# 数据结构与程序设计 (Data Structure and Programming)

前导

北航计算机学院 晏海华





- ■是信息类专业学生最重要的专业技能之一
- 是其它许多计算机重要专业课程(操作系统、编译原理)的基础
- ■是信息类核心专业课程之一

### 作业及参考书



- 教师: 晏海华
- 联系方式: 新主楼G916, Tel: 82328212, 82317625

Email: yhh@buaa. edu. cn

- 参考书:
  - 《C程序设计导引》, 尹宝林编著, 机械工业出版社, 2013
  - 《C程序设计语言》, B. W. Kernighan, D. M. Ritchie, 机械工业出版社 2012, 徐宝文等译
  - 《数据结构教程(第3版)》 唐发根 北京航空航天大学出版社 2017
  - 《数据结构与算法分析(第二版)》—C语言描述》, Mark Allen Weiss著 冯舜玺译 机械工业出版社2013
- 课程网站(课程信息、作业、答疑及考试):

http://judge.buaa.edu.cn (http://115.25.138.223)

- 考核方式:
  - 作业及考试均采用上机方式
  - 作业占20%, 期中考试占30%, 期末占50%

# 课程网站(课程信息、作业、答疑及考试)

■ 网址:

http://judge.buaa.edu.cn

(http://115.25.138.223)

■ 用户名: 学号 密码: 学号后四位

■ 选择课程: 数据结构与程序设计(信息大类 在线考试

■ 请登录后尽快修改密码!

查看及提 交作业

**从据结构与程序设** 基础

欢仰你, 易海华! [个人信息] [注稿]

切換课程: 数据结构与程序设计基础 ▼



排 2017/12/12

明 2017/12/07

年 关于大作业问题致歉和说 明 2017/12/12

年12月4日考核安排 2017/12/04

学辅助平台

CG Online Judge

(7) 课 简介

首页|课程信息|在线作业|在线考试/在线管疑|成绩查询

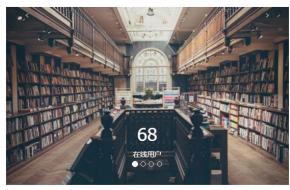
计基础将包括程序设计基础和数据结构两部

作为高等学校计算机专业重要

具备分析问题、解决问题的能力,在40~60分钟内编写并调试通过、能正确运行、一般在40~50行有效代码以内的程序; 效据结构学习中使用C语言来完成课程实践,而且通过学习让学生能够了解结构化程序设计思想以及初步接受到利

实施的基本运算(算法)以及对算法的时间、空间和其他方面进行度量分析几个方面。诵过该课程的学习, 运用课程所讨论的概念和方法更好地进行算法设计与算法分析,掌握计算机进行数据处理的基本原理和技巧 『高程序设计的能力。因此,要求学生熟练掌握常见数据结构的逻辑结构、存储结构以及不同存储结构之间的差 系掌握各种基本运算的实现算法,并能够掌握基本的算法分析方法。学好该课程的主要目的是为学生打好专业基

[编译技术] 2017编译器课程设计 ① 代码生成作业考核情况总结及后续安 [算法设计与分析] 2017-2018学 2 「编译技术」编译器课程设计2017 3 年12月11日考核安排 2017/12/11 「编译技术」2017编译器课程设计 4 语法分析作业完成情况总结和说 [编译技术]编译器课程设计2017 5



学生入口	<u></u>
▲ 学号	
P	
登录	

下载课件和 相关文档、 工具

北京航空航天大学 计算机学院

技术支持或者建议,请加入QQ群: 214085215

### 目标和方法

### 目标

培养学生利用所学知识解决问题的能力,包括解决没有遇到过的问题的能力。(即程序设计能力)

2016. 6. 2国际工程联盟接纳中国为《华盛顿协议》成员。国际认可的工程教育强调的是学生应具有知识的解决问题能力,包括解决没有遇到过的问题的能力。北航已加入了《华盛顿协议》。

### 方法

中国著名桥梁专家、工程教育家茅以升先生提出了: "习而学"、"做中学"(Learning by Doing) 的工程教育思想

■ 获得程序设计能力的唯一途径:

上机实践(编程)!

(Try!!!)





### 忠告



■ 也许有些同学会认为本课程作业太重(需要花很多课外时间),是一个负担,但我给大家的忠告是:

"负担通常会带来痛苦,但是和将来的后悔相比,哪个更痛苦一些?" 这个问题也写在 NBA 凯尔特人队的训练馆里:

"What Hurts More, The Pain of Hard Work or The Pain of Regret?"

