integer> diffWaysToCompute(String expression) {

}
}
```

Python3:

```
class Solution:
def diffWaysToCompute(self, expression: str) -> List[int]:
```