

辉之光

不寄望于奇迹、不依赖于他人、不满足于平庸、不放弃诚信，努力改变现状，尽力做好每一件事。。。

博客园 首页 新闻 新随笔 联系 管理 订阅

038563

昵称: 独孤求败
园龄: 11年8个月
粉丝: 49
关注: 0
+加关注

< 2009年11月 >
日 一 二 三 四 五 六
25 26 27 28 29 30 31
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 1 2 3 4 5

随笔分类(69)
asp.net(14)
C sharp(7)
crystal report(2)
database(7)
Exchange
GIS(7)
JS(2)
oo(1)
Reporting Service(2)
SAAS
sharepoint(4)
WCF(1)
高性能开发(3)
链接备份(1)
摄影知识(3)
生活随笔(14)
数据结构(1)

随笔档案(75)
2014年5月 (1)
2013年6月 (2)
2013年4月 (5)
2013年3月 (3)
2012年11月 (1)
2012年9月 (1)
2011年11月 (1)
2011年8月 (1)
2011年4月 (1)
2009年12月 (2)
2009年11月 (4)
2009年10月 (1)
2009年9月 (1)
2008年11月 (2)
2008年10月 (1)
2008年9月 (1)
2008年7月 (1)
2008年6月 (1)
2008年5月 (1)
2008年3月 (1)
2008年1月 (1)
2007年9月 (1)
2007年8月 (3)
2007年6月 (3)

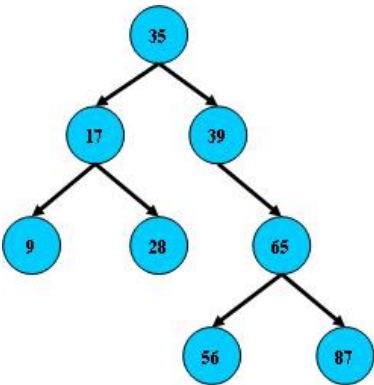
B树、B-树、B+树、B*树

B树

即二叉搜索树：

- 1.所有非叶子结点至多拥有两个儿子（Left和Right）；
- 2.所有结点存储一个关键字；
- 3.非叶子结点的左指针指向小于其关键字的子树，右指针指向大于其关键字的子树；

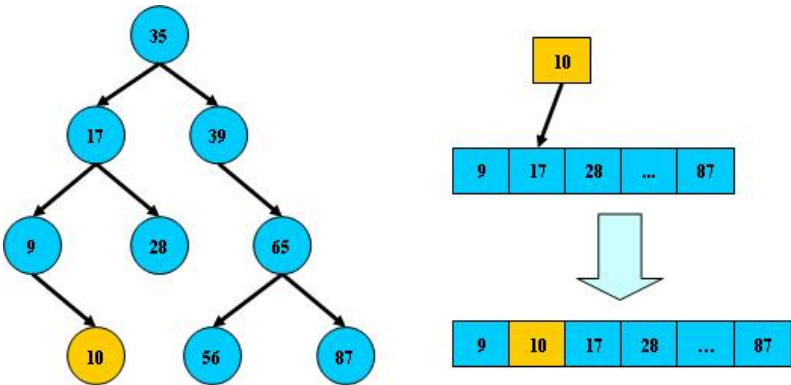
如：



B树的搜索，从根结点开始，如果查询的关键字与结点的关键字相等，那么就命中；否则，如果查询关键字比结点关键字小，就进入左儿子；如果比结点关键字大，就进入右儿子；如果左儿子或右儿子的指针为空，则报告找不到相应的关键字；

如果B树的所有非叶子结点的左右子树的结点数目均保持差不多（平衡），那么B树的搜索性能逼近二分查找；但它比连续内存空间的二分查找的优点是，改变B树结构（插入与删除结点）不需要移动大段的内存数据，甚至通常是常数开销；

如：



但B树在经过多次插入与删除后，有可能导致不同的结构：

2007年5月 (5)
2007年3月 (2)
2007年1月 (2)
2006年11月 (2)
2006年10月 (4)
2006年7月 (2)
2006年6月 (1)
2006年4月 (3)
2006年3月 (1)
2005年12月 (6)
2005年11月 (2)
2005年8月 (2)
2005年7月 (3)

