

```

1  class Solution {
2  public:
3      bool isValid(string s) {
4          // 先用哈希表做一个字符串与数字的映射
5          unordered_map<char, int> hashtable{
6              {'(', 1}, {'{', 2}, {'[', 3}, {')', -1}, {'}', -2},
7              {']', -3},
8          };
9          // 定义一个栈
10         stack<char> char_zhan;
11         // 字符串的长度
12         int N = s.length();
13         for (int i = 0; i < N; i++) {
14             int temp = hashtable[s[i]];
15             if (temp > 0) {
16                 // 左括号入栈
17                 char_zhan.push(s[i]);
18             } else {
19                 // 右括号时, 检查栈是否为空
20                 if (char_zhan.empty()) {
21                     return false; // 栈为空, 表示没有匹配的左括号
22                 }
23                 // 获取栈顶元素
24                 char top_char = char_zhan.top();
25                 int m = hashtable[top_char];
26                 // 判断出栈条件
27                 if (temp + m == 0) {
28                     // 出栈
29                     char_zhan.pop();
30                 } else {
31                     return false; // 不匹配, 返回 false
32                 }
33             }
34             // 判断栈是不是空, 栈空则有效
35             return char_zhan.empty(); // 如果栈空返回 true; 否则返回
36         }
37         return false;
38     };
39 }

```