#### 啥佛图书馆墙壁训言:

"学习时的苦痛是暂时的,未学到的痛苦是 终生的。"(Time the study pain is temporary, has not learned the pain is life-long.)





### 程序设计教材与参考书

### 教材:

# 《C程序设计导引》

尹宝林编著, 机械工业出版社, 2013

### 参考书

《C程序设计语言》, B. W. Kernighan, D. M. Ritchie, 机械工业出版社 2012, 徐宝文等译

### 数据结构教材与参考书



### 教材:

# 《数据结构教程》

第3版 唐发根 编著 北京航空航天大学出版社 2017

### 参考书1\*

《数据结构与算法分析-C语言描述》, M. A. Weiss著 冯舜玺译, 机械工业出版社 2013

#### 参考书2\*

《C++ 致据结构与算法分析(第4版)》, M. Drozdek 著 徐丹 吴伟敏译, 清华大学出版社 2014

#### 参考书3

《大话数据结构》,程杰著,清华大学出版社 2011

一本通俗读本

#### 少专节4

《数据结构(C语言版)》, 严蔚敏 吴伟民编著,清华大学出版社

### 课程作业要求



### ■普通作业

- 共七套(基础1、基础2、线性表、栈和队、树、图、查 找与排序),每套有一组选择填空概念题,多个应用编 程题
- 普通应用题主要考查学生对单一知识点的应用能力

### ■ 综合性能 (Project) 作业

- 综合编程题(只有一个),占整个作业20分中的5分,其 测试数据规模较大,主要考查学生对数据结构综合应用 的能力;
- 评判要求:正确占50%,性能占50%。即在给定时间(100秒)内正确完成,得50%分,性能部分分数以运行最快的程序为基准(其为满分),依次计算得分。运行结果不正确或在给定时间内没有运行结束,则不得分。



### ■ 应在规定时间内提交;

注: 每套普通作业大约开放3~4个星期

- 一定要按照题目要求提交,比如:输入、输出数据格式,提交文件名称等等;
- 严禁抄袭,编程作业查出抄袭要处理!
  - > 不能拷贝提交他人源代码(提交同学的代码)
  - > 不能下载提交网上源代码(提交网上的代码)
  - > 不要将自己的源码拷贝给他人(可能被抄袭)



### 程序设计与程序设计语言

- 程序设计(Programming): 为计算机解决问题所需的分析、设计、编写及调试程序过程。(The process of planning, writing, testing, and correcting the steps required for a computer to solve a problem or perform an operation.)
- 程序设计语言(Programming Language): 用来表达程序的计算机能够执行的人工语言,是程序设计过程中用到的一种工具。如,Fortran, C, C++, Java, C#等。

可以这理解两者之间的关系:

程序设计语言是我们解决问题过程(程序设计)中用到的工具之一(可能还有其它工具,如分析与设计工具。

# 少 北京航空航天大學

### 为什么要学C语言程序设计

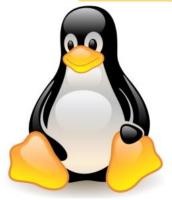
- C语言是一种简洁精练的语言,所涉及的概念比较少,语法和结构也简单,主要特点:
  - 表达能力强,支持结构化程序设计;
  - 语言简洁;
  - 代码效率高: C编写的程序仅比用汇编语言编写的程序 仅相差20% (C语言是运行效率最高的高级语言);
  - 可移植性好:
  - 特别适合编写操作系统、编译程序、数据库系统、嵌入式软件及图形/图象处理等对性能要求高的软件;
- C语言仍是目前广泛使用的编程语言,其应用领域遍及系统软件(如Windows,UNIX, Linux...)、嵌入式软件(飞机机载、导弹弹载、轨道交通、通信、...)、图形图像处理软件、游戏软件...
- C语言是目前广泛流行的面向对象语言C++、C#及Java的基础

### C语言历史



Android是一种基于Linux的自由及开放源代码的操作系统,主要使用于移动设备,如智能手机和平板电脑,由Google公司和开放手机联盟领导及开发。其底层核心用C/C++开发。