.net开发

Allen Lee's Magic
architects
Artech
CareySon
codeplex
Csharp demo
dudu
huangxincheng
scottgu
shanyou
wangiqngpei557
yanyangtian
道法自然
豪情
路过秋天
圣殿骑士
汤姆大叔
汤雪华的博文
伍华聪
阳光铭睿
邀月工作室
张善
邹建

sps开发

Rickie Lee's blog

web前端

bootsnipp
Monk
wrapbootstrap
梦想天空
前端网址
司徒正美
王福朋
张鑫旭

备用网址

分布式计算论坛

朋友

ss blog
被无视的伊谢尔伦

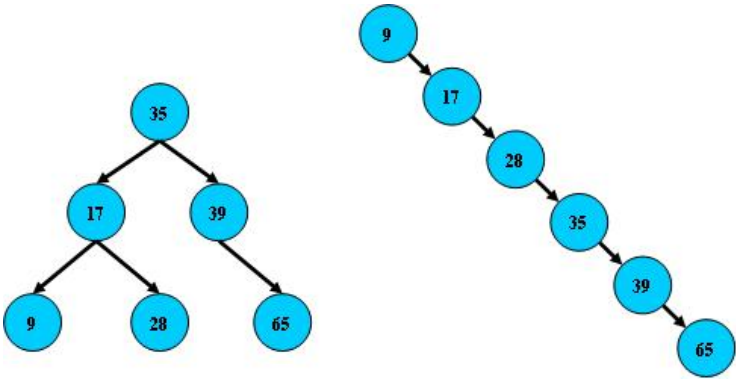
最新评论

1. Re:B树、B-树、B+树、B*树
@独孤求败要不是看了另一篇博客，还不知道这个错误！楼主还是改了吧，加上注释也行，原文肯定是印刷错误，这文章百度排名很高，不知误导多少人！...
--果然如此

2. Re:B树、B-树、B+树、B*树
@失落映画二叉搜索树是 BST，not BT，不要搞错了...
--cnb_yangwei

3. Re:B树、B-树、B+树、B*树
@rocky_24二叉搜索树是 BST，not BT，不要搞错了...
--cnb_yangwei

4. Re:B树、B-树、B+树、B*树
坑爹啊
--livend



右边也是一个B树，但它的搜索性能已经是线性的了；同样的关键字集合有可能导致不同的树结构索引；所以，使用B树还要考虑尽可能让B树保持左图的结构，和避免右图的结构，也就是所谓的“平衡”问题；

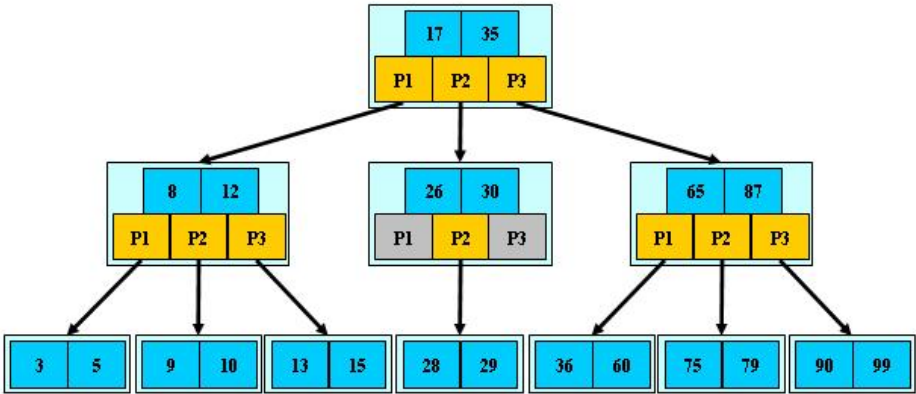
实际使用的B树都是在原B树的基础上加上平衡算法，即“平衡二叉树”；如何保持B树结点分布均匀的平衡算法是平衡二叉树的关键；平衡算法是一种在B树中插入和删除结点的策略；

