



数据结构和抽象数据类型

主讲教师：贡正仙

学习目标



掌握数据结构的定义



了解数据与数据元素



掌握数据的逻辑结构和存储结构



理解数据的逻辑结构与存储结构之间的关系



掌握数据类型和抽象数据类型的关联



1.1 问题引入：大型超市

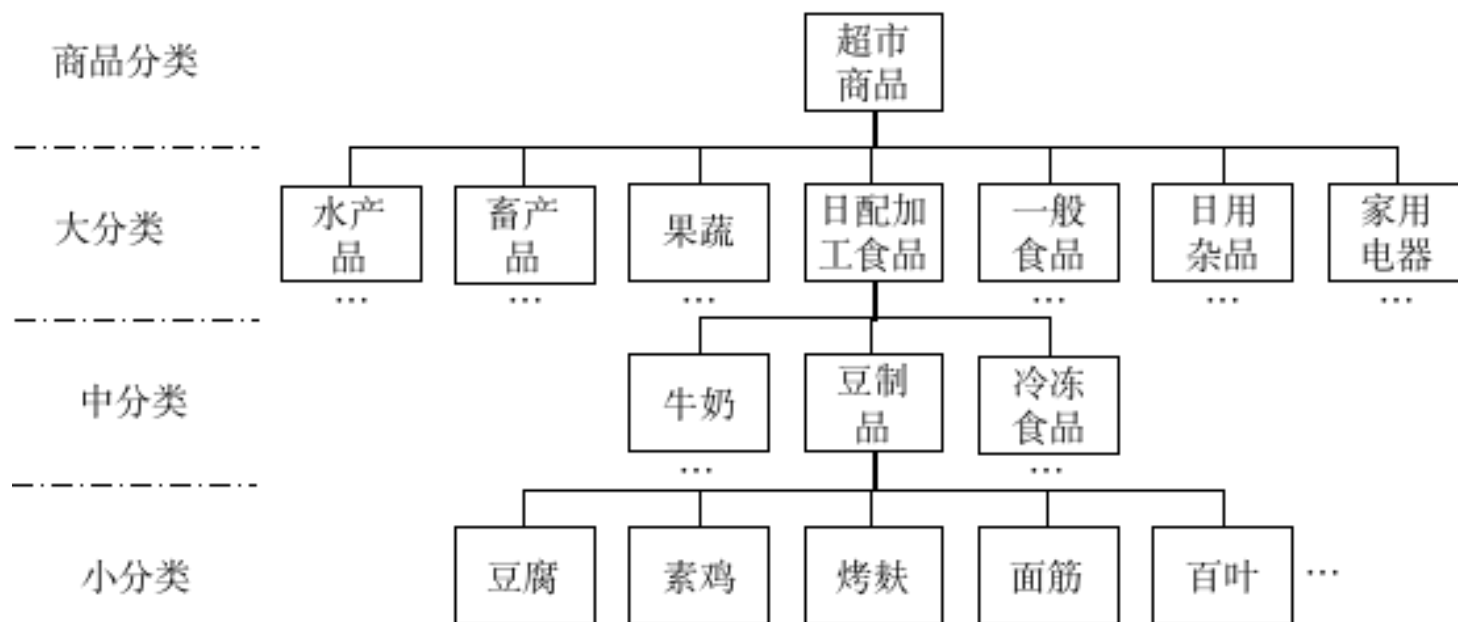
问题：顾客如何能快速找到想要购买的商品，超市又是如何实现方便补货呢？

关键：如何陈列商品？

商品分类 + 商品陈列



商品分类：



商品陈列：

商品分类陈列原则：按照商品的分类层次，大区域 → 中区域 → 小区域

价格按序排列原则：由上至下、由左向右，价格由低到高陈列

先进先出陈列原则：对于同一种商品，先摆放的，客户先取到

特价区：无序（乱放）



1.1 问题引入：大型超市

问题：

如何对商品信息（数据）进行合理的组织（商品分类）、存储（商品陈列）、以及提供必须的操作（商品补架、下架及查找商品）？



如何管理数据以及管理数据的时间和空间的有效性？
(数据结构课程需要研究的两个重要问题)



