

组合出合法最小数

知识点 [数组排序](#) 

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

给一个数组，数组里面都是代表非负整数的字符串，将数组里所有的数值排列组合拼接起来组成一个数字，输出拼接成的最小的数字。

输入描述：

一个数组，数组不为空，数组里面都是代表非负整数的字符串，可以是0开头，例如：["13", "045", "09", "56"]。

数组的大小范围：[1, 50]

数组中每个元素的长度范围：[1, 30]

输出描述：

以字符串的格式输出一个数字，如果最终结果是多位数字，要优先选择输出不是“0”开头的最小数字；如果拼接出的数字都是“0”开头，则选取值最小的，并且把开头部分的“0”都去掉再输出；如果是单位数“0”，可以直接输出“0”

示例1

输入：

20 1

输出：

120

说明：

["20", "1"]能组成 201和120， 其中120比较小

示例2

输入：

08 10 2

输出：

10082

说明：

["08", "10", "2"]能组成 08102、 08210、 10082、 10208、 20810、 21008等数字， 其中"0"开头的08102和08201排除， 选择最小的10082输出

示例3

输入：

01 02

输出：

102

说明：

["01", "02"]能拼接成0102和0201， 都是"0"开头， 选取较小的0102， 去掉前面的0， 输出102

解题思路：

使用经典的全排列算法。

处理完之后需要注意是否以0开头。

```

public class Main{

    public static List<Integer> list = new ArrayList<>();    //非 0 开头的数字
    public static List<Integer> listZero = new ArrayList<>();    //0 开头的数字

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String[] strings = sc.nextLine().split(" ");

        quanpailie(strings, 0, strings.length);

        if(list.size() == 0){    //说明都是以 0 开头的
            Collections.sort(listZero);
            System.out.println(listZero.get(0));
        }else {
            Collections.sort(list);
            System.out.println(list.get(0));
        }
    }

    public static void swap(String[] strings,int a,int b){

        String temp = strings[a];
        strings[a] = strings[b];
        strings[b] = temp;

    }

    /**
     * 经典全排列回溯    （不理解的可以强行背诵，用的地方很多）
     * @param strings    需要排列的数组
     * @param index        处理的索引
     * @param end        数组长度
     */
    public static void quanpailie(String[] strings, int index, int end){

        if(index == end){    //所有的数据都遍历完成
            String res = "";
            for(int i=0;i <strings.length; i++){
                res += strings[i];
            }
        }
    }
}

```

```
        if(res.startsWith("0")){    //以 0 开头的需要分开放
            listZero.add(Integer.valueOf(res));
        }else {
            list.add(Integer.valueOf(res));
        }
    }else {
        for(int i=index; i<strings.length; i++){

            swap( strings, i, index);
            quanpailie( strings,index+1, end);
            swap( strings, i, index);

        }
    }
}
```