

luoxn28

一个纯洁的Java程序员，兼容C/C++。关注架构、互联网技术、分布式、金融、历史等。
https://github.com/luoxn28

博客园 首页 管理

随笔- 70 文章- 0 评论- 33

关于我

一个纯洁的Java程序员，兼容C/C++。
关注架构、互联网技术、分布式、金融、历史等。
[我的GitHub](#) [我的网站](#)

昵称：luoxn28
园龄：2年8个月
粉丝：132
关注：22
[+加关注](#)

<	2018年3月						>
日	一	二	三	四	五	六	
25	26	27	28	1	2	3	
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	

最新随笔

- 1. Redis 基础数据结构与对象
- 2. Redis 内存管理与事件处理
- 3. 【Java核心技术】类型信息(Class对象 反射 动态代理)
- 4. MyBatis框架及原理分析
- 5. Spring的IoC容器
- 6. 如何生成后缀表达式
- 7. 重温设计模式
- 8. Dubbo学习之简单环境搭建
- 9. Linux网络编程-IO复用技术
- 10. Linux进程环境

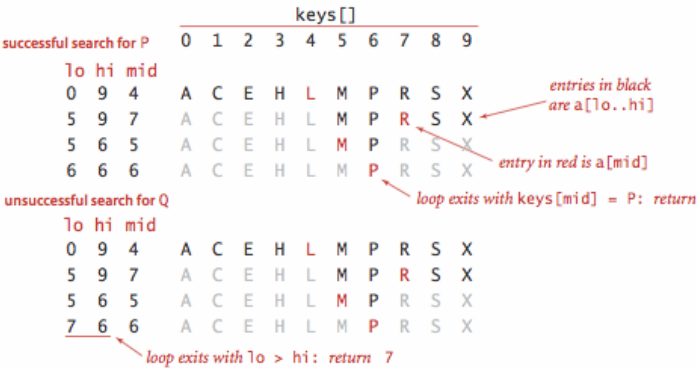
随笔分类

- 【C/C++核心技术】
- 【Java并发实践】
- 【Java核心技术】(1)
- 【SSM框架分析】
- 【Web服务器探索】
- 【高性能MySQL】
- 【后端分布式探索】(2)
- 【技术之外】
- 【开源代码研究】
- 【深入理解JVM】
- 【深入理解OS】
- 【深入探索Linux】
- 【数据结构算法】
- Android(1)
- C/C++(4)
- Java(24)
- OS/网络(7)
- Sql/NoSql(3)
- Unix/Linux(4)
- Web(2)
- 开源代码(8)
- 设计模式&架构(5)
- 数据结构与算法分析(7)

你真的会写二分查找吗

1 二分查找

二分查找是一个基础的算法，也是面试中常考的一个知识点。二分查找就是将查找的键和子数组的中间键作比较，如果被查找的键小于中间键，就在左子数组继续查找；如果大于中间键，就在右子数组中查找，否则中间键就是要找的元素。



(图片来自《算法-第4版》)

```
/**
 * 二分查找，找到该值在数组中的下标，否则为-1
 */
static int binarySearch(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] == key) {
            return mid;
        }
        else if (array[mid] < key) {
            left = mid + 1;
        }
        else {
            right = mid - 1;
        }
    }

    return -1;
}
```

每次移动left和right指针的时候，需要在mid的基础上+1或者-1，防止出现死循环，程序也就能正确的运行。

注意：代码中的判断条件必须是while (left <= right)，否则的话判断条件不完整，比如：array[3] = {1, 3, 5};待查找的键为5，此时在(low < high)条件下就会找不到，因为low和high相等时，指向元素5，但是此时条件不成立，没有进入while()中。


2 二分查找的变种

关于二分查找，如果条件稍微变换一下，比如：数组之中的数据可能可以重复，要求返回匹配的数据的最小（或最大）的下标；更进一步，需要找出数组中第一个大于key的元素（也就是最小的大于key的元素的）下标，等等。这些，虽然只有一点点的变化，实现的时候确实要更加的细心。

二分查找的变种和二分查找原理一样，主要就是变换判断条件（也就是边界条件），如果想直接看如何记忆这些变种的窍门，请直接翻到本文最后。下面来看几种二分查找变种的代码：

2.1 查找第一个与key相等的元素

查找第一个相等的元素，也就是说等于查找key值的元素有好多个，返回这些元素最左边的元素下标。




```
// 查找第一个相等的元素
static int findFirstEqual(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] >= key) {
            right = mid - 1;
        }
        else {
            left = mid + 1;
        }
    }


    if (left < array.length && array[left] == key) {
        return left;
    }

    return -1;
}
```



2.2 查找最后一个与key相等的元素

查找最后一个相等的元素，也就是说等于查找key值的元素有好多个，返回这些元素最右边的元素下标。




```
// 查找最后一个相等的元素
static int findLastEqual(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] <= key) {
            left = mid + 1;
        }
        else {
            right = mid - 1;
        }
    }


    if (right >= 0 && array[right] == key) {
        return right;
    }


    return -1;
}
```



