

公告: CSDN迁址后第一次网友沙龙活动圆满结束!

用户操作

[留言] [发消息] [加好友]

订阅我的博客



mgoann的公告

文章分类

- DB2
- java
- Linux
- shell脚本
- 生活见闻
- 协议

存档

- 2010年06月(12)
- 2010年02月(15)
- 2010年01月(3)
- 2009年06月(2)
- 2009年05月(6)
- 2009年04月(22)
- 2008年11月(3)
- 2008年10月(5)

关于HashMap排序的一点浅见 收藏

## 关于HashMap排序的一点浅见

我们在对hashmap进行排序的时候存在一定的误解，首先在这更正下~

我在遇到这个问题的时候也上网查了相关的资料，发现大家在进行hashmap进行排序的时候存在一定的了解误区。

有很多人认为把hashmap遍历一下，然后对其中的key和value进行排序，然后打印出来~

大家想一想，这样的hashmap排序根本就不是对hashmap进行排序~最多也只能称为hashtable遍历，hashmap的遍历问题很简单，网上也有很多相关的资料，下面我给出一个例程：

```
1. public static void itHashMap(Map<Object, Object> m) {
2.     Set set = m.entrySet();
3.     Iterator it = set.iterator();
4.     while(it.hasNext()) {
5.         Map.Entry<Object, Object> me = (Map.Entry<Object, Object>)it.next();
6.         System.out.println("key:" + me.getKey() + "----value" + me.getValue());
7.     }
8. }
```

这个方法可以将hashmap遍历，并打印出相应的key和value！

其实，单纯的hashmap是无法实现排序的，我这里的排序是指，我们将键值对按照一定的顺序put进hashmap里，然后在进行取键值对的操作的时候，在按照put进去的顺序把键值对取出来。

下面我来浅显的解释下为什么不能排序。

```
1. HashMap<String, Date> hm = new HashMap<String, Date> ();
2.     hm.put("1", new Date(8,10,1));
3.     hm.put("3", new Date(8,10,3));
4.     hm.put("2", new Date(8,10,2));
5.     hm.put("5", new Date(8,10,5));
6.     hm.put("9", new Date(8,10,9));
7.     hm.put("8", new Date(8,10,8));
8.     hm.put("6", new Date(8,10,6));
9.     hm.put("7", new Date(8,10,7));
```

我在**hashmap**里按照以上的顺序放入**9**个键值对，然后我遍历**hashmap**打印出上面**hm**里的键值对：

```
key: 3----value: Tue Nov 03 00:00:00 CST 1908
key: 5----value: Thu Nov 05 00:00:00 CST 1908
key: 7----value: Sat Nov 07 00:00:00 CST 1908
key: 2----value: Mon Nov 02 00:00:00 CST 1908
key: 9----value: Mon Nov 09 00:00:00 CST 1908
key: 8----value: Sun Nov 08 00:00:00 CST 1908
key: 6----value: Fri Nov 06 00:00:00 CST 1908
key: 1----value: Sun Nov 01 00:00:00 CST 1908
```

我们从上面的结果可以看出，取出来的顺序并非我们想要的顺序！

**hashmap**在**put**的时候是根据**key**的**hashCode**进行**hash**然后放入对应的地方！通过**Debug**可以看到**HashMap**的存放位置，网上有很多人说是随即存放，随即取，其实不然，放的时候是根据**hashCode**经过**hash**算法进行存放的，而我们取的时候也是想过**key**来取**value**。所以你取出来拍完序，在放入**HashMap**的时候哦，它又把顺序打乱了~~（这里说的打乱也不是完全没有规律可循的，主要是你根据你的**key**来放的）！

所以我们要实现**HashMap**排序的时候就要借用别的集合来完成这个功能！

基本的思路是这样的，首先遍历**HashMap**取出相应的键值对，然后根据**key**或者是**value**进行排序，然后放入一个有序的集合当中，这样就实现了排序！

