

Java-阿里巴巴找黄金宝箱(IV)-一贫如洗的樵夫阿里巴巴

题目描述:

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从  $0 \sim N$  的箱子，每个箱子上面贴有一个数字，箱子排列成一个环，编号最大的箱子的下一个是编号为  $0$  的箱子。

请输出每个箱子贴的数字之后的第一个比它大的数，如果不存在则输出  $-1$ 。

输入描述:

输入一个数字字符串，数字之间使用逗号分隔，例如:  $1,2,3,1$

字符串中数字个数  $\geq 1$ ， $\leq 10000$ ；每个数字值  $\geq -100000$ ， $\leq 100000$

输出描述:

下一个大的数列表，以逗号分隔，例如:  $2,3,6,-1,6$

补充说明:

示例 1

输入:

$2,5,2$

输出:

$5,-1,5$

说明:

第一个  $2$  的下一个更大的数是  $5$ ；

数字  $5$  找不到下一个更大的数；

第二个  $2$  的下一个最大的数需要循环搜索，结果也是  $5$

示例 2

输入:

$3,4,5,6,3$

输出:

$4,5,6,-1,4$

说明:

```
import java.util.Arrays;
import java.util.LinkedList;
import java.util.Scanner;
import java.util.StringJoiner;
```

// 注意类名必须为 Main，不要有任何 package xxx 信息

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```

        int[] array =
Arrays.stream(scanner.nextLine().split(",")).mapToInt(Integer::parseInt).toArray();
        System.out.println(getResult(array));
    }
    public static String getResult(int[] array){
        LinkedList<int[]> stack = new LinkedList<>();
        int[] newArray = new int[array.length];
        Arrays.fill(newArray,-1);
        findBigOne(array,stack,newArray);
        if (stack.size() != 1) {
            findBigOne(array,stack,newArray);
        }
        StringJoiner stringJoiner = new StringJoiner(",");
        for (int v : newArray){
            stringJoiner.add(v + "");
        }
        return stringJoiner.toString();
    }
    public static void findBigOne(int[] array,LinkedList<int[]> stack ,int[] newArray){
        for (int i = 0; i < array.length; i++) {
            int num = array[i];
            while (true){
                if (stack.size() == 0){
                    stack.add(new int[]{num,i});
                    break;
                }else {
                    int[] number = stack.get(stack.size()-1);
                    int number0 = number[0];
                    int number1 = number[1];
                    if (num > number0){
                        newArray[number1] = num;
                        stack.removeLast();
                    }else {
                        stack.add(new int[] {num,i});
                        break;
                    }
                }
            }
        }
    }
}

```