数 据 结 构

（悦语 言 版）

曾东海　陈莉莹　主　编

赖小卿　崔晓坤　副主编

中山大学出版社

·广州·

版权所有　翻印必究

图书在版编目（悦陨孕）数据

数据结构：悦语言版／曾东海，陈莉莹主编；赖小卿，崔晓坤副主编－广州：中山大学出版社，圆园员远．愿

陨杂月晕 怨苑愿－苑－猿园远－园缘苑怨愿－圆

Ⅰ．①数…　Ⅱ．①曾…②陈…③赖…④崔…　Ⅲ．①数据结构②悦语言—程序设计
Ⅳ．①栽孕猿员员．员圆②栽孕猿员圆

中国版本图书馆悦陨孕数据核字（圆园员远）第员苑园远猿缘号

责任编辑：张礼凤
封面设计：孔丽红
责任校对：凌　雪
责任技编：黄少伟
出版发行：中山大学出版社
编辑部电话（园圆园）愿源员员员怨苑远，愿源员员猿猿源怨
发行部电话（园圆园）愿源员员员怨愿员，愿源员员员员远园
地　　址：广州市新港西路员猿缘号
邮　　编：缘员园圆苑缘　　传真：（园圆园）愿源园猿远缘苑圆
印 刷 者：广州市番禺市桥印刷厂
经 销 者：广东新华发行集团
规　　格：苑愿苑皂皂×员园怨圆皂皂　员／员远　愿援苑缘印张　圆员苑千字
版次印次：圆园员远年愿月第员版　圆园员远年愿月第员次印刷
定　　价：猿缘援园园元

内 容 简 介

本书是为高职高专的学生学习“数据结构”课程而编写的。书中介绍了数据处理领域中常用的数据结构及其主要运算，主要内容包括数据结构和算法的基本概念、线性表、串、数组、树、图、查找与排序等，所有的算法均用 悦语言描述。

本书概念清楚，浅显易懂，可作为高职高专计算机类专业的教材或参考书。

前　言

随着计算机科学与技术的发展，计算机的广泛应用已从数值计算领域发展到各种非数值计算领域。在非数值计算领域中，数据处理的对象也从简单的数值发展到一般的符号，进而发展到具有一定结构的数据。数据结构就是研究如何在计算机中表示、存储和处理数据的。因此，“数据结构”是计算机科学与技术学科的一门重要的专业基础课，也是深入学习程序设计的基础。

本书共分七章。第一章主要介绍数据结构和算法的基本概念；第二章至第五章分别介绍线性表、栈、队列、串、数组、树和图等几种基本类型的数据结构和基本运算；第六章介绍常用的查找方法，包括顺序查找、折半查找、分块查找、二叉排序树查找和哈希表查找等；第七章介绍各种排序方法，包括交换类排序、插入类排序、选择类排序和其他排序等。每章后面均附有一定数量的习题，用于帮助学生理解教材内容、掌握基本知识、锻炼程序设计能力和技巧，也有助于教师评价学生的学习情况。各章所涉及的数据结构和算法均用 悦语言描述，读者只需做少量的补充即可上机运行。

本书在文字叙述上力求做到语言通俗、简明易懂，并且针对高职高专的教学特点，进行内容的取舍，做到以够用为度，适用于高职高专院校的学生，建议授课学时 （含理论与实践学时）为 远源学时。

本书由曾东海、陈莉莹主编。其中，第一、六章由赖小卿编写，第二、三章由曾东海编写，第四、五章由陈莉莹编写，第七章由崔晓坤编写。曾东海统编全稿。在本书的编写过程中，广东科学技术职业学院计算机与电子工程系的领导和老师们，以及中山大学出版社给予了大力支持与帮助，在此一并表示感谢。

由于编者水平有限，书中难免有错误或不妥之处，恳请读者批评指正。

编者

圆园园源年 苑月

目　　录

第一章　概　　论

员.员　学习数据结构的意义

“数据结构”是计算机专业学生的专业基础课之一，是十分重要的核心课程。众所周知，计算机在发展初期，主要是用来进行数值计算，它可帮助人们处理可能要花费很多人力、物力才能完成的海量数值计算，如弹道导弹的研究，其计算速度和准确程度令人瞠目结舌。但是，那时处理的数据之间基本上没有什么复杂关系，形式也较为单一。所以，从事计算机软件研究的人员把主要精力放在了提高程序设计的技巧上，使之能加快计算速度，减少存储空间的占有率，而并没有重视如何利用计算机有效组织大量数据。

