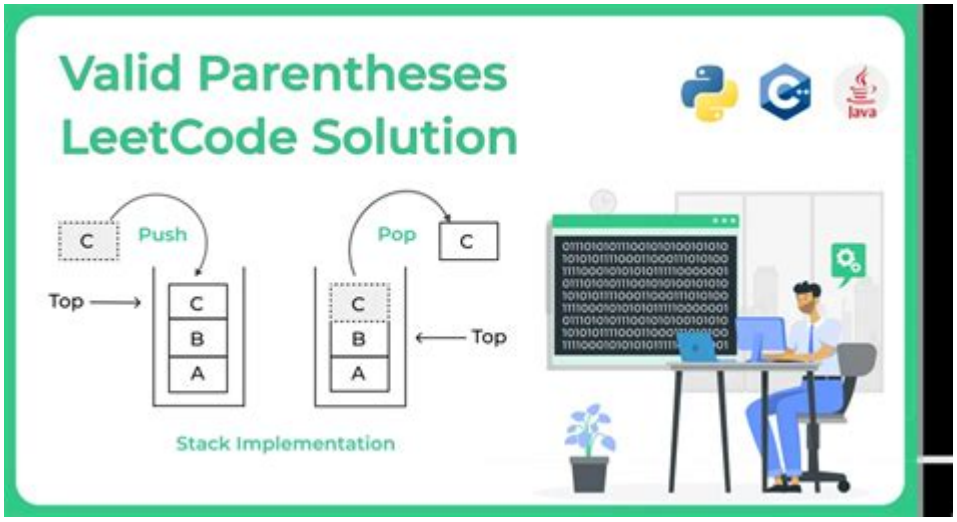
Valid Parentheses Leetcode Solution



Valid Parentheses LeetCode Solution

The problem of validating parentheses is a common interview question that can be found on platforms like LeetCode. It challenges candidates to develop a solution that verifies whether a string containing parentheses is valid. A string is considered valid if every opening parenthesis has a corresponding closing parenthesis and the pairs are correctly nested and ordered. In this article, we will explore the problem in detail, discuss various approaches to solving it, and provide a comprehensive solution using different programming languages.

Understanding the Problem

Before diving into the solution, it is essential to grasp the problem's requirements. The string can contain three types of parentheses:

1. Round brackets: `()`
2. Square brackets: `[]`
3. Curly braces: `{}`

To determine if the string is valid, the following conditions must be met:

- Every opening parenthesis must have a matching closing parenthesis.
- Parentheses must close in the correct order. For instance, `({[]})` is valid, while `([)]` is not.

Examples

To clarify the requirements, let's look at some examples:

- Input: `()`
- Output: `true`

- Input: `()[]{}`
- Output: `true`

- Input: `(]`
- Output: `false`

- Input: `([)]`
- Output: `false`

- Input: `{[]} `
- Output: `true`

Approaches to Solve the Problem

There are several approaches to solving the valid parentheses problem, including:

1. Stack Data Structure
2. Using a Counter
3. Recursion

Let's delve into the most common and efficient approach, which is using a stack.

Stack Data Structure

The stack data structure is particularly suitable for this problem because it operates on a Last In First Out (LIFO) principle. This allows us to easily manage the order of opening and closing parentheses.

Algorithm:

1. Initialize an empty stack.
2. Iterate through each character in the input string:
- If it is an opening parenthesis (`(`, `{`, or `[`), push it onto the stack.
- If it is a closing parenthesis (`)`, `}`, or `]`):
- Check if the stack is empty. If it is, return `false` because there is no corresponding opening parenthesis.
- Pop the top element from the stack and check if it matches the current closing parenthesis. If it does not match, return `false`.
3. After iterating through the string, check if the stack is empty. If it is, the parentheses are valid; otherwise, they are not.

Time Complexity:
- The time complexity of this approach is O(n), where n is the length of the input string, since we are iterating through the string only once.

Space Complexity:
- The space complexity is O(n) in the worst case when all characters are opening parentheses.

Implementing the Solution

Now that we've established the algorithm, let’s implement it in several popular programming languages: Python, Java, and JavaScript.

Python Implementation

```python
def isValid(s: str) -> bool:
stack = []
mapping = {')': '(', '}': '{', ']': '['}

for char in s:
if char in mapping:
top_element = stack.pop() if stack else ''
if mapping[char] != top_element:
return False
else:
stack.append(char)

return not stack
```

Explanation:
- We use a dictionary to map closing parentheses to their corresponding opening ones.
- The loop checks if the character is a closing parenthesis; if so, it checks the top of the stack. If it’s not a match, it returns `false`.
- Finally, we return `True` if the stack is empty, indicating all parentheses were matched.

