



Data Structures

绪论 Introduction

张明强

2024年8月27日

学而不厌 诲人不倦

► 培养计划

Data Structures

课程名称：数据结构

英文名称：Data Structures

课程代码：302006

开设专业：物联网工程，网络空间安全

课程类型：专业核心课程

课程学分：4+1

先行课程：《高等数学》（301001）、《计算机系统概论》（300006）、《C语言程序设计》（301010）

内容简介：《数据结构》是物联网、网络空间安全等专业的专业核心课程。本课程主要介绍线性表、栈和队列、串与数组、树、图等基本数据结构，以及经典检索、排序算法。通过本课程的学习，掌握常用数据结构的基本概念、逻辑结构、存储结构及其基本运算特点和典型应用场景，能够熟练使用基础数据结构进行算法程序设计。

—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**等的学科

研究数据**组织**、**存储**和**运算**的一般方法的学科

➤ 从科学研究角度

研究数据的特性、数据之间的关系（逻辑结构）及其存储表示（物理结构）、数据的操作方法（算法）。

➤ 从知识储备角度

计算机基础、C/C++语言程序设计、高等数学、概率论

➤ 从后续课程角度

**矩阵分析、信号处理与通信（图像、语音、视频等）信息理论
操作系统、编译原理、数据库管理系统、软件工程、人工智能**

➤ 以发展的眼光

数据结构这门课可以帮助你在计算机科学、信息科学等领域的道路上走的更远一些……



主要参考教材

➤ 教学内容

1. 数据的逻辑结构
2. 数据的物理存储结构
3. 对数据的操作方法（算法）

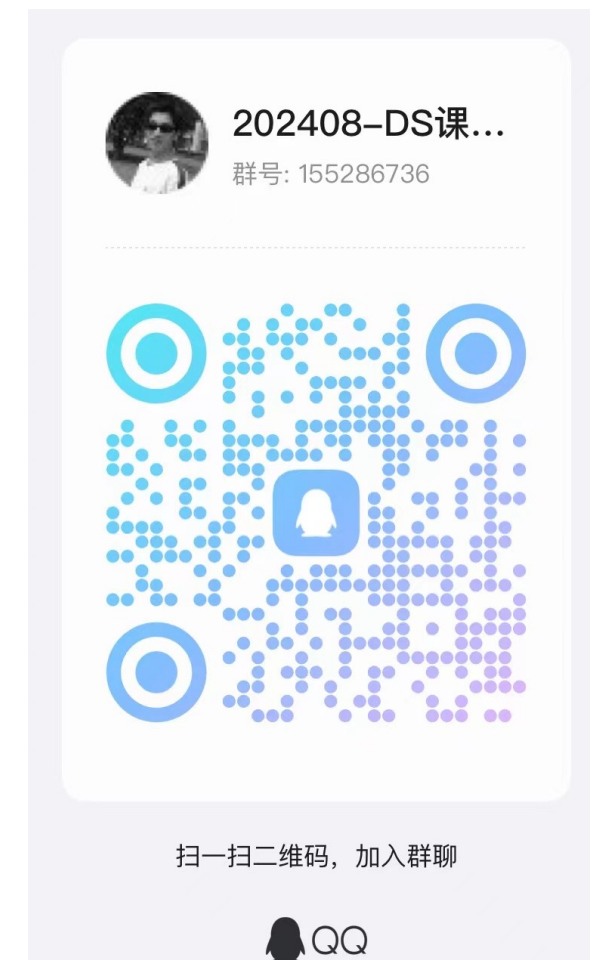
➤ 课程目标

1. 掌握基本的数据结构和算法，会复用、修改、重组
2. 提高算法设计能力、程序设计能力
问题求解过程：问题→想法→算法→程序
程序设计研究的层次：算法→方法学→语言→工具
3. 培养和训练算法分析能力，会评价算法、改进算法

➤ 考核方式

课堂教学（含作业）20%、实验教学与报告 20%
期中测试 20%、 期末测试40%

课程QQ群：155286736



- ➡ **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**

1.1 问题的求解与程序设计

1-1-1 程序设计的一般过程

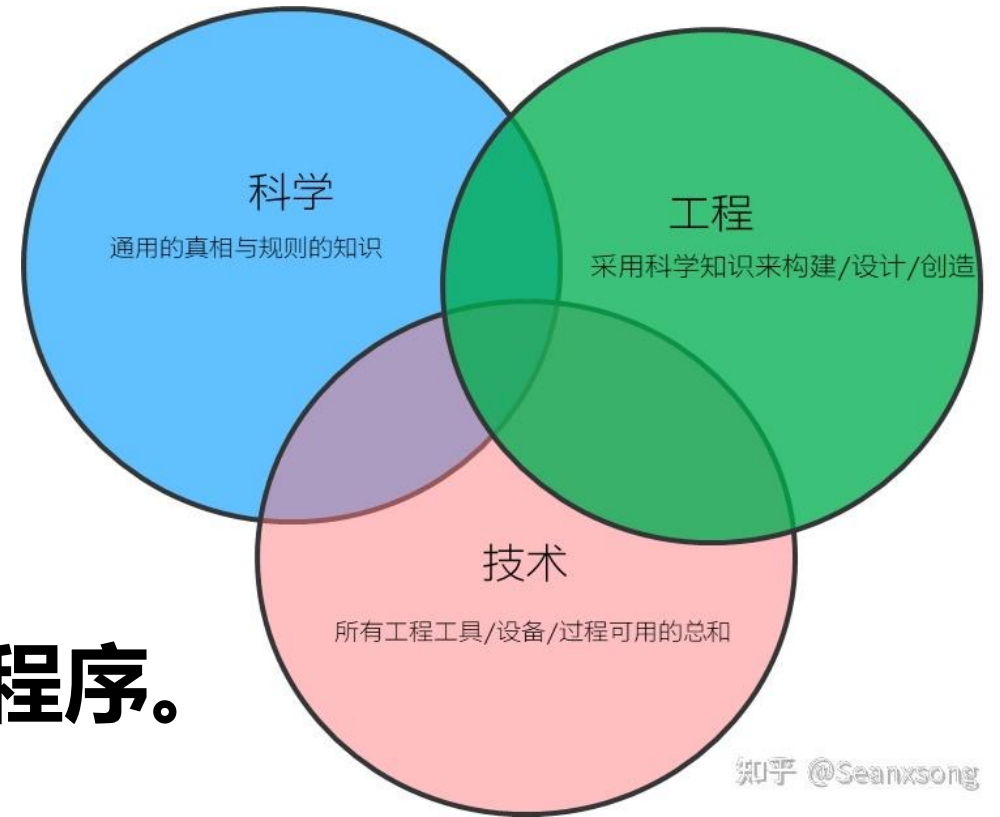
为什么要写程序？

科学： 是什么？ 为什么？

技术： 怎么做？

工程： 怎么做的多快好省？

应用： 直接重复地解决用户问题的程序。

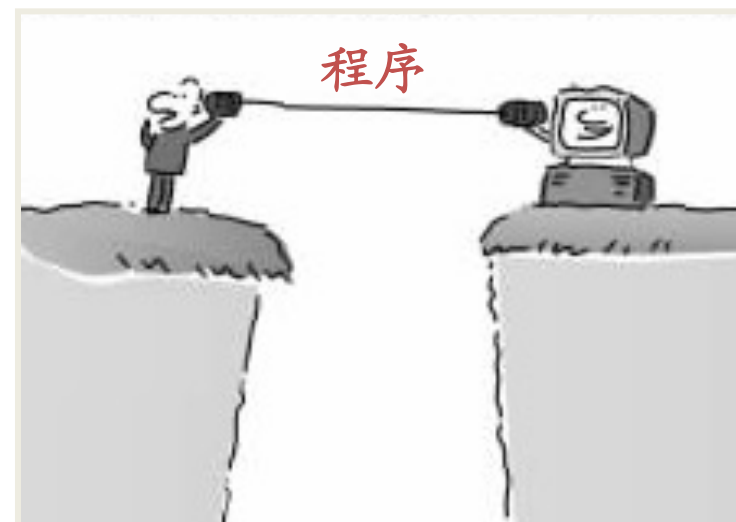


为什么要写程序？

🕒 为什么要写程序？程序有什么用呢？



二进制的
数字世界



人要和计算机有效地交流，必须通过程序

程序=算法+数据结构

软件=程序+文档

➤ 问题分析与描述

Problem Analysis and Specification

➤ 设计

Design

➤ 编码

Implementation (Coding)

