

数据结构

- 主讲：张曼 讲师
 - 邮箱：mzhang@xidian.edu.cn
 - 电话：+86-29-8820-2457
- 课件下载
 - 邮箱：ds_1413014@163.com
 - password: 1413014
- 提交作业
 - 邮箱：ds_assignment@163.com
 - 作业格式要求：
邮件主题：作业序号（两位数字）_学号_姓名（例如：
02_*****_张三）
附件格式：word文档 名称如：02_*****_张三.doc

资料

教材

- 数据结构（C语言版）
严蔚敏，吴伟民. 清华大学出版社. 2003.7.
- 数据结构题集（C语言版）
严蔚敏，吴伟民. 清华大学出版社. 2003.7.

考核

评价结构

| | |
|---------------------|-----|
| Assignments | 10% |
| Projects | 15% |
| Class participation | 5% |
| Exam | 70% |

课程特点

- 很强的理论性

本课程不是以掌握应用性知识为目的，而是以掌握基本理论，基本方法，基本技能为目的。让学生把握解决什么样的问题，用什么思想，采用什么样方法解决，以及用什么方法最优解决等一系列问题。

- 很强的概念性

本课程要求学生不但应该深刻理解某些概念的所有要素，要领，精神实质。同时也要求理解为什么要引入某些概念，某些概念的形成过程，引入某些概念是解决什么样的问题。

课程特点（续）

- 很强的连贯性

本课程结构紧凑，每部分所述问题层层推进，逐步深入，逐步解决。全课程始终是以数据间的关系即“结构”为主线索展开。其中“基本数据结构”部分围绕数据结构三要素即逻辑结构、物理结构、运算特性展开，辅以一定该数据结构基本应用的讲述；而“应用数据结构部分”以基本概念、基本方法、性能分析的讲解顺序展开，使全课程大量庞杂的内容条理分明，轮廓分明。

- 很多的混淆性

本课程中首先有很多容易混淆的基本概念。其次有很多算法，状态等等一系列问题都容易混淆。比如要解决某类问题，也许有很多方法和很多途径，每种方法和途径适用于什么场合，各自存在什么优缺点（例如“内部排序”这一章中各中内排方法的比较与应用），都容易产生相互混淆。

