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

An input string is valid if:

* Open brackets must be closed by the same type of brackets.
* Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

**Example 1:**

Input: "()"

Output: true

**Example 2:**

Input: "()[]{}"

Output: true

**Example 3:**

Input: "(]"

Output: false

**Example 4:**

Input: "([)]"

Output: false

**Example 5:**

Input: "{[]}"

Output: true

Possible solution:

def isValid(s):

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

open\_par = set(["(", "[", "{"])

stack = []

for i in s:

if i in open\_par:

stack.append(i)

elif stack and i == bracket\_map[stack[-1]]:

stack.pop()

else:

return False

return stack == []

Computational thinking:

Pattern recognition: There is a pattern regarding the number of brackets and numbers of combinations. Also, the way brackets are ordered has a pattern.