

Java库谜题65：一种疑似排序的惊人传奇 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_Java_E5_BA_93_E8_B0_9C_c97_138628.htm 下面的程序使用定制的比较器，对一个由随机挑选的Integer实例组成的数组进行排序，然后打印了一个描述了数组顺序的单词。回忆一下

，Comparator接口只有一个方法，即compare，它在第一个参数小于第二个参数时返回一个负数，在两个参数相等时返回0，在第一个参数大于第二个参数时返回一个整数。这个程序是展示5.0版特性的一个样例程序。它使用了自动包装和解包、泛型和枚举类型。那么，它会打印出什么呢？

```
import java.util.*;
public class SuspiciousSort {
    public static void main(String[] args) {
        Random rnd = new Random();
        Integer[] arr = new Integer[100];
        for (int i = 0; i < arr.length; i++)
            arr[i] = rnd.nextInt();
        Comparator cmp = new Comparator() {
            public int compare(Integer i1, Integer i2) {
                return i2 - i1;
            }
        };
        Arrays.sort(arr, cmp);
        System.out.println(order(arr));
    }
    enum Order {
        ASCENDING,
        DESCENDING,
        CONSTANT,
        UNORDERED
    };
    static Order order(Integer[] a) {
        boolean ascending = false;
        boolean descending = false;
        for (int i = 1; i < a.length; i++)
            ascending |= a[i] > a[i-1];
            descending |= a[i] < a[i-1];
        if (ascending && !descending)
            return Order.ASCENDING;
        if (descending && !ascending)
            return Order.DECENDING;
        if (!ascending && !descending)
            return Order.CONSTANT;
        // All elements equal
        return Order.UNORDERED;
        // Array is not sorted
    }
}
```

该程序的main方法创建了一个Integer实例的数组，并用随机数对其进行了初始化，然后用比较器cmp对该数组进行排序。这个

比较器的compare方法将返回它的第二个参数减去第一个参数的值，如果第二个参数表示的是比第一个参数大的数值，其返回值就是正的；如果这两个参数相等，其返回值为0；如果第二个参数表示的是比第一个参数小的数值，其返回值就是负的。这种行为正好与compare方法通常的做法相反，因此，该比较器应该施加的是降序排列。在对数组排序之后，main方法将该数组传递给了静态方法order，然后打印由这个方法返回的结果。该方法在数组中所有的元素都表示相同的数值时，返回CONSTANT；在数组中每一对毗邻的元素中第二个元素都大于等于第一个元素时，返回ASCENDING；在数组中每一对毗邻的元素中第二个元素都小于等于第一个元素时，返回DESCENDING；在这些条件都不满足时，返回UNORDERED。尽管理论上说，数组中的100个随机数有可能彼此都相等，但是这种奇特现象发生的非常小： $2^{32} \times 99$ 分之一，即大约 5×10^{953} 分之一。因此，该程序看起来应该打印DESCENDING。如果你运行该程序，几乎可以肯定你将看到它打印的是UNORDERED。为什么它会产生如此的行为呢？order方法很直观，它并不会说谎。Arrays.sort方法已经存在许多年了，它工作得非常好。现在只有一个地方能够发现bug了：比较器。乍一看，这个比较器似乎不可能出错。毕竟，它使用的是标准的惯用法：如果你有两个数字，你想得到一个数值，其符号表示它们的顺序，那么你可以计算它们的差。这个惯用法至少从1970年代早期就一直存在了，它在早期的UNIX里面被广泛地应用。遗憾的是，这种惯用法从来都没有正确地工作过。本谜题也许应该称为“白痴一般的惯用法的案例”。这种惯用法的问题在于定长的整数没有大到可以

保存任意两个同等长度的整数之差的程度。当你在做两个int或long数值的减法时，其结果可能会溢出，在这种情况下我们就会得到错误的符号。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com