课程学习方法

- 循序渐进学习法

由于本课程很强的理论性、概念性和连贯性，所以学习过程中要从概念入手，逐段、逐节、逐章深刻理解和掌握，层层推进，从基础到应用，最后达到完全掌握该课程内容。

- 概括提炼学习法

每学完一节、一章内容，都要从中概括提炼出本部分内容要点、重点、非重点。一则可以达到内容总结、有效复习的目的，二则可以自检学习中存在的问题。

课程学习方法（续）

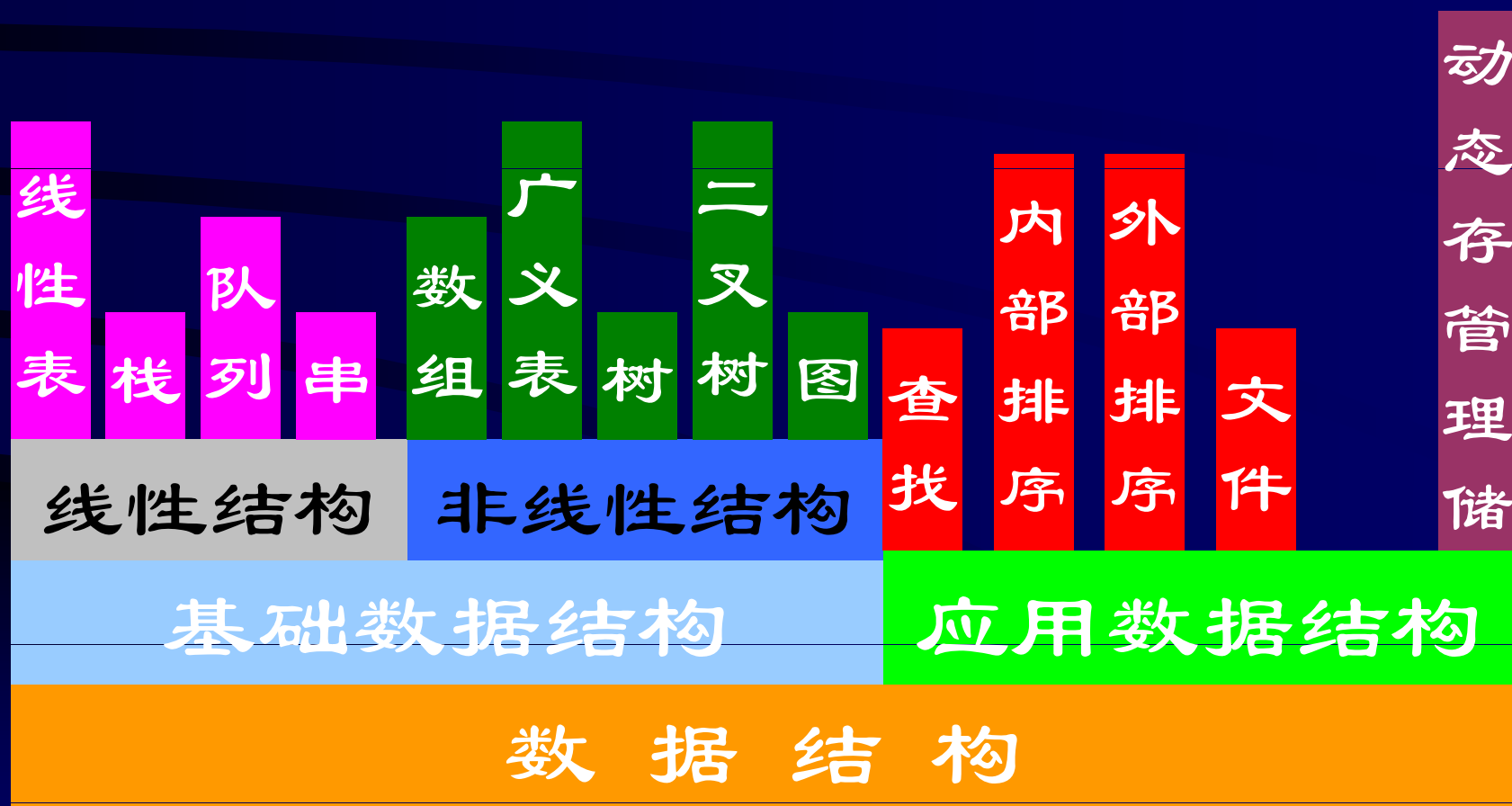
- 归纳对比学习法

针对课程中容易混淆的概念以及课程中同类、非同类容易混淆的问题，进行归纳并加以有效的比较，从中找出它们的异同点，各自的优缺点，各自针对要解决的问题。这种方法不仅能搞清楚容易混淆的问题，而且能更深刻理解本课程的内容实质。

- 循环学习法

由于课程中许多基本概念和复杂算法在一次顺序学习过程中并不能准确和透彻的理解，有些概念和方法可以应用在多种场合，他们的学习就需要循环往复，借助后续内容的信息来全面把握。