B-树

是一种多路搜索树（并不是二叉的）：

- 1.定义任意非叶子结点最多只有M个儿子；且M>2；
- 2.根结点的儿子数为[2, M]；
- 3.除根结点以外的非叶子结点的儿子数为[M/2, M]；
- 4.每个结点存放至少M/2-1（取上整）和至多M-1个关键字；（至少2个关键字）
- 5.非叶子结点的关键字个数=指向儿子的指针个数-1；
- 6.非叶子结点的关键字：K[1], K[2], ..., K[M-1]；且K[i] < K[i+1]；
- 7.非叶子结点的指针：P[1], P[2], ..., P[M]；其中P[1]指向关键字小于K[1]的子树，P[M]指向关键字大于K[M-1]的子树，其它P[i]指向关键字属于(K[i-1], K[i])的子树；
- 8.所有叶子结点位于同一层；

如：（M=3）



B-树的搜索，从根结点开始，对结点内的关键字（有序）序列进行二分查找，如果命中则结束，否则进入查询关键字所属范围的儿子结点；重复，直到所对应的儿子指针为空，或已经是叶子结点；

B-树的特性：

- 1.关键字集合分布在整颗树中；
- 2.任何一个关键字出现且只出现在一个结点中；

5. Re:B树、B-树、B+树、B*树
妈蛋，还好看评论，不然只是随便看一下，被坑了，都不知道找谁哭去
--星空不远

- 3.搜索有可能在非叶子结点结束；
- 4.其搜索性能等价于在关键字全集内做一次二分查找；
- 5.自动层次控制；

由于限制了除根结点以外的非叶子结点，至少含有M/2个儿子，确保了结点的至少利用率，其最底搜索性能为：

$$\begin{aligned} O_{Min} &= O[\log_2(\lceil \frac{M}{2} - 1 \rceil) \times \log_{\frac{M}{2}}(\lceil \frac{N}{\frac{M}{2} - 1} \rceil)] \\ &= O[\log_2(\frac{M}{2})] \times O[\log_{\frac{M}{2}}(\frac{N}{\frac{M}{2}})] \\ &= O[\log_2(\frac{M}{2}) \times (\log_{\frac{M}{2}} N - 1)] \\ &= O[\log_2 N - \log_2(\frac{M}{2})] \\ &= O[\log_2 N] - O[C] \\ &= O[\log_2 N] \end{aligned}$$

其中，M为设定的非叶子结点最多子树个数，N为关键字总数；
所以B-树的性能总是等价于二分查找（与M值无关），也就没有B树平衡的问题；
由于M/2的限制，在插入结点时，如果结点已满，需要将结点分裂为两个各占M/2的结点；删除结点时，需将两个不足M/2的兄弟结点合并；

B+树

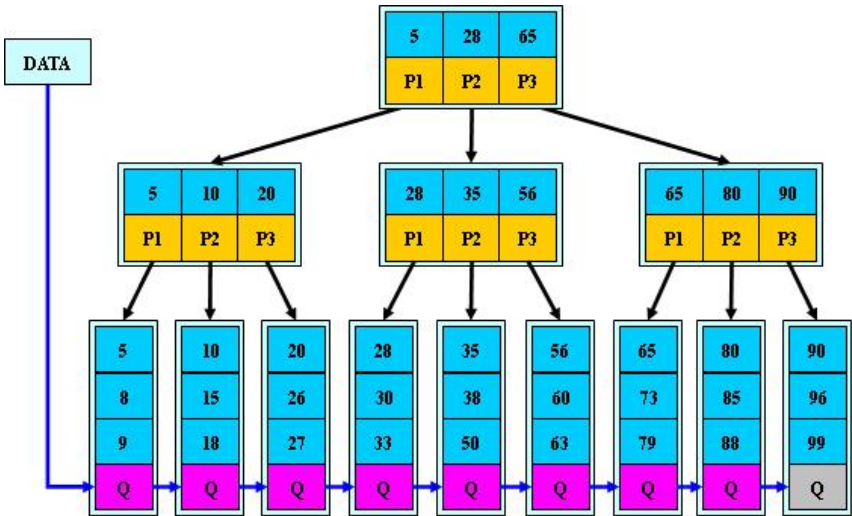
B+树是B-树的变体，也是一种多路搜索树：

- 1.其定义基本与B-树同，除了：
- 2.非叶子结点的子树指针与关键字个数相同；
- 3.非叶子结点的子树指针P[i]，指向关键字值属于[K[i], K[i+1])的子树

