

链表排序程序设计的算法解析

唐嵩明

摘要 本文介绍了链表排序程序设计的3种算法:(1)链表简单排序法。(2)链表选择排序法。(3)链表指针插入排序法。3种排序方法的时间复杂度都是 $O(n^2)$ 。如果链表节点内容很多,3种排序算法中运行时间最节省是链表指针插入排序法,它只交换节点地址,没有交换节点内容。

关键词 链表排序 程序设计 算法解析

在日常数据处理中对数据的排序经常碰到,大多数采用数组或结构体数组数据结构进行排序,对链表排序程序设计介绍较少,而用链表处理数据以其承受机开辟数据个数、节省

内存空间和节省运行时间的优点,被软件开发者大量采用。既然链表处理数据有其独特优点,故对链表排序算法有必要进行研究。下面对链表排序程序设计用 Turbo C 语言解析了3种算法。

一、链表简单排序法^{[1],[3]}

假定链表节点只5个,每个节点有3个成员,学号、总分和指向下1个节点的指针,要求链表按总分由高到低进行排序。见图1。

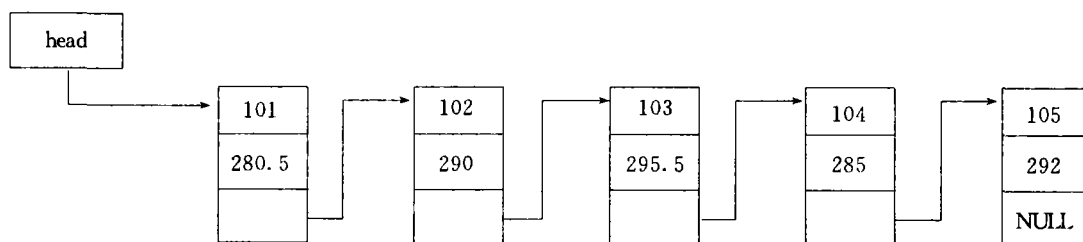


图1 链表结构示意图

步骤1:当前节点指针从第1节点开始,将链表当前节点的总分与其后面节点的总分相比较,如果当前节点总分低,就将后面总分高的节点内容(学号,总分)与之对换,地址不变。然后,当前节点指针指向下1个节点,继续进行比较,一直比较到链表结束。第一趟(共4次比较),已将总分最大值调到第1个节点。

步骤2:当前节点指针从第2节点开始,再进行步骤1类似的比较与交换,经第二趟(共3次比较),第2节点已存放次高的总分值。

步骤3:如此比较下去,可以推知,5个节点要比较4趟。第一趟共进行两两比较4次,在第二次趟只要比较3次,——第四趟比较1次。如n个节点,则要进行n-1趟比较。

实例:输入学生的学号、总分(学号为0时输入结束),然后按总分由大到小排序输出。算法见图3链表简单排序法N-S结构流程图

```
#include "stdlib.h"
#define NULL 0
#define LEN sizeof (struct student)
struct student
{ long num; /* 学号 */
```

```
float total; /* 总分值 */
struct student next; /* 指向下1个节点指针 */
};
int n; /* 存放学生人数的全局变量 */
struct student creat() /* 创建新链表函数,输入学号、总分(学号为0时输入结束) */
{ struct student * head;
  struct student * p1, * p2;
  char ss[20];
  n = -1;
  head = 0;
  do
  { if (n == -1)
    p2 = p1 = (struct student *) malloc(LEN); /* n = -1 就开辟链表头地址 */
    else
    { p2 = p1;
      p1 = (struct student *) malloc(LEN); } /* n <> -1 就开辟新的节点地址 */
    printf("学号 总分 (学号为0时输入结束)\n");
```

```

scanf("%ld",&p1->num);
gets(ss);
p1->total=atof(ss); /* 通过 atof()函数将字符串
ss 转换成单精度值 */
}
n=n+1; /* 累计链表节点个数 */
if (n==0) head=p1; /* 如 n=0 就是链表头,将第 1 节点
地址赋给链表头 */
else p2->next=p1; /* 否则将新节点地址赋给前 1 个节
点 next 地址 */
}while (p1->num!=0); /* 当学号为 0 输入结束 */
p2->next=NULL; /* 将 NULL 赋给链尾 */
return(head); /* 返回链表头地址 */
}

void prin(head) /* 打印链表函数 */
struct student * head;
{ struct student * p;
printf("\nNow, These %d records are:\n",n);
printf("学号 总分\n");
p=head; /* p 指针指向链表头 */ while (p!=NULL)
/* 如 p 指针不为空,就循环打印,直到 p 指针为空退出
*/
{ printf("%-8ld %6.1f\n",p->num,p->total);/*
打印学号、总分值 */
p=p->next; /* 将指向 1 个节点地址赋给 p 指针 */
}
}

struct student * sort1(head) /* 链表简单排序函数 */
struct student * head;
{ struct student * p1, * p2, * temp;
int num;
float total;
temp=(struct student *)malloc(LEN); /* 开辟临时存放
1 个节点地址 */
for(p1=head;p1!=NULL;p1=p1->next) /* 第一层循
环,从链表第 1 个节点开始到链表尾 */
for(p2=p1->next;p2!=NULL;p2=p2->next)/* 第
二层循环,从链表第 2 个节点开始,直到链表尾 */
if(p1->total<p2->total) /* 若总分少于后面的总分则
交换两个节点内容,地址不变 */
{ temp->num=p1->num; /* 把 p1 节点的内容(学号,
总分)赋给 temp 临时节点 */
temp->total=p1->total;
p1->num=p2->num; /* 把 p2 节点的内容(学号,总
分)赋给 p1 节点 */
p1->total=p2->total;

