

Problem 2

Valid Parentheses

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

Constraints:

- $1 \leq s.length \leq 10^4$
- *s* consists of parentheses only '()[]{}'.

Link: <https://leetcode.com/problems/valid-parentheses/>

```
class Solution {
public boolean isValid(String s) {
    Stack<Character> 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();
}
}

```

The screenshot displays the LeetCode interface for the 'Valid Parentheses' problem. The left sidebar shows the problem list with 'Valid Parentheses' selected. The main area shows the problem description and the user's submission status as 'Accepted'. The right sidebar displays performance metrics: Runtime 2 ms, Beats 81.43%, Memory 40.7 MB, and Beats 44.64%. Below the metrics is a code editor showing the Java solution for the problem.

Problem: Valid Parentheses - LeetCode

Submission Status: Accepted

Next question: 21. Merge Two Sorted Lists

More challenges: 22. Generate Parentheses, 32. Longest Valid Parentheses, 301. Remove Invalid Parentheses

Performance Metrics:

- Runtime: 2 ms
- Beats: 81.43%
- Memory: 40.7 MB
- Beats: 44.64%

Notes: Write your notes here

Related Tags: Select tags (0/5)

Code Editor:

```

class Solution {
    public boolean isValid(String s) {
        Stack<Character> 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();
    }
}

```

Console: Run Submit