

构造函数

中文注释的源码如下:

```
* 存储索引集合.
   private int[] mKeys;
5
    * 存储对象集合.
   private Object[] mValues;
10
11
    * 存储的键值对总数.
13
   private int mSize;
14
15
16
    * 采用默认的构造函数,则初始容量为10.
```

告别知识焦虑.

免费云主机试用一年 短信验证码接口

```
this(10);
20
21 }
22
23
      使用指定的初始容量构造SparseArray.
24
25
    * @param initialCapacity 初始容量
26
    */
27
   public SparseArray(int initialCapacity) {
29
       if (initialCapacity == 0) {
           // Effective Java中第43条:返回零长度的数组或者集合,而不是:null
30
31
           mKeys = ContainerHelpers.EMPTY_INTS;
           mValues = ContainerHelpers.EMPTY_OBJECTS;
32
33
       } else {
           // 构造initialCapacity大小的int数组和object数组
34
35
           mKeys = new int[initialCapacity];
36
           mValues = new Object[initialCapacity];
37
       }
38
       // 设置SparseArray存储的<key, value>键值对个数为0.
39
40
       mSize = 0;
41 }
```

和HashMap的数据结构不同,HashMap是使用数组+链表的数据结构存储键值对,而SparseArray只是用了两个数组进行存储.

ContainerHelpers类

之所以SparseArray在存储key为整形的键值对的效率高于HashMap,很大一部分原因是整形key的查找过程中ContainerHelpers类提供了二分查找算法,从而降低了时间复杂度.接下来,我们分析一下ContainerHelpers类的二分查找算法实现:

AI黑科技

告别知识焦虑,即刻启程

免费云主机试用一年
android binder 短信验证码接口 下载 Binder

```
13
14
                if (midVal < value) {</pre>
15
                    lo = mid + 1;
                } else if (midVal > value) {
16
                    hi = mid - 1;
17
18
                } else {
                    return mid; // value found
19
20
                }
21
            }
22
            return ~lo; // value not present
23
       }
24 }
```

这个二分算法的精髓在于:当二分查找没有找到相应元素时,返回的是lo值取反.

因为正数取反为负数,binarySearch返回正数代表查找成功,返回负数代表查找失败.

同时,binarySearch查找失败后,lo代表了当前元素按照升序排序应该插入的下标,后续取反后可以直接获取插入位置.

put()函数

put函数的中文注释源码如下:

```
1 /**
    * 在SparseArray中存储键值对.
 3
   public void put(int key, E value) {
 5
       // 通过二分查找算法计算索引
6
       int i = ContainerHelpers.binarySearch(mKeys, mSize, key);
8
       if (i >= 0) {
           // key已经存在对应的value,则直接替换value.
9
10
           mValues[i] = value;
11
       } else {
12
           i = \sim i;
13
14
           if (i < mSize && mValues[i] == DELETED) {</pre>
               // 特殊的case,直接存储key-value即可
15
               mKeys[i] = key;
16
17
               mValues[i] = value;
18
               return;
```

AI黑科技

告别知识焦虑,即刻启程

免费云主机试用一年 短信验证码接口 下载

注册 登录

X

```
if (mGarbage && mSize >= mKeys.length) {
21
22
               // 如果有元素被删除,并且目前容量不足,先进行一次gc
23
               gc();
24
25
               // Search again because indices may have changed.
26
               i = ~ContainerHelpers.binarySearch(mKeys, mSize, key);
           }
27
28
29
           // 扩容
30
           if (mSize >= mKeys.length) {
               // 获取扩容的数组大小
31
32
               int n = mSize + 1;
33
34
               int[] nkeys = new int[n];
               Object[] nvalues = new Object[n];
35
36
37
               // 数组拷贝最好使用System.arraycopy,而不是自己重撸一遍
38
               System.arraycopy(mKeys, 0, nkeys, 0, mKeys.length);
               System.arraycopy(mValues, 0, nvalues, 0, mValues.length);
39
40
41
               mKeys = nkeys;
               mValues = nvalues;
42
43
           }
44
45
           // i为插入位置,如果i<mSize,则i之后的元素需要依次向后移动一位.
           if (mSize - i != 0) {
46
47
               System.arraycopy(mKeys, i, mKeys, i + 1, mSize - i);
48
               System.arraycopy(mValues, i, mValues, i + 1, mSize - i);
           }
49
50
           // 设置值,存储数量+1
51
52
           mKeys[i] = key;
53
           mValues[i] = value;
54
           mSize++;
       }
55
56 }
```

通过源码,我们来总结一下put函数的步骤:

- 1. 通过二分查找算法,计算key的索引值.
- 2. 如果索引值大于0,说明有key对应的value存在,直接替换value即可.

