

初识数据结构

学习目标

- 【知识目标】
1. 掌握数据结构的定义及其相关术语。

2. 理解学习数据结构的意义。

3. 理解数据结构在程序设计中的作用。

4. 了解算法的度量方法。
- 【能力目标】
1. 能够针对实际问题设计简单的算法。

2. 能够分析算法的时间复杂度，从而评估算法效能。
- 【素质目标】
1. 坚定“四个自信”。

2. 激发爱国热情，培养创新能力。

3. 培养学生发现问题、解决问题的能力。

4. 培养学生自主学习的能力。

学习效果

知 识 内 容		掌 握 程 度	存 在 疑 问
1. 什么是数据结构	数据结构的定义		
	数据结构的常用术语		
2. 数据结构的用途	学习数据结构的意义		
	数据结构的作用		
3. 数据结构与算法	算法的含义		
	算法的设计要求		
	算法的度量		

1.1 学籍档案管理系统的数据组织——什么是数据结构

勤 学 篇

1.1.1 案例说明

一个简单的学籍档案管理系统中需存储学生的学号、姓名、性别、出生日期、班级等信息，可以对信息进行增加、删除、修改或者查询操作，并且查询条件应该是多样化的。例如：按照班级查询、按照专业名称查询等。这些信息如何组织和存储才能最大限度地发挥系统优势，为用户提供最高效快捷的服务呢？请根据系统需求做好系统数据的组织和准备工作。

1.1.2 知识储备

1. 数据结构的定义

数据结构是在整个计算机科学与技术领域中被广泛使用的术语。它用来反映数据的内部构成，即数据由哪些成分构成，以什么方式构成，呈现什么样的结构。因此，可以将数据结构定义为相互之间存在一种或者多种特定关系的数据元素的集合。



数据结构的定义

数据的逻辑结构反映数据之间的逻辑关系，也就是数据的前驱和后继关系，这种关系与数据在计算机中的存储位置无关。逻辑结构包括：集合结构、线性结构、树形结构和图形结构，如图 1.1 所示。

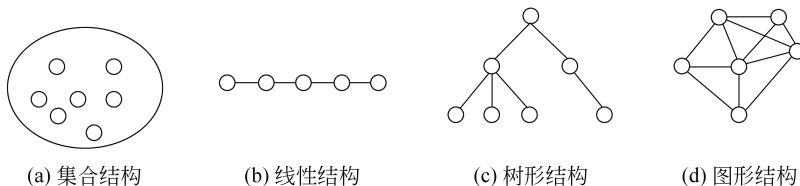


图 1.1 数据的逻辑结构

数据的物理结构反映数据在计算机内的存储方式，又称存储结构。常见的存储结构有顺序存储结构、链式存储结构、索引存储结构和散列存储结构。

2. 数据结构常用的术语

(1) 数据。数据（data）是外部世界信息的载体，是能够被计算机识别、加工、存储的符号的集合。在现实生活中可以将数据比喻为加工某种产品的原材料。常见数据分为数值数据和非数值数据。计算机中的数值数据包括整数、实数等，非数值数据包括字符、图片、影音资料等。



数据结构
常见术语

(2) 数据元素。数据元素（data element）是数据的基本单位，在计算机处理的过程中通常作为一个整体来处理。

(3) 数据项。数据项（data item）是一个数据元素中的一部分。也就是说，一个数据

元素通常由一个或多个数据项组成。例如，在存储学生学籍档案的数据表（见表 1.1）中，每行数据就是一个数据元素，其中学号、姓名等字段就是数据项。

表 1.1 学生学籍档案数据表

学号	姓名	性别	出生年月	班级	...
20121160101	白三	男	1990.4.1	1 班	...
20121160102	朱四	男	1990.6.1	1 班	...
20121160103	沙五	女	1990.12.25	1 班	...
...	...	...	...	...	...