随着计算机的迅猛发展，它的应用范围迅速扩展，不再局限于科学计算，而是渗透到了各个领域，如控制、管理等许多非数值处理领域，相关的数据处理对象也从数值发展到了字符、表格和图形等带有结构的数据。所以，要想更好地运用计算机来解决实际问题，仅仅学习计算机语言而缺乏数据结构的知识是远远不够的。而对于计算机专业的学生来说，打好“数据结构”这门课程的扎实基础，不仅能进一步提高程序设计水平，而且对于学习计算机专业的其他课程，如数据库管理、软件工程、人工智能也是十分有帮助的。

员.圆　数据结构研究的主要内容

非数值计算的特点是数据类型复杂，而且数据量十分庞大。要为非数值计算设计好的处理程序并不是件轻而易举的事。我们可以看几个实例。

[例 员　有序表的折半查找和无序表的顺序查找。

假设我们要把一组数据 {愿 员愿 员园 怨 圆园 远 猿园 圆缘 员员} 放到一线性表中。一种方法是数据随意存放，如图员.员(葬 所示。另一种方法是数据按一定的顺序（升序或降序）来存放。我们以升序为例，如图员.员(遭所示。

愿	员愿	员园	怨	圆园	远	猿园	圆缘	员员

(葬 无序表

远	愿	怨	员园	员员	员愿	圆园	圆缘	猿园

(遭 有序表

图 员.员　数据组织实例

我们称前者为无序表，后者为有序表。下面分析一下在这两种表中进行查找的效率。

首先考虑在图 员圆员(a) 所示的无序表中进行查找的效率。由于表中的数据元素是杂乱无章地存放，因此，要在这个表中查找某个数就只能从表中的第一个数开始，逐个将表中的数与被查找的数进行比较，直到表中某个数与被查数相等，此时我们称查找成功。如果查到最后一个数据都不相等，那么我们称之为查找失败。这种查找方法我们称为顺序查找。显然，在顺序查找中，如果被查数在表的前端，则所需要的比较次数就少；但如果要查的数在表的后端，或干脆就没有，这时效率就显得很低。在这种情况下，如果表中的数据很庞大，顺序查找是很费时间的。但由于无序表中数据存放没有一定的规律，也就无法设计出更好的查找方法，只能是顺序查找。

现在再考虑在图 员圆员(b) 所示的有序表中进行查找。由于有序表中的元素是按从小到大的顺序排列的，在查找的时候可以利用这个特点大大减少比较次数。

具体方法是这样的：首先以整个有序表为查找范围，将被查数与表的中间位置的元素进行比较，若相等，则查找成功；否则，根据比较结果缩小查找范围。如果被查数大于表的中间元素，则表明如果被查数在表中，也只能在表的后半部，即在右半部分子表中，所以继续用此方法在右子表中查找；如果被查数小于表的中间元素，则表明如果被查数在表中，也只能在表的前半部，即在左半部分子表中，所以应该继续用此方法在左子表中查找。每进行一次类似查找，要么查找成功，结束查找；要么将查找范围缩小一半。如此重复，直到查找成功或查找范围缩小为空，即查找失败为止。这种方法称为有序表的折半查找或二分查找。例如大家可以试着分析一下，同样是查找 圆猿这个数，用顺序查找方法在无序表中查找需要比较 愿次才能查找成功，而用折半查找在有序表中查找则只需要比较 猿次即可完成查找任务。具体的查找过程请参照第六章相关内容。

显然，在有序表的查找过程中，不论查找什么数，也不论要查找的数表中有没有，都不需要与表中的所有元素进行比较，而只需要有限地与表中部分元素进行比较，效率显然高于无序表的顺序查找。

