

<https://leetcode.com/problems/valid-parentheses/>

```
class Solution {
public:
    bool isValid(string s) {
        stack<char> res;
        for(char i: s){
            switch(i){
                case '(':
                    res.push(i);
                    break;
                case ')':
                    if (res.empty()!=1 && res.top()=='('){
                        res.pop();}
                    else{
                        res.push(i);
                    }
                    break;
                case '[':
                    res.push(i);
                    break;
                case ']':
                    if (res.empty()!=1 && res.top()=='['){
                        res.pop();}
                    else{
                        res.push(i);
                    }
                    break;
                case '{':
                    res.push(i);
```

```

        break;

    case '}':

        if (res.empty()!=1 && res.top()!=''){

            res.pop();}

        else{

            res.push(i);

        }

        break;

    }

}

return res.empty();

}

};

```

The screenshot displays the LeetCode interface for the 'Valid Parentheses' problem (Problem 20). The problem description states: 'Given a string s 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 brackets. 2. Open brackets must be closed in the correct order. 3. Every close bracket has a corresponding open bracket of the same type.' Examples provided are: Example 1: Input: s = "()", Output: true; Example 2: Input: s = "([)]", Output: false; Example 3: Input: s = "{[]}", Output: true.

The submission status is 'Accepted', submitted by 'coderr' on Jul 17, 2024, at 15:30. The runtime is 0 ms, beating 100.00% of submissions. The memory usage is 7.88 MB, beating 26.11% of submissions. A bar chart shows the runtime distribution across different time limits (1ms, 2ms, 3ms).

The C++ code for the solution is as follows:

```

1  class Solution {
2  public:
3      bool isValid(string s) {
4          stack<char> res;
5          for (int i = 0; i < s.length(); i++) {
6              if (s[i] == '(' || s[i] == '{' || s[i] == '[') {
7                  res.push(s[i]);
8              } else {
9                  if (res.empty() != 1 && res.top() == s[i]) {
10                     res.pop();
11                 } else {
12                     return false;
13                 }
14             }
15             if (i == s.length() - 1 && !res.empty()) {
16                 return false;
17             }
18             if (s[i] == '(' && res.top() == ')') {
19                 res.pop();
20             } else if (s[i] == '{' && res.top() == '}') {
21                 res.pop();
22             } else if (s[i] == '[' && res.top() == ']') {
23                 res.pop();
24             } else {
25                 return false;
26             }
27             if (res.empty() != 1 && res.top() == s[i]) {
28                 res.push(s[i]);
29             } else {
30                 return false;
31             }
32             if (res.empty() != 1 && res.top() == s[i]) {
33                 res.push(s[i]);
34             } else {
35                 return false;
36             }
37             if (res.empty() != 1 && res.top() == s[i]) {
38                 res.push(s[i]);
39             } else {
40                 return false;
41             }
42             return res.empty();
43         }
44     }
45 };

```