```

```

p2->num=temp->num; /* 把 temp 临时节点的内容
(学号,总分)赋给 p2 节点 */
p2->total=temp->total;
}
return(head); /* 返回链表头地址 */
}

main()
{ struct student * head;
head=creat(); /* 调用创建链表函数 */
head=sort1(head);/* 调用链表排序函数 */
prin(head); /* 调用链表打印函数 */
}

```

二、链表选择排序法^{[2],[3]}

假定链表节点只 5 个,每个节点有 3 个成员,学号、总分和指向下 1 个节点的指针.要求链表按总分由高到低进行选择法排序。

步骤 1:把第 1 节点的地址赋给临时指针 pmax,总分赋给临时变量 ktemp.当前指针从链表的第 2 节点开始,将 ktemp 与当前节点的总分相比较,如大于 ktemp 的总分值就将地址 p2 赋给临时节点指针 pmax,总分值赋给临时变量 ktemp.然后,当前节点指针指向下 1 个节点,继续进行比较,直到链尾.总分最大值即为 ktemp 的值,地址即 pmax.若链表第 1 节点地址不等于指针 pmax,就将 pmax 所指节点内容(学号,总分)与 p1 所指节点内容对换.第一趟,共进行 4 次比较,每次比较记住总分最大值的节点地址与其最大值.第一趟比较结束后,将总分最大值调到第 1 个节点。

步骤 2:把第 2 节点的地址赋给临时指针 pmax,总分赋给临时变量 ktemp.当前指针从链表的第 3 节点开始,进行步骤 1 类似的比较与交换.第二趟(共进行 3 次比较)结束后,已将总分次最大值调到第 2 个节点。

步骤 3:如此比较下去,方法如上至链表尾.如 n 个节点,则要进行 n-1 趟比较.与链表简单排序法比较,优点减少节点内容交换次数。

```

struct student * sort2(head) /* 链表选择排序法函数 */
struct student * head;
{ struct student * p1, * p2, * temp, * pmax; /* temp 当
前存放临时指针节点 */
float ktemp; /* * pmax 是最大值节点地址 */
temp=(struct student *)malloc(LEN); /* 开辟临时存
放 1 个节点地址 */
for (p1=head;p1!=NULL;p1=p1->next) /* 第一
层循环,从链表第 1 个节点开始至链尾 */
{ pmax=p1;
ktemp=p1->total; /* 先将当前节点地址 p1 存放