AI黑科技

告别知识焦虑,即刻启程

免费云主机试用一年

短信验证码接口

下载 Binder 搜索

登录

注册

X

get()函数

get函数就是利用二分查找获取key的下标,然后从object[] value数组中根据下标获取值. 之所以SparseArray号称比HashMap有更好的性能:

- 1. SparseArray更加节约内存,一个int[]数组存储所有的key,一个object[] 数组存储所有的value.
- 2. HashMap遇到冲突时,时间复杂度为O(n).而SparseArray不会有冲突,采用二分搜索算法,时间复杂度为O(lgn).

中文注释源码:

```
1 /**
    * 根据指定的key获取value.
   public E get(int key) {
       return get(key, null);
5
6 }
7
    * 利用二分查找算法根据key获取指定的value.
    */
10
   public E get(int key, E valueIfKeyNotFound) {
12
       int i = ContainerHelpers.binarySearch(mKeys, mSize, key);
13
       if (i < 0 || mValues[i] == DELETED) {</pre>
14
15
           return valueIfKeyNotFound;
16
       } else {
17
           return (E) mValues[i];
18
       }
19 }
```

delete()函数

SparseArray中,remove函数最终也是调用delete函数进行(K,V)删除的.

而delete操作只是根据二分算法查找出key对应的下标,然后将object[] value中的对应下标值设置为DELETED.

源码如下:

1 /**

AI黑科技

告别知识焦虑,即刻启程

免费云主机试用一年

短信验证码接口 下载 Binder

登录

注册

 \times

```
public void delete(int key) {
5
       int i = ContainerHelpers.binarySearch(mKeys, mSize, key);
 6
7
       if (i >= 0) {
8
            if (mValues[i] != DELETED) {
9
               // 标记i的值为private static final Object DELETED = new Object();
               mValues[i] = DELETED;
10
               // 设置gc标记为true.
11
12
               mGarbage = true;
13
           }
14
       }
15
16
17
    * Alias for {@link #delete(int)}.
18
    */
19
   public void remove(int key) {
21
        delete(key);
22 }
```

gc()函数

delete()函数中将被删除的key对应的value设置为DELETED,并设置gc标志mGarbage为ture,那SparseArray是神马时候执行gc的呢?如果仔细看了上面的文章,应该还是会有印象.SparseArray是在put函数的时候执行了gc:

```
1 if (mGarbage && mSize >= mKeys.length) {
   2
         // 如果有元素被删除,并且目前容量不足,先进行一次gc
   3
         gc();
   4
   5
         // Search again because indices may have changed.
   6
         i = ~ContainerHelpers.binarySearch(mKeys, mSize, key);
   7 }
gc的实现如下:
   1 private void gc() {
   2
         int n = mSize;
   3
         int o = 0;
   4
         int[] keys = mKeys;
                                   免费云主机试用一年
AI黑科技
                                                      短信验证码接口
                                                                 下载
```

 \times

注册

登录

```
Object val = values[i];
8
9
10
            if (val != DELETED) {
                if (i != o) {
11
12
                    keys[o] = keys[i];
13
                    values[o] = val;
                    values[i] = null;
14
15
                }
16
17
                0++;
18
            }
        }
19
20
21
        mGarbage = false;
22
        mSize = o;
23 }
```

gc函数的原理:遍历一遍数组,将非DELETED资源全部移动到数组前面.

小结

SparseArray vs HashMap:

- 1. 首先,这是两种完全不同的数据结构.SparseArray是两个数组:int[]和Object[], HashMap是数组+链表.
- 2. 查找效率上: 首先,SparseArray不需要对key进行hash运算,并且通过二分查找保证查询效率为O(lgn).而HashMap在未冲突的情况下是O(1),冲突的情况下是O(n).

想对作者说点什么?

下载

Binder

我来说一句

Android性能优化之谈谈SparseArray,SparseBooleanArray和SparseIntArray

② 2223

Android性能优化之谈谈SparseArray,SparseBooleanArray和SparseIntArray 标签: 性能优化android开发 2015-04-14 00:24 ...