**java**在**JDK1.4**以后提供了**LinkedHashMap**来帮我们实现了有序的**HashMap**！

**LinkedHashMap**取键值对的时，是按照你放入的顺序来取的！

具体的排序算法有很多种~也可以自己写个冒泡排序取排列~~

再排完序以后，我们要找到**key**对应的**value**或是**value**对应的**key**！找**key**对应的**value**的时候很简单，网上有很多资料，我这里就不说了~~关键说说找**value**对应的**key**的时候，有的朋友就不知道该怎么做了~~

下面我提供两种思路

一种是我们可以把**value**放入一个数组，**key**也放入一个数组，这样在排序的时候按照**value**排序的时候，我们把**key**的位置也交换一次，这样就保证了键值对的完整性，见例程：

```
1. private static void sortHashMap(HashMap<String, Date> hm) {
2.
3.     List<Date> dateListSorted = new ArrayList<Date> ();
4.
5.     int size = hm.size();
6.     Date[] dates = new Date[size];
7.     Set set = hm.keySet();
8.     Iterator iteratorSet = set.iterator();
9.     int k = 0;
10.    while(iteratorSet.hasNext()) {
11.        String keyStr = (String)iteratorSet.next();
12.        System.out.println("key from set----" + keyStr);
13.        if(keyStr!=null && !(keyStr.length()<1)) {
14.            dates[k] = hm.get(keyStr);
15.            k++;
16.        }
17.        System.out.println("value from hashmap----" + hm.get(keyStr));
18.    }
19.    iteratorList.next();
20.    int sizeList = dates.length;
21.    Object[] keyObj = set.toArray();
22.    for(int i=0;i<sizeList;i++) { //冒泡排序
23.        for(int j=i;j<sizeList;j++) {
24.            if(dates[j].after(dates[i])) {
```

```

25.         Date temp = dates[i]; //交换value
26.         dates[i] = dates[j];
27.         dates[j] = temp;
28.         Object objectTemp = keyObj[i]; //交换key
29.         keyObj[i] = keyObj[j];
30.         keyObj[j] = objectTemp;
31.     }
32.
33.     }
34. }
35. for(int i=0;i<keyObj.length;i++) {
36.     System.out.println("key from array sorted----" + keyObj[i]);
37. }
38. for(int i=0;i<dates.length;i++) {
39.     System.out.println("value from list sorted----" + dates[i]);
40. }
41. LinkedHashMap lhm = new LinkedHashMap();
42. for(int i=0;i<keyObj.length;i++) { //把经过排序的键值对放入Linked中
43.     lhm.put(keyObj[i], dates[i]);
44. }
45. /*
46.  *遍历Linkedhashmap打印出结果
47.  */
48.     Set set1 = lhm.entrySet();
49.     Iterator it = set1.iterator();
50.     while(it.hasNext()) {
51.         Map.Entry me = (Map.Entry)it.next();
52.         System.out.println("key:" + me.getKey() + "----value" + me.getV
           alue());
53.     }
54. }

```

这种方法比较直观，可是效率比较低，而且对内存也很浪费，你想想，在这个过程中我们new了两个长度是lhm.size () 长度的数组，用于存放key和value，排序也是我们自己写的冒泡排序法~~

下面我介绍第二种方法：

第二种方法的思路是这样的，我们用`hm.values()`放回`hm`的`values`集合，再把它转换成一个数组，然后调用`jdk`给我们提供的`Arrays.sort()`方法去帮我们排序，这样效率会比自己写的冒泡排序法效率高得多！

在调用`sort`方法的时候我们需要写一个比较器，比较器的代码如下：

```
1. package com.mgoann.test1;
2.
3. import java.util.Comparator;
4. import java.util.Date;
5.
6. public final class DateComp implements Comparator {
7.
8.     public int compare(Object o1, Object o2) {
9.         Date date1 = (Date)o1;
10.        Date date2 = (Date)o2;
11.        return date2.compareTo(date1);
12.    }
13.
14. }
15.
```