```

```

pmax 中,总分值存放 ktemp * /
for (p2=p1->next;p2!=NULL;p1=p1->next) /*
第二层循环,从链表第 2 个节点开始直到链表尾,在未排序
的链表中查找 * /
if (ktemp<p2->total) /* 如当前节点总分大于
ktemp, * /
{ pmax=p2; /* 将当前最大值节点地址存放 pmax 中 * /
ktemp=p2->total; /* 将当前总分最大值赋给 ktemp
临时变量 * /
} /* 反复进行循环找到该趟最大值节点 * /
if (p1!=pmax) /* 如果当前 p1 节点地址不等于最大值
pmax 节点,则交换两节点内容,地址不变。 * /
{ temp->num=pmax->num; /* 把 pmax 节点中的
内容赋给 temp 节点 * /
temp->total=pmax->total;
pmax->num=p1->num; /* 把 p1 节点中的内容
赋给 pmax 节点 * /
pmax->total=p1->total;
p1->num=temp->num; /* 把 temp 节点中的内容
赋给 p1 节点 * /
p1->total=temp->total;
}
return(head); /* 返回链表头 * /

```

三、链表指针插入排序法^{[1],[3]}

假定链表节点只 5 个,每个节点有 3 个成员,学号、总分和指向下 1 个节点的指针,要求链表按总分由高到低进行指针插入法排序。

在排序时把链表分成两部分:一部分是已排序好的部分,这部分范围从链表第 1 个节点到指针 phead 所指向的节点;第二部分是未排序的链表,这部分范围从指针 p4 所指向的节点到链表结束,即 p4 总是指向未排序链表的第 1 个节点。另外,让指针 p1 总是指向已排序链表中总分比 p4 大的节点; p2 指向已排序中总分比 p4 小的节点。

先假设第 2 个节点为 p4,然后根据 p4 是否为空来判断排序是否结束。将 p2 指向已排序链表头, p2 节点总分与未排序节点 p4 总分比较,如总分大于新节点 p4 总分,将 p1, p2 前移 1 个节点。直至链表中某节点总分少于 p4 节点总分,则循环停止。 p1, p2 指针记住已排序的链表节点中比 p4 总分大的节点和比 p4 总分小的节点位置,把 p4 插入到它们中间,即插入法。 p4 又指向未排序第 1 个节点,再重复以上操作。见图 2。(图中实线表示初始指针位置,虚线表示经过一次排序后的指针位置)。

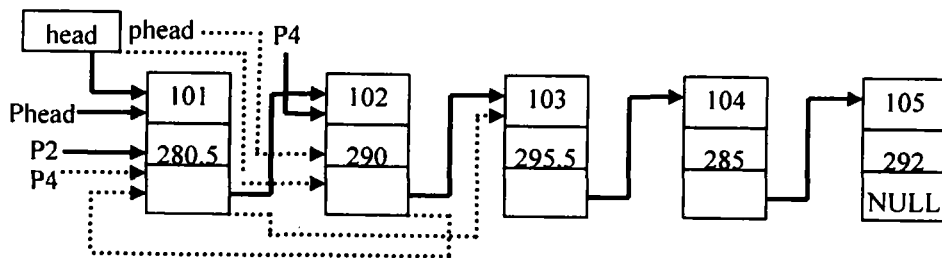


图 2 链表指针插入排序法结构示意图

```

struct student *sort3(head) /* 链表指针插入排序法函数
* /
struct student * head
{ struct student * p1, * p2, * phead=head, * p4=phead->
next;
while (p4!=NULL) /* p4 为空表示排序结束,不为空则
进行排序 * /
{
p2=head; /* 从链表头开始比较 * /
while (p2->total>p4->total) /* 将已排序链表 p2 从
头到尾与新节点 p4 总分比较,如总分大于 p4,将 p1, p2 前移
1 个节点,再反复比较。如已排序链表中有节点总分小于 p4
总分则循环停止 * /
{p1=p2;p2=p2->next;}

```

```

if (p2!=p4) /* 如果 p4 所指节点不是 p2 所指节点 * /
{phead->next=p4->next; /* 重新定义未排序的头节
点 * /
if (p2==head)
{ p4->next=p2; head=p4; } /* 如果 p2 是头节点,将 p4
作为头结点 * /
else {p1->next=p4; p4->next=p2;} /* 插入 p4 节点
* /
}
phead=p4; p4=p4->next; /* 未排序链表头节点下移
1 个,把下 1 个节点加入排序 * /
return(head);
}