SparseArray到底哪点比HashMap好

⊚ 6579

SparseArray是android里为这样的Hashmap而专门写的class,目的是提高效率,其核心是折半查找函数(binarySearch)。 HashMap底层是一个Hash表,...

sparseArray 遍历

◎ 1万

1. 看来我找到了解决办法。我还没有正确的人注意kevAt(index)功能。 所以我会用这样下去 int kev = 0: for(int i = 0: i 2. 如果你不关心键。然后val

免费云主机试用一年 android binder

短信验证码接口

搜索

登录

注册

 \times

SparseArray浅析

🔐 ◎ 2万

HashMap是java里比较常用的一个集合类,我们一般用来缓存一些处理后的结果。但当你做一个Android项目时,在代码中定义这样一个变量,实例化时...

Android编程之SparseArray<E>详解

⑩ ◎ 9万

最近编程时,发现一个针对HashMap的一个提示: 翻译过来就是:用SparseArray来代替会有更好性能。 那我们就来看看源码中SparseArray到底做了哪…

Android为什么推荐使用SparseArray来替代HashMap?

● 2383

SparseArray也许你没听过,那正好今天就来学习一下咯,这也是Android官方推荐使用的,所以我们需要了解一下他的优势和劣势在哪些地方。 首先Spar...

intellij老是警告的SparseArray是什么 - HashMap的替代者

₫ • 1156

序言身为一个有代码洁癖的程序员,在写Android应用的时候,我总是会去注意 代码规范(Google Android Guideline)能一行搞定的代码,绝不写两行 ...

Android SparseArray源码详解

© 1546

尊重原创,转载请标明出处 http://blog.csdn.net/abcdef314159 在Android开发中如果使用key为Integer的HashMap,就会出现黄色警告,提示使用Sp...

虐哭java面试官--聊一聊hashmap

⊚ 3672

Java hashmap的数据结构,开发的时候从来用不到那么深,MD,每个面试官都要问一遍。 别人恶心我的时候,我要比他更恶心才行。 放心,技术一般…

深入理解HashMap (精华必看)

🥙 🧿 1.9万

原文地址:http://www.iteye.com/topic/539465 Hashmap是一种非常常用的、应用广泛的数据类型,最近研究到相关的内容,就正好复习一下。网上...

文 章 执 词 android cpu 个数 android 子进程 进程 android调用系统复制 android 引导页设置 android 精简应用

博主推荐

♡ 换一批



hfreeman2008

Danny_拿



七个核树

12篇文章

SparseArray不适合数据量大的情况的原因

№ © 556

SparseArray不适合数据量较大的根本原因。

SparseArray的常用方法解析

⊚ 112

告别知识焦虑,即刻启程

免费云主机试用一年 android binder

id binder

短信验证码接口

下载 Binder

搜索

登录

注册

SparseArray与HashMap相比优缺点

120

SparseArray(稀疏数组).他是Android内部特有的api.标准的jdk是没有这个类的.在Android内部用来替代HashMap<Integer,E&gt;这种...

Android内存优化之SparseArray源码解析

⊚ 22

Android内存优化之SparseArray源码解析 一.概述 当遇到性能问题时,可以使用SparseArray来代替HashMap进行一些操作。 在源代码中(API26)中有如...

SparseArray和ArrayMap相关源码解析

⊚ 144

内存优化:Android API当中提供了一些优化过后的数据集合工具类,如SparseArray,SparseBooleanArray,以及LongSparseArray等,使用这些API可以让...

SparseArray的使用及实现原理

◎ 1209

转载自: http://extremej.itscoder.com/sparsearray_source_analyse/ 序言 相信大家都用过HashMap用来存放键值对,最近在项目中使用...

Android SparseArray 分析

1130

我们平时在Android平台上开发应用的时候经常会使用Java中的api去处理一些东西,但是由于Android手机的内存,cpu的处理能力等等原因,java上的api...

SpareArray 原理 及其 和HashMap的性能比较

401

SpareseArray 也是通过键值对存储数据,只是key为整形int,类似于key = Interger 的HashMap, 但是SpareseArray 的key 为 int 非 Inter...

SparseArray与SparseArrayCompat区别

⊚ 858

了解SparseArrayCompat需要先了解SparseArray SparseArrayCompat 是什么?目前网上介绍这个类的很少。几乎也找不到SparseArray与SparseArra..