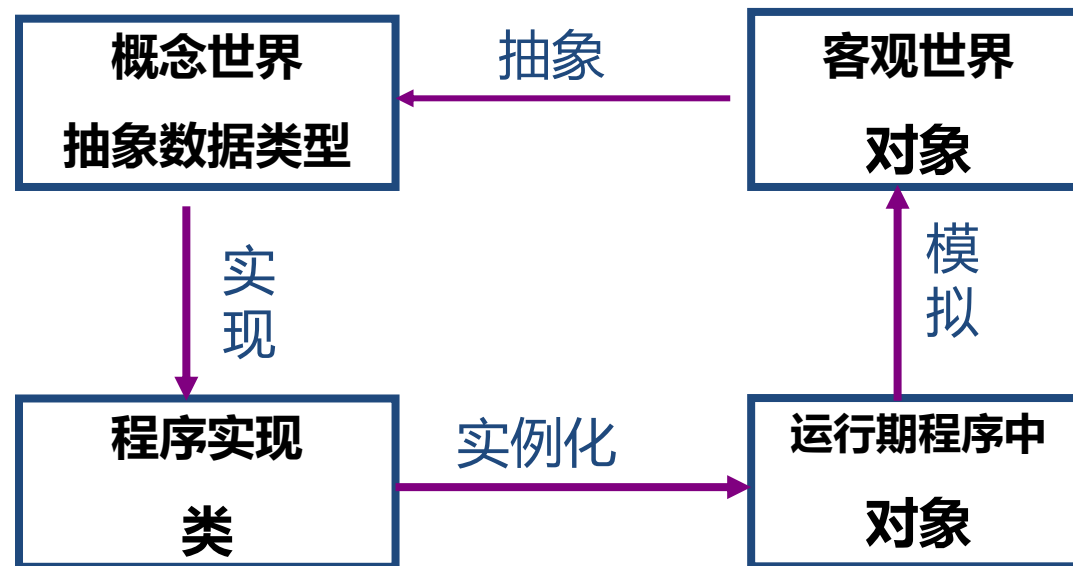
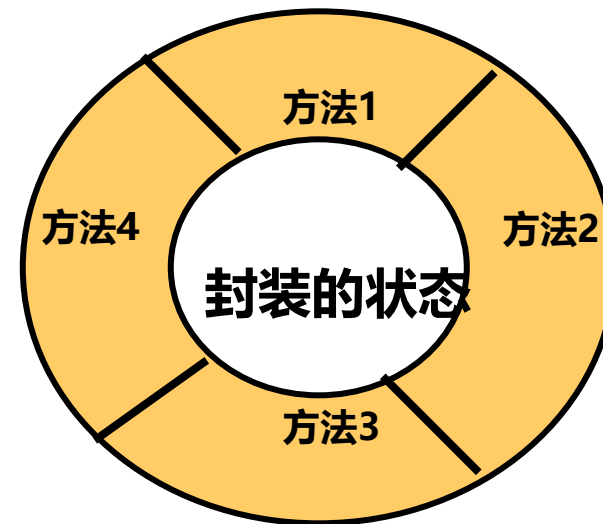
➤ 软件测试、调试

Testing, Execution and Debugging

➤ 软件维护

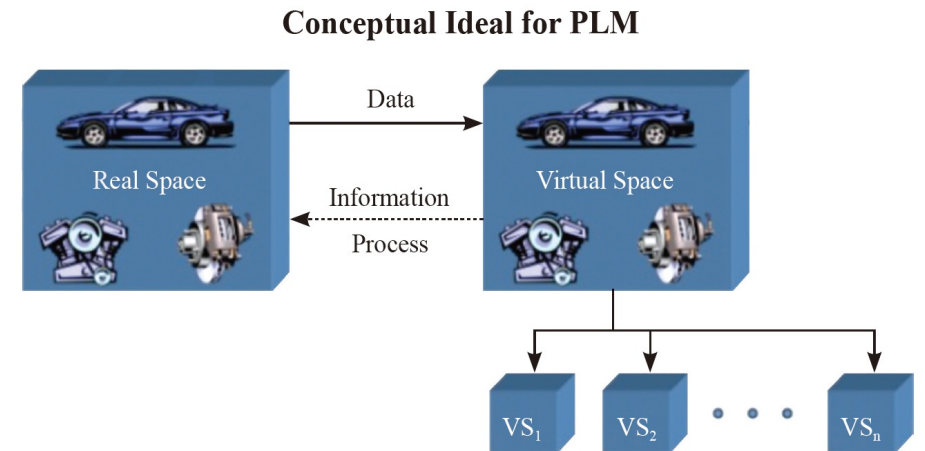
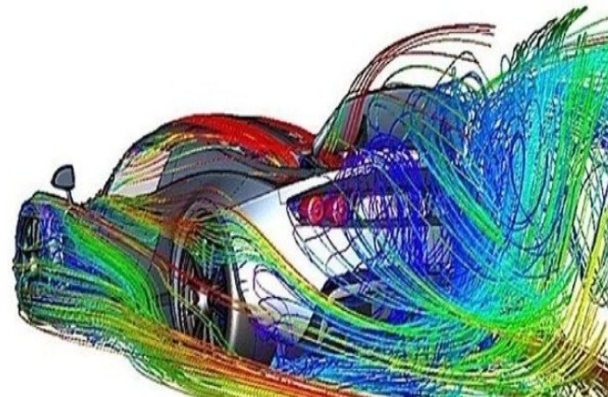
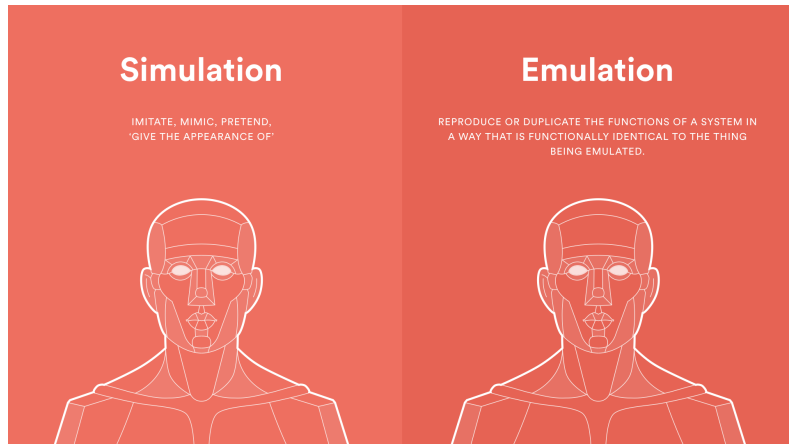
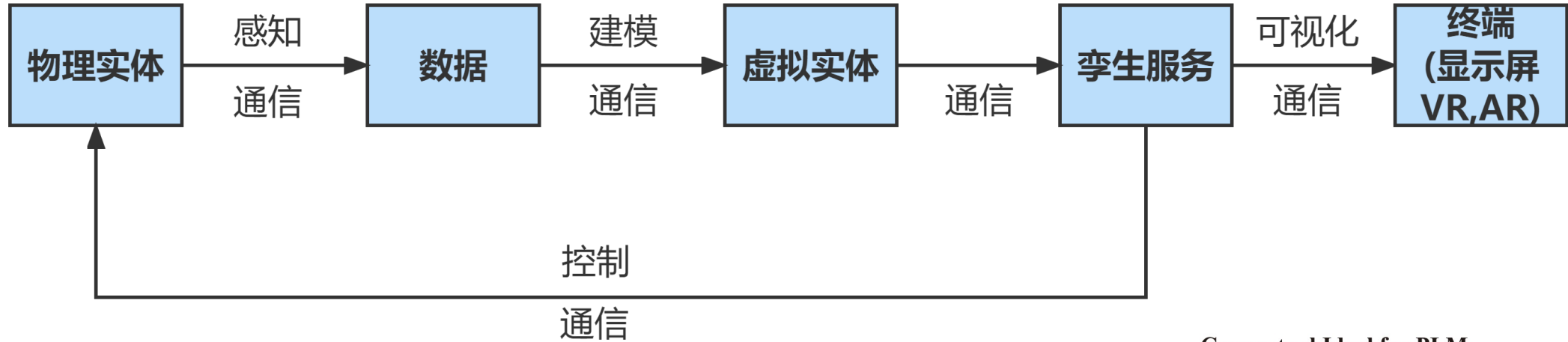
Maintenance

