

# 孩子兄弟树查找双亲结点的算法<sup>①</sup>

袁关伟

(69027 部队, 乌鲁木齐 830000)

**摘要:** 为完善孩子兄弟树抽象数据类型中的基本操作, 进而更好地服务于教学科研和日常应用, 解决了在孩子兄弟树中查找双亲结点这一重要的、复杂的和缺少深度研究的操作。通过数学方法分析得知, 该算法的时间复杂度和空间复杂度均为  $O(n)$ 。同时, 通过对该算法的应用潜力进行挖掘, 发现其在地名查询系统中具有很好应用前景。

**关键词:** 孩子兄弟树; 查找; 双亲结点

## Algorithm of Searching for Parents Node in Child-Sibling Tree

YUAN Guan-Wei

(69027 Troops, Urumqi 830000, China)

**Abstract:** In order to complete the basic operation in abstract data type of Child-Sibling Tree and then better serves for teaching and daily application, the paper solves a important and complex operation of the lack of depth research in Child-Sibling Tree——searching parents node. By the means of mathematical techniques analysis that the time and space complexity of the algorithm are both  $O(n)$ . It is known by excavating its potential that the algorithm has good application prospects in the place-name searching system.

**Key words:** child-sibling tree; search; parents node

树是图论<sup>[1]</sup>、离散数学<sup>[2-9]</sup>和数据结构<sup>[10-21]</sup>研究的重要内容, 可以用来描绘现实世界中广泛存在的层次结构关系<sup>[1-21]</sup>。如人类社会的族谱和各种社会组织机构<sup>[1-21]</sup>、操作系统文件系统的目录结构<sup>[1-22]</sup>、编译程序源程序的语法结构<sup>[1-21]</sup>、基因系列索引结构<sup>[23]</sup>、空间数据库的索引结构<sup>[24-35]</sup>等等。树的存储结构有双亲存储结构、孩子链存储结构、孩子兄弟链存储结构等<sup>[10-21]</sup>, 其中孩子兄弟链存储结构因为易于实现与二叉树的相互转换(借鉴处理二叉树的方法来处理树)、节省存储空间等优点而成为最重要(使用最多)的树的存储结构。这些优点和事实, 国内外经典《数据结构》教材均有论述。例如, 与爱因斯坦《相对论》并称的美国斯坦福大学 Knuth 教授(加州理工学院博士, 已退休)的图灵奖级名著《计算机程序设计艺术》认为以孩子兄弟链为存储结构的树(以下简称孩子兄弟树, 英文缩写 CST)易于实现与二叉树的相互转换, 并深刻剖析了其存储

结构的科学内涵<sup>[10]</sup>; 有“20 世纪最重要的 30 本计算机教材之一”美誉的美国佛罗里达国际大学 Weiss 教授(普林斯顿大学博士)的权威名著《数据结构与算法分析—C 语言描述》认为树的孩子兄弟链存储结构具有节省存储空间的优点, 并论证了其科学内涵<sup>[11]</sup>; 列入世界著名计算机教材精选的美国南加州大学 Horowitz 教授(威斯康星大学麦迪逊分校博士)的经典名著《数据结构基础(C 语言版)》通过对比书中所述树的两种存储结构间接地说明了树的孩子兄弟链存储结构具有节省存储空间的优点, 并以图文并茂的形式说明了孩子兄弟树易于实现与二叉树的相互转换<sup>[12]</sup>; 美国佛罗里达大学 Sahni 教授(康奈尔大学博士)编著的《数据结构算法与应用—C++ 语言描述》图文并茂地说明了孩子兄弟树易于实现与二叉树的相互转换<sup>[13]</sup>; 北京大学张铭教授等编著的《数据结构与算法实验教材》将树的孩子兄弟链存储结构归为重点掌握问题<sup>[14]</sup>; 技术专家左飞

① 收稿时间:2016-03-02;收到修改稿时间:2016-05-19 [doi:10.15888/j.cnki.csa.005519]

