



教育部高职高专计算机教指委规划教材

项目实训型

数据结构与算法

SHUJU JIEGOU YU SUANFA

主 编/田 晶 金 鑫



中国人民大学出版社

教育部高职高专计算机教指委规划教材

数据结构与算法

主 编 田 晶 金 鑫
副主编 陈显刚 陈 锐 王 品
主 审 姜惠民

中国人民大学出版社

• 北京 •

图书在版编目 (CIP) 数据

数据结构与算法/田晶等主编. —北京: 中国人民大学出版社, 2011. 5
教育部高职高专计算机教指委规划教材
ISBN 978-7-300-13434-5

I. ①数… II. ①田… III. ①数据结构 ②算法分析 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2011) 第 032184 号

教育部高职高专计算机教指委规划教材
数据结构与算法

主 编 田 晶 金 鑫
副主编 陈显刚 陈 锐 王 品
主 审 姜惠民

出版发行	中国人民大学出版社	
社 址	北京中关村大街 31 号	邮政编码 100080
电 话	010-62511242 (总编室)	010-62511398 (质管部)
	010-82501766 (邮购部)	010-62514148 (门市部)
	010-62515195 (发行公司)	010-62515275 (盗版举报)
网 址	http://www.crup.com.cn http://www.ttrnet.com (人大教研网)	
经 销	新华书店	
印 刷	北京昌联印刷有限公司	
规 格	185 mm×260 mm 16 开本	版 次 2011 年 7 月第 1 版
印 张	17	印 次 2011 年 7 月第 1 次印刷
字 数	420 000	定 价 29.00 元

中国人民大学出版社
教育部高职高专计算机教指委规划教材
编委会委员名单

主任委员 温 涛

委 员 (按姓名拼音排序)

崔连和	褚建立	高爱国	姜广坤	靳广斌
李 红	李明革	李万龙	李永平	连为民
刘昌明	邵冬华	申亚宁	时武略	宋维堂
王 伟	王跃胜	吴建宁	向 隅	肖 玉
谢伟红	许文宪	严仲兴	杨淑清	易著梁
尹敬齐	张 辉	章忠宪	赵晓玲	甄立常
周 奇				

秘 书 董本清 孙 琳



总 序

近年来,我国高等教育取得了跨越式发展,毛入学率由1998年的8%迅速增长到2010年的25%,已经进入到大众的发展阶段,这其中,高等职业教育对实现“形成全民学习、终身学习的学习型社会”、“构建终身教育体系”的宏伟目标,发挥着其他教育形式不可替代的作用。

质量是职业教育的生命,社会需求是职业教育发展的终极动力。新颁布的《国家中长期教育改革和发展规划纲要》特别强调通过推进教育教学改革来提高质量。《纲要》要求通过课程、教材、教学模式和评价方式的创新,推进就业创业教育,实现人才培养方式转变,着力提高学生的职业道德、职业技能和就业创业能力。

实际上,为了适应我国高等职业教育的发展,全面提高教育教学质量,教育部主管部门先后启动了“国家精品课程建设”和“国家示范性高等职业院校建设计划”,经过四年的建设,无论是办学条件、人才培养模式,还是学生的就业质量都取得了显著进步;同时,也涌现出了一批高水平的优秀课程和优秀教材,为传播优秀教学理念、教学方法和教学内容起到了重要作用,为提高教学质量奠定了坚实基础。

为进一步深化教育教学改革和精品课程建设,进一步挖掘优秀的课程和教材,推广优秀的教育成果,扩大精品课程的受益面,在教育部高等学校高职高专计算机类专业教学指导委员会的指导下,中国人民大学出版社组织召开了计算机类专业的教材研讨会,并成立了教材编审委员会,计划在未来两三年内陆续推出百种高职高专计算机系列精品教材。

此套教材的作者大都是有着丰富的职业教育教学经验和较高专业学术水平的专家和教

授。教材内容的选择克服了追求理论“大而全”的不足，做到了少而精，有针对性，突出了能力的训练和培养；教材体例的安排突出了学习的弹性和灵活性，形成文字教材和多媒体教程相结合的立体化教材，加强了教师对学生学习过程的指导和帮助，形象生动、灵活方便，更能适应学员在职、业余自学，或配合教师讲授时使用，相信会起到很好的教学效果。为满足教师在实际教学中的需求，本套教材在编写体例形式上不拘一格，具备“任务引领型”、“案例型”、“项目实训型”等写作特点，其目的是让学生在学中练、练中学，在实际动手练习中掌握理论知识和专业技能。