以对象为中心的设计



为什么要写程序？

对现实世界事物或问题的模拟和仿真，探究事物的一般规律或演进过程。



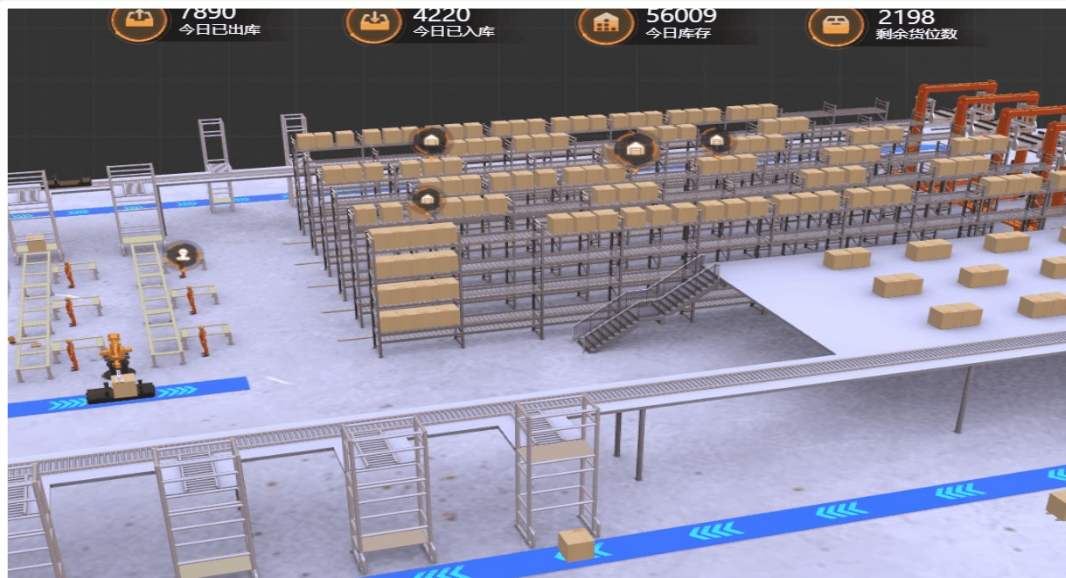
Product Lifecycle Management

为什么要写程序？

➤ 数字孪生助力打造智慧仓储中心，极大地提高了仓库空间利用率和经营效率



➤ 数字孪生系统与物理仓储中心实时数据交互并记录，可视化组件将数据生动展示，使数据呈现更加清晰



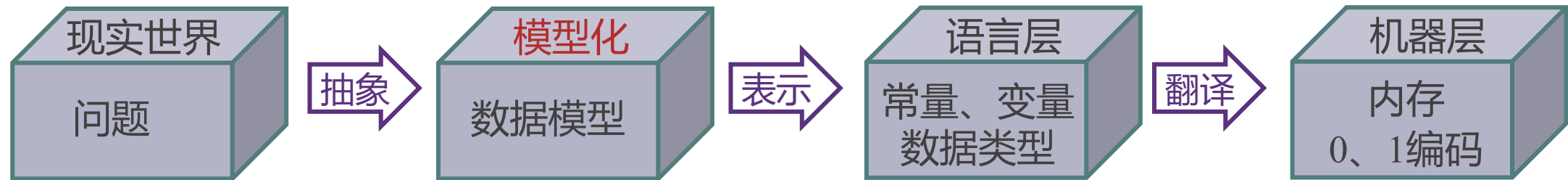
➤ 库房环境、货型、货量一目了然，能耗、人工、吞吐量清晰明白，帮助仓储负责人作出正确空间规划