（B-树是开区间）；

- 5.为所有叶子结点增加一个链指针；
- 6.所有关键字都在叶子结点出现；

如：（M=3）



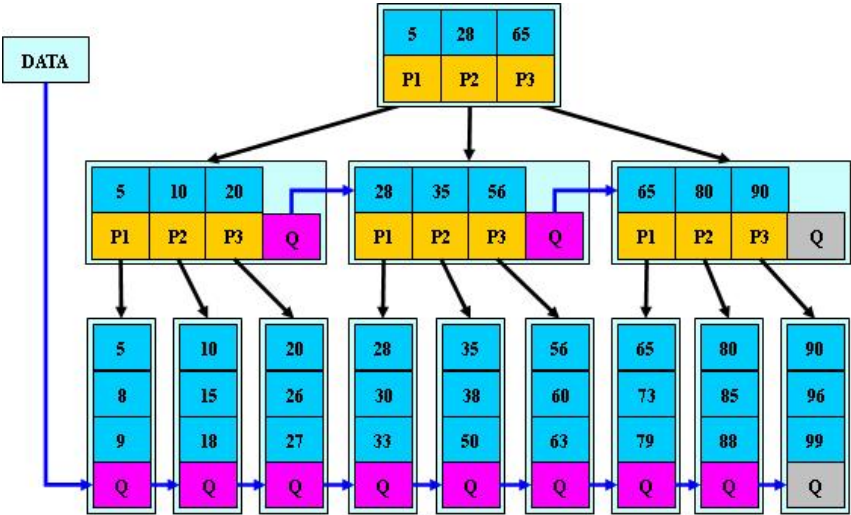
B+的搜索与B-树也基本相同，区别是B+树只有达到叶子结点才命中（B-树可以在非叶子结点命中），其性能也等价于在关键字全集做一次二分查找；

B+的特性:

- 1.所有关键字都出现在叶子结点的链表中（稠密索引），且链表中的关键字恰好是有序的；
- 2.不可能在非叶子结点命中；
- 3.非叶子结点相当于是叶子结点的索引（稀疏索引），叶子结点相当于是存储（关键字）数据的数据层；
- 4.更适合文件索引系统；

B*树

是B+树的变体，在B+树的非根和非叶子结点再增加指向兄弟的指针；



B*树定义了非叶子结点关键字个数至少为 $(2/3)*M$ ，即块的最低使用率为2/3（代替B+树的1/2）；

B+树的分裂：当一个结点满时，分配一个新的结点，并将原结点中1/2的数据复制到新结点，最后在父结点中增加新结点的指针；B+树的分裂只影响原结点和父结点，而不会影响兄弟结点，所以它不需要指向兄弟的指针；

B*树的分裂：当一个结点满时，如果它的下一个兄弟结点未满，那么将一部分数据移到兄弟结点中，再在原结点插入关键字，最后修改父结点中兄弟结点的关键字（因为兄弟结点的关键字范围改变了）；如果兄弟也满了，则在原结点与兄弟结点之间增加新结点，并各复制1/3的数据到新结点，最后在父结点增加新结点的指针；

所以，B*树分配新结点的概率比B+树要低，空间使用率更高；

小结

B树：二叉树，每个结点只存储一个关键字，等于则命中，小于走左结点，大于走右结点；

B-树：多路搜索树，每个结点存储M/2到M个关键字，非叶子结点存储指向关键字范围的子结点；

所有关键字在整颗树中出现，且只出现一次，非叶子结点可以命中；

B+树：在B-树基础上，为叶子结点增加链表指针，所有关键字都在叶子结点中出现，非叶子结点作为叶子结点的索引；B+树总是到叶子结点才命中；

B*树：在B+树基础上，为非叶子结点也增加链表指针，将结点的最低利用率从1/2提高到2/3；

原文地址 <http://blog.csdn.net/manesking/archive/2007/02/09/1505979.aspx>

分类: [数据结构](#)

好文要顶

关注我

收藏该文

[独孤求败](#)
关注 - 0
粉丝 - 49
[+加关注](#)