我们期待，这套高职高专计算机精品教材能够为促进我国高校 IT 职业教育的教学质量做出积极的贡献；我们也相信，这套教材必将在实践中日臻完善、追求卓越！

教育部高等学校高职高专计算机类专业教学指导委员会 主任委员

大连东软信息学院院长 温涛教授

二〇一〇年六月



前言

“数据结构”是一门讨论非数值性程序设计的学科，它为设计系统软件和应用软件提供必要的基础知识和方法。因此，它不仅是大专院校计算机专业和计算机应用专业的一门主要课程，也是工程界众多从事计算机应用工作的、非计算机专业技术人员自修的一门课程。从课程性质上讲，“数据结构”是一门专业技术基础课。

算法是程序设计的灵魂，所谓算法设计就是将解决问题所需要的操作步骤用合适的方法表达出来。

本书是数据结构与算法的有机结合，全面论述了数据结构和算法的内容，从一定深度上涵盖了算法的诸多方面，同时其讲授和分析方法又兼顾了各个层次读者的接受能力。各章内容自成体系，可作为独立单元学习。

考虑到多数院校的前期课程是C语言及C程序，故全书采用类C语言编写的伪代码作为数据结构和算法的描述语言。本书可作为高职高专院校计算机类专业的教材，也可以作为信息类相关专业的选修教材，建议开设学期为第二学期，96学时。

编写本书时，通过对多年从事软件开发、设计的企业人员的调研，分析软件设计工作任务、提取典型工作任务，进而分析得出软件企业常用的数据结构及算法，结合多年的讲义，由经验丰富的教师编写而成。

考虑到读者层次，编者在语言组织上尽量简单明了，除明确学生必须掌握的知识点外，更强调利用算法解决实际问题的思想；在内容组织上，有针对性地增加了以下几项内容：

- ◇ 教学重点——将本章重点内容提炼成关键字，简单明了，使读者印象深刻。
- ◇ 项目实训——综合本章重点算法，设计一个程序，深入理解算法含义。

◇ 专项练习

- ✓ 基本题——精心选取典型试题，有针对地进行练习，增加自主学习能力；
- ✓ 国家计算机等级考试试题——历年国家二级考试真题；
- ✓ 专升本试题——全国各省招收专升本考生（包括自学考试）院校的历年真题。

除此之外，本书有配套的习题答案和源程序文件，可登录 <http://www.crup.com.cn/jiaoyu> 获取。

本书由长春职业技术学院的田晶、金鑫任主编，陈显刚、陈锐、王品任副主编；齐志、丁磊、佟彤、尹茜茜、信建杰、丛路卫、于艳波、曹丽英（吉林农业大学）、刘孝国（吉林农业大学）、赵晓丹（吉林省经济管理干部学院）、任凤娟、乔丹参与了本书的编写工作。姜惠民担任本书的主审工作，给予了指导性的建议。同时，长春职业技术学院嵌入式系统工程教研室主任宋海兰对本书的编写也提出了许多宝贵的意见，在此一并表示感谢！

由于时间仓促以及编者水平有限，本书肯定会有不少缺点和不足，热切期望得到专家和读者的批评指正。

编 者

2011年4月

目 录

第1章 绪论	1	3.3 栈的链表存储结构	48
1.1 什么是数据结构	1	3.3.1 栈的定义	48
1.1.1 数据结构的定义	1	3.3.2 顺序栈的基本运算实现	48
1.1.2 逻辑结构及存储结构 类型	2	3.4 应用实例的实现	50
1.1.3 基本概念和术语	3	3.4.1 表达式中括号配对的合法性 检查	50
1.2 抽象数据类型的表示与实现	3	3.4.2 表达式求值	50
1.3 算法和算法分析	3	3.4.3 递归调用	52
1.3.1 什么是算法	3	3.5 队列的结构及其运算	56
1.3.2 算法描述	4	3.5.1 队列的基本操作	56
1.3.3 算法分析	5	3.5.2 链队的存储结构及其运算	61
专项练习1	6	3.6 栈的应用举例	62
第2章 线性表	9	3.7 栈与队列比较	64
2.1 线性表的基本概念	9	项目实训	64
2.2 线性表的顺序存储——顺序表	10	专项练习3	71
2.2.1 定义顺序表	10	第4章 串	75
2.2.2 顺序表上的运算及其实现	11	4.1 串的基本概念	75
2.3 线性表的链式存储——单链表	14	4.1.1 串的定长顺序存储及运算 实现	76
2.3.1 线性表的链式存储 ——链表	14	4.1.2 串的堆式动态存储及运算 实现	80
2.3.2 单链表的定义	15	4.1.3 串的块链存储表示	83
2.3.3 单链表上的基本运算及其 实现	15	4.2 串的模式匹配算法	84
2.4 线性表的链式存储二——双链表	20	4.2.1 串的简单模式匹配算法	84
2.5 循环链表	22	4.2.2 一种改进的模式匹配算法	86
项目实训	22	项目实训	90
专项练习2	41	专项练习4	94
第3章 栈和队列	45	第5章 数组和广义表	98
3.1 栈的定义及基本运算	45	5.1 数组的定义	98
3.2 顺序栈的存储结构及算法实现	46	5.2 数组的顺序表示和实现	99
3.2.1 顺序栈	46	5.3 矩阵的压缩存储	99
3.2.2 顺序栈的基本运算实现	46	5.3.1 特殊矩阵	100