编著的《C++数据结构原理与经典问题求解》认为树的孩子兄弟链存储结构是目前已知的最节省存储空间的树的存储结构<sup>[15]</sup>;武汉大学李春葆教授等编著的《数据结构教程(第2版)》、西安交通大学朱成立教授编著的《数据结构——使用C语言(第4版)》认为树的孩子兄弟链存储结构的最大优点是可以方便地实现树和二叉树之间的相互转换<sup>[16,17]</sup>。……孩子兄弟链存储结构易于实现树的抽象数据类型中初始化树 Initiate(T)、查找当前结点的第一个孩子结点 LeftChild(T,curr)、查找当前结点的下一个兄弟结点 RightSibling(T,curr)、遍历树 Traverse(T,Visit())等各种操作,唯有查找当前结点的双亲结点的 Parent(T,curr)操作比较复杂,这一点也得到了国内很多《数据结构》优秀教材的一致认同。例如,清华大学严蔚敏教授等编著的《数据结构(C语言版)》、哈工大廖明宏教授等编著的《数据结构与算法(第4版)》认为要改变查找当前结点双亲结点的 Parent(T,curr)操作的复杂困局,需要在结点中增加一个指向双亲结点的指针域<sup>[18,19]</sup>,这是一种通过改变数据结构来降低算法难度的做法,显然不符合本文旨在特定数据结构——树的孩子兄弟链存储结构(其重要性前面已述)下寻求问题解决方法的构想,但鉴于它们系国内优秀的数据结构教材,作者还是分析一下这种做法的弊端。具体为:一是浪费存储空间,再则增加的这一指针域对除 Parent(T,curr)操作外的任何操作都毫无用处,这也印证了 Knuth 教授在其权威计算机著作《计算机程序设计艺术》中的一句话:向上的链接本身在大多数情况下几乎是不适当的,因为很难说出一个结点是否终结点,或者给出它的任何儿子的地址<sup>[10]</sup>;北京大学张乃孝教授在其著作《数据结构与算法(第2版)》中将孩子兄弟树的 Parent(T,curr)操作归为麻烦问题<sup>[20]</sup>;北京大学张铭教授在其合著《数据结构与算法实验教材》中将孩子兄弟树的 Parent(T,curr)操作归为难点问题<sup>[14]</sup>;武汉大学李春葆教授在其合著《数据结构(第2版)》中将孩子兄弟树的 Parent(T,curr)操作归为麻烦问题<sup>[16]</sup>。……上述谈及的这些经典教材中,无一给出孩子兄弟树 Parent(T,curr)操作的具体算法(含复杂度分析)。再从各大期刊来看,作者通过中国知网检索后发现,与树的孩子兄弟链存储结构有关的论文仅有3篇,按发表时间的先后顺序排列分别是:《基于孩子兄弟树的 FAT32 文件删除恢复算法》<sup>[22]</sup>、《基于扩展前缀编码的左孩子右兄弟结构连接算法》<sup>[36]</sup>、《基于

递归算法在孩子兄弟法表示的树中查找任一结点的双亲结点算法的设计与实现》<sup>[37]</sup>。前2篇论文是关于孩子兄弟树的应用研究,具体应用了孩子兄弟树查找孩子结点的相关算法;第3篇论文是关于孩子兄弟树的基础研究,具体研究内容是孩子兄弟树查找双亲结点的算法。第3篇论文的出现虽然对孩子兄弟树基础理论的完善起到了推动作用,但因为是基于递归算法实现的,存在抽象和算法效率不高的弊端,更重要的是缺少对算法复杂度的分析。综上所述,为了得出方法直观、效率更高、分析彻底的孩子兄弟树 Parent(T,curr)操作的算法,进而完善孩子兄弟树的抽象数据类型中的基本操作,以便更好地服务于教学科研和日常应用,于是本文应运而生。