1.2 问题求解

问题分析：为超市寻求一个合适的商品存放方法和所需的对商品的标准操作

商品分类



商品编码

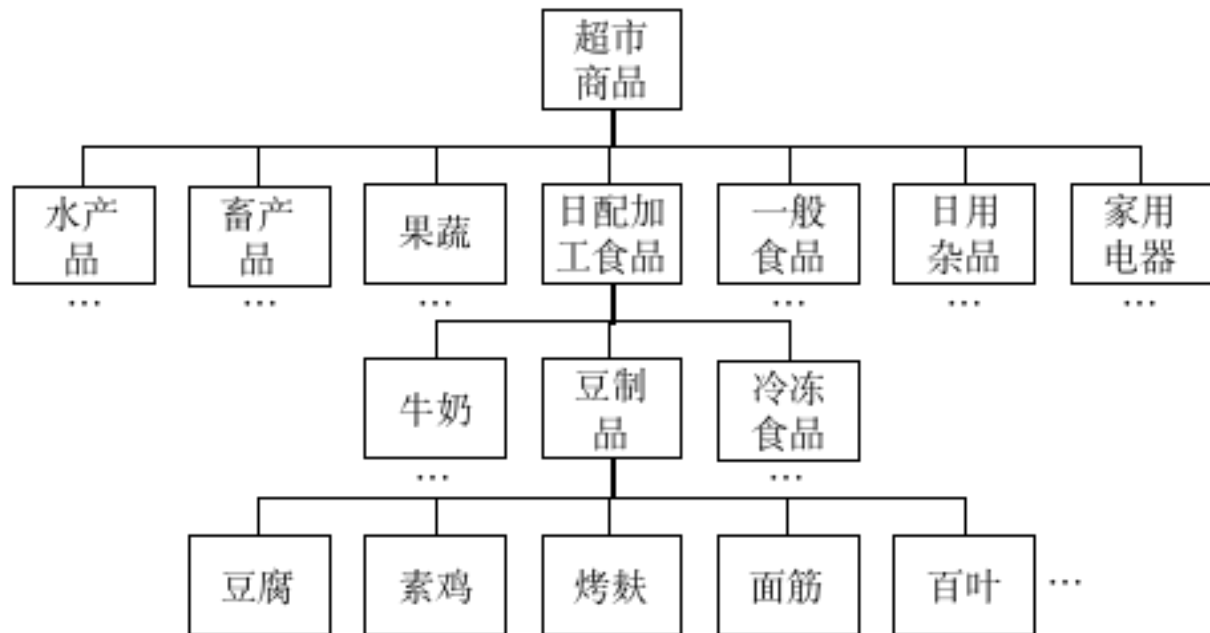


库存&展示管理

查找商品：通过商品编码检索到该商品对应的商品信息

商品信息：商品产地，商品价格，出厂日期，保质日期等

对商品的抽象





存储结构

超市里几乎所有的管理
都不仅与商品有关，还
与超市的空间布局有关



必须把商品的数据信息与超
市的空间布局进行组合，使
商品与其展示位置一一对应

假设某超市将商品划分为A、B、C、D、E、F、G、H八个区域，每个区有9个货架，每个货架有6层。则可设计一个代表物理位置的三位编码：

$$(a_1 \ a_2 \ a_3)$$

其中， a_1 、 a_2 、 a_3 分别代表商品的区域、货架及货架层次，其取值范围分别可以为1~8、1~9、1~6。

如：3 5 4表示商品放在区域C、第5个货架的第6层上。

思考：商品编码取值、取值范围是否还有其他方案？



算法设计

算法设计：针对超市商品的操作及实现的问题。例如，在超市中，最常见的操作为商品的补架和下架等。

商品补架

假设当超市货架上某商品已售出20%左右时，该商品需补架。流程如下：

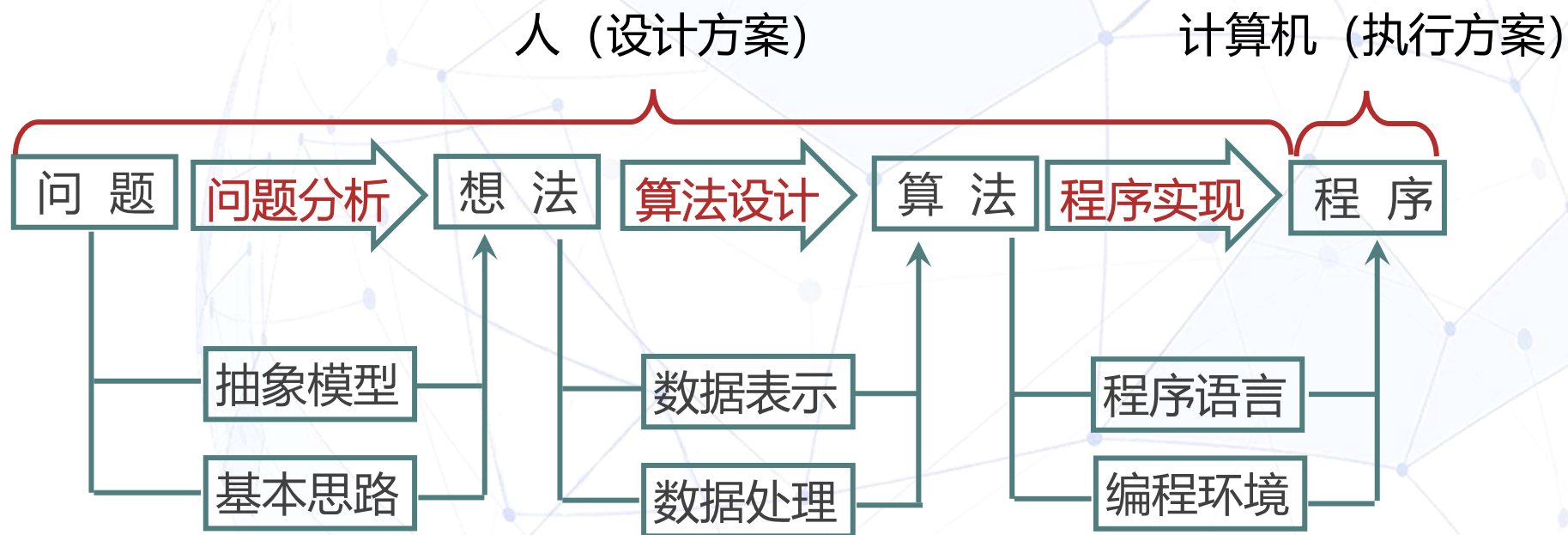
- 1) 如果在货架上某商品已售出20%，则根据该商品的编码，从库存取出该商品满架的20%件数，同时库存减少相应的件数；
- 2) 如商品件数不够，需通知采购补货；
- 3) 将取出的商品放置在指定的货架和层架上，使其满架。

商品下架

假设当超市货架上某商品已临近有效日期或长时间几乎无售出等情况时，该商品将下架。流程如下：

- 1) 将该商品在货架上的剩余件数全部取下，使货架为空；
- 2) 将取下的商品放回库存，并增加相应的库存量；
- 3) 对该商品库存作相应处置，使其编码失效。