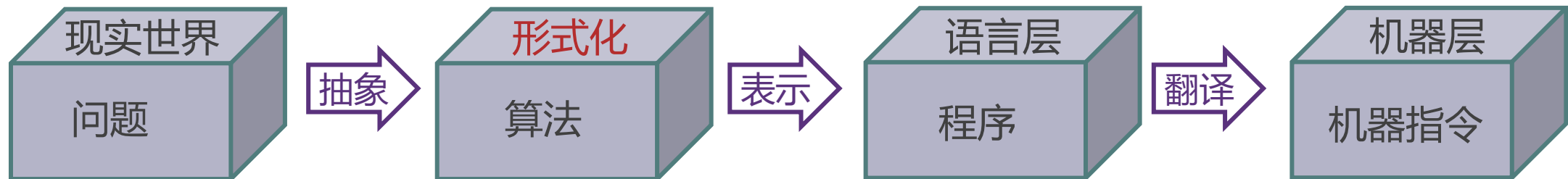
程序设计的关键

🕒 程序设计的关键是什么？

📌 数据表示：从问题抽象出**数据模型**，从机外表示转换为**机内表示**

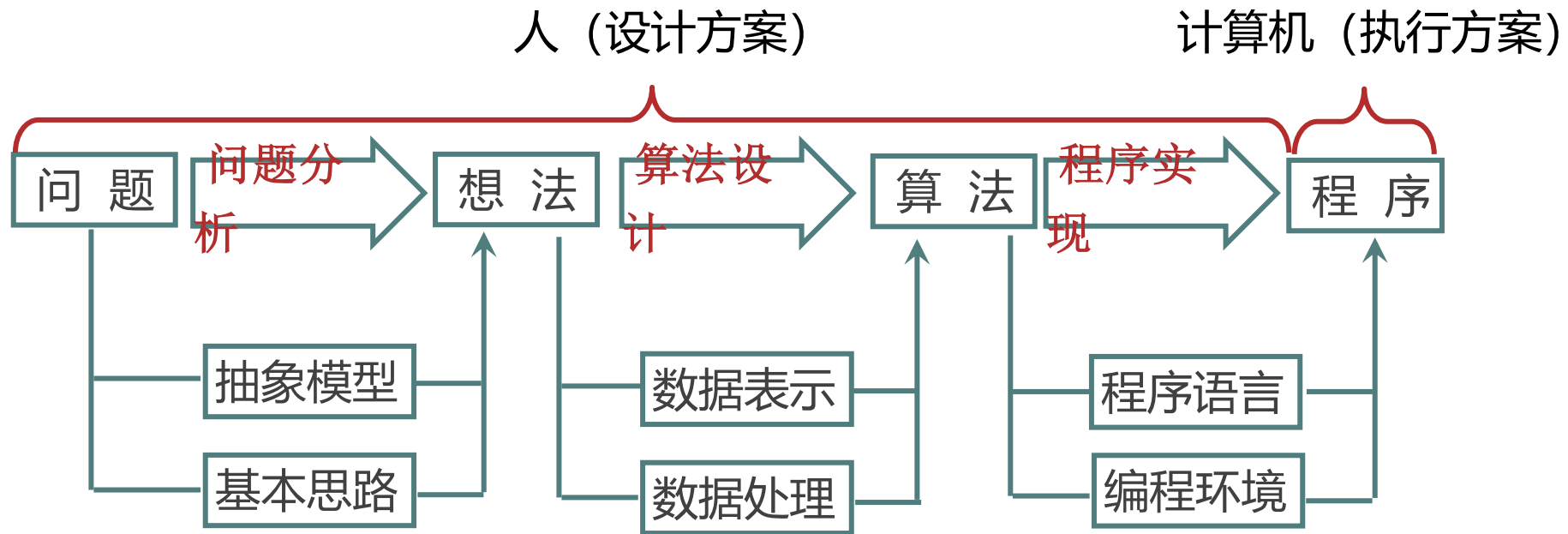


📌 数据处理：设计**算法**，再将算法转换为程序设计语言对应的**程序**



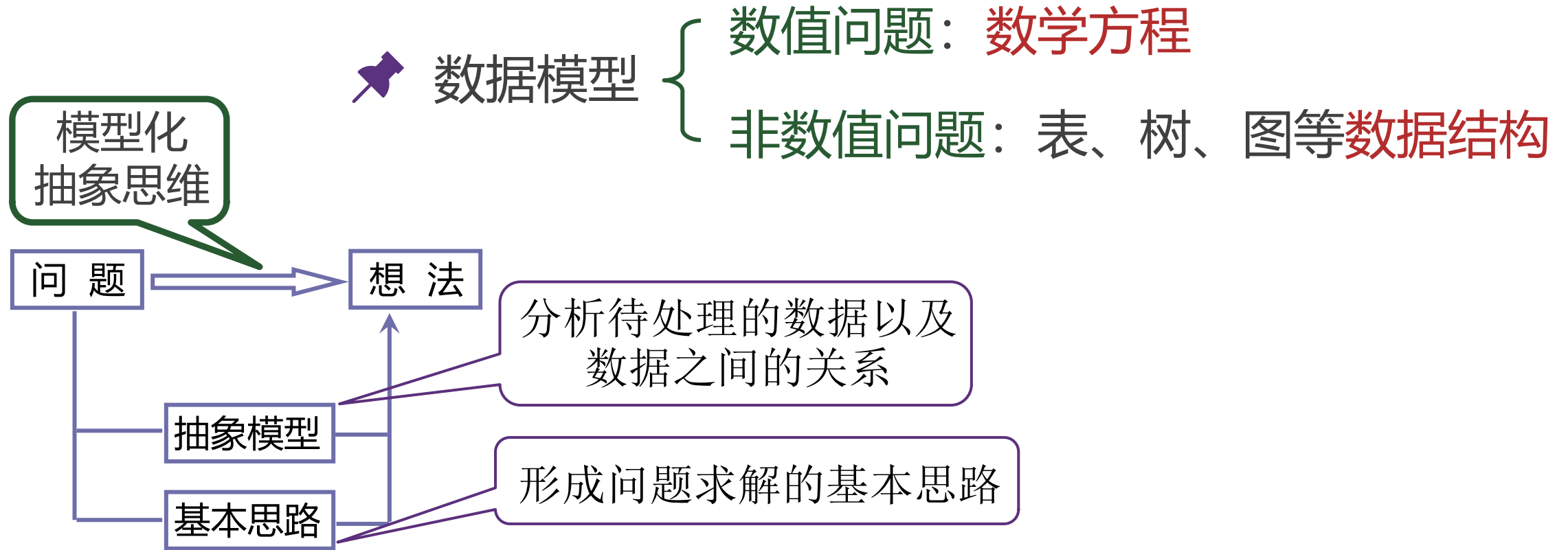
程序设计的一般过程

🕒 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

程序设计的一般过程

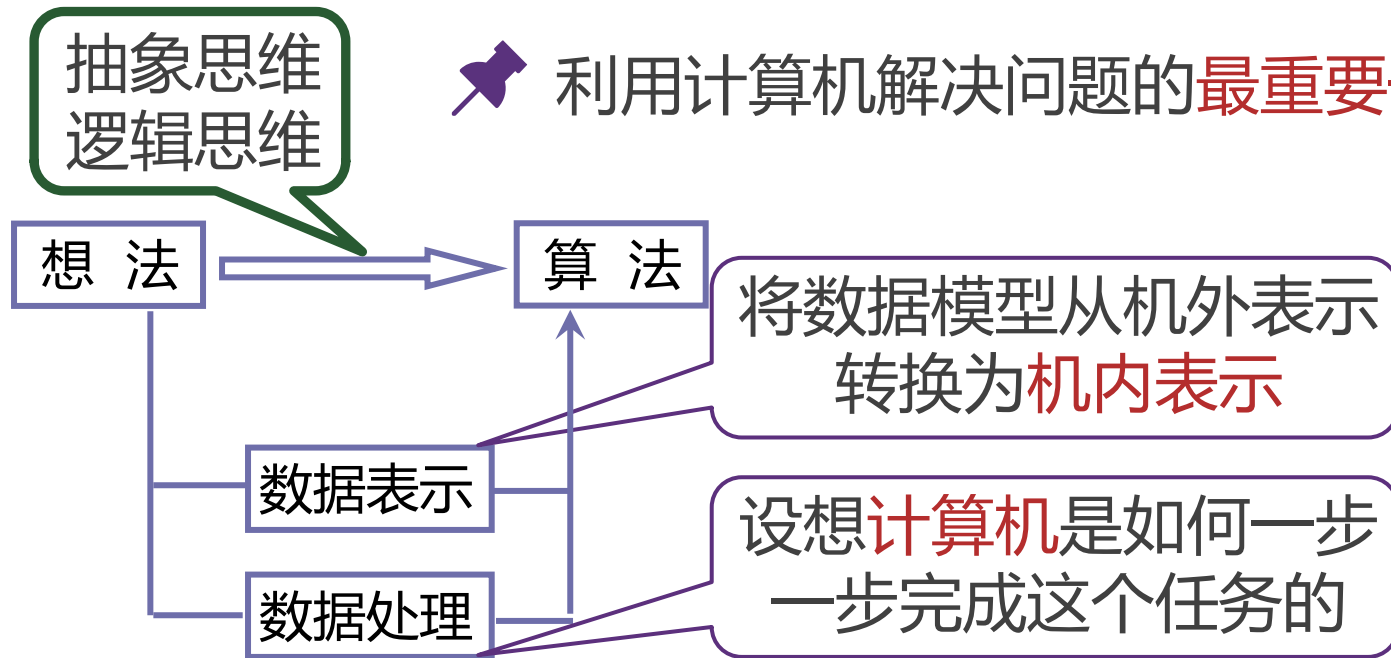


计算思维: 模型化、形式化、逻辑思维、抽象思维

程序设计的一般过程

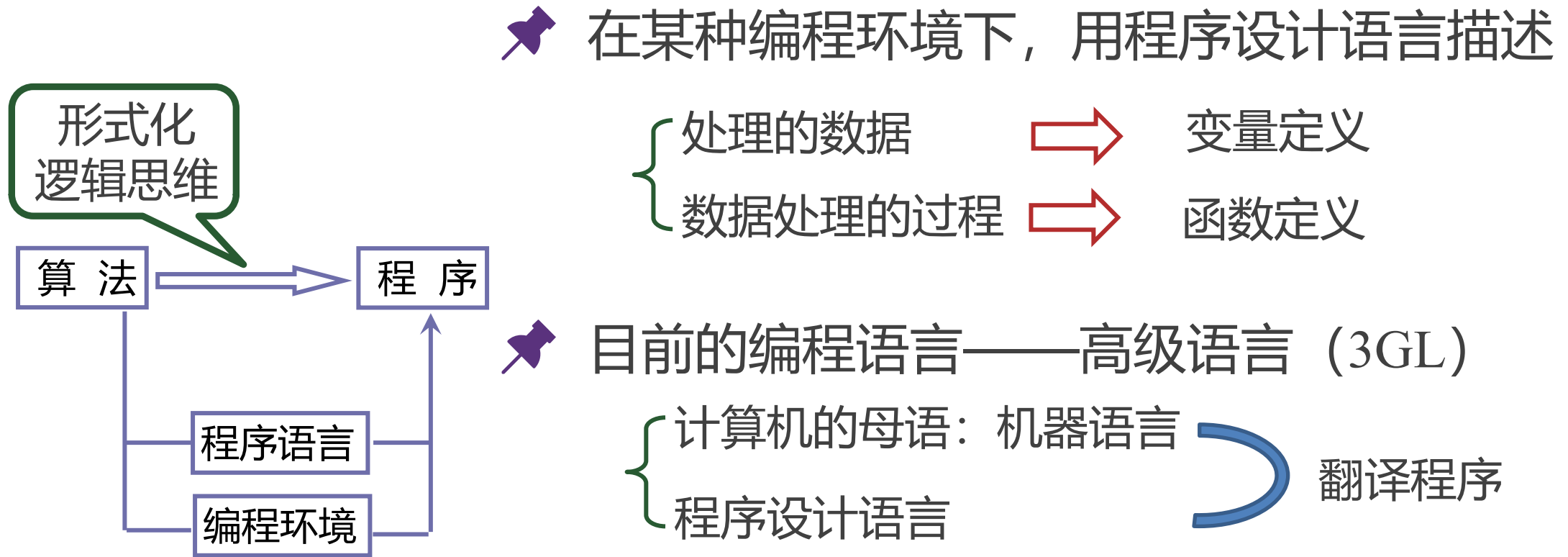
✦ 算法用来描述问题的解决方案，是具体的、机械的**操作步骤**

✦ 利用计算机解决问题的**最重要一步**是将人的想法描述成算法



计算思维：模型化、形式化、**逻辑思维**、**抽象思维**

程序设计的一般过程

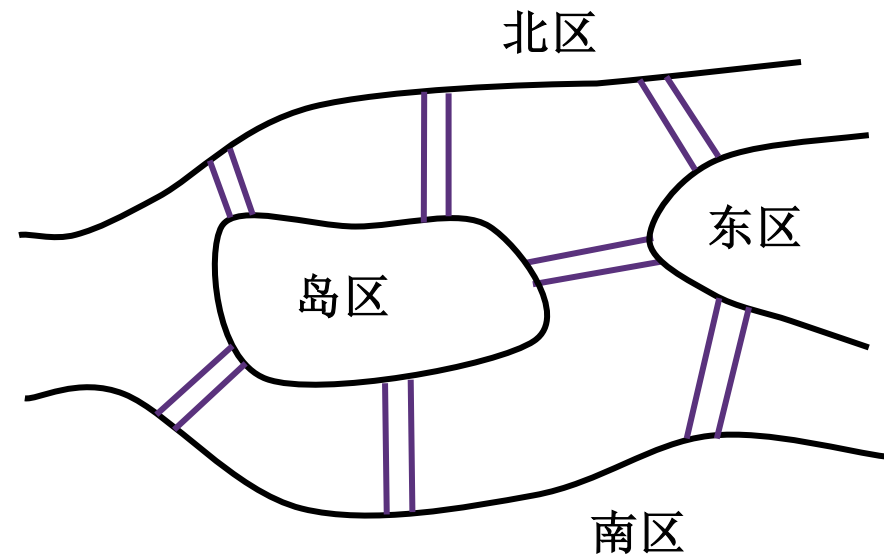


计算思维：模型化、**形式化**、**逻辑思维**、抽象思维

程序设计的一般过程

例 1 哥尼斯堡七桥问题

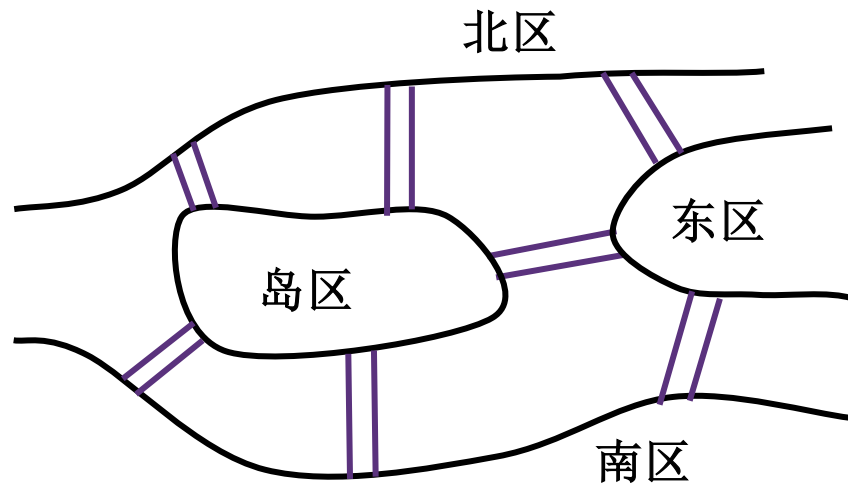
【问题】 17世纪的东普鲁士有一座哥尼斯堡城（现在叫加里宁格勒，在波罗的海南岸），城中有一座岛，普雷格尔河的两条支流环绕其旁，并将整个城市分成北区、东区、南区和岛区 4 个区域，全城共有七座桥将 4 个城区连接起来，于是，产生了一个有趣的问题：一个人是否能在一次步行中穿越全部的七座桥后回到起点，且每座桥只经过一次。



