class Solution:

def isValid(self, s: str) -> bool:

stack = []

# A mapping of closing brackets to their corresponding opening brackets

bracket\_map = {')': '(', ']': '[', '}': '{'}

for char in s:

if char in bracket\_map: # If the character is a closing bracket

# Check if the stack is empty or the top element doesn't match

if not stack or stack.pop() != bracket\_map[char]:

return False

else: # If the character is an opening bracket

stack.append(char)

# After iterating through the string, the stack should be empty for a valid string

return not stack