```

(下转第 64 页)

```
Pause">
</object>
</body>
</html>
```

读者如果直接把上面的 HTML 文件放在服务器的 Web 发布目录下, 远程 IE 用户可以通过在 IE 浏览器地址栏输入发布 Web 服务器的 IP 地址或域名访问该页面, 但是看到的画面仅仅包含 NetMeeting 视频窗口, 因为我们简化了一些代码。读者可以自行增加 JavaScript 或 VBScript 脚本, 安排若干图形来表示按钮, 然后通过单击事件激活 CallToBtn—onclick() 函数实现呼叫, 通话完毕后通过激活 onHangUp() 函数挂机。

这里我们对上述代码的主要部分作一些解释。实际上, 用户开发包含 NetMeeting 组件页面的工作量已经很小了, 通过在 <object> </object> 标识中使用 NetMeeting ActiveX 控件的 CLSID, 就实现了对 NetMeeting 组件的调用。CLSID 后跟着的是 NetMeeting ActiveX 组件的 ID 号, 它是一个全局唯一的注册号, 可直接在 Windows 的注册表中查找到。如果读

(上接第 59 页)

参考文献

- [1] 刘尊全著, 计算机病毒防范与信息对抗技术, 1991 年
- [2] 阿部英志著, マイコンピュ-タ, 1986. (20)
- [3] 姜山等, Windows 9X 硬件中断设备驱动程序的开发, 微型电脑的应用, 2000. 16(1)

(上接第 62 页)

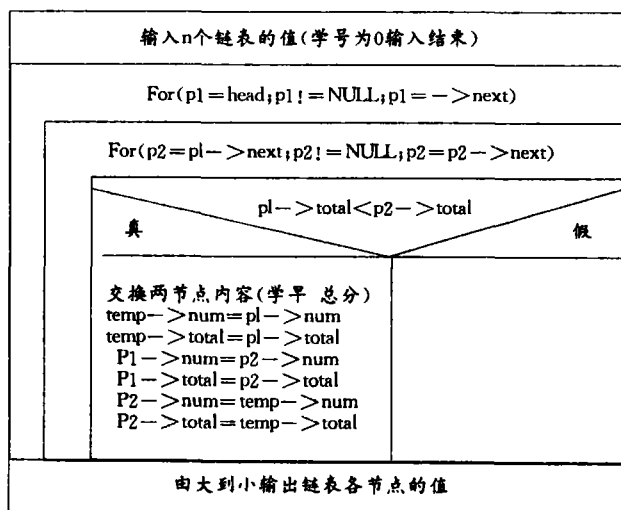


图3 链表简单排序N-S结构流程图

小结: 用链表简单排序法, 算法比较直观简便, 且容易实现, 缺点是节点内容交换太频繁。当待排序节点数量很小时, 是一种很简便的排序方法。而链表选择排序法, 每趟比较, 只

者希望深入研究完全代码, 可参考微软 NetMeeting 开发文档^[2]。

总之, NetMeeting 组件的使用非常方便, 这些组件是作为 Windows 9X/NT/2000/XP 操作系统的内嵌部分在安装时生成的。因此任何上述系统的用户都可以使用 NetMeeting 组件。当然, 在硬件方面用户还要有点投资, 也就是购买一个摄像头和一块视频捕获卡, 或者直接买一个一体化的 USB 接口的摄像头。Web 模式的个人会议系统适合于机动用户对固定用户的访问, 因此在远程教学、远程咨询中能够发挥一定的作用。

参考文献

- [1] 张明德、王永东, 视频会议系统原理与应用, 北京希望电子出版社, 北京, 1999
- [2] <http://www.microsoft.com>, Windows NetMeeting 3 SDK, April 1999

(收稿日期: 2002 年 4 月 15 日)

- [4] 尚涛等, PAID 系统的结构化模型, 微型电脑应用, 2001. 17 (1)
- [5] 中岛信行, 未公开 DOS コール活用术, インタフエス別冊付録, July, 1991
- [6] 李沐荪编译, MS-DOS 操作系统高等教程—汇编语言和 C 语言程序手册, 北京科海总公司培训中心, 1987 年 9 月。

(收稿日期: 2002 年 5 月 26 日)

记住总分最大值的节点地址与最大值, 每趟比较后, 才将未排序链表的当前头节点与查找到总分最大值的节点内容交换。与链表简单排序法比较, 减少节点内容交换次数。而链表指针插入排序法, 算法较复杂。三种方法的时间复杂度都是 $O(n^2)$, 但在节点内容很多情况下, 链表指针插入排序法运行时间最节省, 因为它只交换节点地址, 没有交换节点内容。