剩下的就是更具`value`去找相应的`key`的问题了~~这里我用了一个`for`循环和`Iterator`去查找`hm`里的`key`，找到以后我们就放到`LinkedHashMap`里~，详见例程：

```
1. package com.mgoann.test1;
2.
3. import java.util.Arrays;
4. import java.util.Collection;
5. import java.util.Comparator;
6. import java.util.Date;
7. import java.util.HashMap;
8. import java.util.Iterator;
9. import java.util.LinkedHashMap;
10. import java.util.Map;
11. import java.util.Set;
12.
13. public class Sort {
```

```

14.
15.     /**
16.      * @param args
17.      */
18.     public static void main(String[] args) {
19.         HashMap<String, Date> hm = new HashMap<String, Date>();
20.         LinkedHashMap<String, Date> linkedHM = new LinkedHashMap<String, Date>();
21.         hm.put("1", new Date(8,10,1));
22.         hm.put("3", new Date(8,10,3));
23.         hm.put("2", new Date(8,10,2));
24.         hm.put("5", new Date(8,10,5));
25.         hm.put("9", new Date(8,10,9));
26.         hm.put("8", new Date(8,10,8));
27.         hm.put("6", new Date(8,10,6));
28.         hm.put("7", new Date(8,10,7));
29.         itHashMap(hm);
30.         linkedHM = sortHashMap(hm);
31.         itHashMap(linkedHM);
32.     }
33.
34.     private static LinkedHashMap sortHashMap(HashMap<String, Date> hm) {
35.
36.         LinkedHashMap<String, Date> linkedHM = new LinkedHashMap<String, Date>();
37.         final Comparator DATE_COMP = new DateComp();
38.
39.         int index = 0;
40.
41.         Collection<Date> DateColl = hm.values();
42.         Object[] dates = DateColl.toArray();
43.         Arrays.sort(dates, DATE_COMP);
44.         System.out.println("dates after sorted");
45.         printElements(dates);
46.         Set set = hm.entrySet();
47.         for(int j=0;j<dates.length;j++) {
48.             Iterator it = set.iterator();
49.             while(it.hasNext()) {

```

```

50.         Map.Entry<String, Date> me = (Map.Entry<String, Date>
           )it.next();
51.         if(dates[j].equals(me.getValue())) {
52.             linkedHM.put(me.getKey(), me.getValue());
53.         }
54.     }
55. }
56.     return linkedHM;
57. }
58.
59. public static void printElements(Collection c) {
60.     Iterator it = c.iterator();
61.     while(it.hasNext()) {
62.         System.out.println(it.next());
63.     }
64. }
65.
66. public static void printElements(Object[] dates) {
67.     for(int i=0;i<dates.length;i++) {
68.         System.out.println(dates[i]);
69.     }
70. }
71.
72. public static void itHashMap(Map<String, Date> m) {
73.     Set set = m.entrySet();
74.     Iterator it = set.iterator();
75.     while(it.hasNext()) {
76.         Map.Entry<String, Date> me = (Map.Entry<String, Date>)it.
           next();
77.         System.out.println("key:" + me.getKey() + "----value:" + m
           e.getValue());
78.     }
79. }
80. }

```

这样我们就实现了**HashMap**的排序问题!