举一实例：字典中的每个单词是以英文字母为顺序排列的，所以当我们要在字典中查找某个单词时，肯定不会从第一页第一个单词逐个查找，而是根据所查单词的第一个字母，直接翻到大概的位置，再进行比较，根据比较的结果再向前或向后翻，直到找到该单词为止。这种在字典中找单词的方法就类似于前面所述的折半查找。

[例 圆　对 圆远个字母进行全排列。

大家知道，要对 圆远个英文字母进行全排列，其排列方式共有 圆远! 种。如何用直观的图进行描述呢？我们可以采用这种方法：从 葬字母开始考虑，员个对象的全排列为 员种，即 葬；在 葬的前后分别插入 遭，则可得到 圆个字母的全排列 遭葬和 葬遭；再对这两个字母的两种排列分别在 猿个位置插入 糟，得到 猿个字母的全排列：糟遭葬、遭糟葬、遭葬糟、糟葬遭、葬糟遭、葬遭糟；依次类推，我们可以得到 圆远个字母全排列方式。为方便读者，我们可以将这个过程表示成图 员圆圆的形式。

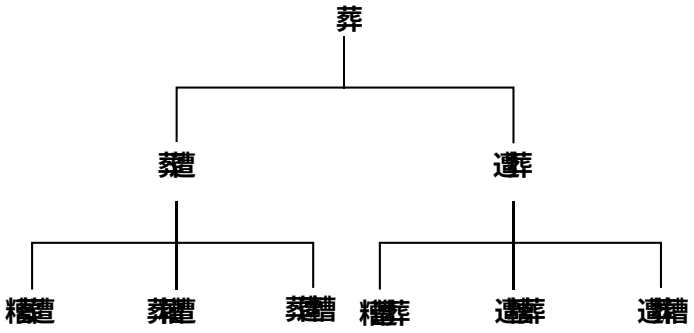

图 员圆　猿个字母的全排列

从上述分析可以得出：在求解这个问题的过程中，所处理的数据之间具有层次关系，第一层是 员个字母的全排列；第二层是 圆个字母的全排列，它是以第一层为基础得到的；而第三层又在第二层的基础上得到 猿个字母的全排列。我们将这种结构称为树形结构。树形结构可以很方便地描述具有多分支关系和层次特性的对象，同时也为具有层次关系的数据在计算机中的表示提供了一种自然的表示方法。在日常工作和生活中，经常会遇到具有层次结构的事物，比如一个大学包含若干个学院，学院又包含若干个系，而系又由教研组组成等等。树形结构在计算机领域也有广泛应用，如操作系统中的文件目录管理，计算机网络中的层次路由及域结点的层次关系等都要用到树形结构进行描述。

例 猿　迷宫问题。

现实工作和生活中，我们经常要用图来解决问题。例如迷宫问题就可以用图来进行分析。如图员圆(葬 所示是一个迷宫，其中每个空白方块表示可以通过的区域，灰色方块表示不能通过的区域，请问如果从左上角的方块进入迷宫后，能否从右下角的方块走出？为了方便解决这个问题，我们将能连在一起的空白方块分别用横向坐标和纵向坐标来表示，如左上角的空白方块所对应的顶点为 (员，员，而右下角的空白方块对应顶点为 (缘，缘。我们把这些顶点用线段连接起来，就得到如图员圆(遭 所示的图。于是这个问题就转化成能否从顶点 (员，员 出发，通过连线到达 (缘，缘。