2.3 查找最后一个等于或者小于key的元素

查找最后一个等于或者小于key的元素，也就是说等于查找key值的元素有好多个，返回这些元素最右边的元素下标；如果没有等于key值的元素，则返回小于key的最右边元素下标。





随笔档案

- 2017年7月 (2)
- 2017年6月 (1)
- 2017年4月 (1)
- 2017年3月 (1)
- 2017年2月 (1)
- 2017年1月 (2)
- 2016年12月 (3)
- 2016年11月 (3)
- 2016年10月 (7)
- 2016年9月 (5)
- 2016年8月 (5)
- 2016年7月 (6)
- 2016年6月 (10)
- 2016年5月 (11)
- 2016年4月 (9)
- 2016年3月 (3)

积分与排名

积分 - 94753
排名 - 3323

最新评论

- 1. Re:网上书城+源码
@luoxn28能介绍下你看的web的书籍吗。什么版本号，什么作者的。我也想买买看。刚学java 学到web阶段了。...
--菜鸟小小苏
- 2. Re:你真的会写二分查找吗
(left+right) / 2会溢出啊亲。。。
--keyganker
- 3. Re:MyBatis框架及原理分析
不错，思路清晰讲解详细。
--阳光泛滥的日子
- 4. Re:你真的会写二分查找吗
不错，清晰易懂
--Z皓
- 5. Re:你真的会写二分查找吗
文章写得nice，不过2.1里面不用判断数组是否越界，因为当attr[middle]==key时，right=middle-1；这个-1会最终让attr[right+1]==key，此时left==r.....
--inuyasha~

阅读排行榜

- 1. 深入理解Java反射(31396)
- 2. 你真的会写二分查找吗(22938)
- 3. 优先队列原理与实现(15974)
- 4. google protobuf安装与使用(13788)
- 5. 深入理解FTP协议(9148)

评论排行榜

- 1. 你真的会写二分查找吗(6)
- 2. 网上书城+源码(6)
- 3. Tomjson - 一个"短小精悍"的 json 解析库(4)
- 4. CleanBlog(个人博客+源码)(3)
- 5. MyBatis框架及原理分析(3)

推荐排行榜

- 1. 你真的会写二分查找吗(10)
- 2. MyBatis框架及原理分析(8)
- 3. 重用设计模式(8)
- 4. 网上书城+源码(6)
- 5. STL"源码"剖析-重点知识总结(5)

```
// 查找最后一个等于或者小于key的元素
static int findLastEqualSmaller(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] > key) {
            right = mid - 1;
        }
        else {
            left = mid + 1;
        }
    }
    return right;
}
```

2.4 查找最后一个小于key的元素

查找最后一个小于key的元素，也就是说返回小于key的最右边元素下标。

```
// 查找最后一个小于key的元素
static int findLastSmaller(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] >= key) {
            right = mid - 1;
        }
        else {
            left = mid + 1;
        }
    }
    return right;
}
```

2.5 查找第一个等于或者大于key的元素

查找第一个等于或者大于key的元素，也就是说等于查找key值的元素有好多个，返回这些元素最左边的元素下标；如果没有等于key值的元素，则返回大于key的最左边元素下标。

```
// 查找第一个等于或者大于key的元素
static int findFirstEqualLarger(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] >= key) {
            right = mid - 1;
        }
        else {
            left = mid + 1;
        }
    }
    return left;
}
```

2.6 查找第一个大于key的元素

查找第一个等于key的元素，也就是说返回大于key的最左边元素下标。

```
// 查找第一个大于key的元素
static int findFirstLarger(int[] array, int key) {
    int left = 0;
    int right = array.length - 1;

    // 这里必须是 <=
    while (left <= right) {
        int mid = (left + right) / 2;
        if (array[mid] > key) {
            right = mid - 1;
        }
        else {
            left = mid + 1;
        }
    }
    return left;
}
```

3 二分查找变种总结

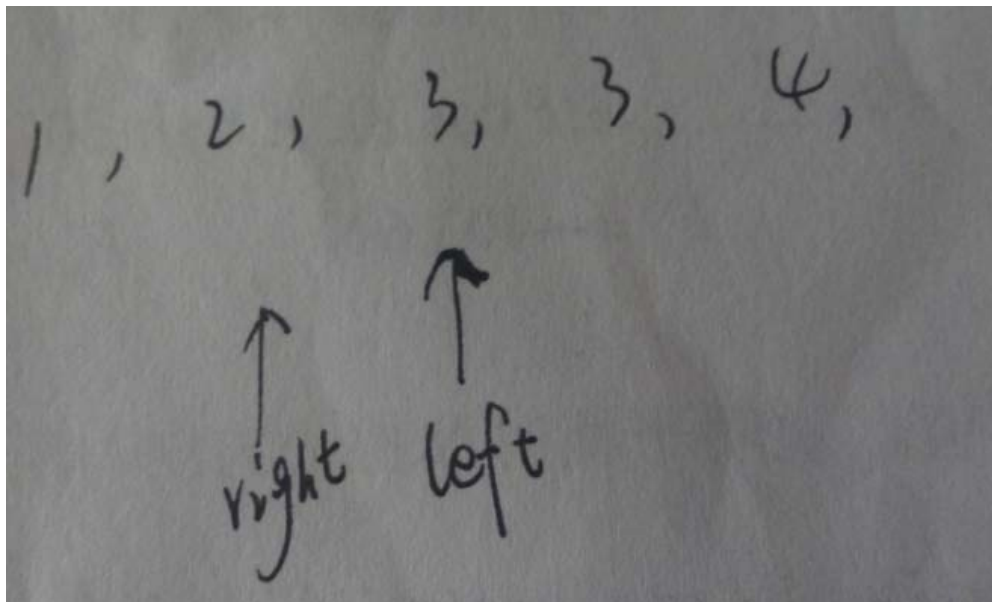
```
// 这里必须是 <=
while (left <= right) {
    int mid = (left + right) / 2;
    if (array[mid] ? key) {
        //... right = mid - 1;
    }
    else {
        // ... left = mid + 1;
    }
}
return xxx;
```