## 1 预备知识

### 1.1 树的孩子兄弟链存储结构

树的孩子兄弟链存储结构为每个结点设计三个域<sup>[10-21]</sup>:一个数据元素域、一个指向该结点的第一个孩子结点的指针域、一个指向该结点的下一个兄弟结点的指针域。这种存储结构的一个显著特征是能在一种结构中平衡兼顾地表达出层次结构关系和线性结构关系——双亲和孩子之间是层次结构关系(通过左子链链接表达),兄弟和兄弟之间是线性结构关系(通过右指针链接表达)。这种特征使得其对应的孩子兄弟树兼具树和单链表的双重特性:一个结点的所有孩子结点本质上是以其第一个孩子结点为头结点的单链表对应的结点。

孩子兄弟链存储结构下树的结点类型定义如下:

```
typedef struct TNode
{
    ElemType data; //ElemType 为通用类型
    struct TNode *firstChild; //结点的长子结点的指针
    struct TNode *nextSibling; //结点的兄弟结点的指针
}CST_Node;
```

树的孩子兄弟链存储结构如图1所示。

### 1.2 队列

环形队列(特殊的顺序队列)因具有节省存储空间,防止“假溢出”现象的产生等优点而被广泛使用<sup>[10-21]</sup>。在环形队列中,为了方便区别队空和队满,通常约定在入队时少用一个数据元素空间,以队尾指针加1等于队首指针来判断队满<sup>[16]</sup>。

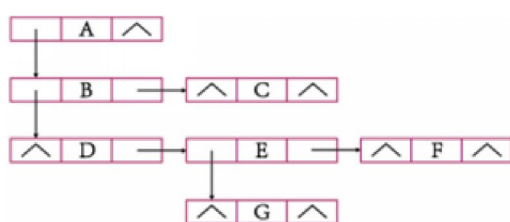


图1 树的孩子兄弟链存储结构

环形队列的结点类型定义如下:

```

typedef struct
{
    ElemType data[MaxSize]; //ElemType 为通用类型
    int front,rear; //队头和队尾指针
}SqQueue;
  
```

设  $q$  是指向数据类型  $SqQueue$  的指针变量, 则在环形队列中判断队满和队空的条件分别如下(队头队尾指针初始化时置为 0, 出队入队时对应加 1)<sup>[16]</sup>:

队满条件:  $(q->rear+1)\%MaxSize==q->front$

队空条件:  $q->rear==q->front$

### 1.3 穷举搜索法

穷举搜索法<sup>[14,38]</sup>, 就是从问题可能的解空间中列举所有可能的解状态, 并用给定的检验条件进行判定, 能使命题成立的, 即为问题的解. 穷举搜索法简单易行, 当变量个数不多且每个变量取值个数也不多的情况下, 使用这种方法是非常有效的. 对于复杂问题, 如果找不到更好的算法, 计算机的速度和容量又允许, 也可以使用穷举搜索法. 穷举搜索法的要点在于“穷举”, 即必须列出所有可能的候选解, 并用逻辑表达式表达清楚, 特别应注意边界条件和组合条件. 当候选解可能取无限个值时, 不能使用穷举搜索法. 当候选解取值虽为有限但取值空间非常大时, 虽然从理论上可以使用穷举搜索法, 但考虑到计算机硬件条件的限制, 可能不得不将方法优化或采用别的方法.

### 1.4 树的性质

树的结点数等于其所有结点的度之和加 1<sup>[1-9,16]</sup>.

证明: 一方面, 根据树的定义可知, 在一棵树中, 除根结点外, 每个结点有且仅有一个前驱结点, 即每个结点与指向它的一个分支一一对应, 所以树中除根结点之外的结点数等于其所有结点的分支数之和. 另一方面, 根据树的度的定义可知, 树的所有结点的分支数之和即树的所有节点的度之和. 综上所述可得, 树的结点数等于其所有结点的度之和加 1.