课程内容框架



教学内容---第一章

1. 绪论

2. 线性表

3. 栈、队列和串

4. 数组

5. 广义表

6. 树和二叉树

7. 图

8. 动态存储管理

9. 查找

10. 内部排序

11. 外部排序

12. 文件

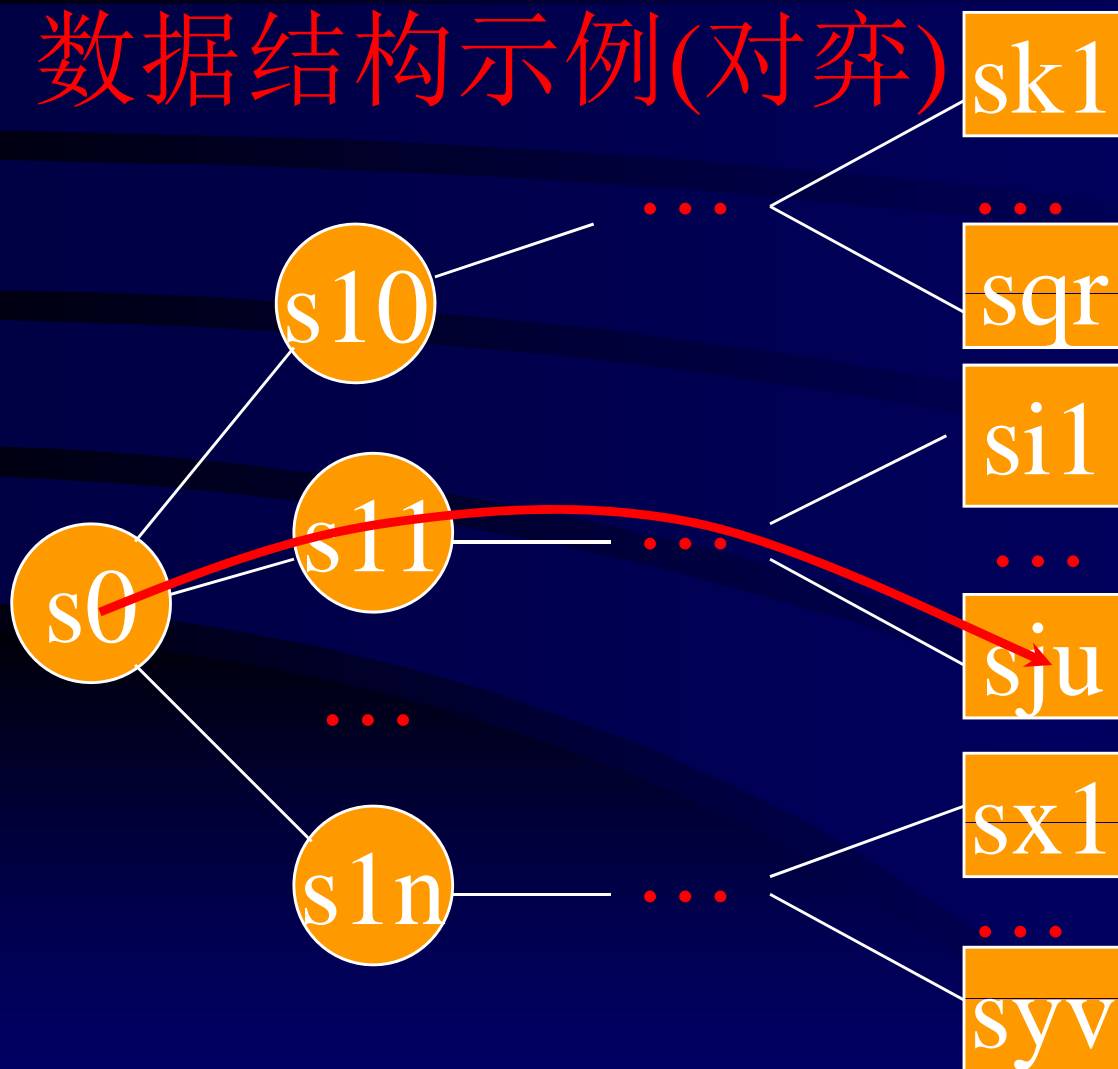
1.1什么是数据结构

数据结构学科发展背景

- 应用领域从科学计算到非数值计算
- 起初数据结构中内容在其他课程中表述
- 1968年美国唐.欧.克努特开创数据结构最初体系。在《计算机程序设计技巧》第一卷《基本算法》系统阐述数据的逻辑结构、存储结构及操作
- 数据结构的两个发展方向：面向专门领域特殊问题的数据结构；从抽象数据类型的观点讨论数据结构