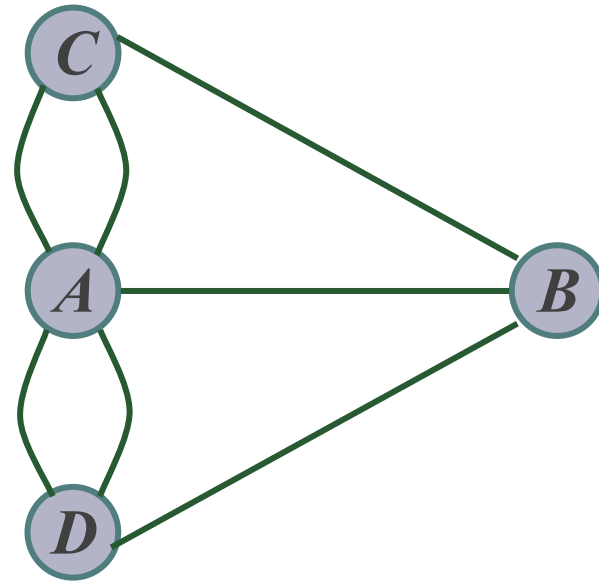
程序设计的一般过程

例 1 哥尼斯堡七桥问题 欧拉回路：经过图中每条边一次且仅一次的回路

【想法——抽象模型】 可以用A、B、C、D表示4个城区，用7条线表示7座桥，将七桥问题抽象为一个图模型



抽象



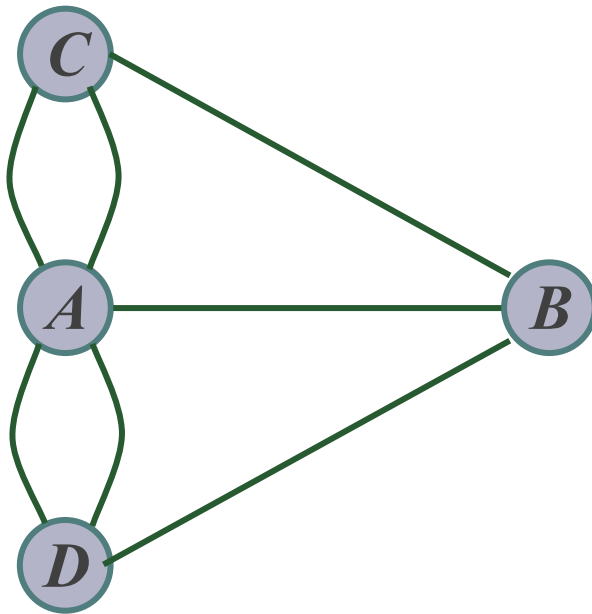
【想法——基本思路】 是否存在欧拉回路的判定规则是：

- (1) 如果通奇数桥的地方多于两个，则不存在欧拉回路；
- (2) 如果没有一个地方通奇数桥，则无论从哪里出发，都能找到欧拉回路。

程序设计的一般过程

例 1 哥尼斯堡七桥问题

【算法——数据表示】 设二维数组 $\text{arc}[n][n]$ 存储七桥问题的图模型



	A	B	C	D
A	0	1	2	2
B	1	0	1	1
C	2	1	0	0
D	2	1	0	0

程序设计的一般过程

例 1 哥尼斯堡七桥问题

【算法——抽象算法】 将求解七桥问题的关键（求与每个顶点相关联的边数）独立出来，设计具体的求解步骤，即设计算法

算法: EulerCircuit

输入: 二维数组 $\text{mat}[4][4]$

输出: 通奇数桥的顶点个数 count

1. count 初始化为 0;
2. 下标 i 从 $0 \sim n-1$ 重复执行下述操作:
 - 2.1 计算第 i 行元素之和 degree ;
 - 2.2 如果 degree 为奇数, 则 $\text{count}++$;
3. 返回 count ;