以上程序在 PII 微机上 Turbo C2.0 集成开发环境下调试运行正确。

参考文献

- [1] 谭浩强编著, C 程序设计, 北京: 清华大学出版社, 1999 12 月 [M]
- [2] 严桂兰、黄思先等编著, C 语言程序设计与应用教程, 厦门: 厦门大学出版社, 2001 年 8 月 [M]
- [3] 付清祥、王晓东编著, 算法与数据结构, 北京: 电子工业出版社, 1998 年 1 月 [M]

(收稿日期: 2002 年 4 月 18 日)

Design and Implementation of Unicom Geographic Information System P46
 Lin Qiaomin Zhuang Weihua Xu Xiaofeng Wei Feng (Department of Computer Science, HeHai University Nanjing 210098)

Abstract This paper introduces a solution to facility management in the area of telecommunication with GIS technology, focusing on its goal, system functions, three-tier structure, data management and user-defined controls.

Keywords geographic information system facility management three tiermodel data management customized control

A Dynamic QoS Management Mechanism for Real-time Multimedia Communication P49
 Fan Chengzhi Wu qingrun Zhang Wenyuan (Department of Automation, Shanghai Jiaotong University Shanghai 200030)
 Song Ning (Department of Electronic Engineering, Shanghai Jiaotong University Shanghai 200030)

Abstract Real-time multimedia communication demands high level of QoS. This paper proposes a management mechanism of dynamic QoS control based on an analytical comparison among current QoS control methods.

Keywords multimedia real-time communication quality of service

A Multiple-tier Distributed System Based on Client/Server P51
 Zhuang Weihua Li Tingting Li Xiaofang (School of Computer Science, Hehai University Nanjing 210098)

Abstract This paper introduces design and implementation of multiple-tier distributed systems based on the Client/Server mode. It illustrates the database design, data distribution, processing methods and data security with a sample of Inhabitant Property Information Management System of the Flood Area in Jiangsu Province.

Keywords property registration database distributed management information data security

An Online Learning System of E-Commerce Simulation Based on Browser/Server P54
 Mao Li (Wuxi Campus, Donghua University Wuxi 214063) Xie Wei (Shanghai Telecom Co. Shanghai 200040) Gao Chuanshan (Department of Computer Science and Engineering, Fudan University Shanghai 200433)

Abstract This paper introduces DHU-EC, an online learning system of E-Commerce simulation. It virtualizes the whole process of foreign trade in the B2B mode of the real world for the students.

Keywords EC B/S mode ASP simulation system for learning

Learners' Garden

How to Recover a "Frozen" System Using "Cloning"? P57
 Wang Yu Wu Chunli Fan Hongwei Ma Yuquan (School of Mechanical Engineering, Shandong University Jinan 250061)

Abstract People usually pay more attention to avoiding catastrophic failures of the computer. But this paper focuses on data restoration in case of damaged storage media. The "cloning" technology is used to recover a system frozen by viruses.

Keywords cloning recovery boot record boot sector

On Chained List Sorting Algorithms P60
 Tang Aiming (Department of Technology and Engineering, Minjiang College Fuzhou 350002)

Abstract This paper introduces three chained list sorting algorithms: direct sort, selection sort and pointer insertion sort. The time complexity of all of them are $O(n^2)$

Keywords chained list sort program design algorithm analysis

A Web-based Personal Conferencing System Based on Microsoft NetMeeting P63
 Zhou Qiang Zhao Luna Luo Yan Huang Dongjun (Test Centre, Zhongnan University Changsha 410083)

Abstract This paper introduces a Web-based personal conferencing system based on Microsoft NetMeeting, which is suitable for remote applications such as online learning and consulting. A sample Webpage containing codes in JavaScript and VBScript is given for illustration.

Keywords NetMeeting component Web model personal conferencing system realization

Address: Room 1504, Floor 15, Bao Zhaolong Library, Shanghai Jiaotong University

1954 Huashan Road, Shanghai 200030, The People's Republic of China

Tel: 86-21-62933230

Fax: 86-21-62933230

Email: smcaa@online.sh.cn

URL: <http://www.smcaa.online.sh.cn> IP: 202.96.210.198

Publisher: Shanghai Microcomputer Application Association

Code Number: 6329M

Distributor: China International Book Trading Corporation (P. O. Box 399, Beijing)