程序运行结果如下:

**key:3----value:Tue Nov 03 00:00:00 CST 1908**

key:5----value:Thu Nov 05 00:00:00 CST 1908  
key:7----value:Sat Nov 07 00:00:00 CST 1908  
key:2----value:Mon Nov 02 00:00:00 CST 1908  
key:9----value:Mon Nov 09 00:00:00 CST 1908  
key:8----value:Sun Nov 08 00:00:00 CST 1908  
key:6----value:Fri Nov 06 00:00:00 CST 1908  
key:1----value:Sun Nov 01 00:00:00 CST 1908  
dates after sorted  
Mon Nov 09 00:00:00 CST 1908  
Sun Nov 08 00:00:00 CST 1908  
Sat Nov 07 00:00:00 CST 1908  
Fri Nov 06 00:00:00 CST 1908  
Thu Nov 05 00:00:00 CST 1908  
Tue Nov 03 00:00:00 CST 1908  
Mon Nov 02 00:00:00 CST 1908  
Sun Nov 01 00:00:00 CST 1908  
key:9----value:Mon Nov 09 00:00:00 CST 1908  
key:8----value:Sun Nov 08 00:00:00 CST 1908  
key:7----value:Sat Nov 07 00:00:00 CST 1908  
key:6----value:Fri Nov 06 00:00:00 CST 1908  
key:5----value:Thu Nov 05 00:00:00 CST 1908  
key:3----value:Tue Nov 03 00:00:00 CST 1908  
key:2----value:Mon Nov 02 00:00:00 CST 1908  
key:1----value:Sun Nov 01 00:00:00 CST 1908

发表于 @ 2008年10月16日 13:47:00 | [评论\(8\)](#) | [举报](#) | [收藏](#)

[新一篇: 南京13路](#)

连续三年举办，备受好评的技术盛宴

**第4届软件开发2.0大会**

首次移师上海

仅有**15天**

**6折特惠价**

11.1-11.15

立即抢票▶

咨询电话：**010-64351**

时间：12月9-10日 地点：上海

[查看最新精华文章](#) [请访问博客首页相关文章](#)

---

[justinavril](#) **EX** □□□Monday, October 20, 2008 11:04:52 [□□](#) [□□](#)



□□ □□□□...

---

[justinavril](#) **EX** □□□Monday, October 20, 2008 11:07:45 [□□](#) [□□](#)

□□ □□□□...





[Bmouth](#) □□□Monday, August 17, 2009 09:02:29 [□□](#) [□□](#)



□□□□□,□□value□□□□□key□□□□□□□□key□□□□□Value□□□□□□□□□  
□□□!? □□□□□□□Map.Entry<String, Date>□□□□□□□□□□Array.sort(Map.  
Entry<String, Date>[])□□□□□□□

[feiazifeiazi](#) □□□Sunday, January 03, 2010 01:43:36 [□□](#) [□□](#)



□□□□□

[flyingg0512](#) □□□Tuesday, January 05, 2010 10:03:57 [□□](#) [□□](#)



□□□□□□□

[□□□□](#) □□□Friday, April 30, 2010 15:13:48 [□□](#) [□□](#)



LINKEDHASHMAP □□ □□□□□□3Q

[lionheihei2](#) □□□Sunday, August 29, 2010 14:25:49 [□□](#) [□□](#)



□□□

[snowrong](#) □□□Monday, September 27, 2010 16:03:48 [□□](#) [□□](#)



发表评论

表情:



评论内容:

用户名: 匿名用户

[登录](#) [注册](#)

热门招聘职位

师, 软件开发工程师, 欲试从速!

【Autodesk】欧特克软件(中国)  
)诚聘软件开发,测试,研究员

【东星软件(上海)有限公司】

诚聘Flash程序员、3D游戏引擎

【深圳好伴电子商务】高薪诚聘  
：PHP、网络前端工程师、网页

【Google】诚招技术精英，史上  
人数最多职位最广！

【敦煌网】诚聘研发类职位：Ja  
va、PHP、网站架构师、运维工

【柯达 (Kodak)】诚聘C++(

---

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#)

北京创新乐知广告有限公司 版权所有，京 ICP 证 070598 号

世纪乐知(北京)网络技术有限公司 提供技术支持

江苏乐知网络技术有限公司 提供商务支持



Email:webmaster@csdn.net

Copyright © 1999-2010, CSDN.NET, All Rights Reserved