利用计算机求解问题的一般过程



计算机不能分析问题并产生问题的解决方案，必须由**人**来分析问题、确定解决方案、编写程序，再让**计算机**执行程序最终获得问题的解

程序

人要和计算机有效地交流，必须通过程序

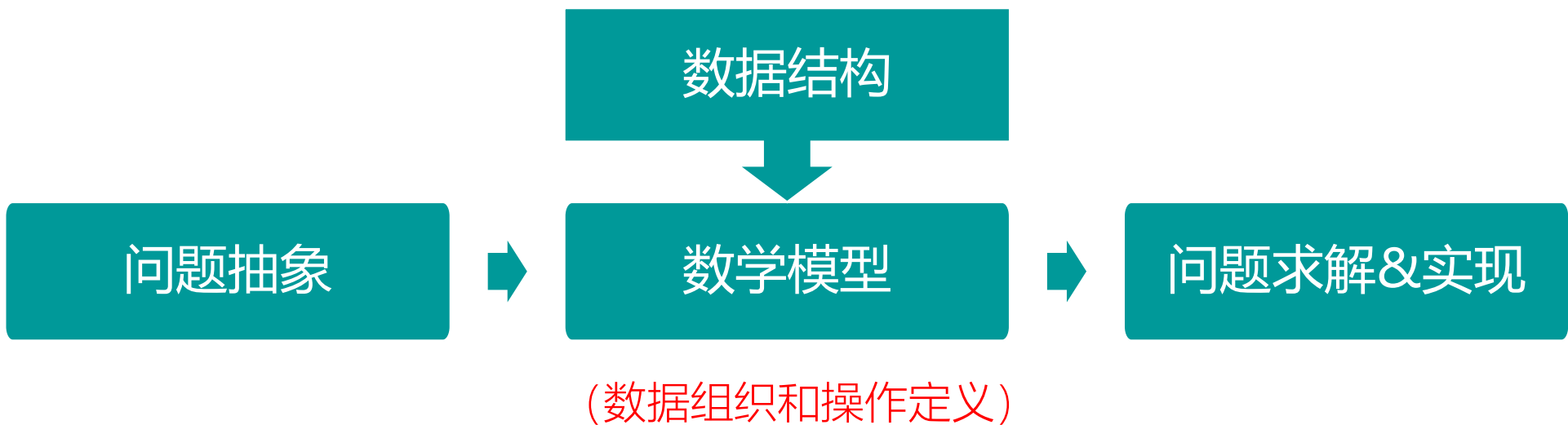
瑞士计算机科学家尼古拉斯·沃斯 (Niklaus Wirth)

计算机领域人尽皆知的名言

算法 + 数据结构 = 程序

Algorithm + Data Structures = Programs



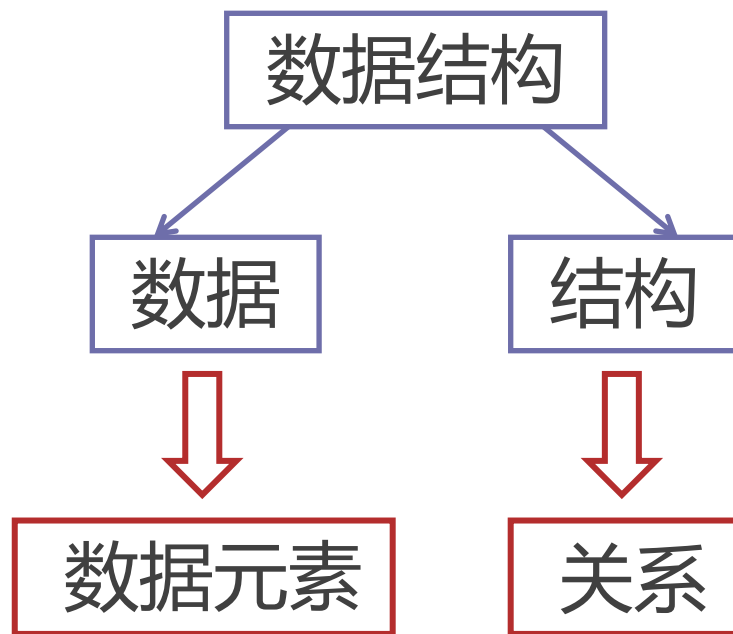




数据结构定义

数据结构：一组具有特定关系的同类数据元素的集合。

在超市的例子中，商品就是数据元素，商品的编码表示商品的存储结构，商品的上架、下架和补架都是对商品的操作定义与实现。



数据?
数据元素?

理解数据

✦ 数据：所有能输入到计算机中并能被程序识别和处理的符号集合

数据 { 数值数据：整数、实数等
非数值数据：图形、图象、声音、文字等

数据是程序的处理对象，严格来说，计算机 = 数据处理机

数据元素

📌 数据元素：数据的**基本单位**，在程序中作为一个整体进行考虑和处理

📌 数据项：构成数据元素的**最小单位**

学号	姓名	性别	出生日期	籍贯
15041001	王 军	男	19970102	吉林省图们市
15041002	李 明	男	19980328	吉林省吉林市
15041003	汤晓影	女	19971116	吉林省长春市
...

数据项

数据元素

通常情况下，数据元素具有相同**个数**和**类型**的数据项

理解数据元素



数据元素是讨论数据结构时的**着眼点**



学籍管理问题，数据元素是什么？ \Rightarrow 表项

学号	姓名	性别	出生日期	籍贯
15041001	王 军	男	19970102	吉林省图们市
15041002	李 明	男	19980328	吉林省吉林市
15041003	汤晓影	女	19971116	吉林省长春市
...