5.3.2 稀疏矩阵	100	7.1.1 图的定义	140
5.4 广义表的定义	104	7.1.2 图的相关术语	141
5.5 广义表的存储结构	105	7.2 图的存储结构	144
项目实训	106	7.2.1 邻接矩阵	144
专项练习 5	109	7.2.2 邻接表和逆邻接表	146
第 6 章 树和二叉树	115	7.3 图的遍历	148
6.1 树的定义和基本术语	115	7.3.1 深度优先搜索	148
6.1.1 树的定义	115	7.3.2 广度优先搜索	150
6.1.2 树的表示	115	7.4 图的应用	151
6.1.3 树的基本术语	117	7.4.1 最小生成树	151
6.1.4 树的术语	117	7.4.2 最短路径	156
6.1.5 树的基本运算	117	7.4.3 拓扑排序	159
6.1.6 树的存储结构	118	项目实训	161
6.2 二叉树	119	专项练习 7	173
6.2.1 二叉树的定义	119	第 8 章 查找	177
6.2.2 二叉树的重要性质	119	8.1 静态查找表	177
6.2.3 二叉树与树、森林之间的 转换	121	8.1.1 顺序表的查找	177
6.3 二叉树的存储结构	122	8.1.2 有序表的查找	178
6.3.1 二叉树的顺序存储结构	122	8.1.3 索引顺序表的查找	180
6.3.2 二叉树的链式存储结构	122	8.2 哈希表	181
6.4 二叉树的遍历和线索二叉树	123	8.2.1 什么是哈希表	181
6.4.1 二叉树遍历的概念	123	8.2.2 哈希函数的构造方法	182
6.4.2 二叉树遍历算法	123	8.2.3 处理冲突的方法	182
6.4.3 二叉树线索的概念	124	项目实训	183
6.4.4 线索化二叉树	125	专项练习 8	186
6.5 二叉树的基本运算及其实现	127	第 9 章 排序	192
6.5.1 二叉树基本运算概述	127	9.1 基本概念	192
6.5.2 二叉树基本运算的算法 实现	127	9.2 插入排序	193
6.6 哈夫曼树及其应用	130	9.2.1 直接插入排序	193
6.6.1 哈夫曼树的概念	130	9.2.2 希尔排序	195
6.6.2 构造哈夫曼树	130	9.3 交换排序	196
6.6.3 哈夫曼编码	131	9.3.1 冒泡排序	196
项目实训	131	9.3.2 快速排序	197
专项练习 6	135	9.4 选择排序	200
第 7 章 图及图的应用	140	9.4.1 简单选择排序	200
7.1 图的基本概念	140	9.4.2 堆排序	201
		9.5 归并排序	207
		9.5.1 归并排序的过程	207

9.5.2 归并排序的算法	207	10.2 顺序文件	221
9.5.3 算法分析	209	10.3 索引文件	222
9.6 各种内部排序的比较	209	10.3.1 索引文件概述	222
项目实训	209	10.3.2 ISAM 文件	223
专项练习 9	215	10.3.3 VSAM 文件	224
第 10 章 文件	220	10.4 哈希文件	224
10.1 文件的基本概念	220	项目实训	225
10.1.1 什么是文件	220	专项练习 10	230
10.1.2 文件的逻辑结构及操作	221	习题答案	233
10.1.3 文件的存储结构	221	参考文献	257