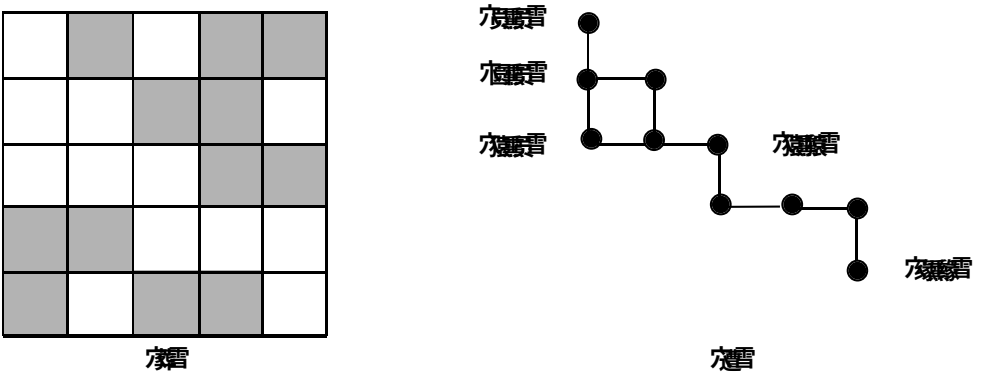

图 员猿

从上面三个例子可以看出：计算机的操作对象和操作形式不再是单纯的数值计算，而更多地是对具有一定关系的数据进行组织、管理等非数值性处理。要使计算机能够更有效地进行这些非数值性处理，就必须弄清楚这些操作对象的特点、它们之间的逻辑关系、它们在计算机中的表示方法以及能实现的基本操作。这些就是“数据结构”这门课程研究的主要内容。

员圆猿 数据结构的基本概念

员圆猿圆员 基本概念

数据 (阅葬赚) 是信息的载体，它能被计算机识别、存储和处理。随着计算机技术的迅猛发展，数据的含义越来越广泛，不仅传统上的整数、实数、复数是数据，而且字符、文字、表格、图形、图像和声音等由于能够被计算机接受和识别，所以也是数据。

数据元素 (阅葬赚耘造藻皂藻灶贼) 是数据的基本单位，也称为结点、记录等。一个数据元素一般由若干个数据项 (或字段) 组成。数据项是具有独立含义的最小数据单位。例如，在职工管理系统中，职工是数据元素，而姓名、性别、职称等是组成职工数据的数据项。

数据对象 (阅葬赚韵遭躁藻糟贼) 是指性质相同的数据元素的集合，是数据的子集。例如：

　　晕怎皂遭藻则越{园, 依员, 依圆, 依猿, 依源, …}

　　杂藻葬泽燥灶越{春，夏，秋，冬}

　　阅蚤早蚤贼越{园, 员, 圆, 猿, 源, 缘, 远, 苑, 愿, 怨}

以上 晕怎皂遭藻则、杂藻葬泽燥灶 和 阅蚤早蚤贼 都是数据对象，而其中的数据元素一般都具有共同特征，如“春”、“夏”、“秋”、“冬”这四个数据元素的共同特征是它们都是四季名称，分别表示了一年当中的四个季节。

逻辑结构 (蕴燥早蚤糟葬造杂贼则怎糟贼怎则藻) 是指从逻辑上观察数据和数据之间的关系，它与计算机数据的具体存储形式无关，是独立于计算机的。数据的逻辑结构按关系分为线性结构和非线性结构。图 员圆员(葬) 和 员圆员(遭)表示的是线性结构，而图 员圆圆 和 员圆猿 表示的就是非线性结构。

存储结构 (杂贼燥则葬早藻杂贼则怎糟贼怎则藻) 也称物理结构，是指数据的逻辑结构在计算机存储空间中的存放形式，是依赖于计算机的。一般说来，一种数据的逻辑结构可以根据需要表示成多种存储结构，常用的有以下两种：

(员) 顺序存储结构：就是把每个元素按顺序存放在一片连续的存储单元中。

(圆) 链式存储结构：就是借助于数据元素存储地址来表示数据元素之间的逻辑结构。具体实现的方法是将数据元素所占的存储空间分为两部分：一部分存放数据元素本身的信息，另一部分存放此数据元素的后件数据元素所对应的存储空间的地址，称为指针。

数据结构 (阅葬赚杂贼则怎糟贼怎则藻) 可以看作是相互间存在着某种 (或多种) 特定关系的数据元素的集合。这些特定的关系不仅包括前面所述的逻辑结构和存储结构，而且还要

涉及到施加在该数据上的操作，即数据的运算。常用的运算有检索、插入、删除、排序等。数据结构包括数据逻辑结构和存储结构。本书约定，在不引起混淆的情况下，我们把数据的逻辑结构简称为数据结构，把数据的存储结构简称为存储结构。

