把数组排成最小的数-比较器Comparator-剑指Offer45

# java.lang.String的compareTo方法源码

String类继承了Comparable接口，实现了**compareTo**方法；一般的排序需要指定一个排序规则，一般需要定义一个比较器Comparator，如Arrays.sort可以传入Comparator。后面介绍如何创建Comparator。

public int **compareTo**(String anotherString) {

int len1 = value.length;

int len2 = anotherString.value.length;

int lim = Math.min(len1, len2);

char v1[] = value;

char v2[] = anotherString.value;

int k = 0;

while (k < lim) {

char c1 = v1[k];

char c2 = v2[k];

if (c1 != c2) {

return c1 - c2;

}

k++;

}

return **len1 - len2**;//若长的包括短的，则长的大于短的。

}

实例： abcd>abc;字符串231>23

# 剑指Offer45:把数组排成最小的数

## 题目描述

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*剑指Offer45:把数组排成最小的数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\* \* 输入一个正整数数组，把数组里所有数字拼接起来排成一个数，

\* \* 打印能拼接出的所有数字中最小的一个。例如输入数组{3，32，321}，

\* \* 则打印出这三个数字能排成的最小数字为321323。

## 思路分析

\* \* 思路：

\* \* 蛮力法：实现全排列，然后获取最下的一个；

\* \* 新方法：设定一个特定的排序规则，然后对数组进行排序;

\* \* 最后按照从小到大顺序拼接起来即可。

\* \* 这里最关键的是：创建这个排序规则，即new一个Comparator。

\* \* 排序规则与String的比较规则不一致，String中231>23,而这里需要231<23;

\*具体规则：长字符串要与和循环短字符串比较，直到找到不相等字符，返回大小关系。

\* 注意:考虑大数问题，超出int范围，采用字符串表示每一个数。

## Java代码

public String PrintMinNumber(int [] numbers) {

if(numbers == null||numbers.length == 0) return "";

String[] numStrs = new String[numbers.length];

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

numStrs[i] = String.valueOf(numbers[i]);

//利用Arrays给定的sort方法，传入比较器

Arrays.sort(numStrs, **new Comparator<String>() {//匿名对象;对String的compareTo方法进行了改进**

**@Override**

**public int compare(String str1, String str2) {**

**// return str1.compareTo(str2);//采用String的compareTo方法不行**

**//String的比较规则是32>3;二者要求32<3**

**char[] chars1 = str1.toCharArray();**

**char[] chars2 = str2.toCharArray();**

**int len1 = chars1.length, len2 = chars2.length;**

**int maxLen = len1 > len2 ? len1 : len2;**

**int k = 0;**

**while (k < maxLen) {**

**char c1 = chars1[k % len1];**

**char c2 = chars2[k % len2];**

**if (c1 != c2) {**

**return c1 - c2;**

**}**

**k++;**

**}**

**//对于321和32有：321<32;**

**//而对于324和32有：324>32**

**return 0;//短的应该大**

**}**

**}**);

System.out.println("Arrays.toString(numStrs) = " + Arrays.toString(numStrs));

StringBuilder sb = new StringBuilder();

for(String str:numStrs) sb.append(str);

return sb.toString();

}

//简化比较器代码，在String的compareTo方法基础上实现（已经验证）

Comparator<String> comparator = new Comparator<String>() {//匿名对象

@Override

public int compare(String str1, String str2) {

String str3 = str1+str2;

String str4 = str2+str1;

return **str3.compareTo(str4);//短的应该大**

}

};