20. Valid Parentheses

Easy

Given a string 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.

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

class Solution {

public:

char mapper(char c){

if(c=='(') return ')';

else if (c=='{') return '}';

else return ']';

}

bool openingBrace(char c){

if(c=='('||c=='{'||c=='[') return true;

return false;

}

bool isValid(string s) {

stack<char> st;

int len=s.size();

for(int i=0;i<len;i++){

char curr=s[i];

if(openingBrace(curr)){

st.push(mapper(curr));

}

else{

if(st.empty()==true||st.top()!=curr) return false;

st.pop();

}

}

if(st.empty()==false) return false;

return true;

}

};

Success

[Details](https://leetcode.com/submissions/detail/205155703/)

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Valid Parentheses.

Memory Usage: 798.7 KB, less than 54.27% of C++ online submissions for Valid Parentheses.