

数据结构与算法：简介

关振扬

September 5, 2025

本章主要内容

- 目标、意义
- 数据结构的起源
- 课程的主要内容
- 算法及算法分析
- 数据结构的基本概念

目标

- 掌握基本的数据结构（原理、实践）
- 培养算法设计能力、程序设计能力
- 适当在解决问题时使用数据结构
- 培养算法分析能力

意义

- 这门课为计算机专业的专业基础课
- 这门课与大部分需要处理大数据的课程都有关系
- 这门课同时培养逻辑思维

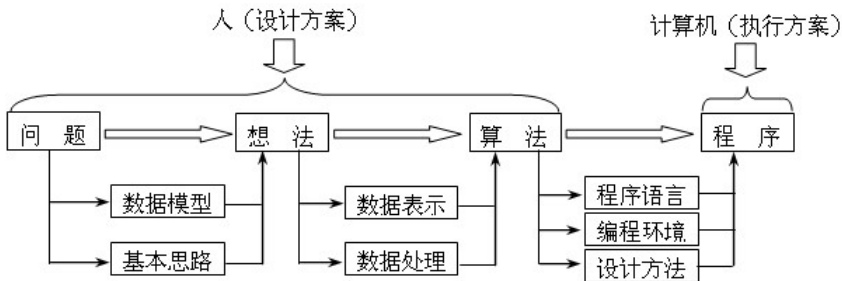
学习建议

- 切忌死背，要理解
- 切忌心浮气躁，要按部就班（尤其除错时）
- 认真做习题及编程题，切忌抄写作业
- 多研究源码

起源

- 编写程序的原因：
自动化数据资料的处理（统计、匿名化处理、等等
……）
- 在这个过程中，我们要有数据存取的方法
- 也需要处理这些数据的方法
- 上述的两个部分都是数据结构 协助解决的问题

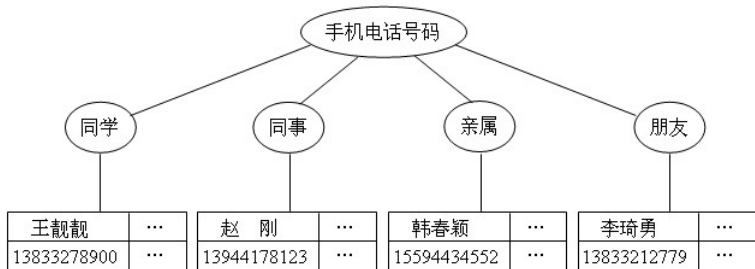
计算机求解问题的一般过程



例子：不同的数据排列方式的影响

比如在数据库中查找电话号码：

姓名	王靓靓	赵 刚	韩春颖	李琦勇	……	张 强
电话	13833278900	13944178123	15594434552	13833212779	……	13331688900



问题的解决

- 计算机求解问题的过程：
问题 → 抽象出问题的模形 → 对模形求解
- 问题的分类：
 - 数值问题：数组、数学公式（数值分析）
 - 非数值问题：数据结构（这门课）

本课程的讨论范围

我们主要集中讨论非数值数据的数据组织及处理，细分为：

- ① 数据的逻辑结构：数据之间的关系；具体有线性表、树、图等。
- ② 数据的存储结构：如何把上述结构存储到计算机的空间里，关心的是有效存储数据及其之间的逻辑关系
- ③ 算法：在上述存储结构中有效地实现插入、删除、查找等操作
- ④ 常用数据处理技术：查找、排序等。

例子：线性结构

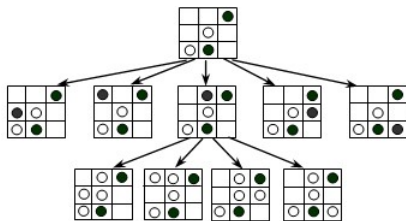
学 号	姓 名	性 别	出生日期	政治面貌
0001	陆 宇	男	1986/09/02	团员
0002	李 明	男	1985/12/25	党员
0003	汤晓影	女	1986/03/26	团员
⋮	⋮	⋮	⋮	⋮