- C语言的产生与UNIX操作系统是密不可分的:
  - UNIX由Bell Lab的K. Thompson和D. M. Ritchie最先在1969年开发的O. S. (它的前身是MIT和AE开发的Multies)。
  - 1970年, V1, V2版在PDP-7机上用汇编语言实现
  - 1971年V3 PDP11/23; 1972年V4 PDP11/45
  - 1972年, D. M. Ritchie 开发出新语言C。 (C ← B ← BCPL ← CPL单数据型语言)
  - 1973年, Ritchie和Thompson 用C改写了UNIX核心(90%)即V5



Linux是一种开放源码的类Unix操作系统。 Linux可安装在各种计算机硬件设备中,从 手机、平板电脑,到台式计算机、大型机 和超级计算机。世界上运算最快的10台超 级计算机运行的都是Linux操作系统。 Linux得名于计算机业余爱好者Linus Torvalds。当然它本身也是由C语言开发的



肯·汤普逊(左)和丹尼斯·里奇(右)

### C语言历史\*



- C语言的原型ALGOL 60语言。(也称为A语言)
- 1963年, 剑桥大学将ALGOL 60语言发展成为CPL (Combined Programming Language)语言。
- 1967年, 剑桥大学的Matin Richards 对CPL语言进行了简化 . 于是产生了BCPL语言。
- 1970年,美国贝尔实验室的Ken Thompson将BCPL进行了修改,并为它起了一个有趣的名字"B语言"。意思是将CPL语言 煮干,提炼出它的精华。并且他用B语言写了第一个UNIX操作系统。而在1973年,B语言也给人"煮"了一下,美国贝尔实验室的D. M. RITCHIE在B语言的基础上最终设计出了一种新的语言,他取了BCPL的第二个字母作为这种语言的名字,这就是C语言。

由此可见,最早的C编译器当然是B语言写的。

# 数据结构

# 十 算法设计与分析

主要研究数据之 间的关系和数据 的组织方式

主要研究常用算法 的设计和算法优劣 的分析(性能)

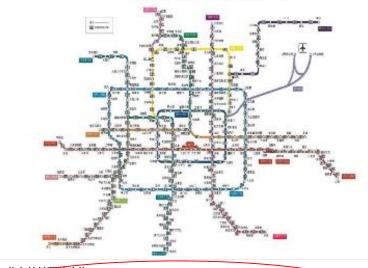
"算法(algorithm)是解决特定问题求解步骤的描述,在计算机中表现 为指令的有限序列, 并且每条指令表示一个或多个操作。

### 数据结构、算法设计与允

数据结构: 研究数据的关系与组织

算法设计与分析: 研究算法的设计和优劣的分析(性能)

#### 北京地铁站间里程全图

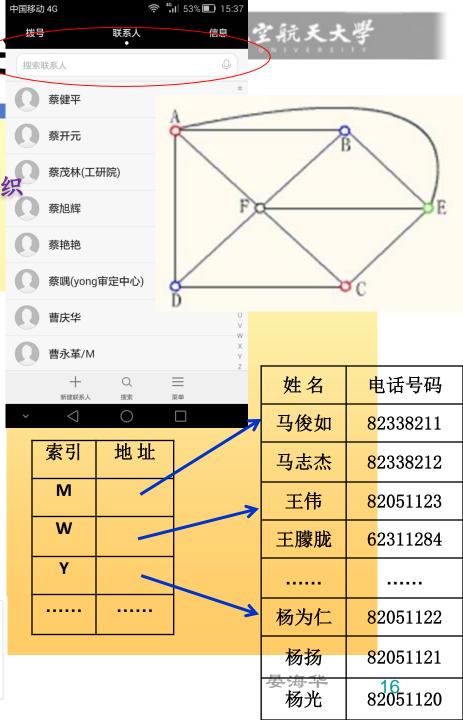


北京地铁票价查询:

请输入地铁起点站 🗕 请输入地铁终点站

开始查询

详情: 起步6公里内每人次3元,6-12公里每人次4元,12-32公里每10公里加1元,32公里以上每20公里加1元,票价不封顶。(北京市发改委)



# 少 北京航空航天大學

### 为什么要学习数据结构

- 计算机软件与理论学科的最重要的专业基础课程 之一
- 后续专业课程学习的必要知识与技能准备
  - 编译技术要使用栈、散列表及语法树
  - 操作系统中使用队列(优先队列)、存储管理表及目录 树
  - 数据库系统运用线性表、多链表、及索引树
  - ...
- 成为一名专业程序员应具备的基本技能(是微软 、Google、百度、腾讯…面试中必不可少的内容)

### 程序设计与数据结构

程序设计:如何使用程序设计语言(工具)来解决问题。



·室內外柱同高 · 內槽高:



山西 五台县 佛光寺大殿 剖面圖

数据结构: 研究数据的构造方法





软件 = ? 程序 = ?