第1章 绪论



教学重点

- 基本概念：数据、数据结构、数据元素、数据对象、数据项、算法；
- 数据结构的分类；
- 抽象数据类型描述；
- 算法的特点及算法分析。

在计算机科学中，数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象（数据元素）以及它们之间的关系和运算等的学科，而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。

“数据结构”作为一门独立的课程在国外是从 1968 年才开始设立的。1968 年美国唐·欧·克努特教授开创了数据结构的最初体系，他所著的《计算机程序设计技巧》第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。

“数据结构”在计算机学科中是一门综合性的专业基础课。数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。数据结构这门课的内容不仅是一般程序设计（特别是非数值性程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序的重要基础。

1.1 什么是数据结构

1.1.1 数据结构的定义

数据结构是计算机存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。通常情况下，精心选择的数据结构可以带来更高的运行或者存储效率。数据结构往往同高效的检索算法和索引技术有关。

数据结构在计算机科学界至今没有标准的定义，个人根据各自理解的不同而有不同的表述：

Sartaj Sahni 在他的《数据结构、算法与应用》一书中称：“数据结构是数据对象，以及存在于该对象的实例和组成实例的数据元素之间的各种联系。这些联系可以通过定义相关的函数来给出。”他将数据对象（Data Object）定义为“一个数据对象是实例或值的集合”。

Clifford A. Shaffer 在《数据结构与算法分析》一书中的定义是：“数据结构是 ADT（抽

象数据类型, Abstract Data Type) 的物理实现。”

Robert L. Kruse 在《数据结构与程序设计》一书中, 将一个数据结构的设计过程分成抽象层、数据结构层和实现层。其中, 抽象层是指抽象数据类型层, 它讨论数据的逻辑结构及其运算, 数据结构层和实现层讨论一个数据结构的表示和在计算机内的存储细节以及运算的实现。

数据结构包括如下三个方面:

(1) 数据元素之间的逻辑关系, 即数据的逻辑结构。

(2) 数据元素及其关系在计算机存储器中的存储方式, 即数据的存储结构, 也称为数据的物理结构。

(3) 施加在该数据上的操作, 即数据的运算。

1.1.2 逻辑结构及存储结构类型

1. 逻辑结构类型

(1) 集合结构。

(2) 线性结构。结点之间关系: 一对一。

特点: 开始结点和终端结点都是唯一的, 除了开始结点和终端结点以外, 其余结点都有且仅有一个前驱结点, 有且仅有一个后继结点。顺序表就是典型的线性结构, 如图 1.1 所示。