（4）数据对象。数据对象（data object）是性质相同的数据元素的集合，是数据的一个子集。例如，字符串集合 {“aa”“bb”“cc”“dd”} 就是一个数据对象。

（5）数据类型。数据类型（data type）是一组性质相同的值的集合，以及定义在这个值上的一组操作的总称。数据类型与数据结构的关系是：数据结构是数据类型的抽象，数据类型是数据结构在计算机内部的具体表现。

1.1.3 巩固基础

1. 数据结构可以定义为存在_____特定关系的_____的集合。

2. 数据的逻辑结构包括_____、_____、_____、_____。

3. 一个数据元素通常由一个或多个_____组成。

4. 从逻辑上可以把数据结构分成（ ）。

- A. 线性结构和非线性结构
- B. 线性结构和树形结构
- C. 动态结构和静态结构
- D. 内部结构和外部结构

5. 想实现对数据的运算，必须确定其（ ）。

- A. 数据对象
- B. 逻辑结构
- C. 存储结构
- D. 数据操作

6. 数据结构在计算机内存中的表示是指（ ）。

- A. 数据的存储结构
- B. 数据结构
- C. 数据的逻辑结构
- D. 数据元素之间的关系

7. 在数据结构中，与所使用的计算机无关的是数据的（ ）结构。

- A. 存储
- B. 物理
- C. 逻辑
- D. 物理和存储

8. 下面关于数据结构的基本操作的叙述中，错误的是（ ）。

- A. 插入操作是在数据结构中添加新的元素
- B. 删除操作是从数据结构中移除指定的元素
- C. 查找操作是在数据结构中查找特定的元素
- D. 基本操作的实现效率与数据结构的存储方式无关



巩固基础

善 询 篇

1.1.4 头脑风暴

请搜集并了解现实生活中的哪些领域、哪项技术用到了数据结构中的内容，这些内容起到了什么样的作用？将心得记录在表 1.2 中，以防遗忘，也可分享出去，以获得更强的思维碰撞。学习中遇到的疑惑也可一并记录，问题是成长的阶梯，解决问题的过程就是思维进步的过程。

表 1.2 什么是数据结构

我的想法	集思广益

笃 行 篇

1.1.5 案例分析

学籍档案管理系统的数据库组织直接影响系统的性能。如果只是存储数据，那么将每个数据元素依次存储在顺序表中即可，线性表可以按照存储位置很方便快捷地实现查询；如果要实现项目说明中提出的增加、删除、修改信息等功能，使用链表的效率会比较高，因为在链表上插入删除数据的时候，不需要像顺序表那样大量地移动元素，详细内容会在后续案例中进行说明；当按照不同条件查询数据时，就要想办法提高检索的效率，这种情况下采用树形结构是比较好的选择。

1.1.6 总结提高

数据结构主要研究 3 方面的内容，即数据的逻辑结构、数据的存储（物理）结构、数据（算法）的运算。通常，算法的设计取决于数据的逻辑结构，算法的实现取决于数据的物理（存储）结构。具体如图 1.2 所示。

对于数据结构的定义以及相关术语不需要死记硬背，重要的是理解。在理解相关概念的基础上可以尝试对一些问题进行初步分析，将理论知识与解决实际问题紧密结合起来。例如，计划到几个城市旅游观光的时候，怎样设计一条最短路径，可以又省时间又省钱呢？为什么 Office 办公软件中撤销功能能够把最近的一次操作恢复回来？大规模球类比赛中的赛制是如何安排的？多思考这些贴近人们现实生活的实例，对于学习数据结构会有很大的帮助。

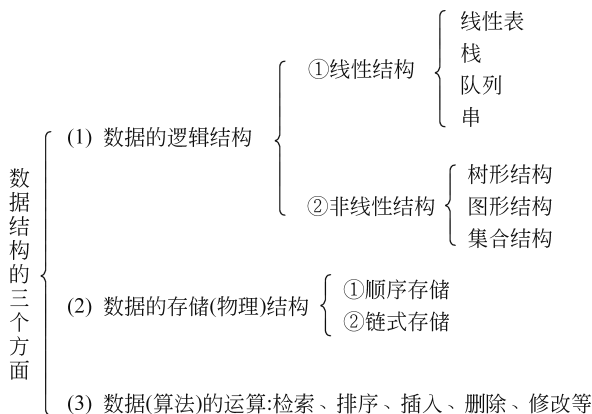


图 1.2 数据结构的研究内容

1.2 五子棋人机对战系统决策分析——数据结构的用途

勤 学 篇

1.2.1 案例说明

在人机对弈系统中，机器会根据人的每一步走子方案，按照某种算法进行分析和判断，最终得到对自己最有利的方案并且执行。尝试模拟这个过程，并且画出相应的数据结构示意图。

1.2.2 知识储备

1. 学习数据结构的意义

“数据结构”在计算机科学中是一门理论性强的专业核心课。它不仅研究计算机的硬件（特别是编码理论、存储装置和存取方法等），而且和计算机软件有着更密切的关系，无论是编译程序还是操作系统，都涉及数据元素在存储器中的组织问题。在研究信息检索时也必须考虑如何组织数据，这样查找和存取数据元素会更为方便。因此，可以认为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。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）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和大型应用程序的重要基础。要想有效地使用计算机、充分发挥计算机的性能，还必须学习和掌握好数据结构的有关知识，目的是了解计算机处理对象的特性，通过算法描述出问题的解决方案，并最终交给计算机进行处理。具体如图 1.3 所示。



