

Data Structures

# 绪论 Introduction

2023年8月31日

学而不厌 诲人不倦

- ➡ **1.1** 问题的求解与程序设计
- ➡ **1.2** 数据结构的基本概念
- ➡ **1.3** 算法的基本概念
- ➡ **1.4** 算法分析

# 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

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
...	...	...	...	...

数据是程序的处理对象，严格来说，**计算机 = 数据处理机**

📌 **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

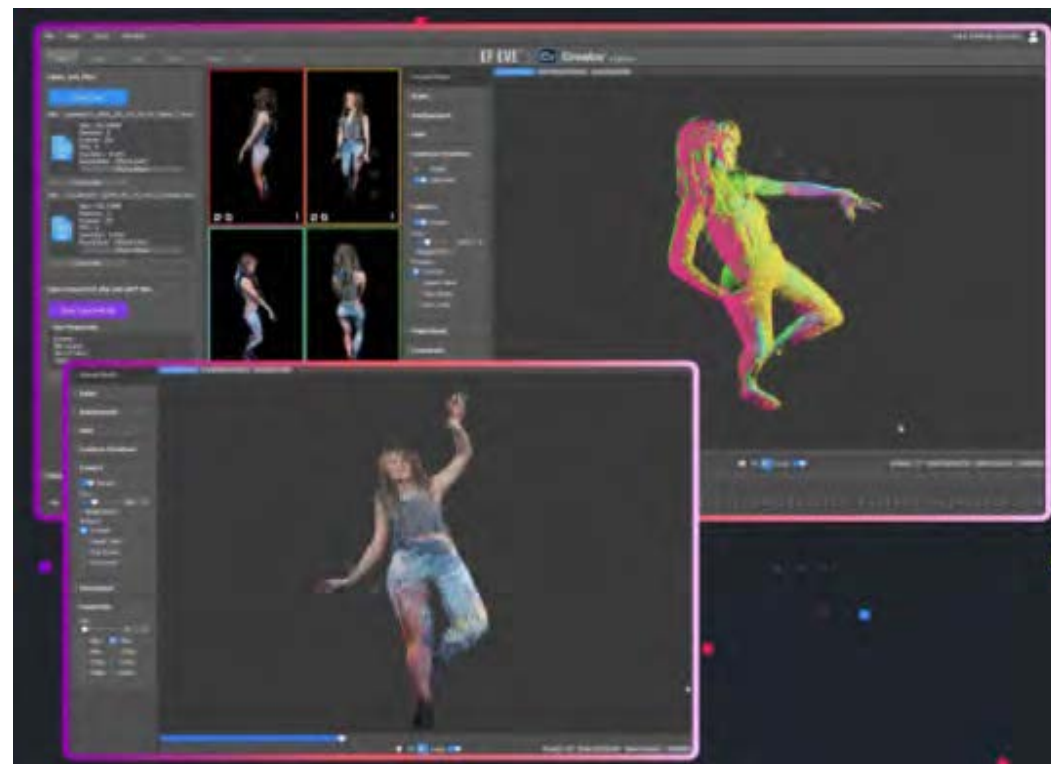
数据是程序的处理对象

Release Date:  
July 12, 2022 11:22AM [E

Read the Release:  
2022-031

Permissions:  
Content Use Policy

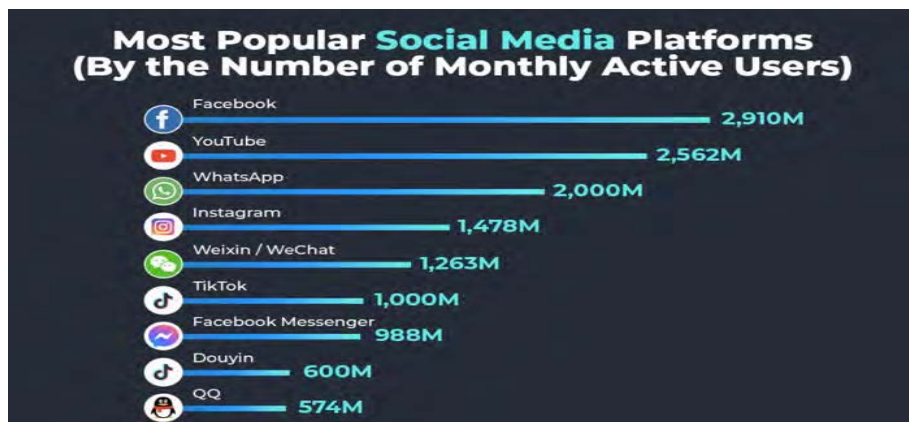
Download Options:  
Full Res, 14575 X 8441, TIF (13  
Full Res, 14575 X 8441, PNG (1  
2000 X 1158, PNG (3.33 MB)  
Text Description, PDF (78.50