软件 = 程序 + 文档

程序 = 数据结构 + 算法 (图灵奖得主N. Wirth(沃斯))

**好数据结构 + 好算法 + 好风格 = 好程序** 

### 数据结构课程的主要内容和目的

### 主要内容

- ▶ 逻辑结构:通过抽象的方法研究被处理数据之间存在何种逻辑关系,即逻辑结构,有两大类:线性结构(线性表、数组、栈、字符串等)和非线性结构(树、图等)。
- ◆ 存储结构:研究每种逻辑关系在计算机内部如何表示,即存储结构或者物理结构,如线性存储结构、链式存储结构、 索引结构和散列结构。
- ◆ 算法:研究在数据各种结构的基础上如何对数据实施有效的操作或处理(创建、清除、插入、删除、搜索、更新、访问、遍历等)。

### 主要目的

掌握数据处理的基本原理和方法,更好地进行算法设计与算法分析,提高程序设计的水平和能力。

# 数据结构

绪论



- 1 什么是数据结构
- 2 算法及其描述
- 3 算法分析的基本概念



# 1 什么是数据结构

### 1.1 名词术语

数据

描述客观事物的数字、字符以及一切能够输入到计算机中,并且能够被计算机程序处理的符号的集合。

数据元素

数据这个集合中的一个一个的元素。

数据对象

具有相同特性的数据元素的集合。

自然数 (1, 2, 3, 4, …)

# 数据

# 数据元素

**3** 

(25, 78, 36, 100, 28, 45

2

'A', 'B', 'C', ..., 'Z')

数列

字母表

3

_ 学 号	姓名	性别	年龄	其 他
99001	张三	女	17	
99002	李四	男	16	
99003	王五	<u>男</u> 女	18	
99004	周六	女	17	
•	•••	•	•	•
99035	刘末	男	19	

数据文件





# 1 什么是数据结构

### 1.1 名词术语

数据

描述客观事物的数字、字符以及一切能够输入到计算机中,并且能够被计算机程序处理的符号的集合

数据元素

数据这个集合中的一个一个的元素。

数据对象

具有相同特性的数据元素的集合。

结构

数据元素之间具有的关系。

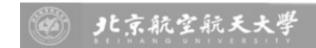


### 1.2 数据结构的定义

1. 数据元素之间的联系称之为结构,数据结构 就是具有结构的数据元素的集合。

2. 数据结构 是一个二元组 Data-Structure=(D,R) 其中, D是数据元素的有限集合,R是D上的关系的集合。

某一数据对象



### 数据结构包括:

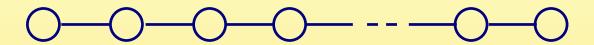
- ■数据的逻辑结构
- ■数据的存储(物理)结构
- ■数据的操作(算法)

### 逻辑结构

### 数据元素之间具有的逻辑关系(结构)

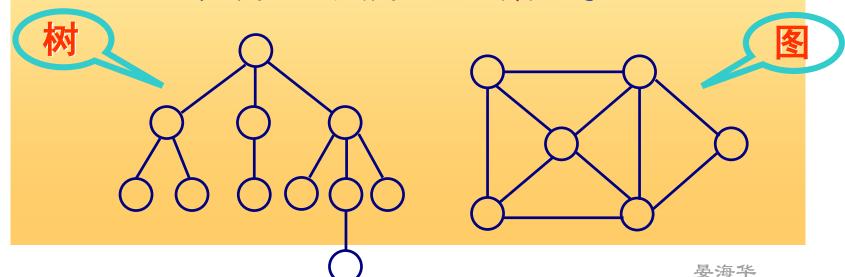
### 线性结构

如线性表、堆栈、队列、串、文件等



### 非线性结构

如树、二叉树、图、集合等



存储结构

# 具有某种逻辑结构的数据在计算机中的存储方式(存储映象)

### 1. 顺序(sequential)存储结构

用一组**地址连续**的存储单元依次存放数据元素,数据元素之间的逻辑关系通过元素的地址直接反映。也就是说数据间的逻辑关系与物理关系是一致的。

元素1 元素2 元素3

---

元素n-1 元素n

数组结构

### 2. 链式(linked)存储结构

用一组地址任意的存储单元依次存放数据元素,数据元素之间的逻辑关系通过指针间接地反映。在这种结构中,每个数据节点由两部分组成,一部分是数据本身(数据字段);另一部分是指针,用于存放后续结构的地址(指针字段)