	A	B	C	D
A	0	1	2	2
B	1	0	1	1
C	2	1	0	0
D	2	1	0	0

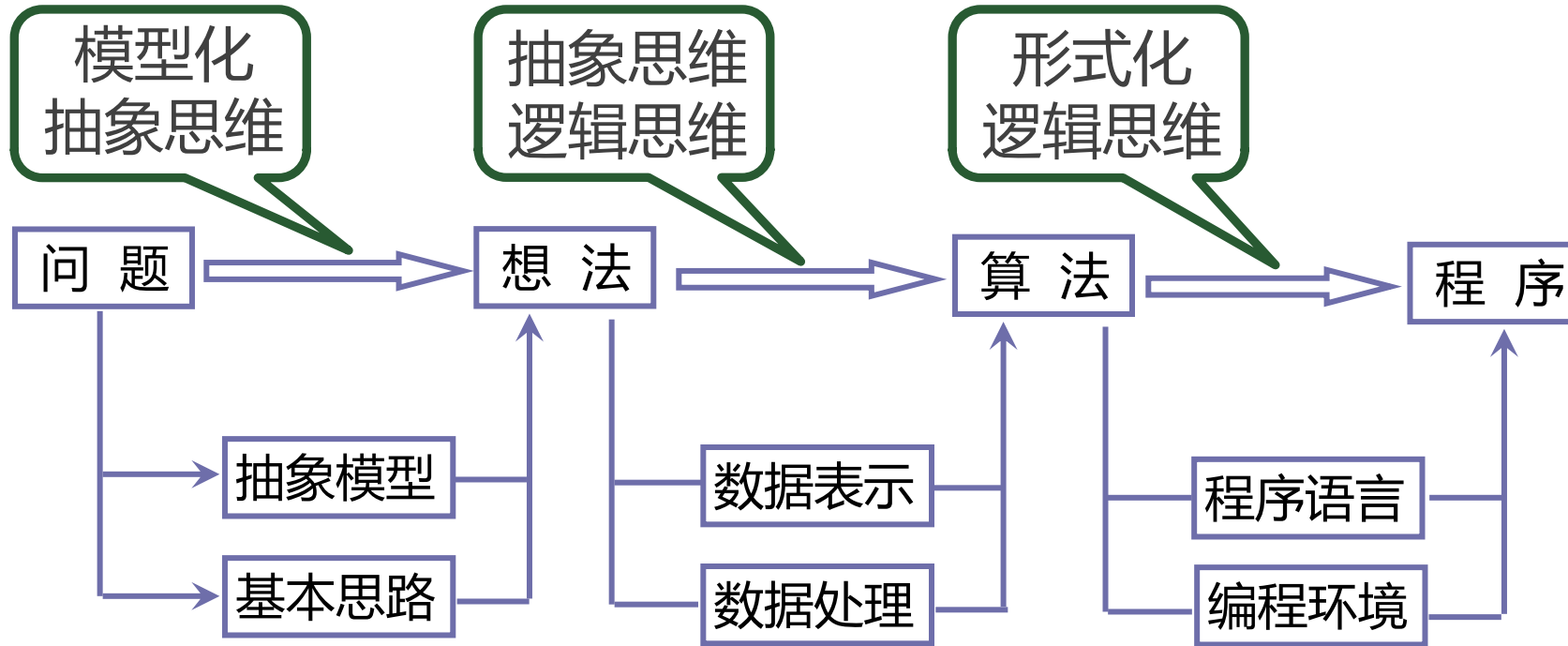
程序设计的一般过程

例 1 哥尼斯堡七桥问题

【程序】 将算法用 C++ 语言的函数定义进行描述

```
int EulerCircuit(int mat[4][4], int n)          /*函数定义，二维数组作为形参*/
{
    int i, j, degree, count = 0;
    for (i = 0; i < n; i++)                      /*依次累加每一行的元素*/
    {
        degree = 0;
        for (j = 0; j < n; j++)
            degree = degree + mat[i][j];          /*将通过顶点i的桥数求和*/
        if (degree % 2 != 0) count++;             /*桥数为奇数*/
    }
    return count;                                /*结束函数，将count返回到调用处*/
}
```

程序设计的一般过程



问题→想法→算法→程序，正是**计算思维**的运用过程

算法训练就像**思维体操**，逐渐提高计算思维能力

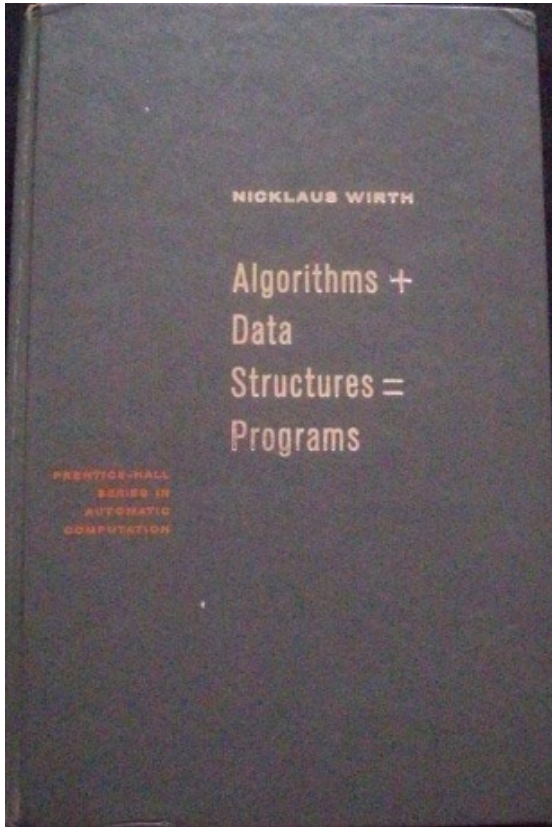
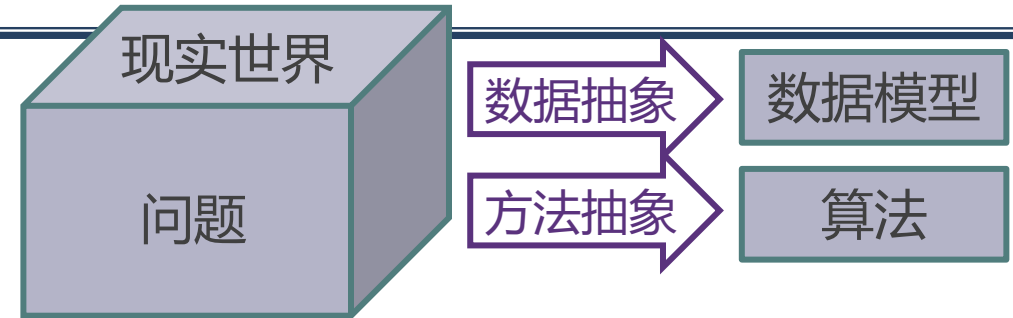
1.1 问题的求解与程序设计

1-1-2 数据结构在程序设计中的作用

编写好程序



怎么才能设计出好的程序呢？



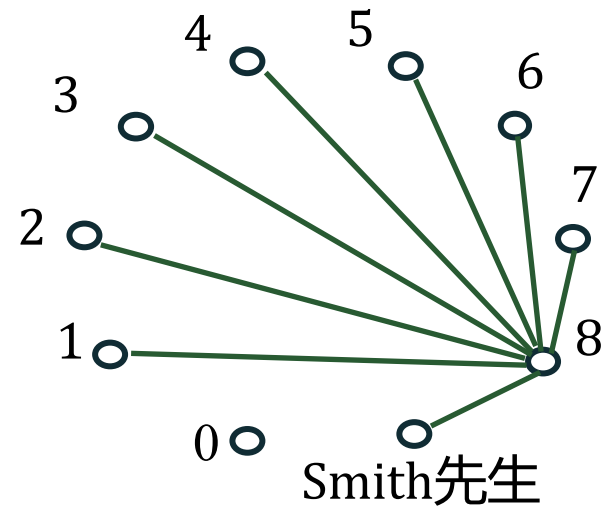
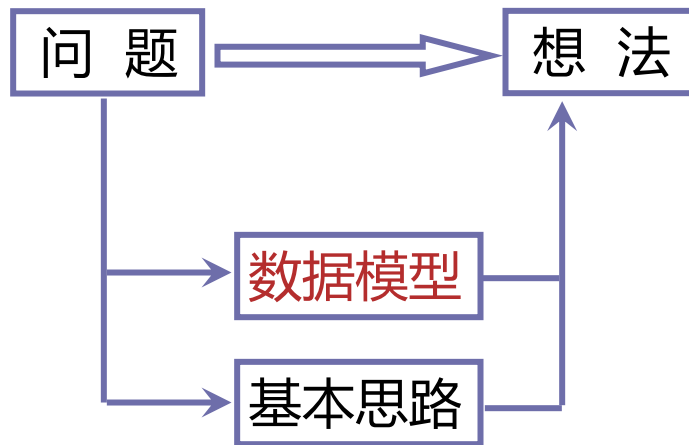
沃思
Wirth