## 2 算法设计

### 2.1 算法思想

本文采用的算法设计方法为穷举搜索法, 问题的解空间为除左指针为空的结点外的所有结点. 具体为: 要在 CST 中查找某一结点  $x$  的双亲结点, 不妨反过来想, 若在 CST 中能够找到结点  $r$ ,  $r$  具有“第一个孩子结点的主关键字值等于  $x$  的主关键字值, 或第一个孩子结点的任一右兄弟结点的主关键字值等于  $x$  的主关键字值”的特征, 那么就一定可将  $r$  确定为  $x$  的双亲结点. 为此, 可按一定的顺序逐一试探 CST 的所有结点, 若试探到  $r$  结点, 说明已找到  $x$  的双亲结点, 立即返回  $r$  的指针, 结束试探; 若试探完 CST 的所有结点, 都没遇到  $r$  结点, 则说明在 CST 中不存在  $x$  的双亲结点, 则返回 NULL. 在 CST 中不存在  $x$  的双亲结点有两种原因: 一种原因是  $x$  的主关键字值和 CST 根结点的主关键字值相等(树根不可能有双亲结点); 另一种原因是  $x$  的主关键字值不等于 CST 中任一结点的主关键字值.

### 2.2 算法实施策略

为实现算法思想中逐一试探的设想, 不妨采用数据结构队列. 具体为: 从 CST 的根结点开始, 对每个结点, 按照“先试探该结点本身, 再试探该结点的第一个孩子结点(若第一个孩子结点存在), 进而试探该结点的第一个孩子结点的系列右兄弟结点(若第一个孩子结点的系列右兄弟结点存在)”的顺序逐一试探, 直到试探到  $r$  结点或试探完 CST 的所有结点为止. 由于队列具有“先进先出”的特点<sup>[10-21]</sup>, 所以它天生就具有维护这种试探次序的能力. 上面已经论述过, 当  $x$  的主关键字值等于根结点的主关键字值时, 则在 CST 中不存在  $x$  的双亲结点, 利用此可优化算法. 因为试探是从根结点开始的, 所以在试探根结点时, 先判断一下其主关键字值是否等于  $x$  的主关键字值, 若是, 则可提前宣布查找失败, 返回 NULL. 另外, 由 CST 的结构不难发现: CST 的结点有资格成为双亲结点的充分必要条件是其左指针域非空. 下一节中算法实现部分提前结束对某个结点检验的理论依据就在于此——既然没有资格做双亲结点, 自然就不会是任何结点的双亲结点了.

## 3 算法实现

本文算法采用 C 语言实现, 以下即为实现函数. 在这个函数中, 结点的主关键字类型被定义为字符串

型, TVN 为孩子兄弟树的结点数, 指针数组大小为 TVN+1(环形队列入队时少用一个数据元素空间, 故加 1)可绝对避免队满现象的发生, 因为入队元素不可能超过孩子兄弟树的结点数。

```
CST_Node *searchParent (CST_Node *root,char x[])
{ /*函数名: CST 寻亲函数 程序员: 袁关伟
   时间: 2014 年 11 月 地点: 北师大*/
  CST_Node *queue[TVN+1],*q;
  int front=0,rear=0;
  if(strcmp(root->data,x)==0)
    return NULL;
  rear=(rear+1)%TVN;
  queue[rear]=root;
  while(front!=rear)
  {
    front=(front+1)%TVN;
    if(queue[front]->firstChild==NULL)
      continue; //该结点无孩子, 提前结束本趟试探
    else if(strcmp(queue[front]->firstChild->data,x)==0)
      return queue[front];
    else
    { //在 CST 的长子结点的系列右兄弟结点中查找
      rear=(rear+1)%TVN;
      queue[rear]=queue[front]->firstChild;
      q=queue[front]->firstChild->nextSibling;
      while(q)
      {
        if(strcmp(q->data,x)==0)
          return queue[front];
        rear=(rear+1)%TVN;
        queue[rear]=q;
        q=q->nextSibling;
      }
    }
  }
  return NULL;
} //结束—寻亲函数
```

