Given a string, find the length of the **longest substring** without repeating characters.

**Example 1:**

**Input:** "abcabcbb"

**Output:** 3

**Explanation:** The answer is "abc", with the length of 3.

**Example 2:**

**Input:** "bbbbb"

**Output:** 1

**Explanation:** The answer is "b", with the length of 1.

**Example 3:**

**Input:** "pwwkew"

**Output:** 3

**Explanation:** The answer is "wke", with the length of 3.

Note that the answer must be a **substring**, "pwke" is a *subsequence* and not a substring.

这是我第一次自己做出Medium题目！

思路：

首先我们可以用Deque，他是双向stack，能用Deque，绝不用stack，因为他不仅能双向，还能contains

for loop循环charAt

如果没有contains这个charAt, push到末尾addLast，更新max

如果contains这个charAt, 把开头一直到这个charAt的全部pop完

class Solution {

public static int lengthOfLongestSubstring(String s) {

Deque<Character> myStack=new ArrayDeque<Character>(); //注意，一定要用arrayDeque

int max=0; //记录历史max

for(int i=0;i<s.length();i++) {

if (myStack.contains(s.charAt(i))){

while (myStack.getFirst()!=s.charAt(i)) {

myStack.removeFirst(); 一直remove

}

myStack.removeFirst(); remove等于的

myStack.addLast(s.charAt(i)); 后续加上

}

else {

myStack.addLast(s.charAt(i));

max=Math.max(max, myStack.size());

}

}

return max;

}

}