#1299. Replace Elements with Greatest Element on Right Side

20. Valid Parentheses

Easy 15480 767 Add to List Share

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

```
Input: s = "()"
Output: true
```

Example 2:

```
Input: s = "()[]{}"
Output: true
```

Example 3:

```
Input: s = "(]"
Output: false
```

Code

```python
class Solution:
    def isValid(self, s: str) -> bool:
        opened = []
        for i in s:
            if i == '(' or i == '{' or i == '[':
                opened.append(i)
            elif len(opened) > 0:
                if i == ')':
                    if opened[-1] != '(':
                        return False
                    opened.pop()
                elif i == '}':
                    if opened[-1] != '{':
                        return False
                    opened.pop()
                elif i == ']':
                    if opened[-1] != '[':
                        return False
                    opened.pop()
            else:
                return False
        if len(opened) == 0:
            return True
        else:
            return False
```

The purpose of the code is to receive a String value which contains parentheses and determine whether the value follows correct grammar or not. The challenging part of the problem was that there are many conditions to consider because it includes three different types of parentheses “()”, “{}“, and “[]”.