例如: 点名册上顺序排列的学生名单等。



图 1.1 线性表的顺序结构

(3) 树形结构。结点之间关系: 一对多。

特点: 开始结点唯一, 终端结点不唯一。除终端结点以外, 每个结点有一个或多个后续结点; 除开始结点外, 每个结点有且仅有一个前驱结点, 如图 1.2 所示。

例如: 各单位的组织结构图、族谱等。

(4) 网状结构。结点之间关系: 多对多。

特点: 没有开始结点和终端结点, 所有结点都可能多个前驱结点和多个后继结点, 如图 1.3 所示。

常用来研究生产流程、施工计划、各种网络建设等问题。

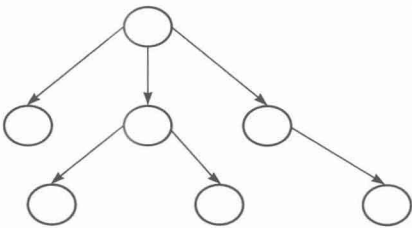


图 1.2 线性表的树形结构

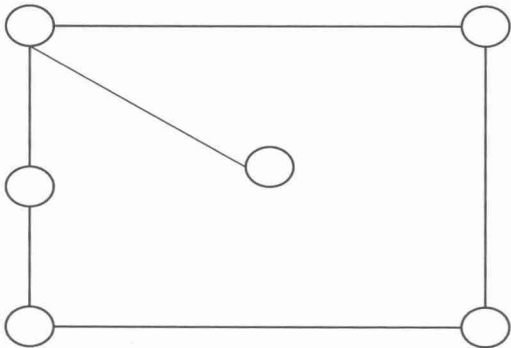


图 1.3 线性表的网状结构

2. 存储结构类型

- (1) 顺序存储方法。
- (2) 链式存储方法。
- (3) 索引存储方法。
- (4) 散列存储方法。

1.1.3 基本概念和术语

数据：是所有能被输入到计算机中，且能被计算机处理的符号的集合。它是计算机操作的对象

的数据的总称，也是计算机处理的信息的某种特定的符号表示形式。

数据元素：是数据（集合）中的一个“个体”，是数据的基本单位。

数据对象：是具有相同性质的若干个数据元素的集合。

数据项：一个数据元素由若干个数据项组成。数据项是数据的不可分割的最小单位。

1.2 抽象数据类型的表示与实现

抽象数据类型指的是用户进行软件系统设计时从问题的数学模型中抽象出来的逻辑数据结构和逻辑数据结构上的运算，而不考虑计算机的具体存储结构和运算的具体实现算法。

抽象数据类型 = 数据元素集合 + 抽象运算

例如，抽象数据类型复数的定义：

```
ADT Complex
{
    数据对象(即上面所说的数据元素集合):
    D = {e1, e2 | e1, e2 均为实数}
    数据关系:
    R1 = {< e1, e2 > | e1 是复数的实数部分, e2 是复数的虚数部分 }
    基本操作(即上面所说的抽象运算):
        AssignComplex(&Z, v1, v2): 构造复数 Z。
        DestroyComplex(&Z): 复数 Z 被销毁。
        GetReal(Z, &real): 返回复数 Z 的实部值。
        GetImag(Z, &Imag): 返回复数 Z 的虚部值。
        Add(z1, z2, &sum): 返回两个复数 z1, z2 的和。
} ADT Complex
```

1.3 算法和算法分析

1.3.1 什么是算法

数据元素之间的关系有逻辑关系和物理关系，对应的操作有逻辑结构上的操作功能和具体存储结构上的操作实现。

通常把具体存储结构上的操作实现步骤或过程称为算法。

程序 = 算法 + 数据结构

算法的五个重要特性如下：

- (1) 有穷性：在有穷步之后结束。
- (2) 确定性：无二义性。

(3) 可行性：可通过基本运算有限次执行来实现。

(4) 有输入：可有零个或多个。

(5) 有输出：至少有一个输出。

例：考虑下列两段描述：

● 描述一

```
void exam1( )
{
    n = 2;
    while (n % 2 == 0)
        n = n + 2;
    printf(" %d\n", n);
}
```

● 描述二

```
void exam2()
{
    y = 0;
    x = 5/y;
    printf(" %d, %d\n", x, y);
}
```

这两段描述均不能满足算法的特征，试问它们违反了哪些特征？

解：(1) 描述一算法是一个死循环，违反了算法的有穷性特征。

(2) 描述二算法包含除零错误，违反了算法的可行性特征。

1.3.2 算法描述

本书采用 C/C++ 语言描述算法。

说明：C++ 语言中提供了一种引用运算符“&”，可以把引用理解成一个变量的别名，当建立引用时，程序用另一个已定义的变量或对象（目标）的名字初始化它，从那时起，引用作为目标的别名而使用，对引用的改动实际上就是对目标的改动。

注意：Turbo C 不支持引用类型。

编写一个函数 swap1(x, y)，当执行语句 swap1(a, b) 时，交换 a 和 b 的值。

```
void swap1(int x, int y)
{
    int temp;
    temp = x; x = y; y = temp;
}
```

注意：a 和 b 的值不会发生交换。

为此，采用指针的方式来回传形参的值，需将上述函数改为：

```
void swap2(int *x, int *y)
{
    int temp;
    temp = *x;          /* 将 x 的值放在 temp 中 */
    *x = *y;            /* 将 x 所指的值改为 *y */
    *y = temp;
}
```



```

    *y = temp;          /* 将 y 所指的值改为 temp */
}

```

上述函数的调用改为 swap2 (&a, &b), 显然远不如采用引用方式简洁。所以本书后面很多算法都采用引用形参。

例如:

```

int a = 6;          /* a 为普通的整型变量 */
int &b = a;          /* b 是 a 的引用变量 */

```

这里说明 b 变量是变量 a 的引用, b 也等于 6, 之后这两个变量同步改变。当 a 改变时 b 也同步改变, 同样, 当 b 改变时 a 也同步改变。

引用常用于函数形参中, 采用引用形参时, 在函数调用时将形参的改变回传给实参, 例如, 有如下函数 (其中的形参均为引用形参):

```

void swap(int &x, int &y)
/* 形参前的 "&" 符号不是指针运算符 */
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}