1.1什么是数据结构（续）

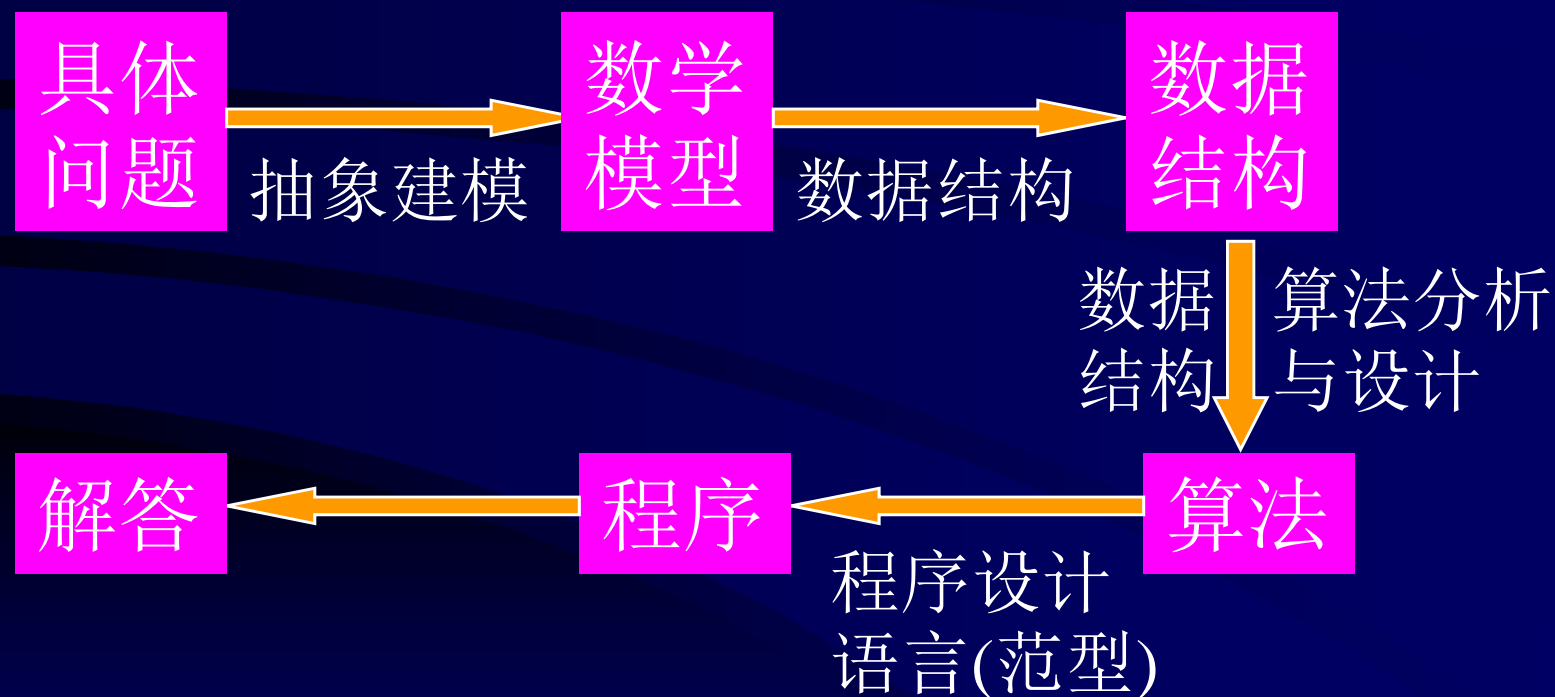
数据结构示例(对弈)



描述非数值计算问题的数学模型不再是数学方程，而是诸如表、树和图之类的数据结构

1.1什么是数据结构（续）

计算机解决问题的过程



1.1什么是数据结构（续）

数据结构学科的地位

- 综合性的专业基础课
- 介于数学、计算机硬件和计算机软件之间的核心课程
- 不仅是一般程序设计的基础，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
- 本课程前继课程：离散数学、C语言
- 本课程后续课程：面向对象程序设计、操作系统、编译原理、数据库系统、人工智能等