199

« 上一篇: [分布式计算 网格计算 并行计算 云计算](#)
» 下一篇: [.net学习资料链接收集整理](#)

posted @ 2009-11-16 17:44 独孤求败 阅读(166212) 评论(34) 编辑 收藏

发表评论

#1楼 2011-09-29 21:15 | Firefly727

弄错了吧！我记得：B树和B-树是同一种树，只不过英语中B-tree被中国人翻译成了B-树，让人以为B树和B-树是两种树，实际上，两者就是同一种树，而你文中所说的B树是二叉查找树，并不叫B树！

支持(42) 反对(0)

#2楼 2011-10-25 14:53 | yonghu86

不错。

支持(0) 反对(0)

#3楼 2012-02-17 09:07 | nightmare23

一楼正解。

支持(0) 反对(0)

#4楼 2012-12-12 09:53 | 康杜

二叉查找树是Binary Search Tree，而不是B Tree。B Tree就是B-Tree。
BST http://en.wikipedia.org/wiki/Binary_search_tree
B-tree <http://en.wikipedia.org/wiki/B-tree>

支持(7) 反对(0)

#5楼 2013-02-25 20:57 | 寒塘渡鹤影

B树就是二叉排序树，楼主说的是对的。

支持(0) 反对(2)

#6楼 2013-03-30 17:37 | A_Crazy_Man

纯属误导人！！

支持(6) 反对(0)

#7楼 2013-05-27 15:55 | 失落映画

@ 寒塘渡鹤影
[引用](#)
B树就是二叉排序树，楼主说的是对的。

别在误导人了！

二叉树 ,英文叫 Binary Tree
B树, 英文叫 B-Tree (注意是连起来的)

国内一些高大纯S.B看到B-Tree,然后还看到 B+ Tree(注意+ 与 Tree之间分隔开的),就翻译为B-树,B+树,纯粹的不懂英文!

支持(6) 反对(0)

#8楼[楼主] 2013-05-28 11:00 | 独孤求败

@ 失落映画
thanks, 原文作者也许在写的时候搞错或者概念理解有问题, B树为b-tree (或者翻译的b-树), 而且之前也有网友出了, 由于是引用的, 本博客也不修改, 相信有下面的评论大家更能有兴趣研究一下, 但是请尊重一下原作者, 注明用语, 在此表示感谢!

支持(0) 反对(0)

#9楼 2013-07-24 11:05 | A_Crazy_Man

本文严重误导很多人, 期待作者可以尽快修正Blog内容.

支持(0) 反对(0)

#10楼 2013-10-10 03:04 | 寒心雪林

B-树其实就是B树, 或者叫B_树, 源于英文的B-tree (B_tree),把tree翻译成中文就成为了通常说的B-树或者B_树, 其实就是B树, 一个东西来的。所谓的二叉搜索树叫BST (binary search tree), 明白了吗?

支持(0) 反对(0)

#11楼 2013-12-02 09:07 | oyld

看到第一句就知道错了。。

支持(0) 反对(0)

#12楼 2014-01-15 21:23 | len_sround

博主不修改吗? 这要误导多少人?

支持(0) 反对(0)

#13楼 2014-03-03 14:34 | ~嘉言懿行~~我是煲仔饭~~

mark, 感谢分享。

支持(0) 反对(0)

#14楼 2014-04-07 14:12 | Brav3heart

Binary Search Tree VS Balance Tree, 通常称后者为B-Tree。
名字无所谓, 讲清楚数据结构就行!

支持(0) 反对(0)

#15楼 2014-04-16 15:19 | Lora_wen

B=B-树, 二叉搜索树又称为BST, 不是B树.....我记得老师讲过, 你可以查查, 改一下吧, 好多人等着看呢(>_<)

支持(0) 反对(0)

#16楼 2014-05-22 11:11 | 凤儿

除了二叉搜索树是B树外这个错误外, 其它内容写的还是很好的, 还是建议楼主修改下

支持(0) 反对(0)

#17楼 2014-07-21 19:19 | Goodyang

楼主, 错误被提出来这么久了, 你不打算修改吗? 这样好吗? 你不知道会误导很多人吗?