数据结构



数据元素

关系

抽象



一般来说，能**独立**、**完整**地描述问题世界的一切实体都是数据元素

理解数据元素

📎 数据元素是讨论数据结构时的**着眼点**

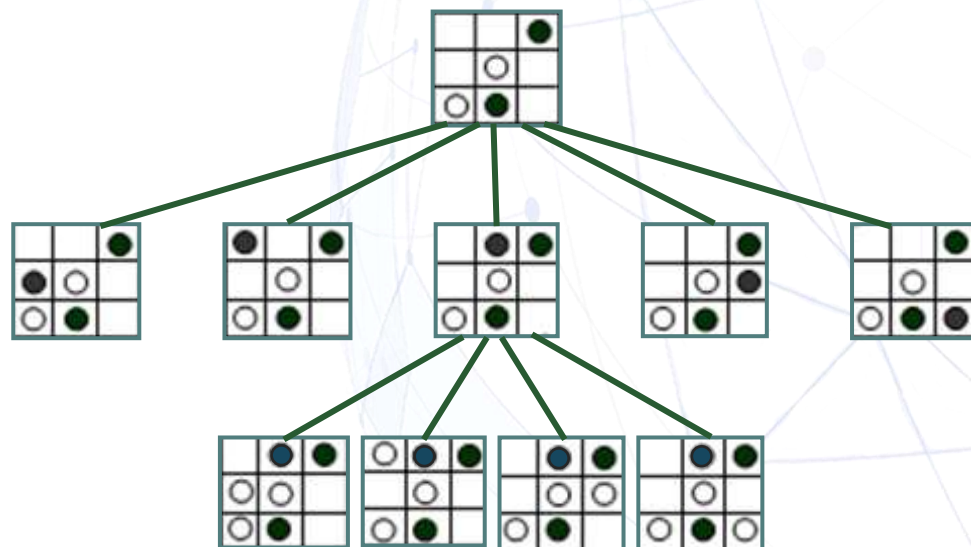
🕒 人机对弈问题，数据元素是什么？ ➡ 格局

数据结构

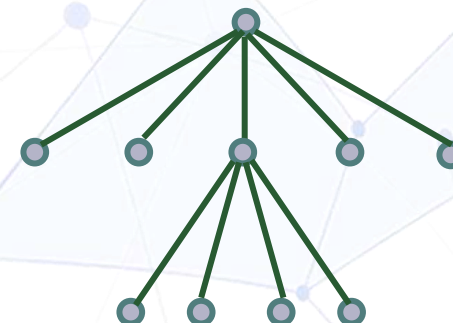


数据元素

关系



抽象 ➡



一般来说，能**独立**、**完整**地描述问题世界的一切实体都是数据元素

理解数据元素



数据元素是讨论数据结构时的**着眼点**



七巧板涂色问题，数据元素是什么？

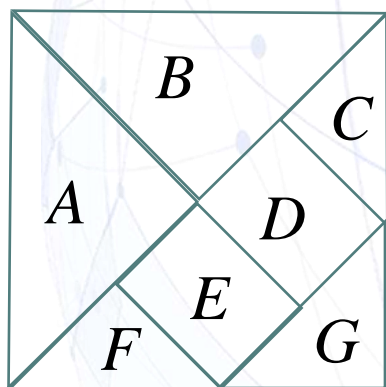


区域

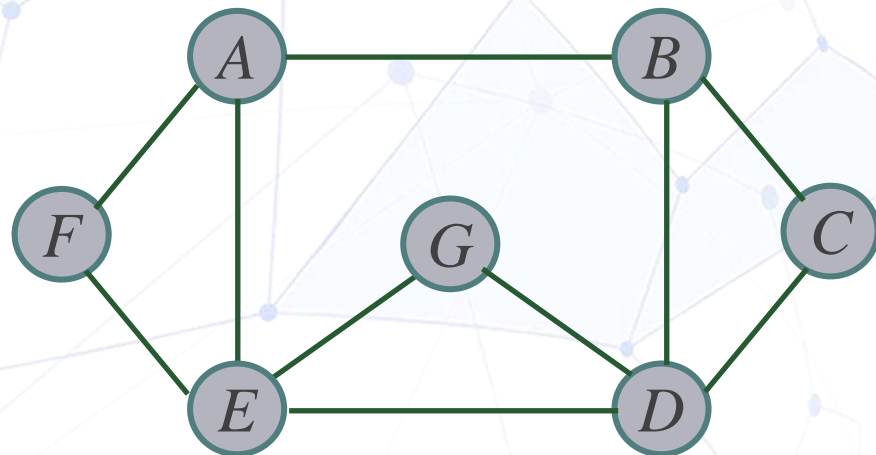
数据元素

关系

数据结构



抽象



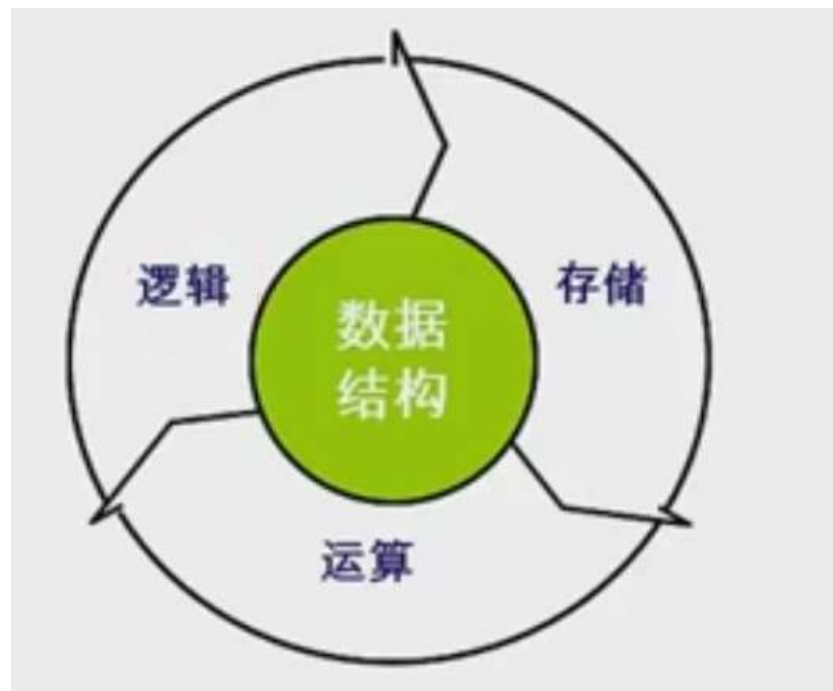
一般来说，能**独立**、**完整**地描述问题世界的一切实体都是数据元素



数据结构定义

数据结构：一组具有特定关系的同类数据元素的集合。

数据结构包括三个要素：**逻辑结构**、**存储结构**及其**操作定义与实现**。



逻辑结构

✦ 数据结构：相互之间存在一定**关系**的**数据元素**的集合

✦ 按照**视点**的不同，分为逻辑结构和存储结构



是否基于内存

✦ 数据的逻辑结构：数据元素之间**逻辑关系**的整体



关联方式或邻接关系



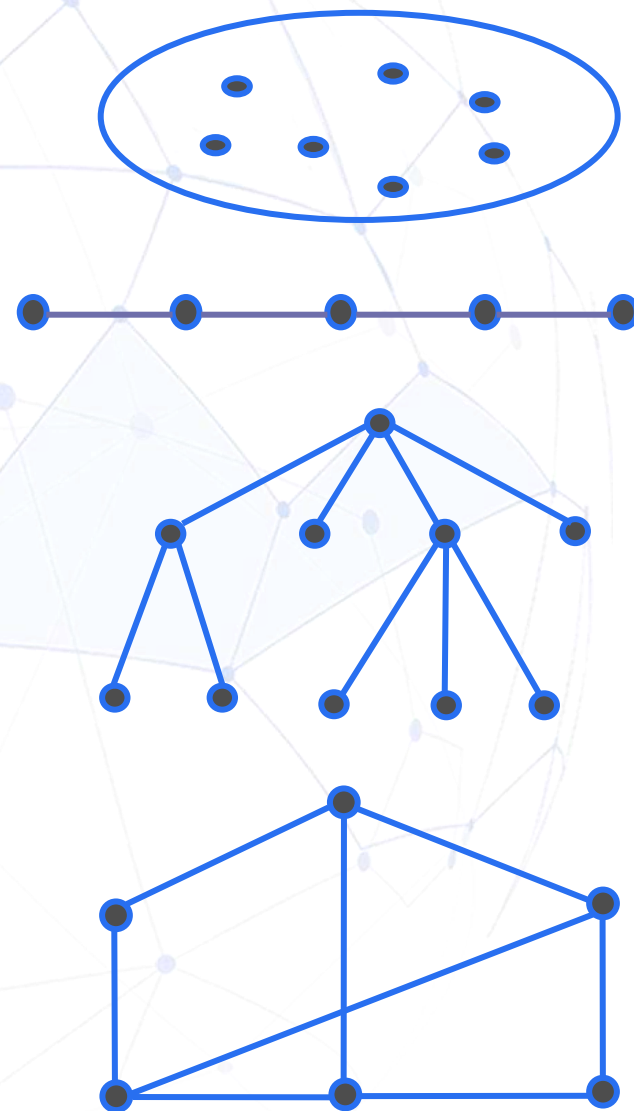
取决于实际问题

理解逻辑结构

✦ 数据结构从逻辑上分为四类：

- (1) 集合：数据元素之间**没有**关系
- (2) 线性结构：数据元素之间是**一对一**的线性关系
- (3) 树结构：数据元素之间是**一对多**的层次关系
- (4) 图结构：数据元素之间是**多对多**的任意关系