Java Implementation

```java
import java.util.Stack;

public class ValidParentheses {
```

```
public boolean isValid(String s) {
Stack stack = new Stack<>();
for (char c : s.toCharArray()) {
if (c == '(' || c == '{' || c == '[') {
stack.push(c);
} else {
if (stack.isEmpty()) {
return false;
}
char top = stack.pop();
if ((c == ')' && top != '(') ||
(c == '}' && top != '{') ||
(c == ']' && top != '[')) {
return false;
}
}
}
return stack.isEmpty();
}
}
```

Explanation:
- Similar to the Python implementation, we utilize a stack to keep track of opening parentheses.
- We check for matches and return `false` if any mismatches occur.

JavaScript Implementation

```javascript
function isValid(s) {
const stack = [];
const mapping = { ')': '(', '}': '{', ']': '[' };

for (let char of s) {
if (char in mapping) {
const topElement = stack.length === 0 ? '' : stack.pop();
if (mapping[char] !== topElement) {
return false;
}
} else {
stack.push(char);
}
}

return stack.length === 0;
}
```

Explanation:
- The JavaScript implementation follows the same logic as the Python and Java versions, using an array as a stack.

Handling Edge Cases

When solving the valid parentheses problem, it’s also crucial to consider edge cases:

1. Empty String: An empty string should return `true`, as there are no unmatched parentheses.
2. Single Parenthesis: Strings such as `(` or `[` should return `false`, as they are unmatched.
3. Long Strings: Performance may be tested with very long strings to ensure that the algorithm runs efficiently.

Conclusion

The valid parentheses problem is a classic example of using a stack to solve problems that involve nested structures. The approach discussed here is both efficient and straightforward, making it an excellent choice for interviews and real-world applications. By mastering this problem, you can gain a deeper understanding of stack data structures and their applications in various programming challenges.

Frequently Asked Questions

What is the problem statement for 'Valid Parentheses' on LeetCode?

The problem asks to determine if a string containing just the characters '(', ')', '{', '}', '[' and ']' is valid. An input string is valid if the brackets are closed in the correct order.

What is the optimal time complexity for solving the 'Valid Parentheses' problem?

The optimal time complexity for solving the 'Valid Parentheses' problem is O(n), where n is the length of the input string.

What data structure is commonly used to solve the

'Valid Parentheses' problem?

A stack data structure is commonly used to solve the 'Valid Parentheses' problem, as it allows for the Last-In-First-Out (LIFO) processing required for matching brackets.

How do you handle edge cases in the 'Valid Parentheses' problem?

Edge cases include checking for an empty string (which is valid) and ensuring that the parentheses are correctly matched and nested. If the count of opening and closing brackets ever mismatches during traversal, the string is invalid.

Can you provide a sample Python code solution for 'Valid Parentheses'?

Sure! Here is a sample solution in Python:

```python
def isValid(s):
stack = []
parentheses_map = {')': '(', '}': '{', ']': '['}
for char in s:
if char in parentheses_map:
top_element = stack.pop() if stack else ''
if parentheses_map[char] != top_element:
return False
else:
stack.append(char)
return not stack
```

What are the common pitfalls to avoid when implementing the solution?

Common pitfalls include not properly handling cases where the stack is empty when trying to pop an element, and not checking if the stack is empty at the end of the function, which indicates unmatched opening brackets.

What is the significance of using a hash map in the solution?

Using a hash map allows for O(1) time complexity when checking for corresponding opening brackets, improving the efficiency of the solution compared to using a linear search.

Valid Parentheses Leetcode Solution

is not a valid integer value 是什么意思 **-** 百度知道

Dec 30, 2024 · “is not a valid integer value”是一个计算机编程中的常见错误提示，意思是输入的值不是一个有效的整数。详细解释：1. 整数与有效值的概念：在计算机编程中，整数是一种基 …

储蓄卡上的validthru是什么意思 - 百度知道

Mar 1, 2022 · 储蓄卡上的validthru是什么意思是有效期的意思。通储蓄卡和信用卡在卡面上有一个比较重大的区别就是，信用卡上印有有效期，通常格式为：Valid From 12/08, Valid Thru …

数据透视时提示“数据透视表字段名无效” - 百度经验

Apr 21, 2014 · 如下图，在插入----数据透视表时，EXCEL提示“数据透视表字段名无效。在创建透视表时，必须使用组合为带有标志列列表的 …

请问大家怎么填写*Date of Birth (MM/DD/YYYY)*? _百度知道

请问大家怎么填写Date of Birth (MM/DD/YYYY)? 提示上写的 你可以选择一下的。

*valid until*是什么意思 - 百度知道

Apr 10, 2024 · valid until的意思是“有效期至”。在日常生活和各种文档中，我们经常能看到“valid until”这个词组。它通常用于表示某个事物、文件或服务的有效期限。这个词组直译过来就是“ …

valid thru是什么意思? _百度知道

Feb 9, 2024 · validthru也有标为goodthru的情况。 valid thru过期会怎样 valid thru一般指信用卡有效截止时间，过了这个时间信用卡就不能使用了。比如valid thru，12/21表示这张信用卡的有 …