支持(0) 反对(0)

#18楼 2014-08-08 18:32 | 蒲公英小帝

话说 B树就是B-tree吧, 二叉查找树是 (BST) 好不好, 误导人呢?

支持(0) 反对(0)

#19楼 2014-09-05 10:48 | 吴浪舟

我觉得 B树和 排序二叉树是有本质区别的; 这两种树建树的方法有天壤之别, 不应该拿过来一起讨论;

支持(0) 反对(0)

2017/3/1	B树、B-树、B+树、B*树 - 独孤求败 - 博客园
#20楼 2015-02-26 16:35 rockZ	误人子弟
	支持(0) 反对(0)
#21楼 2015-07-15 10:39 添经地翼	B树就是B-树，而不是binary tree二叉树。这个只是之前翻译的问题而已。
	支持(0) 反对(0)
#22楼 2015-08-28 20:51 NorthrendSnow	从上到下顺序： 二叉查找树 B树 B+树 B*树 木有B-树的说法
	支持(0) 反对(0)
#23楼 2015-09-07 11:14 rocky_24	恩恩 误人子弟。
	支持(0) 反对(0)
#24楼 2015-09-07 11:15 rocky_24	一楼好人 @Firefly727 <u>引用</u> 弄错了吧！我记得：B树和B-树是同一种树，只不过英语中B-tree被中国人翻译成了B-树，让人以为B树和B-树是两种树，实际上，两者就是同一种树，而你文中所说的B树是二叉查找树，并不叫B树！
	支持(0) 反对(0)
#25楼 2015-09-25 00:22 _____Json	您在介绍 B-tree的时候,第三条说 3.除根结点以外的非叶子结点的儿子数为[M/2, M]; 但是您的图 (也就是b-tree段落第一张图) 的第二个子节点(中间非叶子子节点)的子节点就一个.这是为什么呢. 通过您的图可以看出M=3
	支持(0) 反对(0)
#26楼 2015-09-25 10:24 水边	楼主真够坚持，死性不改啊..... 一定要把误导坚持到底，牛x
	支持(0) 反对(0)
#27楼 2015-10-08 13:32 放作夥	@ Firefly727 很多时候看网上资料还以为存在3种类型的B树，其实是2种，但是楼主让人明白下次你再看见B树就要知道这些人只是在讲二叉，真正意义的B树只有两种。
	支持(0) 反对(0)
#28楼 2015-10-19 10:37 WQZ321123	多谢
	支持(0) 反对(0)
#29楼 2015-11-12 22:09 生活之禅	B树不是二叉搜索树！
	支持(0) 反对(0)

2017/3/1

B树、B-树、B+树、B*树 - 独孤求败 - 博客园

#30楼 2016-02-28 11:35 | 星空不远

妈蛋，还好看了评论，不然只是随便看一下，被坑了，都不知道找谁哭去

支持(0) 反对1

#31楼 2016-07-25 16:25 | livend

坑爹啊

支持(0) 反对1

#32楼 2016-08-13 13:48 | cnb_yangwei

@ rocky_24二叉搜索树是 BST，not BT，不要搞错了

支持(0) 反对1

#33楼 2016-08-13 13:50 | cnb_yangwei

@ 失落映画
二叉搜索树是 BST，not BT，不要搞错了

支持(0) 反对1

#34楼 2016-11-08 15:22 | 果然如此

@ 独孤求败
要不是看了另一篇博客，还不知道这个错误！楼主还是改了吧，加上注释也行，原文肯定是印刷错误，这篇文章百度排很高，不知误导多少人！

支持(0) 反对1

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。

- 最新IT新闻：
- 那些喜欢等反转的人都是缺乏主见的

· Ghost Robotics演示能适应不同地形的机器人Minitaur

· 想成为物联网领域的小米，你必须要看的10个原则

· 联通微博微卡来了：刷微博免流量费

· Google悄然发布全新视频会议应用Meet
- » 更多新闻...
- 最新知识库文章：
- 垃圾回收原来是这么回事

· 「代码家」的学习过程和学习经验分享

· 写给未来的程序媛

· 高质量的工程代码为什么难写

· 循序渐进地代码重构
- » 更多知识库文章...