```

当用执行语句 swap(a, b) 时, a 和 b 的值发生了交换。

1.3.3 算法分析

1. 算法时间复杂度分析

一个算法是由控制结构 (顺序、分支和循环三种) 和原操作 (指固有数据类型的操作) 构成的, 算法的执行时间取决于两者的综合效果。

同一问题可以采用多种算法实现。由于算法的描述语言不同 (即所用的编程语言不同), 算法执行的环境不同 (操作系统不同或不同的计算机) 及其他因素, 所以不能用绝对执行时间单纯进行比较。

算法执行时间大致为基本运算所需的时间与其运算次数 (也称为频度) 的乘积。

被视为算法基本运算的一般是最深层循环内的语句。

在一个算法中, 进行基本运算的次数越少, 其运行时间也就相对越少; 基本运算次数越多, 其运行时间也就相对越多。

所以, 通常把算法中包含基本运算次数的多少称为算法的时间复杂度, 也就是说, 一个算法的时间复杂度是指该算法的基本运算次数。

算法中基本运算次数 $T(n)$ 是问题规模 n 的某个函数 $f(n)$, 记做:

$$T(n) = O(f(n))$$

记号 “O” 读做 “大 O”, 它表示随问题规模 n 的增大算法执行时间的增长率 and $f(n)$ 的增长率相同。“O” 的形式定义为:

若 $f(n)$ 是正整数 n 的一个函数, 则 $T(n) = O(f(n))$ 表示存在一个正的常数 M , 使得当 $n \geq n_0$ 时都满足:

$$|T(n)| \leq M|f(n)|$$

也就是只求出 $T(n)$ 的最高阶，忽略其低阶项和常系数，这样既可简化 $T(n)$ 的计算，又能比较客观地反映出当 n 很大时，算法的时间性能。

例如， $T(n) = 3n^2 - 5n + 10000 = O(n^2)$ / * 这里忽略了低阶项 $5n$ 和常数项 10000 */
一个没有循环的算法的基本运算次数与问题规模 n 无关，记做 $O(1)$ ，也称为常数阶。

一个只有一层循环的算法基本运算次数与问题规模 n 的增长呈线性增大关系，记做 $O(n)$ ，也称为线性阶。

其余常用的还有平方阶 $O(n^2)$ 、立方阶 $O(n^3)$ 、对数阶 $O(\log_2 n)$ 、指数阶 $O(2^n)$ 等。

各种不同数量级对应的值存在着如下关系：

$$O(1) < O(\log_2 n) < O(n) < O(n * \log_2 n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$$

2. 算法空间复杂度分析

类似于算法的时间复杂度，本书以空间复杂度作为算法所需存储空间的量度，记做：

$$S(n) = O(g(n))$$

其中 n 为问题的规模或大小。一个上机执行的程序除了需要存储空间来存储本身所用的指令、常数、变量和输入数据外，还需要一些对数据进行操作的地单元和存储一些为实现计算所需信息的辅助空间。若输入数据所占空间只取决于问题本身，与算法无关，则只需要分析除输入和程序之外的额外空间，否则应同时考虑输入本身所需空间（与输入数据的表示形式有关）。若额外空间相对于输入数据量来说是常数，则称此算法为原地工作。

总之，一个好的算法应同时兼顾时间和空间效率，高效但占用超大空间的算法是不可取的，同样，占用空间少但效率低的算法也不是一个优秀的算法。

专项练习 1

一、基本题

1. 填空题

- (1) 数据的逻辑结构有_____、_____、_____、_____四种。
- (2) 数据的存储结构分为_____、_____、_____、_____四种。
- (3) 数据结构中评价算法的两个重要指标是_____和_____。
- (4) 数据结构是研讨数据的_____和_____，以及它们之间的相互关系，并对与这种结构定义相应的_____，设计出相应的_____。
- (5) 线性结构元素之间是_____对_____的关系；树形结构元素之间的关系是_____；图形结构元素之间的关系是_____。

2. 选择题

- (1) 设有如下遗产继承规则：丈夫和妻子可以互相继承遗产；子女可以继承父亲或母亲的遗产；子女间不能相互继承。表示该遗产继承关系最合适的数据结构应该是()。
- A. 树 B. 图 C. 数组 D. 二叉树
- (2) 在数据结构中，从逻辑上可以把数据结构分成()。
- A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
- C. 线性结构和非线性结构 D. 内部结构和外部结构
- (3) 计算机算法是指()。
- A. 计算方法 B. 排序方法
- C. 解决问题的有限运算序列 D. 调度方法