

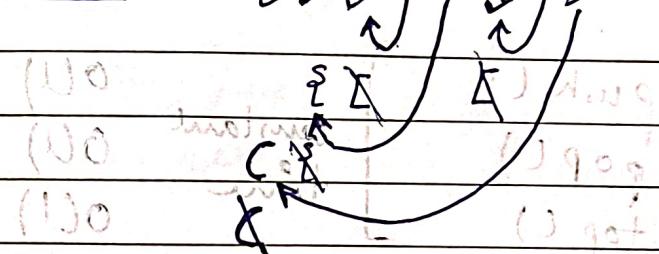
# Valid Parentheses

(LC-20)

( ) { } [ ]

- ① Open brackets must be closed by same type of brackets.
- ② Open brackets must be closed by bracket in correct order.
- ③ Every close bracket has corresponding open bracket of same type.

for eg:- ( { [ ] } [ ] )



1st opening bracket      First closing bracket

Stack (char)

Pseudocode - bool isValid (str) {

    Stack (char) st;

    for (i=0 to n) { // opening

        if (str[i] == {)

            return false;

        else // closing char

            if (str[i] match with st.top())

                st.pop();

            else

                return false;

open      closing

    } return st.size() == 0;

#Code :-

bool isValid (string str) {

stack<char> st;

for (int i=0 ; i < str.size() ; i++) {

if (str[i] == '(' ||

str[i] == '{' ||

str[i] == '[') { st.push(str[i]); }

else {

if (st.size() == 0) { return false; }

if ((st.top() == '(' && str[i] == ')') ||

(st.top() == '{' && str[i] == '}') ||

(st.top() == '[' && str[i] == ']')) {

st.pop();

} else { return false; }.

}

return st.size() == 0;

}