学习数据结构的意义

2. 数据结构的作用

在计算机发展的初期，人们使用计算机的目的主要是处理数值计算问题。当人们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个关于此数学模型的算法，最后编出程序进行调

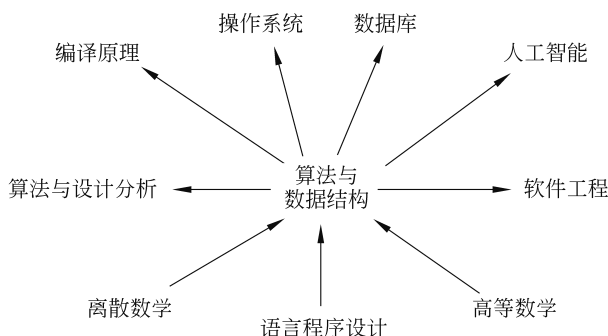


图 1.3 数据结构与其他课程的关系

试、测试，直至得到最终的解答。由于早期所涉及的运算对象是简单的整型、实型或布尔类型数据，所以程序设计者的主要精力都集中于程序的设计上，而没有重视数据结构。随着计算机应用领域的扩大和软、硬件的发展，非数值计算问题显得越来越重要。据统计，当今处理非数值计算性问题占用了 85% 以上的机器时间。这类问题涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用简单的数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要选择合适的数据结构，设计相应的算法，才能有效地解决问题。

举个简单的例子，要管理图书馆中众多的书籍，最直接的方式是按照书籍入馆的顺序依次摆放在书架上，并记录书籍的信息和摆放位置，但是如果有人来借书就会遇到麻烦了，因为书籍的摆放是完全随机的，非常不方便查找，也就是说，书籍这种数据没有经过必要的组织和处理。为了方便读者借书，可以将书籍进行分类摆放以解决上述问题。但是按照什么规则来分类也是很重要的，如按照书名、按照出版社、按照首字母、按照内容分类等。很明显目前绝大多数图书馆采用的都是按照书籍内容来分类的，因为这样可以帮助读者即使在不知道书名、出版社等信息的情况下也能最快地检索到自己想要的图书。

由以上例子可以看出，描述非数值计算问题的数学模型不再是数学方程，而是诸如线性表、树、图之类的数据结构。因此，数据结构课程的主要作用是研究非数值计算的程序设计问题中出现的计算机操作对象以及它们之间的关系。

1.2.3 巩固基础

一、判断题

1. 数据结构主要研究的内容是数值运算问题。 ()
2. 数据结构相关内容只与软件有关。 ()
3. 数据结构相关知识只适用于解决简单的问题。 ()
4. 数据结构是一门可以独立解决问题的学科。 ()

二、选择题

1. 学习数据结构有助于 ()。
A. 提高编程的效率
B. 降低代码的可读性
C. 减少程序的运行速度
D. 增加代码的错误率



巩固基础

2. 以下关于学习数据结构的意义，描述错误的是（ ）。
- A. 能更好地组织和管理数据

B. 使程序变得更复杂难以理解

C. 提高算法的效率和性能

D. 为解决复杂问题提供基础
3. 学习数据结构可以（ ）。
- A. 限制程序的功能扩展

B. 增强解决实际问题的能力

C. 阻碍对算法的理解

D. 降低程序的可维护性
4. 以下（ ）不是学习数据结构带来的好处。
- A. 优化内存使用

B. 降低代码的复用性

C. 提高程序的健壮性

D. 便于代码的调试
5. 学习数据结构的重要意义在于（ ）。
- A. 降低软件开发的成本

B. 增加软件开发的时间

C. 使软件更容易出现漏洞

D. 限制算法的创新

善 询 篇

1.2.4 头脑风暴

请设计一个具体的应用场景，需要用到数据组织管理方式才能更好地解决问题，然后讨论一下哪种解决方案最好。将心得记录在表 1.3 中，以防遗忘，也可分享出去，以获得更强的思维碰撞。学习中遇到的疑惑也可一并记录，问题是成长的阶梯，解决问题的过程就是思维进步的过程。