<https://www.nasa.gov/press-release/nasa-reveals-webb-telescope-s-first-images-of-unseen-universe>



## 数据集

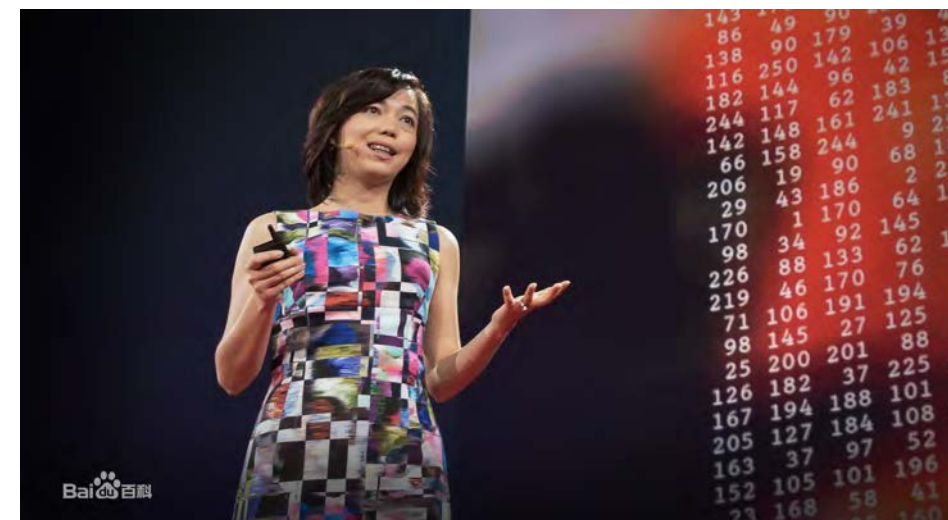


根据性别和年龄计算某部电影的平均得分

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100836 entries, 0 to 100835
Data columns (total 10 columns):
userId      100836  non-null  int64
movieId     100836  non-null  int64
rating      100836  non-null  float64
timestamp   100836  non-null  int64
gender      100836  non-null  object
age         100836  non-null  int64
occupation  100836  non-null  int64
zip         100836  non-null  object
title       100836  non-null  object
genres      100836  non-null  object
dtypes: float64(1), int64(5)
memory usage: 8.5+ MB
```

**MovieLens**

语音、文字、  
