

## Documentation Optimal Division

### 📖 Table of Contents

1. [Problem Statement](#)
2. [Intuition](#)
3. [Key Observations](#)
4. [Approach](#)
5. [Edge Cases](#)
6. [Complexity Analysis](#)
  - [Time Complexity](#)
  - [Space Complexity](#)
7. [Alternative Approaches](#)
8. [Algorithm](#)
9. [Test Cases](#)
10. [Final Thoughts](#)

#### 1. 🚀 Problem Statement

Given a list of integers `nums`, perform float division from left to right ( $a / b / c / d$ ) and place parentheses in such a way that the final result is maximized. Return the string representation of the expression with no redundant parentheses.

#### 2. 💡 Intuition

We want to maximize the result of sequential divisions. Since division is not associative, placing parentheses changes the result. To maximize the outcome, we need to minimize the denominator. Hence, group the entire denominator using parentheses.

### 3. 🔍 Key Observations

- Division is not associative:  $a / (b / c) \neq (a / b) / c$
- Wrapping all but the first number in parentheses gives the highest result.
- The only case that needs parentheses is when the list has more than 2 elements.

### 4. ☐ Approach

- If only one number  $\rightarrow$  return it as string.
- If two numbers  $\rightarrow$  return "a/b"
- If more than two  $\rightarrow$  return "a/(b/c/d/...)" (combine all except first inside one set of parentheses).

### 5. ☐ Edge Cases

- [1000]  $\rightarrow$  Output: "1000"
- [2, 3]  $\rightarrow$  Output: "2/3"
- [2, 3, 4]  $\rightarrow$  Output: "2/(3/4)"

### 6. 📊 Complexity Analysis

#### ☐ Time Complexity

- $O(n) \rightarrow$  Building the output string with join() requires visiting each element once.

#### 📦 Space Complexity

- $O(n) \rightarrow$  Temporary space to store the string representation of numbers.

## 7. 🔄 Alternative Approaches

- Brute Force DFS: Try all parenthesis placements (exponential time).
- ✖ Too slow for  $n = 10$
- Greedy Approach (Optimal Used Here): Only one optimal way due to predictable math structure.

## 8. □ Algorithm

- Check the length of nums.
- If  $\text{length} == 1 \rightarrow \text{return str}(\text{nums}[0])$
- If  $\text{length} == 2 \rightarrow \text{return "a/b"}$
- Else:
  - Convert all numbers from index 1 to end into string.
  - Join them using '/'.
  - Wrap them in () and prefix with  $\text{nums}[0] + '/'$ .

## 9. ✔ Test Cases

Input	Expected Output	Explanation
[1000]	"1000"	Single number
[2, 3]	"2/3"	No parentheses needed
[2, 3, 4]	"2/(3/4)"	Grouped denominator
[1000,100,10,2]	"1000/(100/10/2)"	Grouping maximizes value
[6,2,3,4]	"6/(2/3/4)"	Best grouping for max result

## 10. □ Final Thoughts

This problem teaches a key lesson about operator associativity and how grouping affects computation. Instead of brute force, recognizing mathematical patterns allows a greedy and efficient solution. With a solid grasp of number behavior, the problem reduces to a string formatting task.