{ 线性关系：线性结构
非线性关系：树结构和图结构



理解逻辑结构

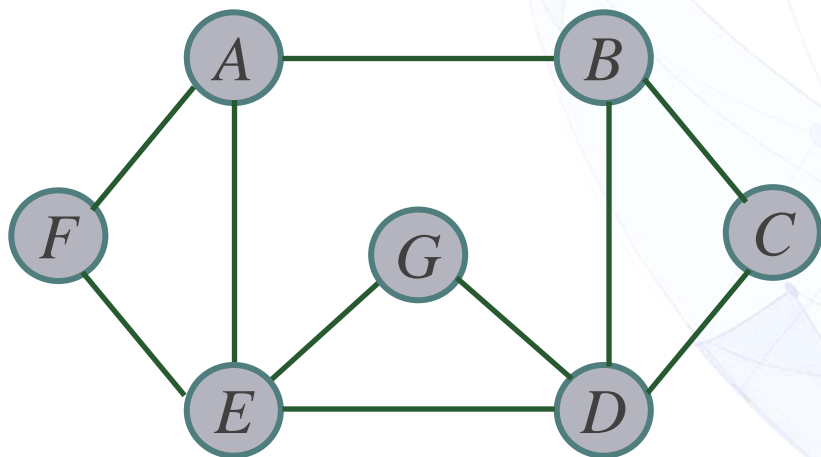
📌 数据结构：相互之间存在一定**关系**的**数据元素**的集合

📌 数据的逻辑结构：数据元素之间**逻辑关系**的整体

📌 数据的逻辑结构在形式上可定义为一个二元组：

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

其中 D 是数据元素的有限集合， R 是 D 上关系的集合



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

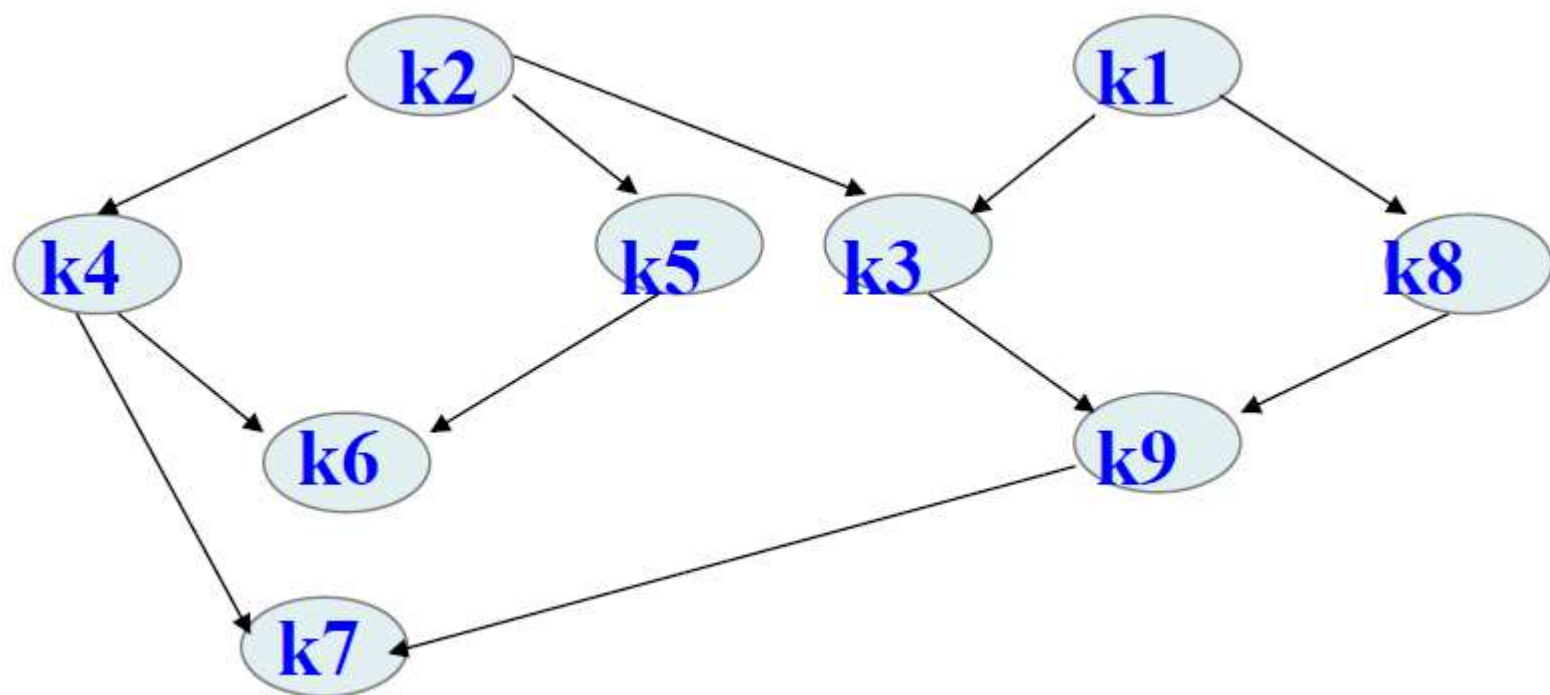
其中 $D = \{A, B, C, D, E, F, G\}$

$R = \{R1\}$, $R1 = \{(A, B), (A, E), (A, F), (B, C), (B, D), (C, D), (D, E), (D, G), (E, F), (E, G)\}$

【例1】 设有数据结构为： $B = (D, R)$ ，其中：

$D = \{k1, k2, k3, \dots, k9\}$