图像、视频.....



<https://www.image-net.org/index.php>

# 理解数据元素

- 📌 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

数据结构



数据元素

关系

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 



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**

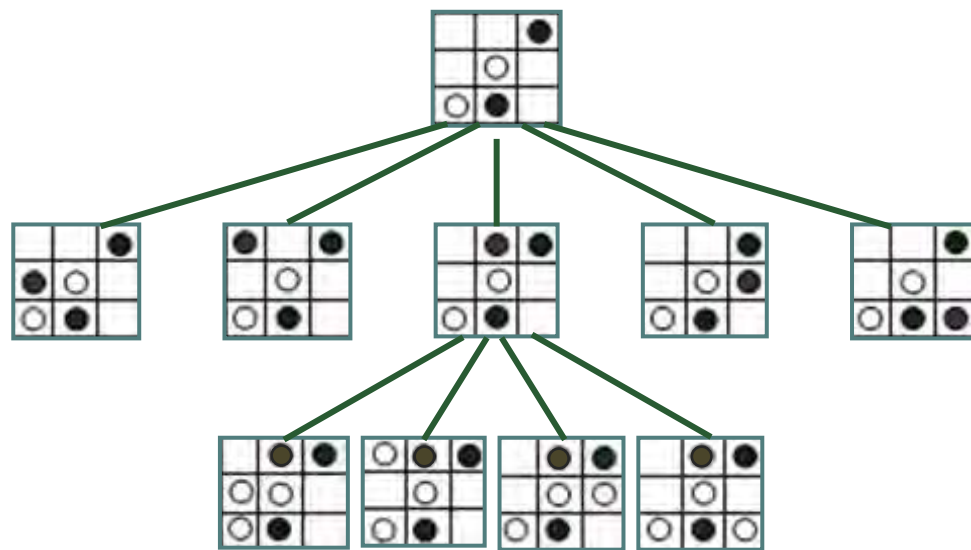
🕒 人机对弈问题，数据元素是什么？ ➡ 格局

数据结构

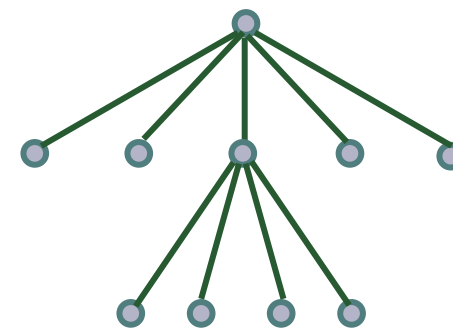


数据元素

关系



抽象



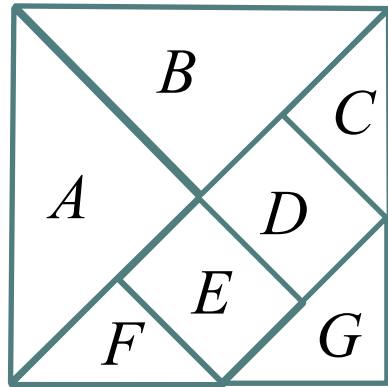
一般来说，能**独立**、**完整**地描述问题世界的一切实体都是数据元素



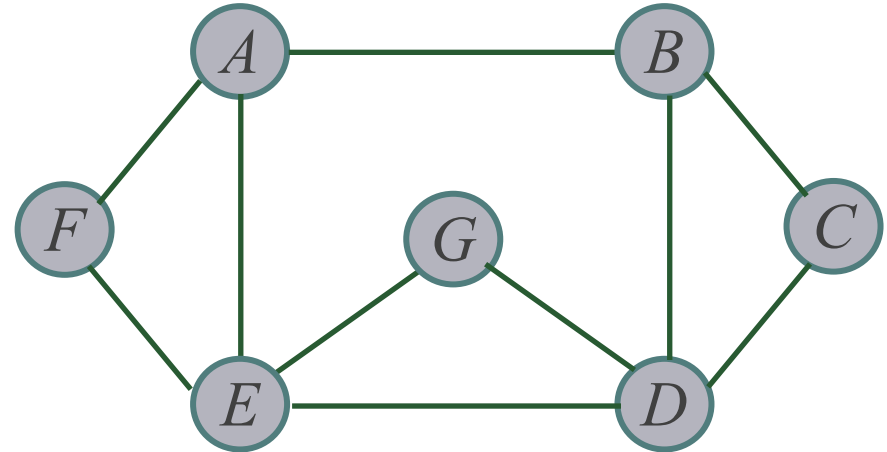
# 理解数据元素

📎 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



抽象

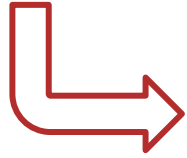


一般来说，能**独立**、**完整**地描述问题世界的一切实体都是数据元素

# 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

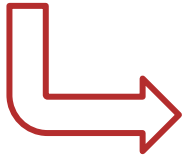