#### 4 算法测试及应用

在日常生活中, 人们常常会遇到听说过某个地名但不知其细节的情况。譬如, 听说过苦鲁勒村(国内前

往世界第二高峰乔戈里峰的必经之地), 但不知道它属于哪个省(自治区、直辖市)哪个市(地区、州)哪个县(市、区)哪个乡(镇), 不知道它是行政村还是村小组(居民点), 不知道若它是行政村, 还下辖哪些村小组(居民点)。这样的需求催生了地名查询系统的诞生。现以云南省地名查询系统为例(图 2、图 3 分别为云南省行政区划图和其所对应的 CST 模型图), 该系统要求输入任一行政区名, 可以查找到其上级行政区名和其所管辖的所有下级行政区名, 且可查到的最高行政区名为省级、最低行政区名为村小组级(本文的实验数据只到县级)。值得注意的是, 为了增强系统的健壮性, 防止地名相同时引发的查找错误, 可以建立同名数组或数据库(数组或数据库中的记录含相同地名数量的字段)。具体为: 在输入一个地名查找时, 先通过同名数组或数据库过滤, 若该地名位列其中, 则提示接下来的正确输入形式为“地名+阿拉伯数字”或“地名+英文字母”, 本质上是符合调用 CST 相关算法的输入条件变为符合的输入条件——输入一个记录的主关键字值。因为地名相同的记录相对于地名不同的记录毕竟很少, 所以一个地名通过同名数组或数据库过滤带来的时空开销几乎可以忽略不计。在 VC++ 6.0 中, 用 C 语言编写测试程序实现上述功能, 其中查找某地名的上级行政区名的功能就是调用本文算法实现的, 测试结果如图 4 所示。



图 2 云南省行政区划图

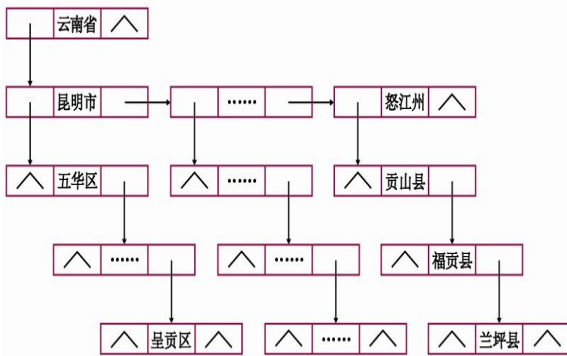


图 3 云南省行政区划 CST 模型图

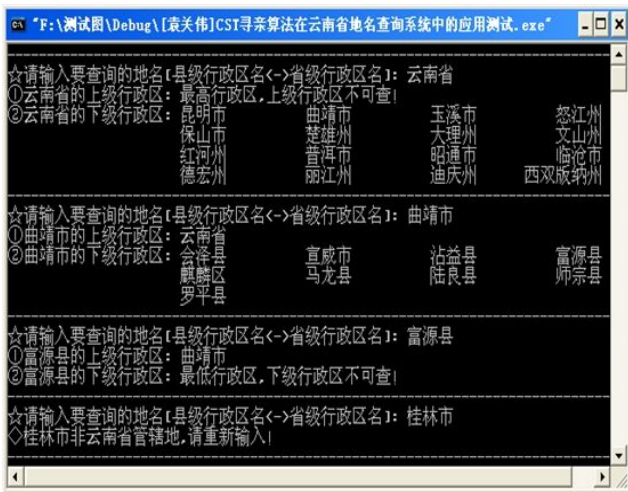


图 4 本文算法在云南省地名查询系统中的应用测试图

### 5 算法分析

设 CST 的结点数为  $n$ , 每个结点本身的个数为  $Num(node\_i)=1(1 \leq i \leq n)$ , 每个结点的度为  $Degree(node\_i)(1 \leq i \leq n)$ ,  $f(n)$  是基本操作重复执行的次数. 从算法实现部分可以清楚地看到, 本文算法的基本操作本质上是访问结点的操作, 同时每个结点  $node\_i(1 \leq i \leq n)$  和其所有子结点是被作为一个整体单元访问的, 最坏情况下(查找失败时), CST 中所有结点及其子结点都要被访问(叶子结点的子结点数视为 0). 于是可得如下关系式(本计算过程利用了 1.4 节的树的性质):