元素1 十 元素2 十 元素3 十 …

→ 元素n-1 → 元素n ^

链表结构

### 3. 索引(indexing)存储结



构

### 构造原理

利用数据元素的索引关系来确定数据元素的存储位置,由数据元素本身与索引表两部分组成。

### 特点

诸如查找、插入和删除等操作的时间效率较高,但存储空间开销较大。

### 4. 散列(hashing)存储结构

以后详细讨论

### 构造原理

通过事先准备好的散列函数关系与处理冲突的方法来确定数据元素的存储位置。

### 特点

诸如查找、插入和删除等操作的时间效率较高,主要缺点是确定好的散列函数比较困难。



姓名 性别民族年龄

(4)	北京	航空航	天大學
	其	他	V E R S I T V

		「エノリ		<b>十四</b> 个	<del>7</del> 16
<b>a</b> <sub>1</sub>	刘晓光	男	汉	16	•••
$\mathbf{a_2}$	马广生	男	口	17	•••
$\mathbf{a}_3$	王 民	男	壮	19	***
:					
$a_3$	张淑华	女	汉	24	•••

0

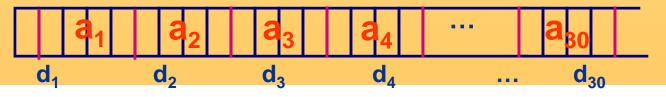
### 逻辑结构:

线性结构 (线性表)

### 存储结构:

1. 顺序存储结构

数据元素之间的逻辑关系通过地址直接反映

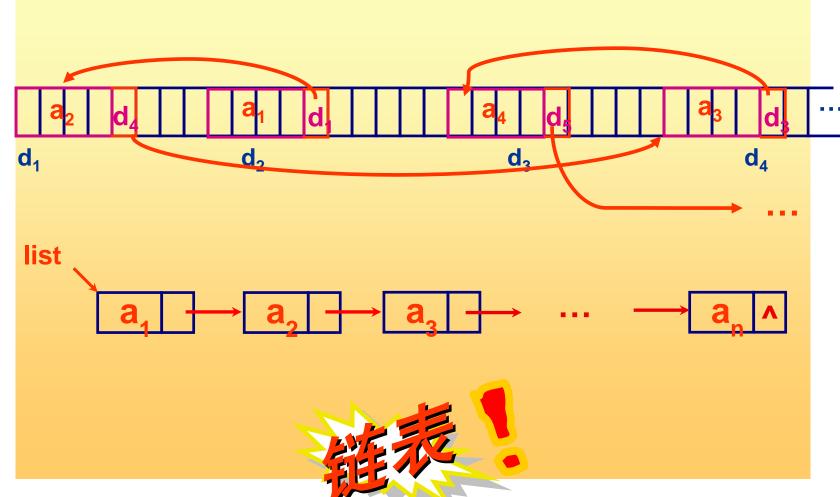




# 用一片地址任意的存储空间



### 2. 链式存储结构



E 22000 00 2 2 2 11

### 操作

### 对具有某种逻辑结构的数据所实施的一系 列统操作**(**算法)

### 基本操作:

- ■构造:构造具有某种逻辑结构的数据集,如构造
  - 一个线性表、链表、树和图等
- 检索: 在已有数据集中查找某一指定元素
- ■插入: 在已有数据集中插入一指定元素
- ■删除:在已有数据集中插入一指定元素
- ■排序:对一数据集中元素按照某一顺序进行排列
- ■遍历:访问数据集中所有元素
- ...