是否基于内存

✦ 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



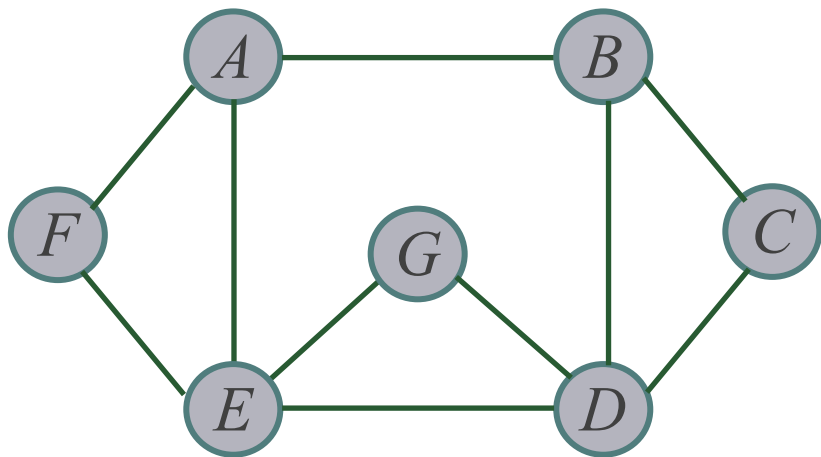
取决于实际问题

# 理解逻辑结构

- ✦ 数据结构：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



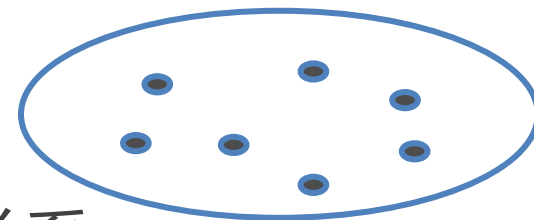
$$\text{Data\_Structure} = (D, R)$$

$$\text{其中 } D = \{A, B, C, D, E, F, G\}$$

$$R = \{R_1\}, \quad R_1 = \{(A, B), (A, E), (A, F), (B, C), (B, D), (C, D), (D, E), (D, G), (E, F), (E, G)\}$$

✦ 数据结构从逻辑上分为四类：

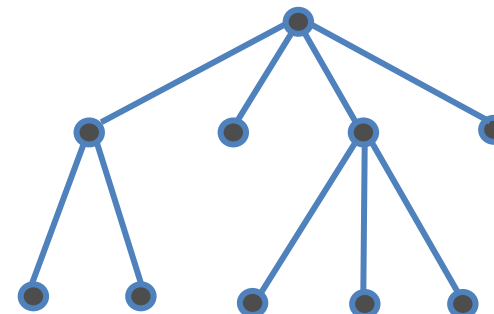
(1) 集合：数据元素之间除了同属于一个集合外，**没有其他**关系



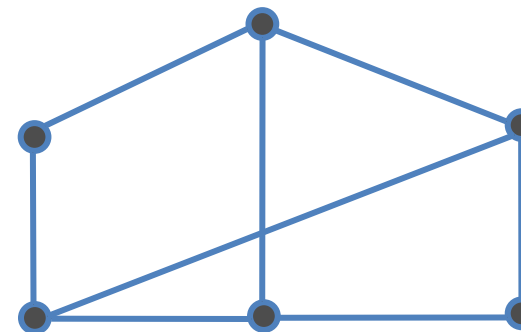
(2) 线性结构：数据元素之间是**一对一**的线性关系



(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

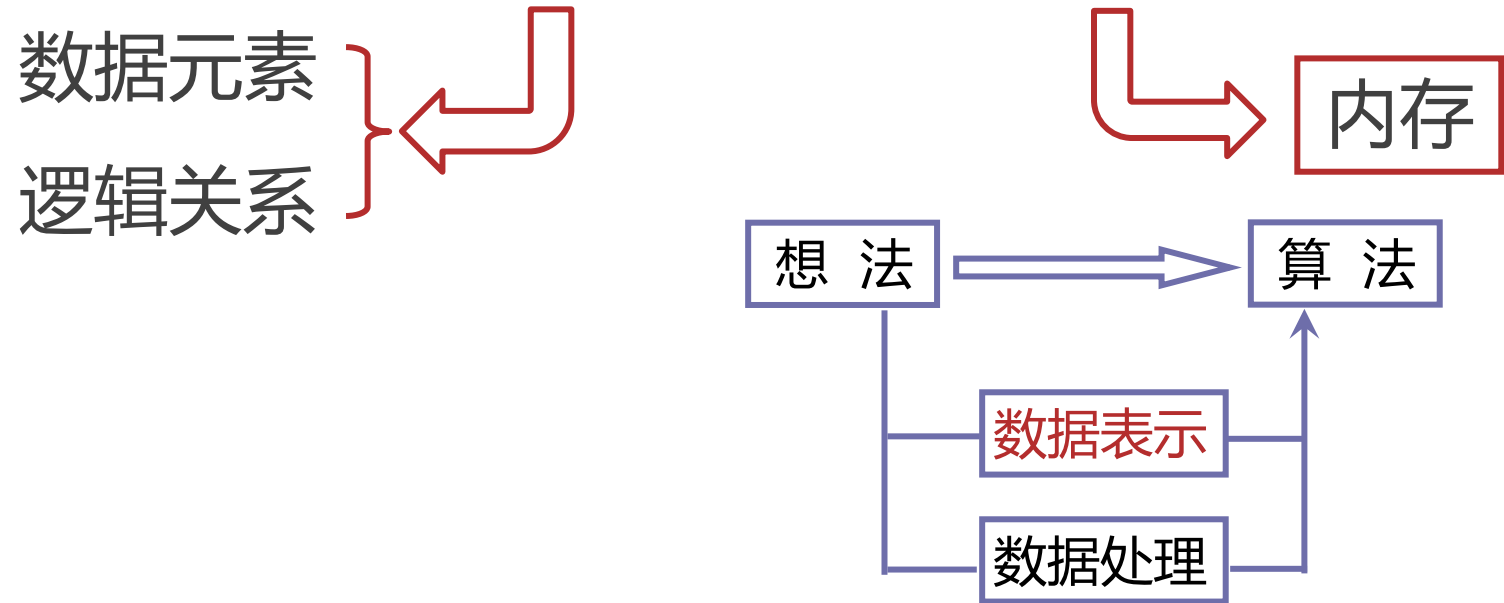