计算机领域人尽皆知的名言
算法 + 数据结构 = 程序

Algorithm + Data Structures = Programs

数据结构的作用

【握手问题】 Smith先生和太太邀请四对夫妻来参加晚宴。每个人来的时候，房间里的一些人都要和其他人握手。当然，每个人都不会和自己的配偶握手，也不会跟同一个人握手两次。之后，Smith先生问每个人和别人握了几次手，他们的答案都不一样。问题是，Smith太太和别人握了几次手？



如果能够将问题抽象出一个合适的数据模型，则问题可能会变得豁然开朗

数据结构的作用

🚀 如果能将问题抽象出一个合适的数据模型，则问题可能会变得豁然开朗

📜 百元买百鸡问题

$$\begin{cases} x + y + z = 100 \\ 5 \times x + 3 \times y + z / 3 = 100 \end{cases}$$

📜 Fibonacci数列问题

$$f(n) = \begin{cases} 1 & n = 1 \\ 1 & n = 2 \\ f(n - 1) + f(n - 2) & n > 2 \end{cases}$$

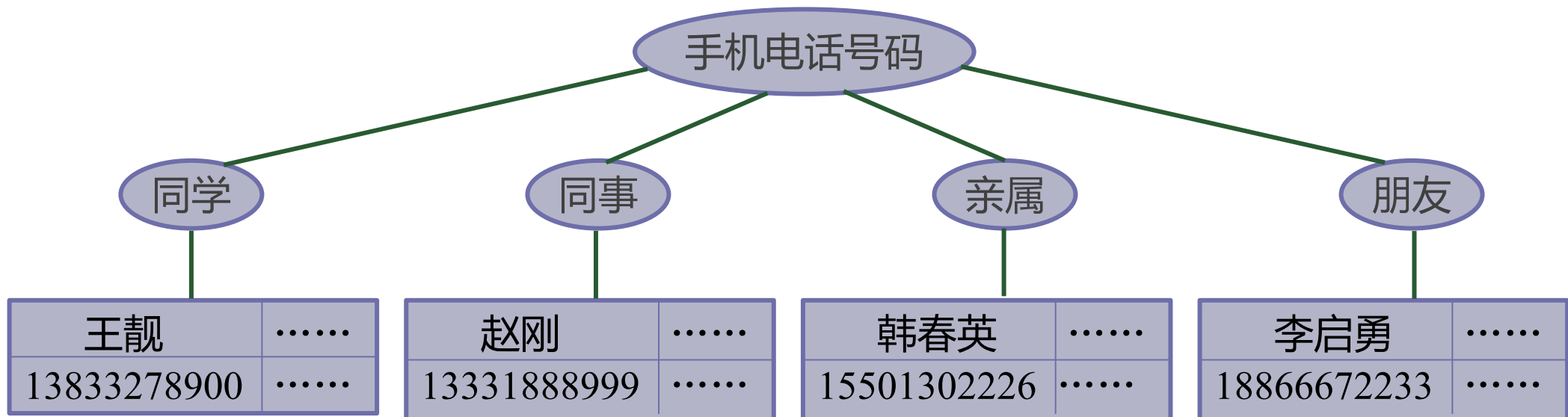
📜 更复杂的问题：人口增长、桥梁应力、股票预测、……

数据结构的作用

基于不同数据模型的算法，其运行效率可能会有很大差别

【电话号码查询问题】 假设某手机中存储了若干电话号码，如何在手机中查找某人的电话号码？

姓 名	王 靓	赵 刚	韩春英	李启勇
电 话	13833278900	13331888999	15501302226	18866672233



1.1 问题的求解与程序设计

1-1-3 算法在程序设计中的作用

算法 + 数据结构 = 程序

Algorithm + Data Structures = Programs

算法的作用

【排序问题】 对整型数组 $r[n]$ 进行非降序排列。

【算法】 有很多排序算法可以解决这个问题，不同排序算法的运行时间有很大差别，起泡排序（请参见8.3.1节）和快速排序（请参见8.3.2节）在不同数据规模的运行时间如下表所示，随着数据规模的增长，起泡排序和快速排序运行时间的差别越来越大。

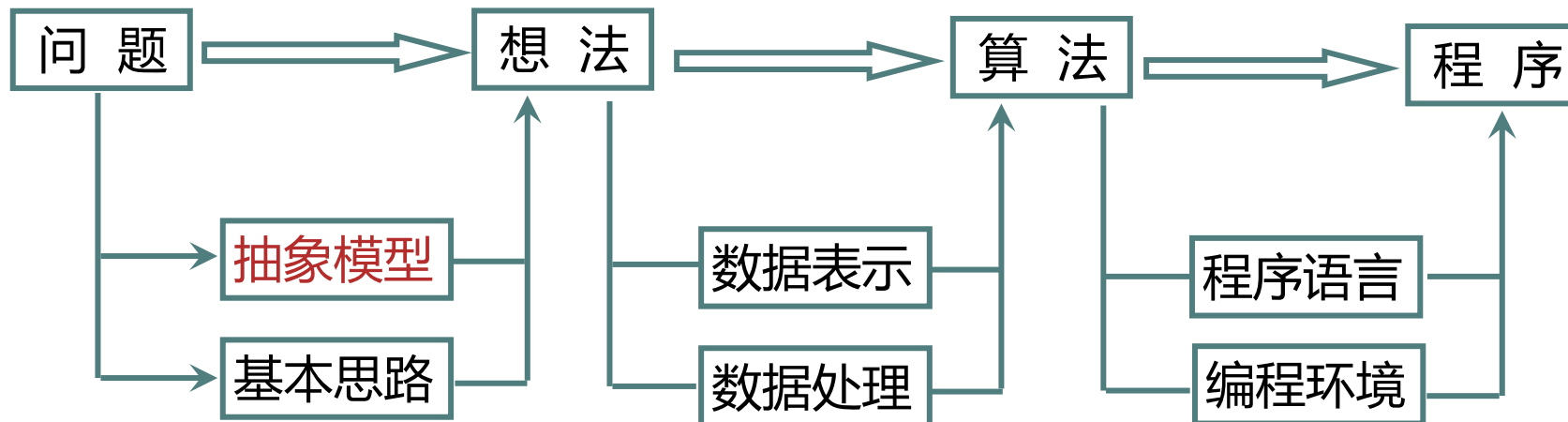
表 1-2 起泡排序和快速排序的运行时间（单位：ms）

数据规模	1000	10 000	100 000	1000 000	10 000 000
起泡排序	3	395	40 276	4 158 440	>100 小时
快速排序	0	1	18	234	5187

1.1 问题的求解与程序设计

1-1-4 本课程讨论的主要内容

数据模型 { 数值问题: 数学方程
非数值问题: 表、树、图等数据结构



例 1 为百元买百鸡问题抽象数据模型（《算经》）

【问题】 鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。
百钱买百鸡，问鸡翁、母、雏各几何？

【想法——数据模型】 设 x 、 y 和 z 分别表示公鸡、母鸡和小鸡的个数，
则有如下方程组成立：

$$\begin{cases} x + y + z = 100 \\ 5 \times x + 3 \times y + z / 3 = 100 \end{cases}$$

$$\begin{cases} 0 \leq x \leq 20 \\ 0 \leq y \leq 33 \\ 0 \leq z \leq 100 \end{cases}$$