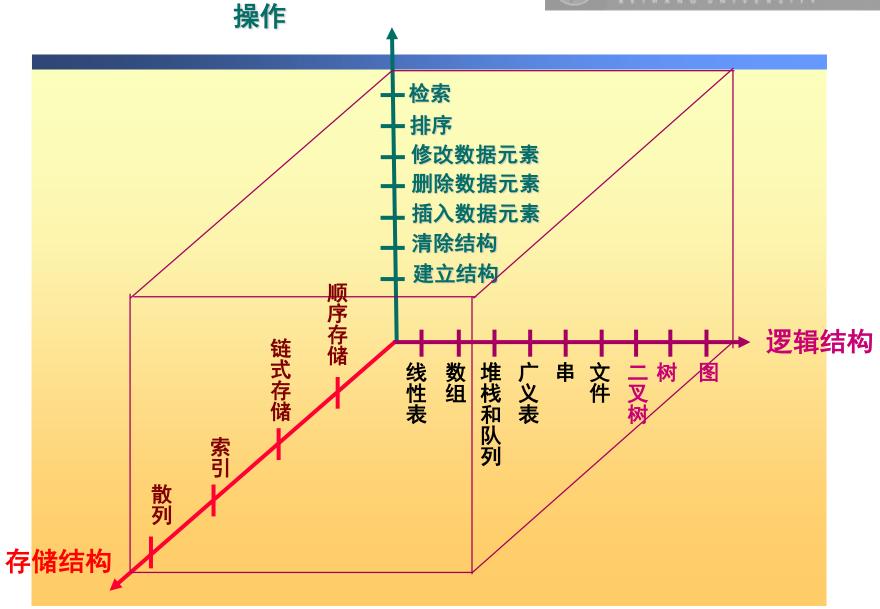
# 1.3 数据结构课程研究的主要内容

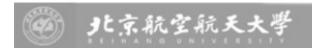
逻辑结构

- 1. 研究数据元素之间的客观联系。
- 2. 研究具有某种逻辑关系的数据在计算机存储器内的存储方式。 *存储结构*
- 3. 研究如何在数据的各种关系(逻辑的和物理的)的基础上对数据实施一系列有效的基本操作。

逻辑结构 + 存储结构 + 算法







## 2 算法及其描述

### 2.1 算法及其性质

# 1.算法的定义

- (1)算法 是用来解决某个特定问题的指令的集合。
- (2)算法是由人们组织起来准备加以实施的一系列有限的基本步骤。
- (3)算法是一组解决问题的清晰指令,它能够对符合一定规范的输入,在有限的时间内获得所需要的输出。



#### 2.算法的性质

#### 一个完整的算法应该具有下面五个基本特性:

输入

由算法的外部提供n≥0个有限量作为算法的输入。

输出

由算法的内部提供n>0个有限量作为算法的输出。

有穷性

算法必须在有限的步骤内能够结束。

确定性

组成算法的每一条指令必须有清晰明确的含义。

有效性

算法的每一条指令必须具有可执行性。

#### 2.2 算法的描述

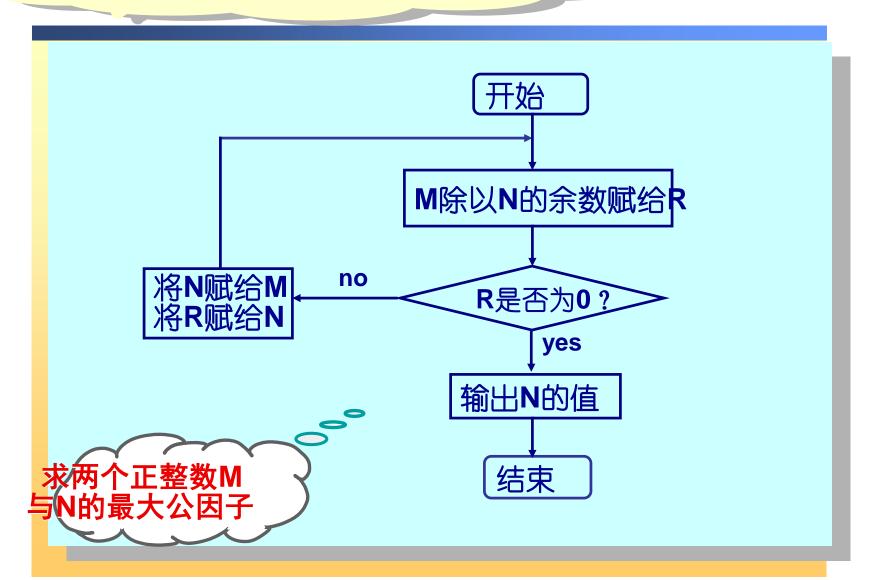
#### 1. 采用自然语言来描述



**阿黎**两个正整数M与N的最大公因子。

- (1) M除以N,将余数赋给中间变量R;
- (2) 判断余数R是否等于零?
  - a) 若R等于零,求得的最大公因子为当前 N的值, 算法到此结束。
  - b) 若R不等于零,则将N赋给M,将R赋给N, 重复步骤(1)和(2)。