{ 线性关系：线性结构  
非线性关系：树结构和图结构

# 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



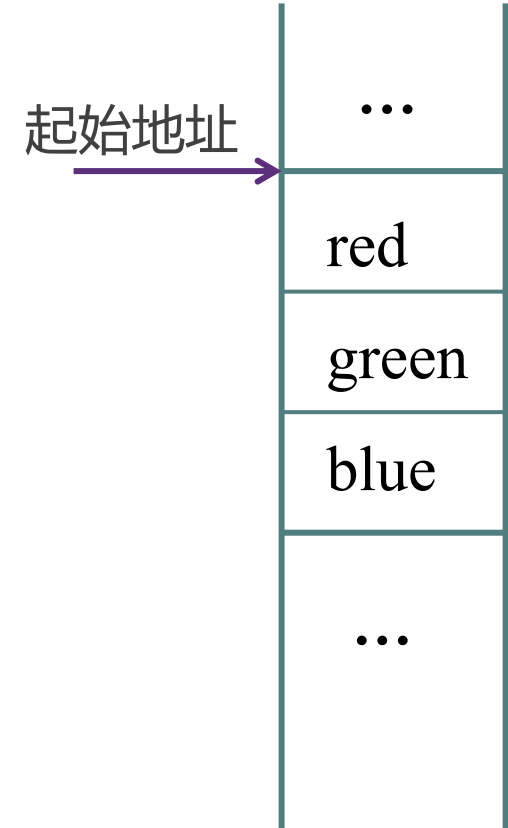


# 理解存储结构

✦ 通常有两种存储结构： 例：(red, green, blue)

(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



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

✦ 通常有两种存储结构： 例：(red, green, blue)

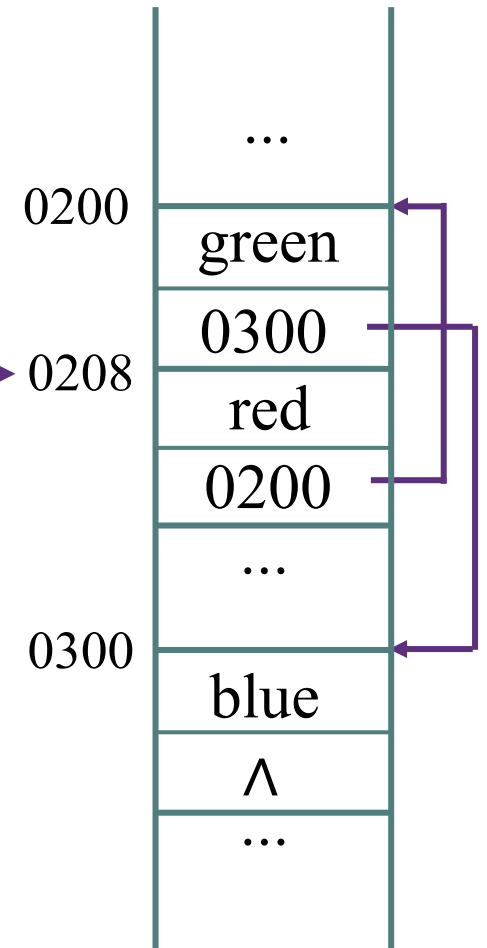
(1) 顺序存储结构：用一组连续的存储单元依次存储数据元素，数据元素之间的逻辑关系由元素的存储位置表示

下标

起始地址

(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



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

## ✦ 链式存储结构：

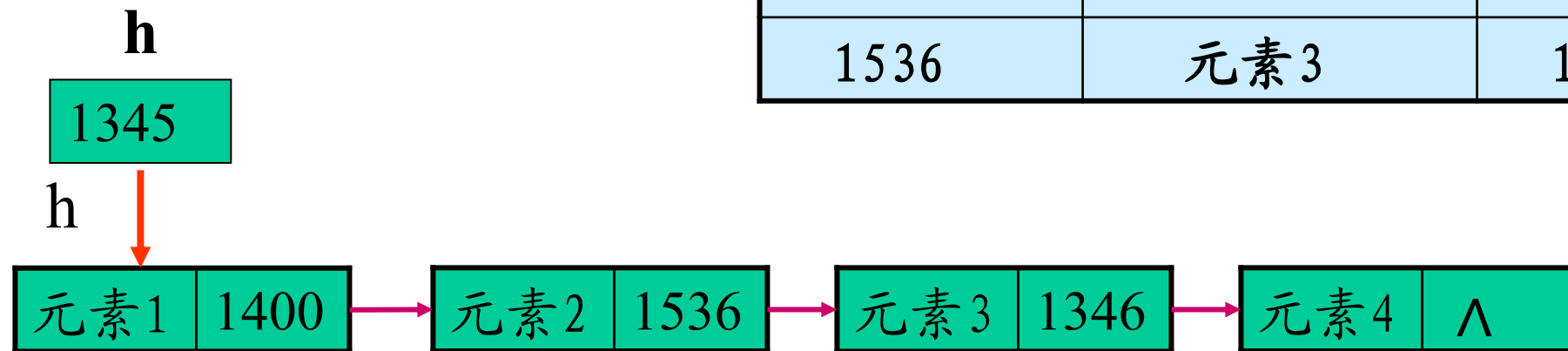
每个节点都由两部分组成：**数据域**和**指针域**。

**数据域**存放元素本身的数据，

**指针域**存放指针。

数据元素之间逻辑上的联系由指针来体现。

存储地址	存储内容	指针
1345	元素1	1400
1346	元素4	Λ