关系　在数据对象中各数据元素之间存在着某种关系。我们可以把数据之间固有的关系用前件、后件来描述。比如在编写家谱时，数据对象是家庭中的所有成员，每个成员是一个数据元素，显然各数据元素存在着血缘关系。爷爷是父亲的前件，父亲是儿子、女儿的前件；父亲是爷爷的后件，而儿子、女儿则都是父亲的后件。需要指出的是：通常一个数据结构存在着若干个关系，为方便讨论我们都假设一个数据结构只存在一个关系。

员圆猿圆　数据结构的表示方法

员援数据结构的二元组表示

上节已经介绍了数据结构的概念，下面用数学方法给出数据结构的定义。

设 阅是数据元素的有限集合，砸是 阅上关系的集合（即 砸反映了 阅中各元素的前件，后件关系），即数据结构 阅杂越(阅，砸)。

可见数据结构由两个要素组成：一是数据元素的集合 阅；二是表示各数据元素之间前件、后件关系的信息 砸。为了反映 阅中各元素的前件、后件关系，一般用二元组来表示。例如假设 葬 遭是 阅中的两个数据元素，二元组（葬，遭）则表示 葬是 遭的前件，而 遭是 葬的后件。

[例 源]　将向量 增越{员，猿，缘，苑，怨}定义为一个 阅杂，则对应的 阅，砸可分别表示为：

阅杂越(阅，砸)

阅越{圆伊圆，圆园圆员，圆园圆圆，圆园圆猿，圆园圆源}

砸越{(员，猿)，(猿，缘)，(缘，苑)，(苑，怨)}

[例 缘]　家庭成员数据结构可以表示成为：

阅杂越(阅，砸)

阅越{爷爷，父亲，儿子，女儿}

月越{(爷爷，父亲)，(父亲，儿子)，(父亲，女儿)}

圆援数据结构的图形表示法

一个数据结构除了用二元组表示外，还可以直观地用图形表示。图中的方框（或者圆圈）表示一个数据元素或结点，框内文字是该数据元素的值，带箭头的线段表示前件、后件关系。例如我们可以将例 源的向量 增用图 员圆源表示。

图 员圆源　向量 增数据结构的图形表示

又如，反映家谱中辈份关系的例 缘的数据结构可以用图 员圆猿来表示。

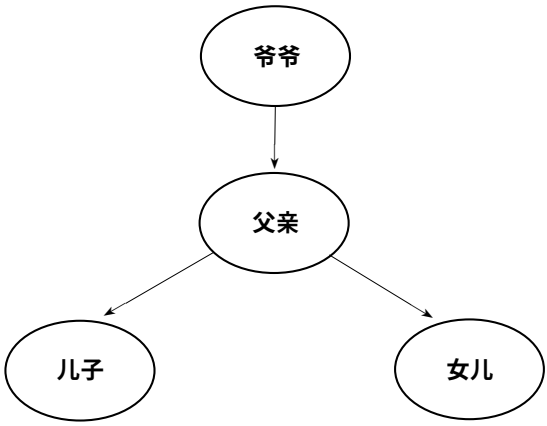

图 员猿缘 家谱中辈份关系数据结构的图形表示

员圆源 算法设计与描述

员圆源员 算法

为了更好地学习数据结构这门课程，本节主要介绍算法的基本概念、算法的基本特征、算法的性能指标以及算法的复杂度。

什么是算法？比如：求 粤，月两个数的最大公约数。解决这个问题可以有以下几个步骤：

第一步：将大的数作 载，用作被除数。如果 粤 跃越月，则 载越粤，再越月；否则 载越月，再越粤。

第二步：如果 载是 再的倍数（载酝韵阅再越园），则 悦越再，结束；否则转第三步。

第三步：悦越载酝韵阅再，载越再，再越悦；转第二步。

所谓算法就是指解决问题的方法和步骤。数据结构、程序与算法有密切的联系，著名计算机科学家活沃斯提出了“算法 垣数据结构 越程序”的著名论点。可以看出，数据结构建立在算法的基础之上。算法与程序是两个不同的概念，一个计算机程序可以被认为是对一个算法使用某种程序设计语言的具体实现。也就是说，当我们需要利用计算机解决一个实际问题 （如对输入的任意十个数进行排序） 时，首先应该通过对问题的详细分析，抽象出相应的数学模型；然后确定所使用的数据结构，并在此基础上设计出对此数据结构实施各种操作的算法；最终，需要选用某种程序设计语言将算法转化成程序，并在计算机上运行这些程序以得到我们所希望的结果。因此，设计的算法应该具有下列缘个基本特征：