#### 2. 采用程序流程图的形式来描述





3. 用一种既脱离某种具体的程序设计。语言,又具有各种程序设计语言的共同特点的形式化语言来描述(吴语言)

类Pascal语言 类C语言 SPARKS语言 (一种类Pascal语言)



#### 4. 利用某种具体程序语言来描述

#### C, C++, Java ...

```
int comfactor( int m, int n )
{
    int r;
    while( 1 ){
        r = m % n;
        if(r == 0)
            return n;
        m = n;
        n = r;
    }
}
```

#### 3 算法分析



#### 算法分析 是指对算法质量(效率)优劣的评价。

时空效率高的算法才是一个好 的算法,它常常是不懈努力和反复 修正的结果

#### 除正确性外,通常从三个方面分析一个算法:

- ▲ 依据算法编写的程序在计算机中运行时间多少的度量,称之为 时间复杂度 反映算法运行的快慢
- ▲ 其他方面。如算法的可读性、可移植性以及易测试性的好坏。



一个程序在计算机中运行时间的多少与诸多 因素有关,其中主要有:

几乎所有算法的时间效 率都与问题的规模有关

1. 问题的规模。

2. 编译程序功能的强弱以及所产生的机器代码质量的优劣。

3. 机器执行一条指令的时间长短。

程序中那些基本语句的执行次数。

直点

对算法运行时间 贡献最大的语句

相关





## 以语句执行的次数的多少作为算法的时间度量的分析方法称为*频度统计法*。

一条**语句的频度**是指该语句被执行的次数,而整个**算法的频度**是指算法中所有语句的频度之和。





#### 求两个n阶矩阵的乘积

#### 算法的频度: t(n) = n<sup>2</sup> + n<sup>3</sup>



# 时间复杂度

假设算法A的语句频度为2n², 算法B的频度为3n+1, 算法C的语句频度为2n²+3n+1,则有下表:

次数	算法 A (2n <sup>2</sup> )	算法 B (3n+1)	算法 C(2n <sup>2</sup> +3n+1)	
n = 1	2	4	6	
n = 2	8	7	15	
n = 5	50	16	66	
n = 10	200	31	231	
n = 100	20 000	301	20 301	
n = 1,000	2 000 000	3 001	2 003 001	
n = 10,000	200 000 000	30 001	200 030 001	
n = 100,000	20 000 000 000	300 001	20 000 300 001	
n = 1,000,000	2 000 000 000 000	3 000 001	200 000 3000 001	

#### 从中可以看出:

判断一个算法的效率时,算式中的常数和其它次要项通常可以 忽略,而更应该关注主要项(最高阶项)的阶数。

#### 关于符号O(Order)的定义

北京航空航天大學

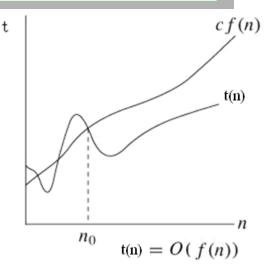
当且仅当存在正整数c和 $n_0$ ,使得 $t(n) \le cg(n)$ 对所有的 $n \ge n_0$ 成立,则称函数t(n)与g(n)同阶,或者说,t(n)与g(n)同一个数量级,记作 t(n) = O(g(n))

称上式为算法的 **渐近时间复杂度** ,或称该算法的时间复杂度为(g(n)) 。 其中, n为问题的规模(大小)的量度。

当问题的输入规模n趋于无穷大时,算法的运行时间表现出固定的增长次数。

$$t(n)=n^3+n^2 g(n)=n^3$$

称算法的时间复杂度为 $O(n^3)$ 。



#### 计算时间复杂度

大**O**表示法关注的是问题规模n增长时,算法执行次数增长的数量级。因此,可以只关注算法中基本语句执行频度,不必对算法的每一个步骤都进行详细的分析。



```
i=1; (1)
while (i<=n) (2)
i=i*2; (3)
```

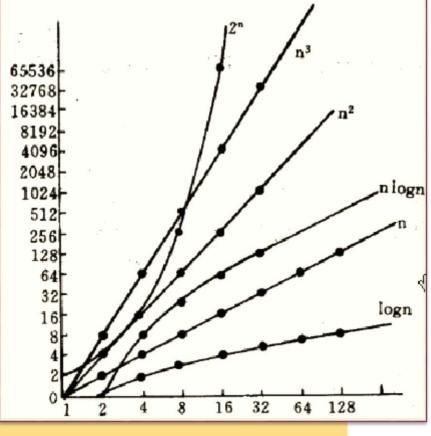
可只关注语句(3)的频度,设为t(n),则有: 2<sup>t(n)</sup><=n