$R = \{ \langle k1, k3 \rangle, \langle k1, k8 \rangle, \langle k2, k3 \rangle, \langle k2, k4 \rangle, \langle k2, k5 \rangle, \langle k3, k9 \rangle, \langle k5, k6 \rangle, \langle k8, k9 \rangle, \langle k9, k7 \rangle, \langle k4, k7 \rangle, \langle k4, k6 \rangle \}$
画出其逻辑结构的图示，并确定相对于关系R，哪些结点是开始结点，哪些是终端结点？



理解存储结构



📌 数据结构：相互之间存在一定**关系**的**数据元素**的集合

📌 数据的存储（物理）结构：数据及其逻辑结构在**计算机**中的表示

数据元素
逻辑关系

内存

🕒 计算机语言如何进行内存分配？

存储结构实质上是内存分配，具体实现时依赖于计算机语言

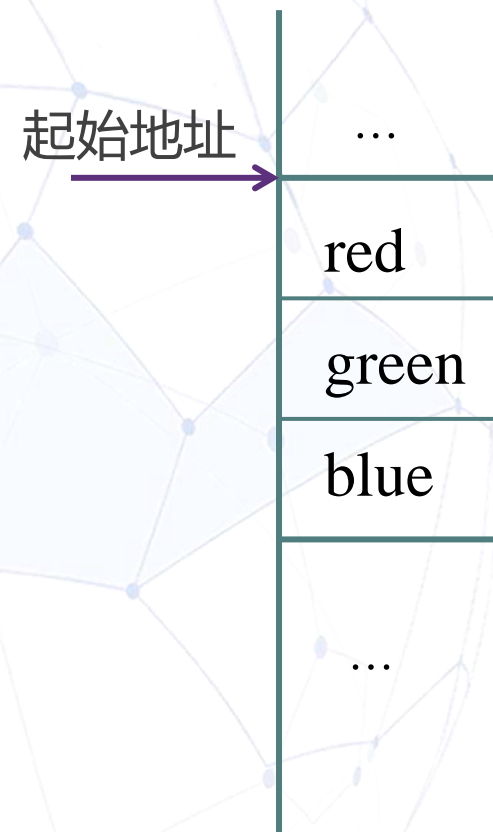
理解存储结构

📌 常见的存储结构:

(1) **顺序存储结构**: 用一组**连续**的存储单元**依次**存储数据元素, 数据元素之间的逻辑关系由元素的**存储位置**表示

↪ 下标

例: (red, green, blue)



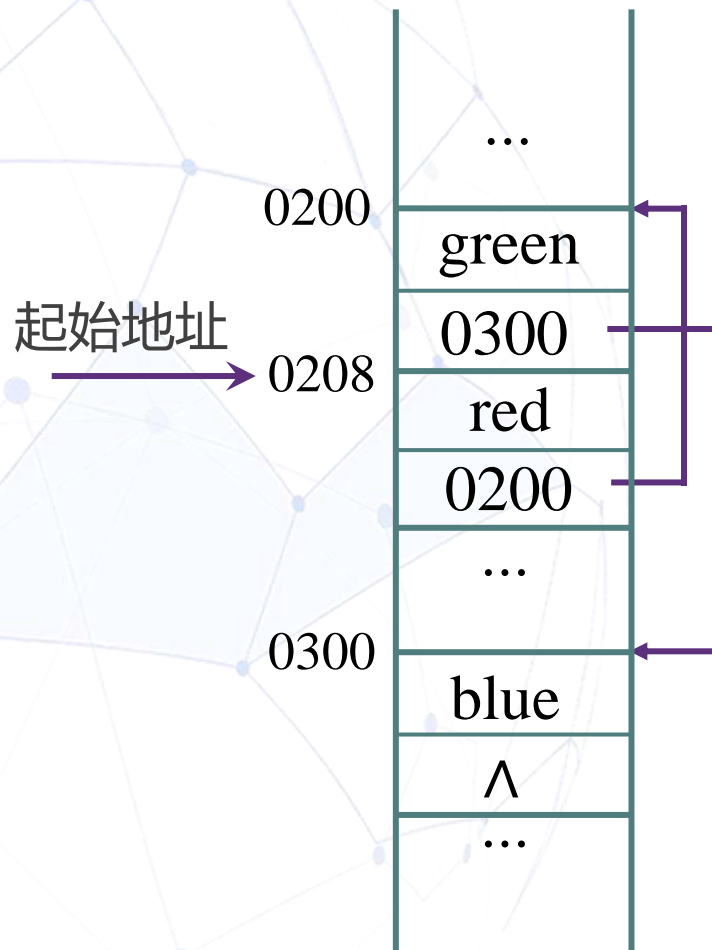
理解存储结构

常见的存储结构:

(2) **链接存储结构**: 用一组任意的存储单元存储数据元素, 数据元素之间的逻辑关系用指针来表示

地址

例: (red, green, blue)

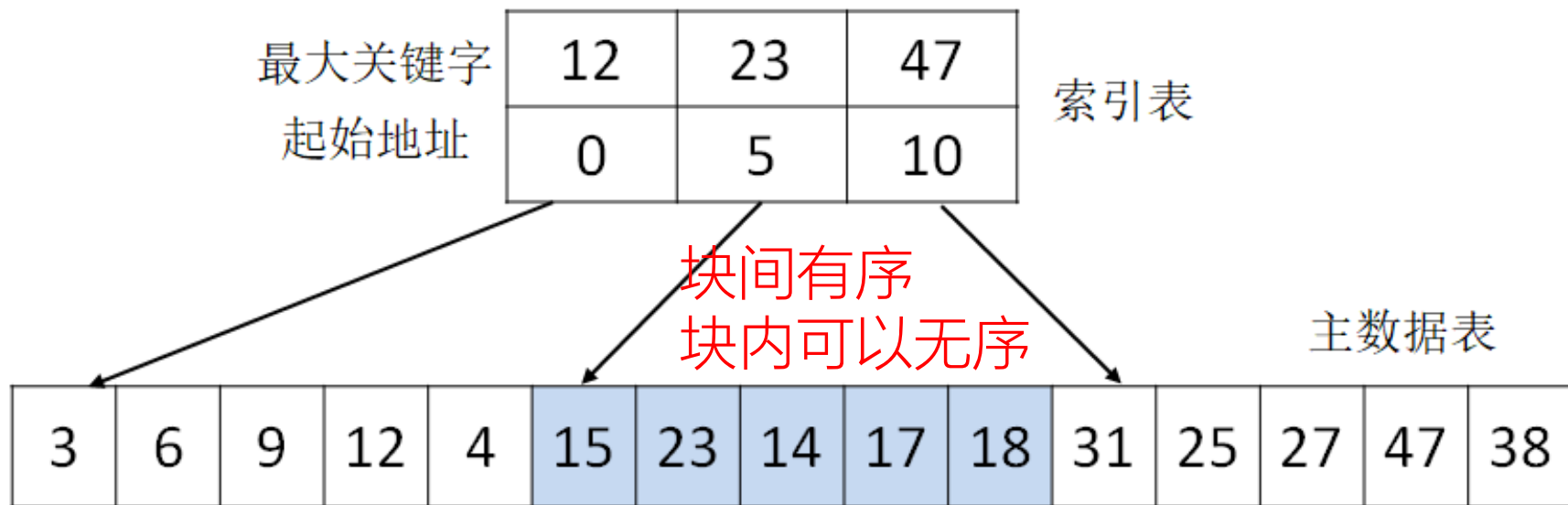


理解存储结构

常见的存储结构:

(3) 索引存储结构:

主数据表用**顺序表**存储元素信息的同时，再建立附加的**索引表**。
该方法主要用于**快速查找**数据元素。



理解存储结构

📌 常见的存储结构：

(4)散列存储（即**哈希存储**）：将数据元素存储在一个连续区域，每一个数据元素的具体存储位置是根据其关键字的值，并通过散列（哈希）函数直接计算出来的

该方法主要用于**快速查找**数据元素。

高级语言中的字典和集合，底层常采用哈希存储结构实现。

城市：上海

2 0 0 2 4 0

上海交通大学闵行校区

邮
政
编
码

逻辑结构和存储结构的关系

 数据的逻辑结构是**用户视图**，**面向问题**  数据本身的构成方式

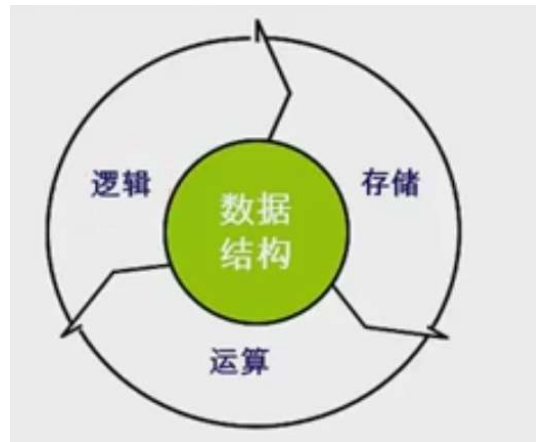
 数据的存储结构是**实现视图**，**面向计算机**  数据在内存的存储表示



一种数据的逻辑结构可以采用多种存储结构来实现，
采用不同的存储结构，其数据处理的**效率**往往是不同的



数据的操作实现



数据的操作（也称**运算或算法**）：包括操作的定义和实现。

操作定义：对现实问题的抽象，它独立于计算机。

操作实现：建立在数据的存储结构之上完成的，它依赖于计算机和具体的程序设计语言。

例：超市里的商品补架和商品下架的描述就是商品（数据）的操作实现。

理解数据类型

🕒 什么是数据类型呢?

