



“十二五”普通高等教育本科国家级规划教材

高等学校计算机专业核心课

名师精品·系列教材

数据结构

(C语言版 第3版)

双色版 | 附微课视频

李冬梅 严蔚敏 吴伟民 编著

DATA STRUCTURES IN C
(3RD EDITION)

符合最新统考考研大纲要求，紧跟最新技术发展
配套立体化教学资源，支持线上线下混合式教学
案例驱动，算法清晰，例题、习题类型丰富



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

数据结构 : C 语言版 : 双色版 / 李冬梅, 严蔚敏, 吴伟民编著. — 3 版. — 北京 : 人民邮电出版社, 2024. — (高等学校计算机专业核心课名师精品系列教材). — ISBN 978-7-115-65125-9

I. TP311.12; TP312.8

中国国家版本馆 CIP 数据核字第 2024AN0707 号

内 容 提 要

本书在选材与编排上, 贴近当前普通高等院校“数据结构”课程的现状和发展趋势, 符合最新的全国硕士研究生统一入学考试计算机科学与技术学科数据结构考试大纲, 内容难度适中, 突出实用性和应用性。全书共 8 章, 内容包括绪论, 线性表, 栈和队列, 串、数组和广义表, 树和二叉树, 图, 查找和排序。全书采用类 C 语言作为数据结构和算法的描述语言。

本书可作为普通高等院校计算机和信息技术相关专业“数据结构”课程的教材, 也可供从事计算机工程与应用工作的科技工作者参考。

-
- ◆ 编 著 李冬梅 严蔚敏 吴伟民
责任编辑 刘 博
责任印制 陈 犇
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <https://www.ptpress.com.cn>
大厂回族自治县聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21.25 2024 年 8 月第 3 版
字数: 560 千字 2025 年 6 月河北第 4 次印刷
-

定价: 59.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315



第 1 章 绪论 1

1.1 数据结构的研究内容	1
1.2 数据结构的基本概念和术语	3
1.2.1 数据、数据元素、数据项和数据对象	3
1.2.2 数据结构	4
1.2.3 数据类型和抽象数据类型	6
1.3 抽象数据类型的表示与实现	7
1.4 算法和算法分析	10
1.4.1 算法的定义及特性	11
1.4.2 评价算法优劣的基本标准	11
1.4.3 算法的时间复杂度	11
1.4.4 算法的空间复杂度	15
1.5 小结	16
习题	17

第 2 章 线性表 19

2.1 线性表的定义和特点	19
2.2 案例引入	20
2.3 线性表的类型定义	22
2.4 线性表的顺序表示和实现	23
2.4.1 线性表的顺序表示	23
2.4.2 顺序表中基本操作的实现	25
2.5 线性表的链式表示和实现	29
2.5.1 单链表的定义和表示	29
2.5.2 单链表基本操作的实现	32
2.5.3 循环链表	38

2.5.4 双向链表	38
------------	----

2.6 顺序表和链表的比较 40

2.6.1 空间性能的比较	40
---------------	----

2.6.2 时间性能的比较	40
---------------	----

2.7 线性表的应用 41

2.7.1 线性表的合并	41
--------------	----

2.7.2 有序表的合并	42
--------------	----

2.8 案例分析与实现 44

2.9 LeetCode 算法练习题 49

2.10 小结 58

习题 59

第 3 章 栈和队列 62

3.1 栈和队列的定义和特点 62

3.1.1 栈的定义和特点	62
---------------	----

3.1.2 队列的定义和特点	63
----------------	----

3.2 案例引入 63

3.3 栈的表示和实现 65

3.3.1 栈的类型定义	65
--------------	----

3.3.2 顺序栈的表示和实现	66
-----------------	----

3.3.3 链栈的表示和实现	68
----------------	----

3.4 栈与递归 70

3.4.1 采用递归算法解决的问题	70
-------------------	----

3.4.2 递归过程与递归工作栈	74
------------------	----

3.4.3 递归算法的效率分析	76
-----------------	----

3.4.4 利用栈将递归转换为非递归的方法	76
-----------------------	----

3.5 队列的表示和实现	77
3.5.1 队列的类型定义	77
3.5.2 顺序队的表示和实现	78
3.5.3 链队的表示和实现	81
3.6 案例分析与实现	83
3.7 LeetCode 算法练习题	90
3.8 小结	101
习题	102