(a) 学生学籍登记表

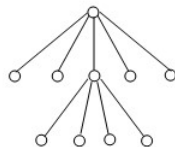


(b) 线性结构

例子：人机对弈问题

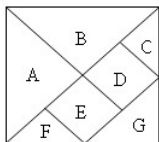


(a) 对弈树的局部

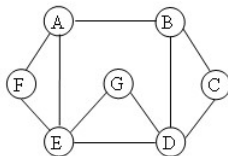


(b) 树结构

例子：七巧板的邻接关系



(a) 七巧板



(b) 图结构

算法的相关概念

Definition

算法 (Algorithm) 是对特定问题求解过程的一种描述，为指令的有限序列。

算法有以下特性：

- ① 输入、输出
- ② 有穷性：正确的算法应该在执行有穷步后结束，且每一步都应该在有穷时间内完成。
- ③ 确定性：相同的输入应该有相同的输出；每条指令需要确切的含义
- ④ 可行性：算法描述的操作均应可通过已经实现的基本操作执行有限次来实现

算法的描述方法

描述方法大致分为以下几种：

- ① 自然语言
- ② 流程图
- ③ 程序语言
- ④ 伪代码

以下我们以一个简单问题为例：使用辗转相除法求两个正整数 m 和 n 的最大公因数。

自然语言（例）

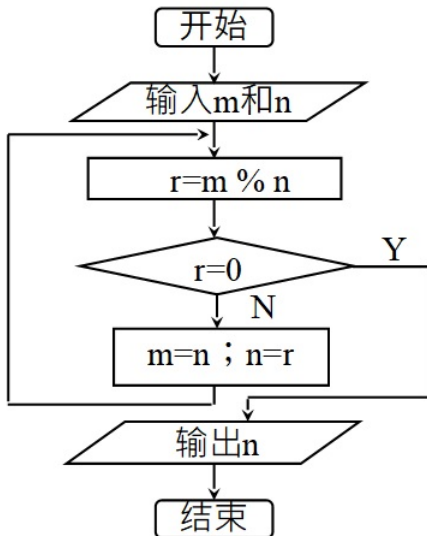
- ① 将 m 除以 n 得到余数 r
- ② 若 r 等于 0，则 n 为最大公因数，算法结束；否则继续下步
- ③ 将 n 的值放至 m 中， r 的值放至 n 中，回到第一步。

自然语言（利弊）

使用自然语言描述算法：

- 优点：容易理解
- 缺点：二义性
- 使用方法：粗线条描述算法思想
- 注意事项：避免写成文章段落
- 较少使用

流程图（例）



流程图（利弊）

使用流程图描述算法：

- 优点：直观
- 缺点：灵活性
- 使用方法：描述简单算法
- 注意事项：注意抽象层次
- 千万不要使用这个方法描述复杂算法；若别无选择，则使用抽象层次。

程序语言（利弊）

使用程序语言描述算法：

- 优点：能由计算机执行
- 缺点：抽象性差
- 使用方法：算法需要验证时
- 注意事项：将算法写成子函数
- 具体例子我们在实验那边进行。

伪代码（例）

- ① $r = m \% n$
- ② 循环直到 r 等于 0 :
 - ① $m = n$
 - ② $n = r$
 - ③ $r = m \% n$
- ③ 输出 n

伪代码（利弊）

使用伪代码 (pseudocode) 描述算法：

- 介于自然语言与程序语言之间，以程序言语为基础，操作描述可结合自然语言
- 优点：表达能力强，抽象性强，较易理解
- 最常用的算法描述方式

算法分析

研究一个算法是否正确，效率等。

关于效率，我们有以下研究方法：

- 事后统计：将算法实现，测算其时间和空间开销（但有一些很大的缺点：环境、实现方法等）
- 事前分析：理论估算算法成效，分为时间复杂度与空间复杂度的分析。我们集中说明前者，后者与数据结构的实现有关