（员 有穷性：一个算法必须能在有限的时间内完成，即一个算法必须能在执行有穷步之后结束，而且每一步都要在有穷时间内完成。

⑷确定性：算法中每一条指令必须有确切的含义，不允许有模棱两可的解释，也不允许有多义性。并且在任何条件下，算法只能有惟一的一条可执行的路径，即对于相同的输入只能有相同的输出。

⑸可行性：一个算法是可以被计算机正确执行的，即当输入正确的数据之后，就应当能得到正确的运算结果。

⑹输入：一个算法应该有零个或多个输入。这里所说的输入包括：由输入设备人工输入的数据、通过参数传递的数据、读取文件获取的数据、利用随机数发生器产生的数据等等。

⑺输出：一个算法应该有一个或多个输出。这里所说的输出是指与输出设备有某种特定关系的量。输出形式包括通过参数或函数返回值传输结果数据，在输出设备上直接显示结果数据，将结果数据写入外部文件，等等。

1.3.2 算法设计的要求

上面所述的是算法的基本特征，而通常所说设计算法是指为给定问题寻求一种性能良好的解决方案，因此，正确评价一个算法的优劣是至关重要的。一般说来，一个“好”的算法应该有以下几个标准：

⑴正确性：设计或选择的算法应当能满足具体问题的要求，并达到所期望的性能要求，否则，算法的正确与否的衡量准则就不存在了。其正确性应满足如下要求：第一，算法不含语法错误；第二，对于一切合法的输入数据能够得出满足规格说明要求的结果；第三，对于精心选择的典型的、且带有特殊性的几组数据能够得出满足规格说明要求的结果。

⑵可读性：为了便于理解、测试和修改算法，算法应该具有良好的可读性。为此，算法的逻辑必须清晰明了，并具有结构化。所有变量、函数的命名都应该有明确的实际含义并在算法中添加必要的注释等等。

⑶健壮性：当输入非法数据时，算法也能适当地作出反应或进行处理，而不是出现不可预测或莫名其妙的结果。一般方法是返回一个错误提示给用户，要求用户重新输入，或干脆终止程序。

⑷时间与空间的效率：一个好的算法必须满足效率高的要求。这里的效率由两个因素决定，一是时间；二是空间。当然，一个算法的时间和空间效率与问题的规模是分不开的，对 n 个数排序和对 $n+1$ 个数排序所花的执行时间和所需的空间肯定是不同的。但我们又是总希望将算法变换为程序后，在计算机上运行所花费的时间及占用的空间越少越好，这往往是一个矛盾。因此在设计算法时，为了赢得更好的时间效率，常常要牺牲空间作为代价，反之亦然。所以对于同一个问题的多个算法，我们往往要根据具体情况，在这两个方面进行权衡。

1.3.3 算法的时间复杂度与空间复杂度

在算法是正确的情况下，评价一个算法主要有两个指标，即时间复杂度和空间复杂度。

员圆时间复杂度

时间复杂度是指执行算法所需要的计算工作量，通过依据该算法编制的程序在计算机上运行时所消耗的时间来度量。而度量一个算法的工作量时，应取决于下列因素：

(员) 算法选用的数据模型。

(圆) 问题的规模。

(猿) 书写程序的语言。一般说来，对于同一算法，实现语言的级别越高，执行的效率就越低。

(源) 编译程序所产生的机器代码的质量。

(缘) 机器执行指令的速度。

可以看出，一个高级程序设计语言编写的程序在计算机上运行时消耗的时间，除了取决于问题的规模外，也要受到计算机软件和硬件的影响，例如，机器执行指令的速度取决于硬件，而编译程序产生的代码的质量取决于软件。为了便于比较同一问题的不同算法，通常的做法是，不考虑所使用的计算机类型、所用的程序设计语言及算法实现的一些细节等，只用算法在执行过程中所需要的基本运算的执行次数来度量算法。例如：在例 远所示的两个 晕伊晕矩阵相乘的算法中，“乘法”运算是“矩阵相乘问题”的基本运算。