第4章 串、数组和广义表 105

4.1 串的定义	105
4.2 案例引入	106
4.3 串的类型定义、存储结构及其运算	107
4.3.1 串的抽象类型定义	107
4.3.2 串的存储结构	108
4.3.3 串的模式匹配算法	109
4.4 数组	118
4.4.1 数组的类型定义	118
4.4.2 数组的顺序存储	119
4.4.3 特殊矩阵的压缩存储	120
4.5 广义表	122
4.5.1 广义表的定义	122
4.5.2 广义表的存储结构	123
4.6 案例分析与实现	125
4.7 LeetCode 算法练习题	126
4.8 小结	134
习题	134

第5章 树和二叉树 137

5.1 树和二叉树的定义	137
5.1.1 树的定义	137
5.1.2 树的基本术语	138
5.1.3 二叉树的定义	139
5.2 案例引入	140
5.3 树和二叉树的抽象数据类型定义	141
5.4 二叉树的性质和存储结构	144
5.4.1 二叉树的性质	144
5.4.2 二叉树的存储结构	146
5.5 遍历二叉树和线索二叉树	147
5.5.1 遍历二叉树	147
5.5.2 线索二叉树	154
5.6 树和森林	158
5.6.1 树的存储结构	158
5.6.2 森林与二叉树的转换	160
5.6.3 树和森林的遍历	161
5.7 哈夫曼树及其应用	162
5.7.1 哈夫曼树的基本概念	162
5.7.2 哈夫曼树的构造算法	163
5.7.3 哈夫曼编码	165
5.8 并查集及其应用	168
5.8.1 并查集的基本概念	168
5.8.2 并查集的实现及优化	169
5.9 案例分析与实现	173
5.10 LeetCode 算法练习题	176
5.11 小结	180
习题	181



第6章 图 184

6.1 图的定义和基本术语 184

6.1.1 图的定义 184

6.1.2 图的基本术语 185

6.2 案例引入 186

6.3 图的类型定义 187

6.4 图的存储结构 188

6.4.1 邻接矩阵 189

6.4.2 邻接表 191

6.4.3 十字链表 193

6.4.4 邻接多重表 194

6.5 图的遍历 196

6.5.1 深度优先搜索 196

6.5.2 广度优先搜索 198

6.6 图的应用 200

6.6.1 最小生成树 200

6.6.2 最短路径 205

6.6.3 拓扑排序 211

6.6.4 关键路径 214

6.7 案例分析与实现 219

6.8 LeetCode 算法练习题 221

6.9 小结 225

习题 227

第7章 查找 230

7.1 查找的基本概念 230

7.2 线性表的查找 231

7.2.1 顺序查找 231

7.2.2 折半查找 232

7.2.3 分块查找 235

7.3 树表的查找 237

7.3.1 二叉排序树 237

7.3.2 平衡二叉树 244

7.3.3 红黑树 249

7.3.4 B- 树 259

7.3.5 B+ 树 267

7.4 散列表的查找 268

7.4.1 散列表的基本概念 268

7.4.2 散列函数的构造方法 270

7.4.3 处理冲突的方法 272

7.4.4 散列表的查找 274

7.5 LeetCode 算法题 277

7.6 小结 284

习题 286

第8章 排序 289

8.1 基本概念和排序方法概述 289

8.1.1 排序的基本概念 289

8.1.2 内部排序方法的分类 290

8.1.3 待排序记录的存储方式 291

8.1.4 排序算法效率的评价指标 291

8.2 插入排序 291

8.2.1 直接插入排序 292

8.2.2 折半插入排序 293

8.2.3 希尔排序 294

8.3 交换排序 296

8.3.1 冒泡排序	296	8.7 外部排序	314
8.3.2 快速排序	298	8.7.1 外部排序的基本方法	315
8.4 选择排序	301	8.7.2 多路平衡归并的实现	316
8.4.1 简单选择排序	301	8.7.3 置换 - 选择排序	317
8.4.2 树形选择排序	303	8.7.4 最佳归并树	321
8.4.3 堆排序	304	8.8 LeetCode 算法题	322
8.5 归并排序	309	8.9 小结	326
8.6 基数排序	311	习题	328
8.6.1 多关键字的排序	311		
8.6.2 链式基数排序	311	参考文献	330

```

for(i=dk+1;i<=L.length;++i)
    if(L.r[i].key<L.r[i-dk].key)           //需将L.r[i]插入有序增量子表
    {
        L.r[0]=L.r[i];                     //暂存在r[0]中
        for(j=i-dk; j>0&& L.r[0].key<L.r[j].key;j-=dk)
            L.r[j+dk]=L.r[j];               //记录后移,直到找到插入位置
        L.r[j+dk]=L.r[0];                   //将r[0]即原r[i],插入正确位置
    }                                        //if
}

void ShellSort(SqList &L,int dt[],int t)
{ //按增量序列dt[0..t-1]对顺序表L进行t趟希尔排序
    for(k=0;k<t;++k)
        ShellInsert(L,dt[k]);               //一趟增量为dt[t]的希尔插入排序
}