例 2 为学籍管理问题抽象数据模型

🕒 如何实现增、删、改、查等功能？ ➡ 数据表示——存储表

🕒 各表项之间是什么关系？ ➡ 表结构

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

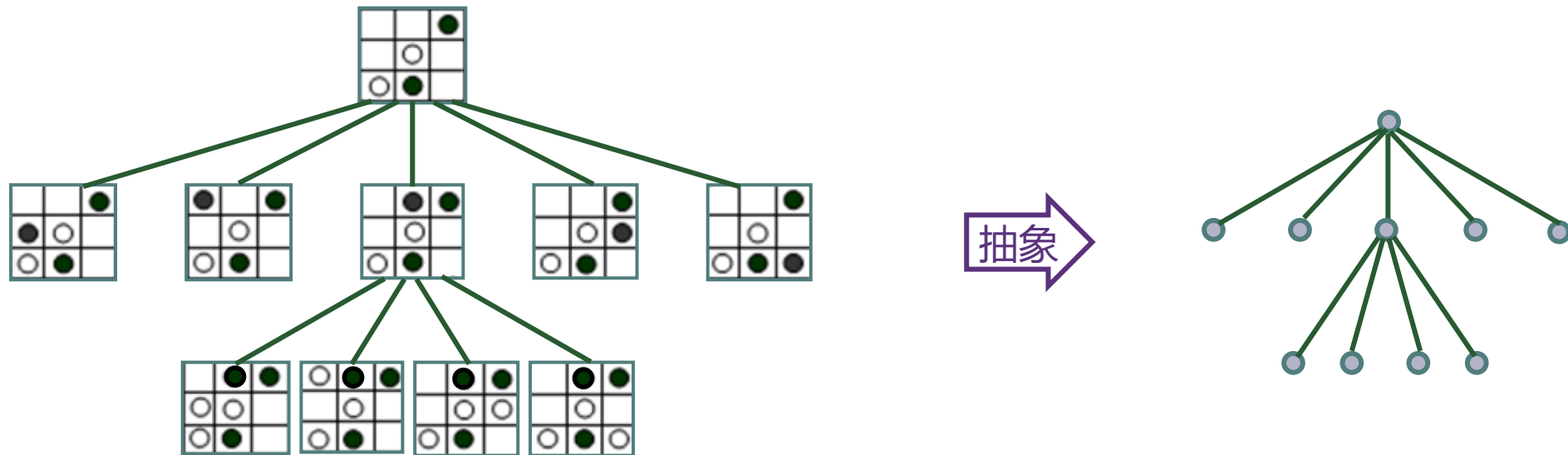


数据模型

例 3 为人机对弈问题抽象数据模型

🕒 如何实现对弈？ \Rightarrow 数据表示——存储对弈树

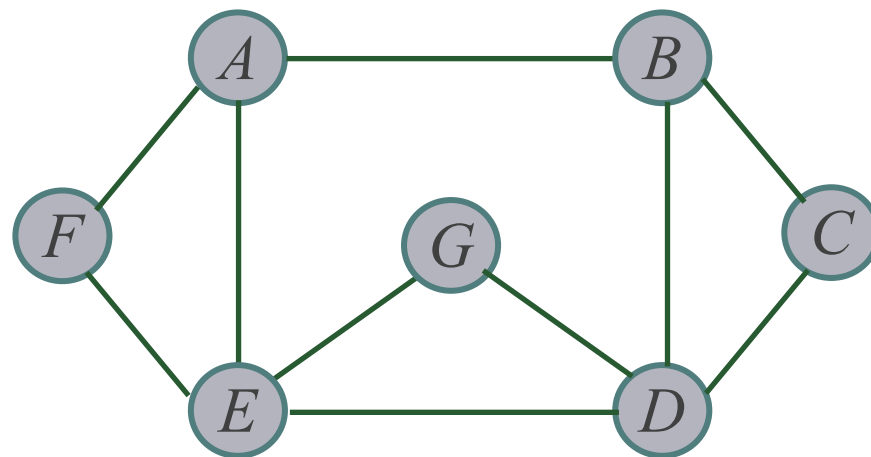
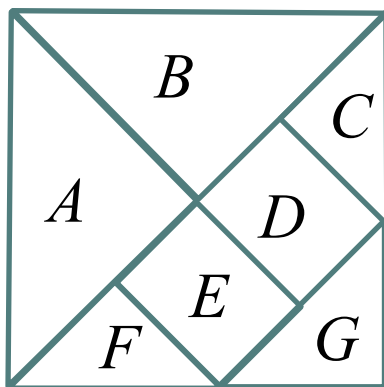
🕒 计算机的操作对象？各格局之间是什么关系？ \Rightarrow 树结构



例 4 为七巧板涂色问题抽象数据模型

🕒 如何实现涂色? \Rightarrow 数据表示——存储七巧板

🕒 如何表示区域之间的邻接关系? \Rightarrow 图结构

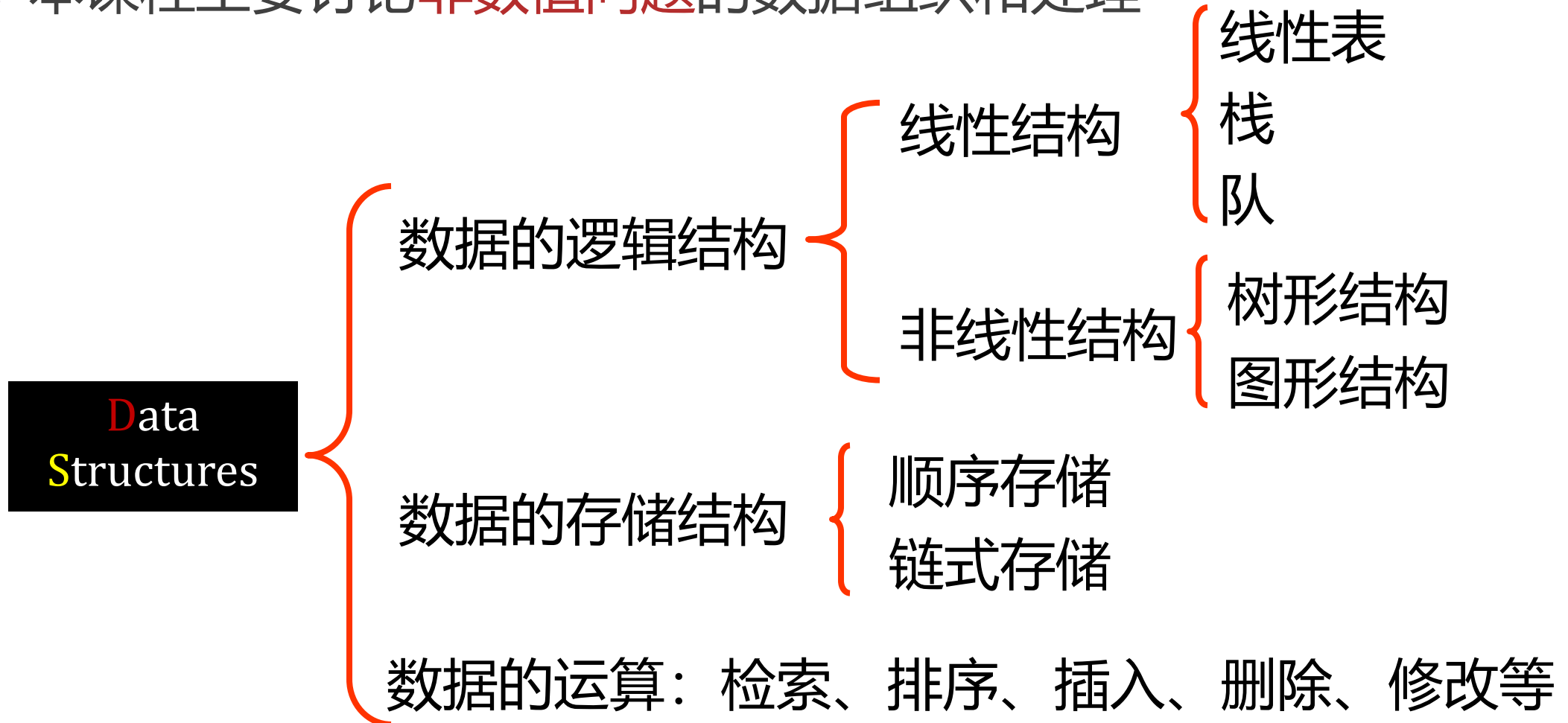


本课程主要讨论非数值问题的数据组织和处理

- (1) 数据的逻辑结构：线性表、树、图等数据结构，其核心是如何组织待处理的数据以及数据之间的关系；
- (2) 数据的存储结构：如何将表、树、图等数据结构存储到计算机的存储器中，其核心是如何有效地存储数据以及数据之间的逻辑关系；
- (3) 算法：如何基于数据的某种存储结构实现插入、删除、查找等基本操作，其核心是如何有效地处理数据；
- (4) 常用数据处理技术：查找技术、排序技术等。

小结

📜 本课程主要讨论非数值问题的数据组织和处理





Thank You !

Q & A