.....	.....	.....
1400	元素2	1536
.....	.....	.....
1536	元素3	1346

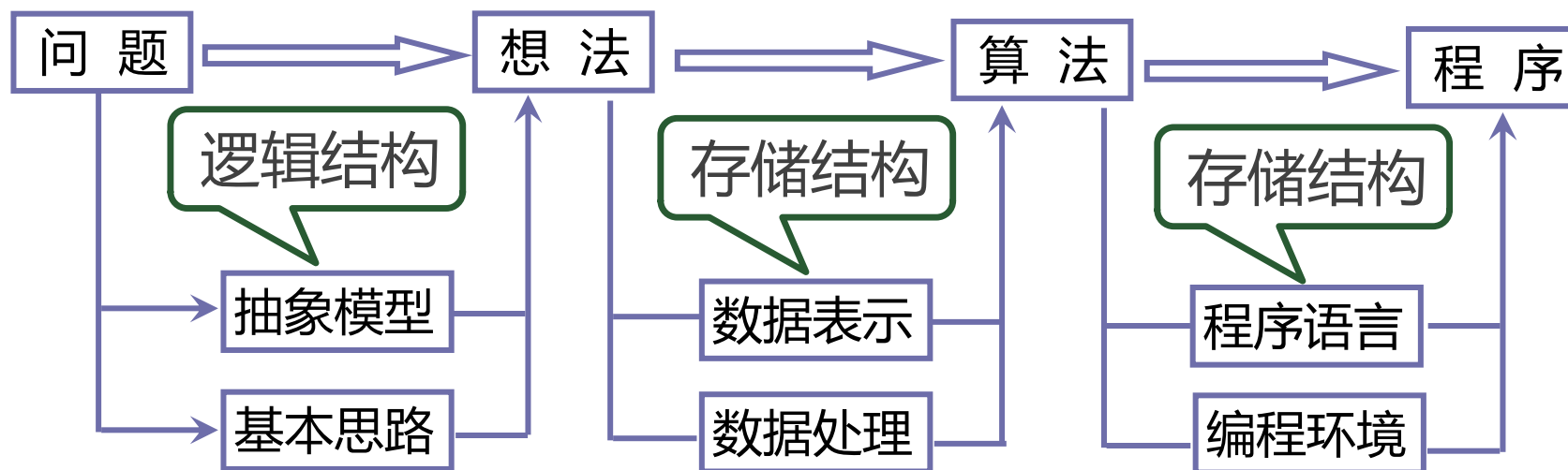


# 逻辑结构和存储结构的关系

- 数据的逻辑结构是**用户视图**，**面向问题**  $\Rightarrow$  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**  $\Rightarrow$  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



## 1.2 数据结构的基本概念

### 1-2-2 抽象数据类型



什么是数据类型？



什么是抽象？







抽象数据类型



# 理解数据类型

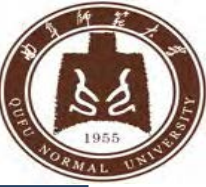
 什么是数据类型呢？

 `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

- ✓ 数据是按照数据结构分类的，具有相同数据结构的数据属同一类。
- ✓ 同一类数据的全体称为一个数据类型。



# 理解数据类型

**数据类型**用来说明一个数据在数据分类中的归属，是数据的一种属性，限定了该数据的变化范围。

**基本数据类型**：int, char, float, double ,

**构造数据类型**：数组、结构体、共用体、枚举等，指针、空(void)类型等。

利用**typedef** 自定义数据类型：      利用 **class** 自定义数据类