```

【算法分析】

(1) 时间复杂度

当增量大于1时,关键字较小的记录就不是一步一步地挪动,而是跳跃式地移动,从而使得在进行最后一趟增量为1的插入排序时,序列已基本有序,只要对记录进行少量比较和移动即可完成排序,因此希尔排序的时间复杂度较直接插入排序的低。但要具体分析,则是一个复杂的问题,因为希尔排序的时间复杂度是所取“增量”序列的函数,这涉及一些数学上尚未解决的难题。因此,到目前为止尚未有人求得一种最好的增量序列,但大量的研究已得出一些局部的结论。如有人指出,当增量序列为 $dt[k]=2^{t-k+1}-1$ 时,希尔排序的时间复杂度为 $O(n^{3/2})$,其中 t 为排序趟数, $1 \leq k \leq t \leq \lfloor \log_2(n+1) \rfloor$ 。还有人在大量的实验基础上推出:当 n 在某个特定范围内,希尔排序所需的比较和移动次数约为 $n^{1.3}$,当 $n \rightarrow \infty$ 时,比较和移动次数可减少到 $n(\log_2 n)^2$ 。

(2) 空间复杂度

从空间来看,希尔排序和前面两种排序方法一样,也只需要一个辅助空间 $r[0]$,空间复杂度为 $O(1)$ 。

【算法特点】

- (1) 记录跳跃式地移动导致排序方法是不稳定的。
- (2) 只能用于顺序结构,不能用于链式结构。
- (3) 增量序列可以有各种取法,但应该使增量序列中的值没有除1之外的公因子,并且最后一个增量值必须等于1。
- (4) 记录总的比较次数和移动次数都比直接插入排序的要少, n 越大时,效果越明显。所以适合初始记录无序、 n 较大时的情况。

8.3 交换排序

交换排序的基本思想是:两两比较待排序记录的关键字,一旦发现两个记录不满足次序要求时则进行交换,直到整个序列全部满足要求为止。本节首先介绍基于简单交换思想实现的冒泡排序,然后给出另一种在此基础上进行改进的排序方法——快速排序。

8.3.1 冒泡排序

冒泡排序(Bubble Sort)是一种最简单的交换排序方法,它通过两两比较相邻记录的关键

字, 如果为逆序, 则进行交换, 从而使关键字小的记录如气泡一般逐渐往上“漂浮”(左移), 或者使关键字大的记录如石块一样逐渐向下“坠落”(右移)。

算法 8.4 冒泡排序

【算法步骤】

① 设待排序的记录存放在数组 $r[1 \cdots n]$ 中。首先将第一个记录的关键字和第二个记录的关键字进行比较, 若为逆序 (即 $L.r[1].key > L.r[2].key$), 则交换两个记录。然后比较第二个记录和第三个记录的关键字。依次类推, 直至第 $n-1$ 个记录和第 n 个记录的关键字进行过比较为止。上述过程称作第一趟起泡排序, 其结果使得关键字最大的记录被安置到最后一个记录的位置上。

② 然后进行第二趟起泡排序, 对前 $n-1$ 个记录进行同样的操作, 其结果是使关键字次大的记录被安置到第 $n-1$ 个记录的位置上。

③ 重复上述比较和交换过程, 第 i 趟是从 $L.r[1]$ 到 $L.r[n-i+1]$ 依次比较相邻两个记录的关键字, 并在“逆序”时交换相邻记录, 其结果是这 $n-i+1$ 个记录中关键字最大的记录被交换到第 $n-i+1$ 的位置上。直到在某一趟排序过程中没有进行过交换记录的操作, 说明序列已全部达到排序要求, 则完成排序。

【例 8.3】已知待排序记录的关键字序列为 $\{49, 38, 65, 97, 76, 13, 27, \overline{49}\}$, 请给出用冒泡排序法进行排序的过程。

冒泡排序过程如图 8.3 所示, 算法如算法 8.4 所示。

49	38	38	38	38	13	13
38	49	49	49	13	27	27
65	65	65	13	27	38	38
97	76	13	27	49	49	
76	13	27	$\overline{49}$	$\overline{49}$		
13	27	$\overline{49}$	65			
27	$\overline{49}$	76				
$\overline{49}$	97					
初始关键字	第一趟排序后	第二趟排序后	第三趟排序后	第四趟排序后	第五趟排序后	第六趟排序后

