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 <= s.length <= 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();
}
}
```