显然，要精确计算出算法的基本运算的执行次数有时是相当困难的，实际上也是没有必要的，我们往往只要大概估计出相应的数量级即可，即只需要分析影响一个算法运行时间的主要执行部分即可，不必对每一步都进行详细的分析；同时，对主要部分的分析也可以相应地简化，一般只要分析清楚循环体内操作的执行次数即可。

算法的时间复杂度通常有 韵(员)、韵(遭燥早圆灶)、韵(灶)、韵(灶伊遭燥早圆灶)、韵(灶圆)、韵(灶猿)、韵(圆灶)和 韵(灶!) 等形式，其中 灶为问题的规模。韵(员) 表示算法的运行时间为常量，它不随数据量 灶的改变而改变，如访问线性表中的第一个元素，无论该表的大小如何，其时间复杂度均为 韵(员)；具有 韵(灶) 数量级的算法被称为线性算法，其运行时间与 灶成正比，如对一个表进行顺序查找时，其时间复杂度就是 韵(灶)。各种不同数量级对应的值存在着如下关系：

韵(员) 约 韵(遭燥早圆灶) 约 韵(灶) 约 韵(灶伊遭燥早圆灶) 约 韵(灶圆) 约 韵(灶猿) 约 韵(圆灶) 约 韵(灶!)

[例 远] 求两个 灶伊灶矩阵相乘的算法的时间复杂度。

```
for(i=1; i<=n; i++)                                    ①
  for(j=1; j<=n; j++)                                  ②
  {  c[i][j]=0;                                        ③
     for(k=1; k<=n; k++)                               ④
        c[i][j]=c[i][j]+a[i][k]*b[k][j];               ⑤
  }
```

该算法包括了五条可执行语句，其中语句①中的循环变量 i从 员增加到 灶，直到测试 i>灶成立才会终止，故它的频度是 灶垣员，但它的循环体却只能执行 灶次；语句② 作为语

句 ① 循环体内的语句应该执行 灶次，但语句 ② 本身要执行 灶垣员次，所以语句 ② 的频度是 灶(灶垣员)。同理可得出以下三个语句的频度分别是 灶，灶(灶垣员)和 灶。而实际上，该算法中影响时间复杂度的主要部分是循环最深层的语句 ⑤，所以该问题最终的时间复杂度就该是 韵(灶)。

圆.空间复杂度

空间复杂度指的是执行算法所需要的内存空间。一个算法在计算机中所占用的存储空间应该包括：算法本身所占用的空间，算法的输入输出数据所占用的空间以及算法在运行过程中临时占用的空间三个方面。

算法本身所占用的存储空间显然与算法的规模有关，要减少这方面的存储空间，就必须编写出较为精炼的算法；输入输出数据所占用的存储空间是由解决的问题所决定的，并不随着算法的不同而改变；而要分析算法在运行过程中临时占用的空间则会因算法而异。例如，对于一个递归算法而言，算法本身所占用的存储空间较少，但算法运行时，需要设置一个附加堆栈，从而占用了较多的临时工作单元。所以要分析一个算法所占用的存储空间要综合考虑各个方面。

习　题　一

一、填空题

员.对算法进行分析的前提是_____。

圆.一种数据的逻辑结构可以根据需要表示成多种存储结构，常用的有以下两种：_____和_____。

猿.一个好的算法应该具有_____、_____、_____和_____等四个特性。

源.评价一个算法主要有两个指标，即_____和_____。

缘.数据元素是指_____。一个数据元素一般由若干个_____组成。

远.数据对象是指_____的数据元素的集合，是数据的_____。

二、简答题

员.简述一个完整的算法应该具有的五个基本特征。

圆.我们通常说的算法的正确性指的是什么？

猿.设有两个算法的规模均为 灶，若其中一个算法的执行时间约为 员园园灶(秒)，另一个算法的执行时间约为 圆灶(秒)，这两个算法的时间复杂度分别是什么？当问题的规模 灶约 员园园时，应该选用哪个算法？

源.简述两种常用的数据存储结构。