valid-parentheses

```
package algorithm.stack;

import java.util.Stack;

/**
 * https://leetcode.com/problems/valid-parentheses/
 *
 Given a string containing just the characters
 '(', ')', '{', '}', '[' and ']',
 determine if the input string is valid.

 The brackets must close in the correct order,
 "()" and "()[]{}" are all valid
 but "(]" and "([)]" are not.

 * @author xiaobaoqiu  Date: 16-7-6 Time:  下午11:15
 */
public class ValidParentheses {
    public static void main(String[] args) {
//        String s = "()[]{}";  //true
//        String s = "([)]";  //false
//        String s = "(";  //false
        String s = ")";  //false
        System.out.println(isValid(s));
    }

    /**
     * 遇到 ([{ 就进栈，遇到 )]} 并且栈顶是对应做括号就出栈，不对应就抛异常
     * 0 ms
     * Your runtime beats 96.72% of java submissions
     */
    public static boolean isValid(String s) {
        if (s == null || s.isEmpty()) return true;

        //simple stack
        int size = 0;
        char[] stack = new char[s.length()];

        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == '(' || s.charAt(i) == '[' || s.charAt(i) ==
            else {
                if (s.charAt(i) == ')') {
                    if (size > 0 && stack[size-1] == '(') --size;
                    else return false;
                } else if (s.charAt(i) == ']') {
                    if (size > 0 && stack[size-1] == '[') --size;
                    else return false;
                } else if (s.charAt(i) == '}') {
                    if (size > 0 && stack[size-1] == '{') --size;
                    else return false;
                }
            }
        }
        return size == 0;
```

```
        }
    }
```