二分查找变种较多，不过它们的“套路”是一样的，以上代码就是其套路，如何快速写出二分查找的代码，只需按照以下步骤即可：

1 首先判断出是返回left，还是返回right

因为我们知道最后跳出while (left <= right)循环条件是right < left，且right = left - 1。最后right和left一定是卡在“边界值”的左右两边，如果是比较值为key，查找小于等于（或者是小于）key的元素，则边界值就是等于key的所有元素的最左边那个，其实应该返回left。

以数组{1, 2, 3, 3, 4}为例，如果需要查找第一个等于或者小于3的元素下标，我们比较的key值是3，则最后left和right需要满足以下条件：



我们比较的key值是3，所以此时我们需要返回left。

2 判断出比较符号

```
int mid = (left + right) / 2;
if (array[mid] ? key) {
    //... right = xxx;
}
else {
    // ... left = xxx;
}
```

也就是这里的 if (array[mid] ? key) 中的判断符号，结合步骤1和给出的条件，如果是查找小于等于key的元素，则知道应该使用判断符号>=，因为是要返回left，所以如果array[mid]等于或者大于key，就应该使用>=，以下是完整代码

```
// 查找小于等于key的元素
int mid = (left + right) / 2;
if (array[mid] >= key) {
    right = mid - 1;
}
else {
    left = mid + 1;
}
```

参考：

- 1、[你真的会写二分检索吗？](#)
- 2、<http://www.cnblogs.com/luoxn28/category/802645.html>

分类: [数据结构与算法分析](#)

好文要顶 关注我 收藏该文 微博 微信



[luoxn28](#)
[关注 - 22](#)
[粉丝 - 132](#)
[+加关注](#)

100

« 上一篇：[深入理解C++对象模型](#)
» 下一篇：[Redis学习总结](#)

posted @ 2016-08-13 16:01 [luoxn28](#) 阅读(22940) 评论(6) [编辑](#) [收藏](#)

评论

#1楼 2017-04-25 10:54 | 搏风雨

@以数组{1, 2, 3, 3, 4, 5}为例，如果需要查找第一个等于或者小于3的元素下标

最后这个例子应该是找第一个大于等于3的元素下标吧

支持(0) 反对(0)

#2楼 2017-05-22 16:26 | 刘毅 (Limer)

<http://www.61mon.com/index.php/archives/187/>，可以看看我的这篇

支持(0) 反对(0)

#3楼 2017-09-05 17:00 | DuMEG

这是我看过的讲二分查找和其变种的最好懂的文章了，赞

支持(0) 反对(0)

#4楼 2017-09-20 15:49 | inuyasha~

文章写得nice，不过2.1里面不用判断数组是否越界，因为当attr[middle]==key时，right=middle-1；这个-1会最终让attr[right+1]==key，此时left==right，然后+1.所以不会越界。

支持(0) 反对(0)

5楼 2017-09-30 15:57 | Z皓

不错，清晰易懂

支持(0) 反对(0)

6楼 2017-11-05 12:31 | keyganker

(left+right) / 2会溢出啊亲。。。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型工控、组态\仿真、建模CAD源码2018！

【活动】杭州云栖·2050大会-全世界年青人因科技而团聚·源点

【抢购】新注册用户域名抢购1元起



最新IT新闻：

- 这73条关于自动驾驶的特写，描绘了一个你可能将会走进的未来世界
 - 谷歌开源FHIR标准协议缓冲工具，利用机器学习预测医疗事件
 - 村里的支付宝：小年轻都在用，中老年有点慌
 - 区块链是妖还是佛？马化腾、李彦宏、张颖都表了态
 - ofo终于等到救命钱
- » [更多新闻...](#)



最新知识库文章：

- [写给自学者的入门指南](#)
 - [和程序员谈恋爱](#)
 - [学会学习](#)
 - [优秀技术人的管理陷阱](#)
 - [作为一个程序员，数学对你到底有多重要](#)
- » [更多知识库文章...](#)