1.1什么是数据结构

什么是数据结构

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以及它们之间的关系和操作等等的学科。

1.2基本概念和术语

基本概念

- **数据**：指所有能输入到计算机中并被计算机程序加工处理的符号的总称。不仅包括数字、字符串，还包括图形、图像、声音、动画、视频等能通过编码而被加工的数据形式。
- **数据元素**：是数据的基本单位，数据集合中的元素。
- **数据项**：是数据的不可分割的最小单位。一个数据元素可由若干个数据项组成。
- **数据对象**：是性质相同的数据元素的集合，是数据的一个子集。

1.2基本概念和术语（续）

结构的定义

- **内涵**：数据元素之间的关系，称为“结构”
- **分类**：
 - *集合：松散的关系
 - *线性结构：一对一的关系
 - *树形结构：一对多的关系
 - *网状结构：多对多的关系

1.2基本概念和术语（续）

数据结构的定义

- 描述性定义：

相互之间存在一种或多种特定关系的数据元素的集合

- 形式化定义：

$\text{Data_Structure} = (D, S)$

$D = \{\text{数据元素的有限集合}\}$

$S = \{D\text{上关系的有限集合}\}$

1.2基本概念和术语（续）

数据结构的分类

- 逻辑结构：

数据元素之间的逻辑关系（本来的关系）

- 存储结构：

数据结构在计算机中的映象。包括数据元素的表示和关系的表示两个方面。

分类：

*顺序存储结构

*链式存储结构

描述方式：

用高级语言中的“数据类型”来描述

1.2基本概念和术语（续）

数据类型

- 内涵：

一个值的集合和定义在这个值集上的一组操作的总称。本质是封装：实现由处理器完成

- 分类：

- *原子类型：

值不可分解，如整型、指针类型等

- *结构类型：

值由若干成分按照某种结构组成，如数组、结构（记录）等

1.2基本概念和术语（续）

抽象数据类型（ADT）与数据类型

ADT指一个数学模型以及定义在该模型上的一组操作。ADT的定义仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。

ADT比数据类型的范畴更广，除了具有固有数据类型的特性之外，还包括用户在设计软件系统时自己定义的数据类型。

1.2基本概念和术语（续）

抽象数据类型形式化定义

$ADT = (D, S, P)$

$D = \{\text{数据对象}\}$

$S = \{D\text{上的关系集}\}$

$P = \{\text{对}D\text{的基本操作集}\}$

定义形式：

ADT 抽象数据类型名 {

 数据对象： ...

 数据关系： ...

 基本操作： ...

} ADT 抽象数据类型名

1.2基本概念和术语（续）

抽象数据类型示例

ADT Triplet {

数据对象: $D=\{e_1,e_2,e_3|e_1,e_2,e_3\text{属于ElemSet}\}$

数据关系: $R_1=\{<e_1,e_2>,<e_2,e_3>\}$

基本操作:

InitTriplet(&T,v1,v2,v3)

DestroyTriplet(&T)

Get(T, i, &e)

Put(&T, i, e)

.....

} ADT Triplet

基本操作的定义格式:

基本操作名 (参数表)

初始条件: <描述>

操作结果: <描述>

1.2基本概念和术语（续）

抽象数据类型分类

分类标准：按照值的不同特性

*原子类型：

变量的值不可分解

*固定聚合类型：

变量的值成分的数目确定

*可变聚合类型：

变量的值成分的数目不确定

*多形数据类型：

变量的值成分不确定（成分类型可变）

1.2基本概念和术语（续）

抽象数据类型表示与实现

- 抽象数据类型表示与实现

抽象数据类型通过固有数据类型来表示和实现。即利用处理器中已存在的数据类型来说明新的结构，用已经实现的操作来组合新的操作。

- 类C语言的描述语法

类C语言精选了C语言的一个核心子集，也做了若干扩充，以利于描述。

如：存储结构用typedef；数据元素类型约定为ElemType；在形参表中，以&打头的参数为引用参数；等等