```
typedef struct
{
    int    num;
    char   name[20];
    float  score;
}STUDENT;
```

```
STUDENT    stu1,stu2, *p;
```

```
class Box
{
    public:
        double length; // 长度
        double breadth; // 宽度
        double height; // 高度
        // 成员函数声明
        double get(void);
        void set( double len, double bre, double hei );
};
```

# 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？  $\Rightarrow$  在一个更高的层次上思考问题

$$1 + 1 = 2$$

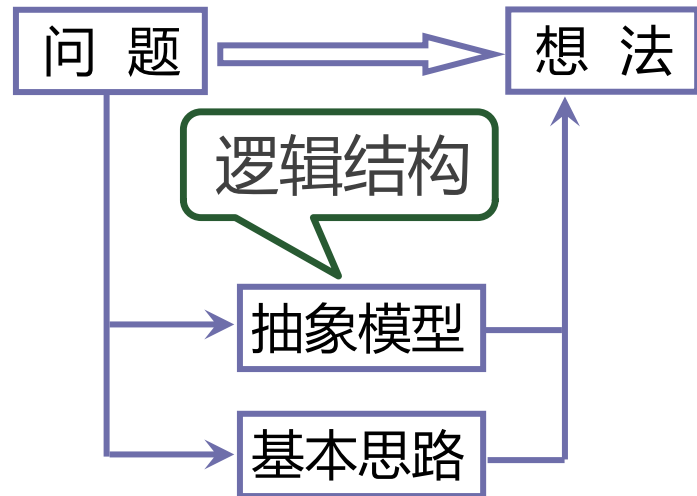
$$2 + 3 = 5 \quad \Rightarrow \quad x + y = z$$

算术运算  $\Rightarrow$  代数运算