图 8.3 冒泡排序过程

待排序的记录总共有 8 个, 但算法在第六趟排序过程中没有进行过交换记录的操作, 则完成排序。

【算法描述】

```
void BubbleSort(SqList &L)
{
    // 对顺序表 L 进行冒泡排序
    m=L.length-1; flag=1;
    while((m>0) && (flag==1))
    {
        flag=0;
        for(j=1; j<=m; j++)
            if(L.r[j].key>L.r[j+1].key)
            {
                flag=1;
                t=L.r[j]; L.r[j]=L.r[j+1]; L.r[j+1]=t; // 交换前后两个记录
            }
        // flag 置为 0, 如果本趟排序没有发生交换, 则不会执行下一趟排序
        // flag 用来标记某一趟排序是否发生交换
        // if
    }
}
```

```

--m;
} //while
} //BubbleSort

```

【算法分析】

(1) 时间复杂度

最好情况(初始序列为正序): 只需进行一趟排序, 在排序过程中进行 $n-1$ 次关键字间的比较, 且不移动记录。

最坏情况(初始序列为逆序): 需进行 $n-1$ 趟排序, 总的关键字比较次数 KCN 和记录移动次数 RMN (每次交换都要移动 3 次记录) 分别为:

$$KCN = \sum_{i=n}^2 (i-1) = n(n-1)/2 \approx n^2/2$$

$$RMN = 3 \sum_{i=n}^2 (i-1) = 3n(n-1)/2 \approx 3n^2/2$$

所以, 在平均情况下, 冒泡排序关键字的比较次数和记录移动次数分别约为 $n^2/4$ 和 $3n^2/4$, 时间复杂度为 $O(n^2)$ 。

(2) 空间复杂度

冒泡排序只有在两个记录交换位置时需要一个辅助空间用于暂存记录, 所以空间复杂度为 $O(1)$ 。

【算法特点】

(1) 稳定排序。

(2) 可用于链式存储结构。

(3) 移动记录次数较多, 算法的平均时间性能比直接插入排序的差。当初始记录无序、 n 较大时, 此算法不宜采用。



冒泡排序

8.3.2 快速排序

快速排序 (Quick Sort) 是由冒泡排序改进而得的。在冒泡排序过程中, 只对相邻的两个记录进行比较, 因此每次交换两个相邻记录时只能消除一个逆序排列。如果能通过两个 (不相邻) 记录的一次交换, 消除多个逆序排列, 则会大大加快排序的速度。快速排序方法中的一次交换可能消除多个逆序排列。

算法 8.5 快速排序

【算法步骤】

在待排序的 n 个记录中任取一个记录 (通常取第一个记录) 作为枢轴 (或支点), 设其关键字为 $pivotkey$ 。经过一趟排序后, 把所有关键字小于 $pivotkey$ 的记录交换到前面, 把所有关键字大于 $pivotkey$ 的记录交换到后面, 结果将待排序记录分成两个子表, 最后将枢轴放置在分界处的位置。然后, 分别对左、右子表重复上述过程, 直至每一子表只有一个记录时, 排序完成。

其中, 一趟快速排序的具体步骤如下。

① 选择待排序表中的第一个记录作为枢轴, 将枢轴记录暂存在 $r[0]$ 的位置上。附设两个指针 low 和 $high$, 初始时分别指向表的下界和上界 (第一趟时, $low = 1$; $high = L.length$)。

② 从表的最右侧位置依次向左搜索, 找到第一个关键字小于枢轴关键字 $pivotkey$ 的记录, 将其移到 low 处。具体操作是: 当 $low < high$ 时, 若 $high$ 所指记录的关键字大于等于 $pivotkey$, 则向左移动指针 $high$ (执行操作 $--high$), 否则将 $high$ 所指记录与枢轴记录交换。

本 / 书 / 特 / 色

本书在选材与编排上, 贴近当前普通高等院校“数据结构”课程的现状和发展趋势, 符合最新的全国硕士研究生统一入学考试计算机科学与技术学科数据结构考试大纲, 内容难度适中, 突出实用性和应用性。全书共 8 章, 内容包括绪论, 线性表, 栈和队列, 串、数组和广义表, 树和二叉树, 图, 查找和排序。全书采用类 C 语言作为数据结构和算法的描述语言。

本书可作为普通高等院校计算机和信息技术相关专业“数据结构”课程的教材, 也可供从事计算机工程与应用工作的科技工作者参考。



本教材的结构框图



用书教师扫码下载
本书配套资源



教材服务热线: 010-81055256

反馈/投稿/推荐信箱: 315@ptpress.com.cn

人邮教育服务与资源下载社区: www.rjiaoyu.com

6037345



扫码获得资源

ISBN 978-7-115-65125-9



9 787115 651259

定价: 59.80元

封面设计: 董志桢