表 1.3 数据结构的用途

我的想法	集思广益

笃 行 篇

1.2.5 案例分析

人工智能是一门综合性很强的科学，计算机人机对弈是其中的一个重要分支。五子棋人机对弈系统中机器决策算法的设计，直接影响对弈的结果。结合图 1.4 来简单分析，在图中第一层的基础上，白子的走子位置有若干选择，于是形成了图中的第二层；从运筹帷幄的角度，针对白子的每一种选择，衍生出黑子的各种对策，即图中的第三层；依此类推，计算机会创建一个表格来计算每种走法的分值，最后在所有决策中执行分值最高的。这个过程需要的数据结构就是树形结构。为了使人机对弈的效果更好，可以穷举决策树中所有的方案供机器选择，这正好利用了树形结构一对多的特点。

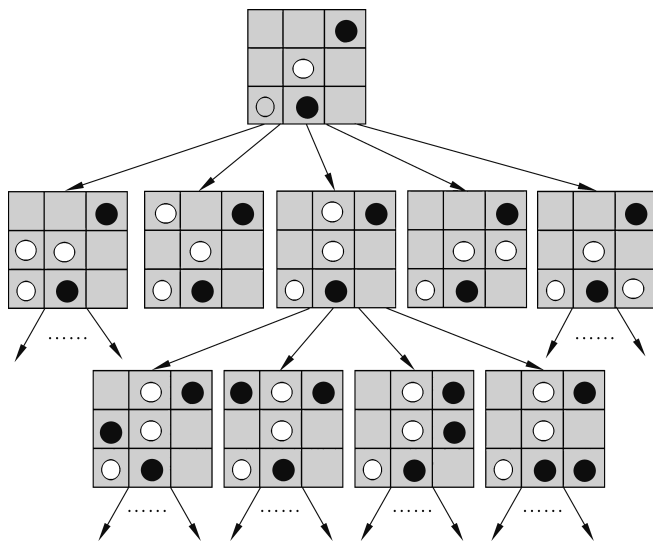


图 1.4 五子棋人机对弈决策树

1.2.6 总结提高

数据结构作为一门独立的学科，在计算机相关的知识体系中起着至关重要的作用，因此有这样一种说法——程序 = 数据结构 + 算法。我们可以毫不夸张地说没有数据结构支撑的程序就像没有灵魂的人一样苍白无力。

通过对数据结构的学习，应该养成“遇到问题先列举解决方法，经过分析选择最佳方案，编写程序交给机器执行”的习惯。对于身边司空见惯、习以为常的问题要多思考、多分析、多比较，用“计算机的思维”方式去处理问题。

1.3 N 个对象全排列——数据结构与算法

勤 学 篇

1.3.1 案例说明

全排列的生成算法就是指对于给定的字符集，用有效的方法将所有可能的全排列无重复无遗漏地列举出来。本节要求设计出 3 个数字全排列的实现方法。

1.3.2 知识储备

1. 算法的含义

广义地说，算法就是为解决问题而采取的步骤和方法，在程序设计中，算法是在有限步骤内求解某一问题所使用的一组定义明确的指令序列，通俗点说即计算机解题的过程。每条指令表示一个或多个操作。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是算



算法的含义

法实现的逻辑推理,后者是算法实现的具体操作。为了表示一个算法,可以采用多种不同的形式,如自然语言、传统流程图、结构化流程图、N-S图、伪代码、计算机语言等。

算法具有以下特征。

(1) 有穷性。一个算法必须保证在执行有限步骤后结束,而且每一步都应该在有限时间内完成。

(2) 确定性。算法中每一条指令必须有明确的含义,不能有歧义。在任何条件下,对于相同的输入只能得到相同的输出。

(3) 可行性。每一个操作步骤都必须在有限的时间内完成。

(4) 输入。一个算法既可以有多个输入,也可以没有输入。

(5) 输出。一个算法可以有一个或多个输出,并且这些输出与输入是有特定关系的,没有输出的算法是没有实际意义的。

2. 算法的设计要求

(1) 正确性。正确性至少包含以下4个层次:程序不含语法错误、程序对于随意的几组合法输入数据能够得出符合要求的结果、程序对于精心设计的典型合法数据输入能够得出符合要求的结果、程序对于所有合法的输入数据都能得出符合要求的结果。