🕒 `int a, b;` ➡ `a = 1000000000000000; a = a % b;`

🕒 `float x, y;` ➡ `x = 1234567.123; x = x % y;`

📌 数据类型：一组值的集合以及定义于这个值集上的一组操作

数据类型（Data Type）

DT是高级语言中的概念

- 用于区分不同性质的数据并对它们做不同操作。
- 某种特定语言中，确定了对象的数据类型，也就确定了该对象的取值范围、存储方法和可进行的操作。

例如C++语言中的整数数据类型用于存储所有整数，采用32位二进制补码进行存储，可进行的操作有加、减、乘、除、求余数等。

理解抽象

✚ 抽象：抽出问题本质的特征而忽略非本质的细节

水果



地图



抽象的好处是什么？



在一个更高的层次上思考问题

$$1 + 1 = 2$$

$$2 + 3 = 5$$



$$x + y = z$$

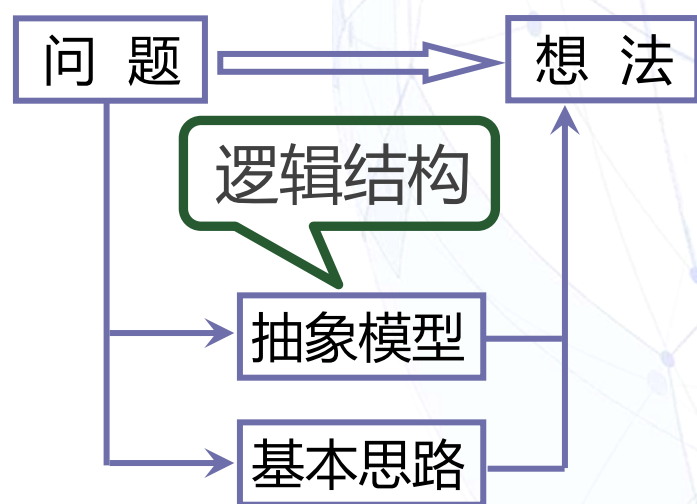
算术运算



代数运算

抽象数据类型的提出

🕒 抽象数据类型把什么抽象掉了呢？



数据模型 = 数据的逻辑结构
强调有哪些数据元素，数据元素之间
满足什么逻辑关系，基于数据模型有
哪些基本操作

抽象数据模型

✳️ 抽象数据类型：一个数据模型以及定义在该模型上的一组操作

📎 抽象数据类型不考虑数据项，把具体的数据类型抽象掉了

📎 抽象数据类型只考虑数据的逻辑结构和基本操作

学号	姓名	性别	出生日期	籍贯
15041001	王 军	男	19970102	吉林省图们市
15041002	李 明	男	19980328	吉林省吉林市
15041003	汤晓影	女	19971116	吉林省长春市
...

抽象

抽象数据类型的实现



如何实现抽象数据类型 (abstract data type, 简称ADT) 呢?

抽象层

- 数据模型 (逻辑结构)
- 操作集合

(a) 定义——ADT定义

设计层

- 数据表示 (存储结构)
- 算法

(b) 设计——数据结构设计

实现层 (C、...)

- 自定义数据类型
- 自定义函数

实现层 (C++、Java、...)

- 成员变量
 - 成员函数
- } 类

(c) 实现——程序语言实现



抽象数据类型

抽象数据类型：一个与数据元素及在数据元素之上的实现无关的数据类型，它们对使用者来说无需知道数据元素的类型，只需知道数据元素之间的逻辑关系，也不用关心是怎么实现的。

如：两个数的相加，可以是两个整数的相加，也可以是两个浮点数的相加。这时需要针对两个不同的类型的数据元素定义并实现两个加法操作。

ADT 抽象数据类型名 {

数据元素： <数据元素的定义>

数据关系： <数据关系的定义>

基本操作： <基本操作的定义>

}

例子：“矩阵”的抽象数据类型

- 类型名称：矩阵 (Matrix)
- 数据对象集：一个 $M \times N$ 的矩阵 $A_{M \times N} = (a_{ij})$ ($i=1, \dots, M; j=1, \dots, N$)由 $M \times N$ 个三元组 $\langle a, i, j \rangle$ 构成，其中 a 是矩阵元素的值， i 是元素所在的行号， j 是元素所在的列号。
- 操作集：对于任意矩阵 $A, B, C \in \text{Matrix}$ ，以及整数 i, j, M, N
 - `Matrix Create(int M, int N)`：返回一个 $M \times N$ 的空矩阵；
 - `int GetMaxRow(Matrix A)`：返回矩阵A的总行数；
 - `int GetMaxCol(Matrix A)`：返回矩阵A的总列数；
 - ElementType `GetEntry(Matrix A, int i, int j)`：返回矩阵A的第 i 行、第 j 列的元素；
 - `Matrix Add(Matrix A, Matrix B)`：如果A和B的行、列数一致，则返回矩阵 $C=A+B$ ，否则返回错误标志；
 - `Matrix Multiply(Matrix A, Matrix B)`：如果A的列数等于B的行数，则返回矩阵 $C=AB$ ，否则返回错误标志；
 -

二维数组？一维数组？十字链表？

先按行加？先按列加？什么语言？

讨论数据结构时的着眼点是（ ）。

- ☐ A 数据
- ☒ B 数据元素
- ☐ C 数据对象
- ☐ D 数据项

提交

逻辑结构与数据元素本身的内容和形式无关。

☒ A 正确

☐ B 错误

提交

理论上讲，存储结构与程序设计语言无关。

☐ A 正确

☒ B 错误

提交

5. 按照视点的不同，数据结构分为逻辑结构和存储结构，其中的视点指的是（ ）。

- ☒ A 内存
- ☐ B 外存
- ☐ C 文件
- ☐ D 问题

提交

顺序存储结构中数据元素之间的逻辑关系是由（ ）表示的。

- ☒ A 存储位置
- ☐ B 指针
- ☐ C 线性结构
- ☐ D 非线性结构

提交

基于某种逻辑结构之上的基本操作，其实现是唯一的。

☐ A 正确

☒ B 错误

提交

井字棋游戏中操作对象的逻辑结构是：

- ☐ A 线性表
- ☒ B 树
- ☐ C 图
- ☐ D 集合

提交

课堂小结

数据结构的定义

数据元素和数据项

逻辑结构和存储结构

抽象数据类型

逻辑结构种类

- 线性
- 树
- 图
- 集合

存储结构种类

- 顺序结构
- 链式结构