$$\begin{aligned}
 f(n) &= \sum_{i=1}^n (Num(node\_i) + Degree(node\_i)) \\
 &= \sum_{i=1}^n Num(node\_i) + \sum_{i=1}^n Degree(node\_i) \\
 &= n + (n - 1) = 2n - 1 = O(n)
 \end{aligned}$$

所以, 本文算法在查找失败时的时间复杂度为  $O(n)$ . 显然, 查找成功时的时间复杂度一定小于查找失败时的时间复杂度, 所以本文算法的最坏时间复杂

度为  $O(n)$ . 同时, 很容易看出, 本文算法的最好时间复杂度为  $O(1)$ , 如待查找结点为根结点或根结点的第一个孩子结点时即为最好情况下时间复杂度的例子.

由于实施本文算法的辅助空间主要由循环队列决定, 而循环队列的大小为 CST 的结点数, 所以本文算法的空间复杂度为  $O(n)$ .

### 6 结语

一方面, 本文阐明了在孩子兄弟树中查找任一结点的双亲结点这一算法的完整科学体系——设计、实现、实验、应用和分析, 彻底解决了孩子兄弟树抽象数据类型中一个重要的、复杂的、缺少深度研究的操作, 这对过去学习《数据结构与算法》时对此问题费解过的学生、现在学习《数据结构与算法》时对此问题正在费解的学生以及将来学习《数据结构与算法》时对此问题可能费解的学生来说都是雪中送炭, 这一点曾经经历过这种费解的作者是刻骨铭心的; 另一方面, 本文集如何选用算法设计方法、如何运用数据结构、如何利用树的性质以及如何采用逆向思维设计算法和整体思维分析算法(的复杂度)于一体, 知识集成度高, 是《数据结构与算法》不可多得的综合案例, 可供学习和研究《数据结构与算法》的读者作训练用. 这两方面决定了本文的学术价值. 对于运用价值, 作者抛砖引玉, 仅结合自身专业背景客观分析了本文算法在地名查询系统中有较好的运用, 相信各位读者结合自身专业背景和工作实践, 还将有更多更好的运用发现. 当然, 使用穷举搜索法设计的所有算法都存在效率难达最佳的弊端, 本文算法也不例外, 所以作者也衷心期望日后有读者对此问题能从算法设计方法上寻求突破, 以取得更好的时间和空间复杂度.

### 参考文献

- 1 王树禾.图论.北京:科学出版社,2004.
- 2 Grimaldi RP. 林永钢译.离散数学与组合数学.第 5 版.北京:清华大学出版社,2007.
- 3 Johnsonbaugh R. 石纯一,金涛,等,译.离散数学.第 5 版.北京:人民邮电出版社,2003.
- 4 Dossey JA, Otto AD, Spence LE, et al. 章炯民,王新伟,曹立,译.离散数学.第 4 版.北京:清华大学出版社,2005.
- 5 Dossey JA, Otto AD, Spence LE, et al. 章炯民,王新伟,曹立,译.离散数学.第 5 版.北京:机械工业出版社,2007.