即  $t(n) \leq \log_2^n$ ,取最大值 $t(n) = \log_2^n$ 

即: 时间复杂度为 $O(\log_2^n)$ 

#### 对于算法分析具有重要意义的:

n	log <sub>2</sub> n	n nl	og <sub>2</sub> n	n²	n3
10	3.3	101	3.3×10	)1	102
100	6.6	102	6.6×10	<b>)</b> 2	104
1000	10	<b>10</b> 3	1.0×10	<b>)</b> 4	106
10000	13	104	1.3×10	<b>)</b> 5	108
		•			



#### 常见的时间复杂度

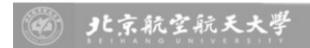
 $O(1) < O(\log_2 n) < O(n) < O(n\log_2 n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$ 

**O(1)** 

——表示算法的复杂度为常量,不随问题规模n的大小而改变。







#### 最坏情况与平均情况

- ◆ 最坏情况运行时间是一种保证,那就是运行时间将不会再坏了。在实际应用中,这是一种最重要的需求。如机载软件关键软件中必须要做最坏情况时间分析。
- ◆ 平均运行时间是所有情况中最有意义的,因为它是期望的运行时间。
- ◆ 对算法的分析,一种是平均时间复杂度;另一种是最坏时间复杂度。一般在没有特殊说明的情况下,都是指最坏时间复杂度。



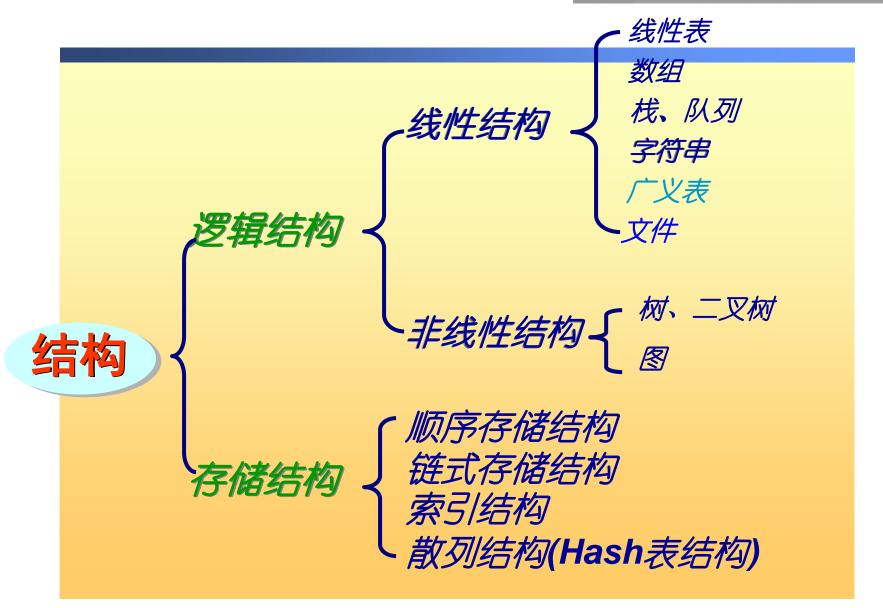
#### 延伸学习\*-算法空间复杂度

算法空间复杂度通过计算算法运行时所需的存储 空间来实现。

> 在写程序时,经常会用空间来换时间。算法空间复杂 度是衡量算法效率的另一个 重要指标。









#### 算法的定义

- 用来解决某个特定课题的指令的集合。是由人们组织起来准备加以实施的一系列 有限的基本步骤。

#### 算法的基本特征

输入、输出、有穷性、确定性、有效性

- 可以采用自然语言或程序设计框图的形式。
- 可以采用某一种自定义的符号语言。
- 可以采用某一种具体的程序设计语言。

- 什么是算法分析?
- 算法分析的目的是什么?
- 算法分析的前提是什么?
- 通常从哪几个方面对算法进行分析?







### 算法

#### 线性结构

线性表、数组、 堆栈、 队列、 广义表、串

#### 非线性结构

二叉树树、图

构造、增、删、查等基本操作

检索排序

## 本讲结束