*valid percent*和*Cumulative Percent*是什么 - 百度知道

Aug 25, 2013 · valid percent和Cumulative Percent是什么ValidPercent:有效百分比 (ValidPercent):各频数占有效样本数的百分比.cumulativepercent:1.累积比 …

怎样填这些? 信用卡号码、**cvv2**、截止日期**YY\MM** 。_百度知道

怎样填这些? 信用卡号码、cvv2、截止日期YY\MM。信用卡号码、cvv2、截止日期YY\MM都能在信用卡上找到，找到后填写清楚即可。信用卡号码印在信用卡正面，一般为16位数字组 …

validthru是什么意思 **-** 百度知道

validthru是什么意思validthru：网络释义：卡片正面的一般是用于信用卡有效期。卡面上一般都会显示：validthru 09/12，那就表示，这张卡的有效期截止到2012年09月（30日24：00）在信 …

中国农业银行借记卡上写（**MONTH/YEAR**）和（**VALID THRU**）是 **…**

中国农业银行借记卡上写（MONTH/YEAR）和（VALID THRU）是什么意思? 农业银行的信用卡卡面上会有该标识，分别代表：月/年和有效期，旁边显示的4位数字代表的是您卡片的有效期。

is not a valid integer value 是什么意思 **-** 百度知道

Dec 30, 2024 · “is not a valid integer value”是一个计算机编程中的常见错误提示，意思是输入的值不是一个有效的整数。详细解释：1. 整数与有效值的概念：在计算机编程中，整数是一种基本数据类型，表示没有小数部分的数值。当程序或系统要求用户输入一个整数时，它会期望得到一个符合整数规范的数值。而 …

储蓄卡上的**validthru**是什么意思 **-** 百度知道

Mar 1, 2022 · 储蓄卡上的validthru是什么意思是有效期的意思。通储蓄卡和信用卡在卡面上有一个比较重大的区别就是，信用卡上印有有效期，通常格式为：Valid From 12/08, Valid Thru 12/11。表示这张信用卡的有效期是从200

数据透视时提示“数据透视表字段名无效” - 百度经验

Apr 21, 2014 · 如下图，在插入----数据透视表时，EXCEL提示“数据透视表字段名无效。在创建透视表时，必须使用组合为带有标志列列表的 ...

请问大家怎么填写Date of Birth (MM/DD/YYYY)？_百度知道

请问大家怎么填写Date of Birth (MM/DD/YYYY)？提示上写的 你可以选择一下的。

valid until是什么意思 - 百度知道

Apr 10, 2024 · valid until的意思是“有效期至”。在日常生活和各种文档中，我们经常能看到“valid until”这个词组。它通常用于表示某个事物、文件或服务的有效期限。这个词组直译过来就是“有效直到”，意味着在这个指定的日期或时间之前，该事物、文件或服务是有效的。一旦超过了这个日期或时间，它就 ...

valid thru是什么意思？_百度知道

Feb 9, 2024 · validthru也有标为goodthru的情况。 valid thru过期会怎样 valid thru一般指信用卡有效截止时间，过了这个时间信用卡就不能使用了。比如valid thru，12/21表示这张信用卡的有效期到2021年12月。过期后用户还能继续还款，但是不能刷卡消费了。

valid percent和Cumulative Percent是什么 - 百度知道

Aug 25, 2013 · valid percent和Cumulative Percent是什么ValidPercent:有效百分比 (ValidPercent):各频数占有效样本数的百分比.cumulativepercent:1.累积比率cumulativefrequency累积频率cumulativepercent累积比率2.累积百分比...

怎样填这些？信用卡号码、cvv2、截止日期YY\MM 。_百度知道

怎样填这些？信用卡号码、cvv2、截止日期YY\MM。信用卡号码、cvv2、截止日期YY\MM都能在信用卡上找到，找到后填写清楚即可。信用卡号码印在信用卡正面，一般为16位数字组成；cvv2在信用卡背面，签名栏那7位

validthru是什么意思 - 百度知道

validthru是什么意思validthru：网络释义：卡片正面的一般是用于信用卡有效期。卡面上一般都会显示：validthru 09/12，那就表示，这张卡的有效期截止到2012年09月（30日24：00）在信用卡卡面上都明确的标有该信

中国农业银行借记卡上写（*MONTH/YEAR*）和（*VALID THRU*）是 *...*

中国农业银行借记卡上写（MONTH/YEAR）和（VALID THRU）是什么意思？农业银行的信用卡卡面上会有该标识，分别代表：月/年和有效期，旁边显示的4位数字代表的是您卡片的有效期。

Discover the best valid parentheses LeetCode solution with step-by-step explanations and code examples. Master this challenge today! Learn more now!

Back to Home