题目

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:**

**Input:** "{[]}"

**Output:** true

**Input:** "([)]"

**Output:** false

题目：就是让左括号与右括号match，中间可以添加左括号，不能用不一样的右括号结尾

**public** **static** **boolean** isValid(String s) {

Deque<Character> a= **new** ArrayDeque<Character>();

HashMap<Character,Character> mymap=**new** HashMap();

mymap.put('}', '{');

mymap.put(']', '[');

mymap.put(')','(');

**for**(**char** c: s.toCharArray()) {

**if**(c==('(')||c==('{')||c==('[')) {

a.push(c);

}

**else** **if** (c==(')')||c==('}')||c==(']')) {

**if**(a.isEmpty()) {

**return** **false**;

}

**if**(a.peek()==(mymap.get(c))) {

a.pop();

}

**else**

**return** **false**;

}

}

**return** (a.isEmpty());

}

思路：

这就是很简单的一个栈问题

如果当前字符为左半边括号时，则将其压入栈中

如果遇到右半边括号时，分类讨论：

1）如栈不为空且为对应的左半边括号，则取出栈顶元素，继续循环

2）若此时栈为空，则直接返回 false

3）若不为对应的左半边括号，反之返回 false

知识点：

不要用stack,stack完全可以用 Deque代替

Deque<Character> a= **new** ArrayDeque<Character>();