SparseArray 源码解析及hashMap性能对比

SparseArray数据结构: private int[] mKeys; // int 数组 保存 key private Object[] mValues; // Object 数组 存va...

android开发笔记之通过辅助类解决findViewByld需要对返回值强制类型转换的问题

⊚ 1104

1.android中findViewByld需要对返回值强制类型转换的问题描述 findViewByld的返回值是view类型,通常开发中,我们需要将其强制转换成实际类型…

分别讲解gson和fastjson解析json格式

1470

gson是谷歌开发支持的,谷歌GSON这个Java类库可以把Java对象转换成JSON,也可以把JSON字符串转换成一个相等的Java对象。*服务器端将数据转换...

安卓性能优化-使用ArrayMap与SparseArray

1503

性能优化是我们做开发的必须要熟练掌握的技能,所以我打算写一个性能优化专题,把平时用到的一些优化方法记录下来,以便忘记的时候可以快速查找...

告别知识焦虑,即刻启程

免费云主机试用一年

短信验证码接口

下载 Binder 搜索

登录

注册

 \times

SparseArray介绍 26 结构方面: 它内部则是通过两个数组来进行数据存储的,一个存储key,另外一个存储value,为了优化性能,它内部对数据还采取了压缩的方式来表示稀... 数据结构HashMap (Android SparseArray 和ArrayMap) 3921 HashMap也是我们使用非常多的Collection,它是基于哈希表的 Map 接口的实现,以key-value的形式存在。在HashMap中,key-value总是会当做一个整... 安卓SparseArray的indexOfValue方法遇到的坑 984 在数据量比较少的情况下使用SparseArray可以提高效率,最近一个项目利用它存储int型数据,利用indexOfValue去获取value的索引,本以为value大小范... Android 中的SparseArray——来源于源代码(翻译) SparseArray 结构: public classSparseArray android.util 类概述: Spars... 在activity中通过Parcelable传递数据 20 private String a = ""; private String b = "";新建一个类 MyClass。实现接口 Parcelable。实现所有... 下载 源码解析alamofire 11-21 源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码解析alamofire源码。 alibaba fastJson框架快速解析复杂有重复性质的ison 287 fastJson 源自著名的阿里巴巴开源项目,是一个性能很好的 Java 语言实现的 JSON 解析器和生成器。 主要特点: 快速FAST (比其它任何基于Java的解析... 下载 STRUTS2源码解析 09-15 STRUTS2源码解析STRUTS2和MTP 面试必备: SparseArray源码解析 2417 想看我更多文章:【张旭童的博客】http://blog.csdn.net/zxt0601 想来gayhub和我gaygayup:【mcxtzhang的Github主页】https://githu... SparseArray实现原理 SparseArray SparseArray原理: SparseArray采用两个数组,用来存放key以及value值的,核心思想是通过折半查找来找到key对应的位置,然后取出值,或... SparseArray用法 94 /** * SparseArrays 利用integer去管理object对象。不像一个正常的object对象数组,它能在索引数中快速的查找到所需的结果。(这 * 句话是音译,原意...

搜索

Binder

AI黑科技

告别知识焦虑,即刻启程

免费云主机试用一年

短信验证码接口

 \times

注册

登录

android sparseArray源码解析

android sparseArray源码解析step1:背景今天偶然看到了sparseArray,发现其主要是针对的类型进行了优化,何为sparse?是稀疏的意思。指的是对稀疏...

下载 dubbo源码解析2

05-14

dubbo源码解析2.dubbon原码解析2.dubbonnededubbo

SparseArray源码中的G点

通过上一章 小探Android中的SparseArray和HashMap了解了SparseArray和HashMap的一点区别,以及SparseArray的使用场景。这一章主要来记录一下查...

下载 PHPWIND 解析

09-14

⊚ 542

下载 spring实战

03-30

spring实战,源码解析。spring实战,源码解析。spring实战,源码解析。spring实战,源码解析。spring实战,源码解析。

下载 Spring源码深度解析

09-12

Spring源码深度解析 Spring源码深度解析 Spring源码深度解析 Spring源码深度解析 Spring源码深度解析 Spring源码深度解析

没有更多推荐了,返回首页



告别知识焦虑,即刻启程

免费云主机试用一年

短信验证码接口 下载 搜索

登录

注册

X