Problem 553: Optimal Division

Problem Information

Difficulty: Medium
Acceptance Rate: 62.47%
Paid Only: No
Tags: Array, Math, Dynamic Programming

Problem Description

You are given an integer array `nums`. The adjacent integers in `nums` will perform the float division.

* For example, for `nums = [2,3,4]`, we will evaluate the expression `"2/3/4"`.

However, you can add any number of parenthesis at any position to change the priority of operations. You want to add these parentheses such the value of the expression after the evaluation is maximum.

Return _the corresponding expression that has the maximum value in string format_.

Note: your expression should not contain redundant parenthesis.

Example 1:

Input: nums = [1000,100,10,2] **Output:** "1000/(100/10/2)" **Explanation:** 1000/(100/10/2) = 1000/((100/10)/2) = 200 However, the bold parenthesis in "1000/(**(** 100/10**)** /2)" are redundant since they do not influence the operation priority. So you should return "1000/(100/10/2)". Other cases: 1000/(100/10)/2 = 50 1000/(100/(10/2)) = 50 1000/100/10/2 = 0.5 1000/100/(10/2) = 2

Example 2:

Input: nums = [2,3,4] **Output:** "2/(3/4)" **Explanation:** (2/(3/4)) = 8/3 = 2.667 It can be shown that after trying all possibilities, we cannot get an expression with evaluation greater than 2.667

Constraints:

* `1 <= nums.length <= 10` * `2 <= nums[i] <= 1000` * There is only one optimal division for the given input.

Code Snippets

C++:

```cpp
class Solution {
public:
string optimalDivision(vector<int>& nums) {


}
};
```

Java:

```java
class Solution {
public String optimalDivision(int[] nums) {


}
}
```

Python3:

```python
class Solution:
def optimalDivision(self, nums: List[int]) -> str:
```