(2) 易读性。一个好的算法往往是与他人共享的,晦涩难懂的算法既不易与人交流,又会造成修改、调试维护的极大困难。

(3) 高效性。在人的交往中,办事效率高的人更容易受到赏识。算法也一样,对于同一问题的不同解决方案,运行时间越少的效率越高,特别是在大型程序设计中,如果每个算法都具有高效性,那么对于缩短整个程序运行时间是非常有帮助的。

(4) 健壮性。当接收非法的数据时,算法也能够做出适当的处理,而不是停止运行或者崩溃。

(5) 可维护性。一个好的算法应该保持后期维护的低投入。

3. 算法效率的度量

算法的效率主要由算法的运行时间和存储所需空间来决定。其中,算法在整个运行过程中所耗费的时间称为算法的时间复杂度,算法整个运行过程占用的存储空间叫作空间复杂度。

算法的时间复杂度记作 $T(n)=O(f(n))$,其度量方法主要有如下两种。

(1) 事后统计的方法。因为很多计算机内部有计时功能,有的甚至可以精确到毫秒级,所以不同算法对应的程序可通过一组或几组相同的统计数据来分辨优劣。但这种方法有两个缺陷:一是必须先运行依据算法编制的程序;二是所得时间的统计量依赖于计算机硬件、软件等环境因素,因此,人们常采用事前分析估算方法。

(2) 事前分析估算的方法。事前分析估算的方法是一种通过对算法中不同语句序列的分析,得出算法中所有语句执行次数的相对大小,从而判断算法的运行时间长短的估算方法。这只是一个相对概念,不是绝对的。

以下面的这段代码为例来说明时间复杂度的计算方法。

```
public static Boolean seqSearch(char key){
    int i=0,counter=0;
    char[] data;
```



算法效率的度量

```

while(i<10){
    System.out.print(" 结果 ");
    if(key==data[i])
        counter++;
    i++;
}
}

```

计算机执行这个算法时，第 1、2 条语句定义并赋初值语句都各执行 2 次简单操作，第 3 条 while 循环语句以及循环体语句除语句 counter++ 执行的次数和 key 在 data 数组中出现的次数一致外，均执行 10 次，把所有语句的执行次数加起来就得到 44+(key 在 data 数组中出现的次数)。

当上例中的循环次数不是 10，而是 n 次，则该例中每条语句的时间复杂度为 2, 2, n , n , n , n , 即时间复杂度的总和为“ $f(n)=4 \times n+4+(\text{key 在 data 数组中出现的次数})$ ”，其复杂度表示为 $T(n)=O(n)$ 。

算法的空间复杂度是一个算法在运行过程中临时占用存储空间大小的量度。算法在计算机存储器内占用的存储空间主要包括 3 个部分，即算法源代码本身占用的存储空间、算法输入/输出数据所占用的存储空间、算法运行过程中临时占用的存储空间。其中，算法源代码本身所占用的存储空间与算法的长短成正比，要想节省这部分空间，就要尽量编写简洁的源代码；算法输入/输出数据所占用的存储空间是由算法所要解决问题的规模决定的，它不随算法的不同而改变；算法在运行过程中临时占用的存储空间则随算法不同而有所区别，好的算法在程序运行过程中占用的临时存储空间不随数据输入规模的不同而不同。

在对算法进行时间复杂度和空间复杂度的分析时，往往不能二者兼顾，考虑好的时间复杂度，就得牺牲空间复杂度的性能；反之亦然。因此，要从算法使用的频率、处理的数据规模等多方面进行综合考虑。

1.3.3 巩固基础

1. 算法的计算量的大小称为计算的（ ）。
A. 效率 B. 复杂性
C. 现实性 D. 难度
2. 算法指的是（ ）。
A. 计算方法 B. 解决问题的步骤和方法
C. 调度方法 D. 排序方法
3. 在下面的程序段中，对 x 的赋值语句的频度为（ ）。

```

for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        x=x+1

```

- A. $O(2n)$ B. $O(n)$ C. $O(n^2)$ D. $O(\log_2 n)$
4. 算法的特征除了输入和输出之外，还包括（ ）。
A. 有穷性、正确性、可行性



巩固基础