时间复杂度分析

- 算法的执行时间 = 每条语句执行时间之和。
- 一般我们关心的是问题的规模与执行时间的关系（这里规模与输入挂钩），因此一般我们假定每个基本语句的时间一致。
- 因此，我们关心的是到底算法里基本语句被执行了多少次。

简单例子：

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


大 O 符号

Definition

对于两个函数 $f, g: \mathbb{N} \rightarrow \mathbb{R}$ ，若存在实数 c 与整数 n_0 使

$$\forall n \geq n_0, |f(n)| \leq c|g(n)|,$$

则记为 $f(n) = O(g(n))$ 。

- 若 $f(n) = \sum_{h=0}^m a_h n^h$ ，则有 $f(n) = O(n^m)$
- 若 $f(n) = \sum_{h=0}^m a_h n^h$ ，则有 $f(n) = O(e^n)$

例子

- ① `++x;`
- ② `for(i=1;i<=n;i++) x++;`
- ③ `for(i=1;i<=n;i++) for(j=1;j<=i-1;j++) x++;`
- ④ `for(i=1;i<=n;i=2*i) ++x;`
- ⑤ 辗转相除法

最好、最坏、平均情况

- 不是所有时候算法分析都很简单的；甚至有时候与输入的数据有关
- 解决方案：最好、最坏、平均情况
- 不同情况会让我们对上述三个方案有不同程度的关心：
 - 最好：若不少情况都可以归为此类
 - 最差：实时系统
 - 平均：若直到输入数据的概率分布、偏向规律

数据结构的定义

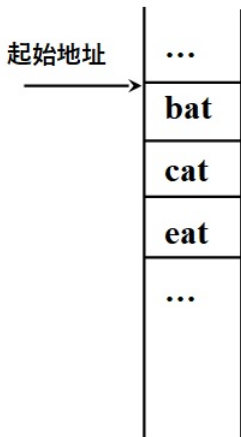
Definition

数据结构由数据与结构组成。数据之间的关系即为其结构。结构的视点分为逻辑结构与存储结构，前者为数据之间的逻辑关系（如前面有看过的例子）；后者又称物理结构，为数据及逻辑结构在计算机中的表示。

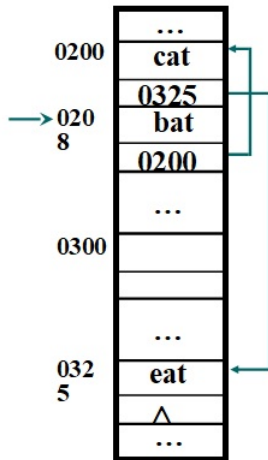
物理结构一般分为两种：顺序存储结构与链接存储结构。

物理结构（图示）

例：(bat, cat, eat)



例：(bat, cat, eat)

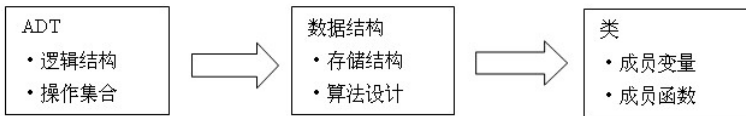


逻辑结构与存储结构之间的关系

- 逻辑结构是用户视角，面向问题的。
- 存储结构是实现视角，面向计算机的。
- 一种数据的逻辑结构并不限定使用单种存储结构
- 不同存储结构会影响数据处理的效率

面向用户：抽象数据类型

- 数据类型：一组值的集合以及定义于这个集上的一组操作的总称。（例：整数类型）
- 抽象：抽象出本质的特征而略过细节。
- 抽象数据类型 (Abstract Data Type, ADT)：一个数据结构及对该结构上的一组操作的总称。



(a) 使用视图——ADT 的定义

(b) 设计视图——ADT 的设计

(c) 实现视图——ADT 的实现

抽象数据类型的书写方法)

ADT 抽象数据类型名

Data

数据元素之间逻辑关系的定义

Operation

操作1

前置条件：执行此操作前数据所必须的状态

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

后置条件：执行该操作后数据的状态

操作2

.....

.....

操作n

.....

endADT

数据结构基本概念（小结）

