Valid Parentheses : Given a string s containing just the characters '(', ')', '{', '}', '[', ']', determine if s is valid.

s 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.

★ Stack: Last in / First Out (LIFO)

Ex) s = "()"
output: True

s = "()[]{}"
output: True

s = "(]"
output: False

s = "([{}])" ([{}])
out: True

1. stack : (
2. " : ([
3. " : ([{
4. '}' is not in mapping, check } == mapping[stack.pop()]
5. ']' "
6. ')' "

Idea:

① Create mapping (+stack) : mapping = {')':'(', ']':'[', '}':'{'} : space: O(n)

② for i in s: time complexity: O(n)

check i is in mapping

if yes, push to stack

else, pop from stack and check i == mapping[stack.pop()]

③ If stack is empty, return True.