- 6 耿素云,屈婉玲,王捍贫.离散数学教程.北京:北京大学出版社,2002.
- 7 耿素云,屈婉玲.离散数学.修订版.北京:高等教育出版社,2004.
- 8 尹宝林,何自强,许光汉,等.离散数学.第3版.北京:高等教育出版社,2011.
- 9 王义和.离散数学引论.第3版.哈尔滨:哈尔滨工业大学出版社,2007.
- 10 Knuth DE. 苏运霖.译.计算机程序设计艺术(第1卷 基本算法).第3版.北京:国防工业出版社,2002.
- 11 Weiss MA. 冯舜玺.译.数据结构与算法分析——C语言描述.北京:机械工业出版社,2004.
- 12 Horowitz E, Sahni S, Anderson-Freed S. 朱仲涛.译.数据结构基础.第2版.北京:清华大学出版社,2009.
- 13 Sahni S. 汪诗林,孙晓东,等.译.数据结构算法与应用——C++语言描述.北京:机械工业出版社,2005.
- 14 张铭,赵海燕,等.数据结构与算法实验教程.北京:高等教育出版社,2011.
- 15 左飞.数据结构原理与经典问题求解.北京:电子工业出版社,2008.
- 16 李春葆,尹为民,等.数据结构教程.第2版.北京:清华大学出版社,2007.
- 17 朱成立.数据结构——使用C语言.第4版.北京:电子工业出版社,2009.
- 18 严蔚敏,吴伟民.数据结构(C语言版).北京:清华大学出版社,1997.
- 19 廖明宏,郭福顺,等.数据结构与算法.第4版.北京:高等教育出版社,2007.
- 20 张乃孝.算法与数据结构——C语言描述.第2版.北京:高等教育出版社,2006.
- 21 陈媛,何波,卢玲,等.算法与数据结构.第2版.北京:清华大学出版社,2011.
- 22 张华,刘乃琦,郭建东.基于孩子兄弟树的FAT32文件删除恢复算法.计算机应用研究,2009,26(3):1116-1118.
- 23 皇甫永伟.LM-Suffix:基于后缀树的基因序列索引结构研究[学位论文].昆明:云南大学,2014.
- 24 邬伦,刘瑜,等.地理信息系统——原理、方法和应用.北京:科学出版社,2001.
- 25 刘南,刘仁义.地理信息系统.北京:高等教育出版社,2002.
- 26 胡鹏,黄杏元,华一新.地理信息系统教程.武汉:武汉大学出版社,2002.
- 27 张宏,温永宁,刘爱利,等.地理信息系统算法基础.北京:科学出版社,2006.
- 28 崔铁军.地理空间数据库原理.北京:科学出版社,2007.
- 29 刘义,景宁,陈萃,熊伟.MapReduce 框架下基于 R-树的 k-近邻连接算法.软件学报,2013,24(8):1836-1851.
- 30 刘义,陈萃,景宁,熊伟.基于 R-树索引的 Map-Reduce 空间连接聚集操作.国防科技大学学报,2013,35(1):136-141.
- 31 龚俊,柯胜男,朱庆,张叶廷.一种集成 R 树、哈希表和 B\* 树的高效轨迹数据索引方法.测绘学报,2015,44(5):570-577.
- 32 何婧,吴跃,杨帆,尹春雷,周维.基于 KD 树和 R 树的多维云数据索引.计算机应用,2014,34(11):3218-3221,3278.
- 33 王世杰,艾明耀,马捷,严俊.基于 KD 树的 LiDAR 点云索引方法及其在公路勘测中的应用.地理空间信息,2015,13(1):140-142.
- 34 黄志.基于多级格网与 STR 树的混合索引研究[硕士学位论文].杭州:浙江大学,2013.
- 35 惠子怡.GR~\*树空间索引方法研究[硕士学位论文].北京:中国地质大学(北京),2014.
- 36 文思,文贵华.基于扩展前缀编码的左孩子右兄弟结构连接算法.计算机工程与设计,2010,31(10):2312-2315.
- 37 袁关伟,赵家刚.基于递归算法在孩子兄弟法表示的树中查找任一结点的双亲结点算法的设计与实现.现代计算机(专业版),2011(19):3-6.
- 38 张友生.系统分析师教程.第2版.北京:清华大学出版社,2010.