# 抽象数据类型

🕒 抽象数据类型把什么**抽象**掉了呢？

📌 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**



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**

# 抽象数据类型

🕒 抽象数据类型把什么**抽象**掉了呢？

📌 **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

 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

## 如何定义抽象数据类型呢？

ADT 抽象数据类型名

DataModel

数据元素之间逻辑关系的定义

Operation

操作 1

输入：执行此操作所需要的输入

功能：该操作将完成的功能

输出：执行该操作后产生的输出

操作接口（函数原型）

操作 2

.....

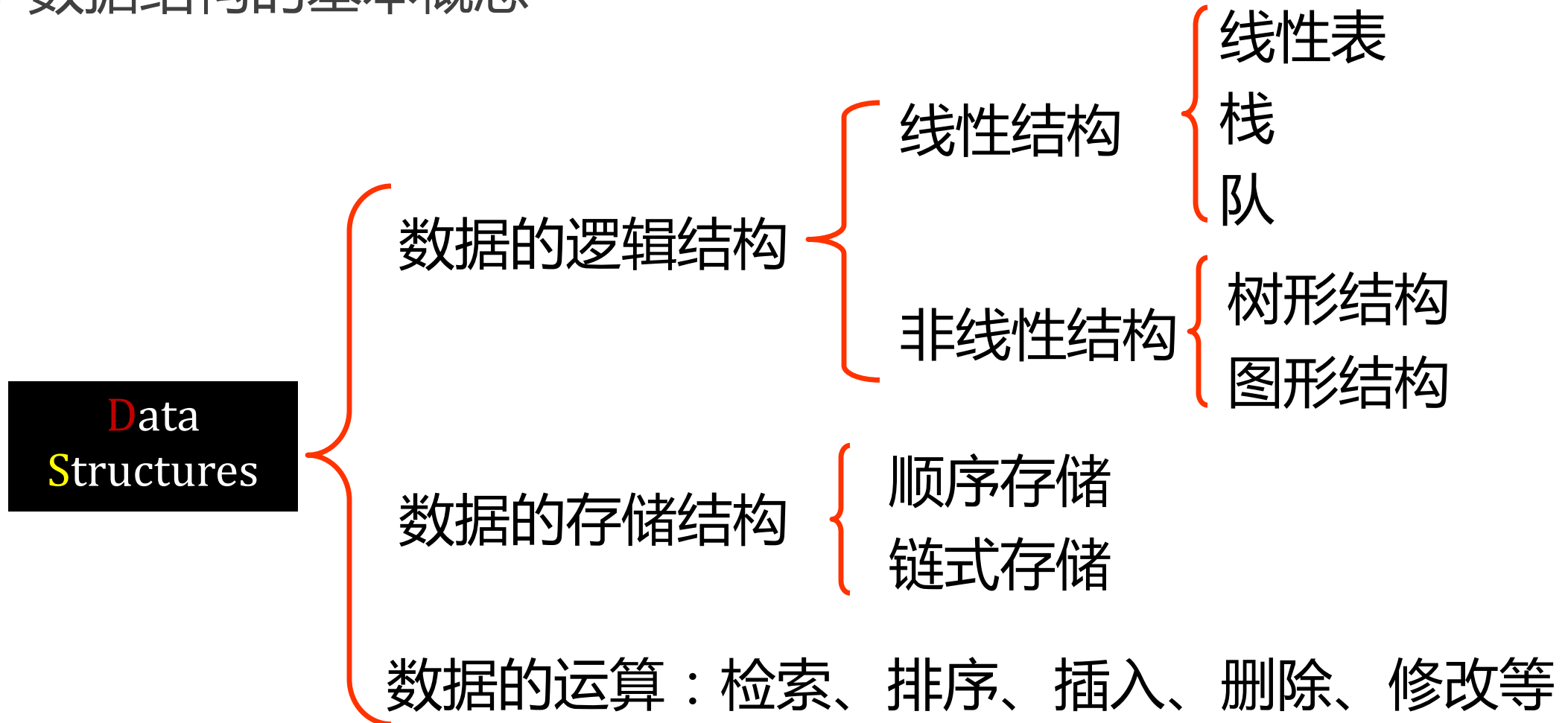
操作  $n$

.....

endADT

# 小结

## 📜 数据结构的基本概念





*Thank You !*

*Q & A*