第一章 绪论

计算机学院

杨震

yangzhen@bupt.edu.cn

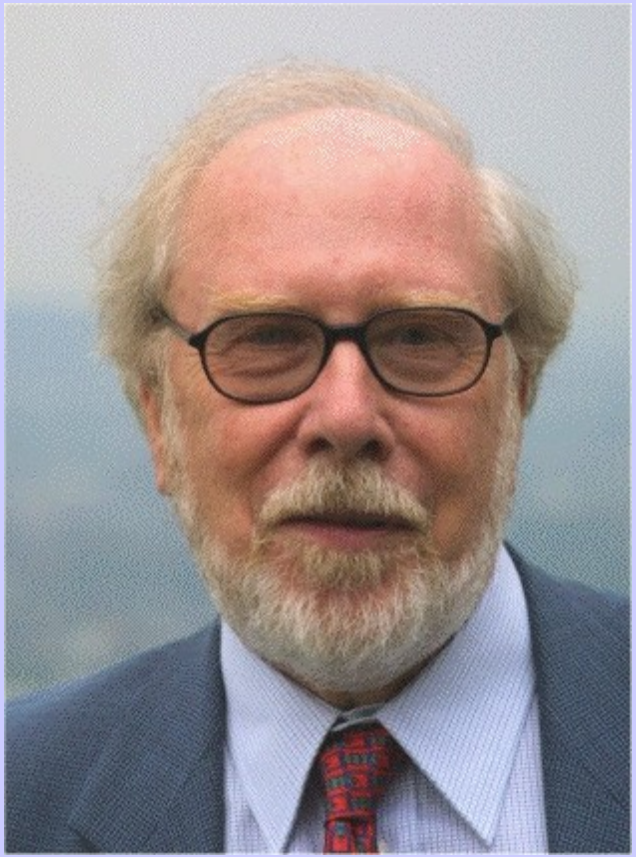
课程说明

- 选修课
  - 32学时
  - QQ群：1134482124
  - 总评成绩中，平时成绩(作业+课堂)占40%，期末考试占60%
- 教材
  - 1）《数据结构与算法》；张晓丽等；机械工业出版社
  - 2）《数据结构(C 语言版)》；胡学刚等；高等教育出版社
- **参考书目:**
  - 《数据结构（C 语言版）》；严蔚敏，吴伟民；清华大学出版社

本章内容

- 1.1 学习数据结构的意义
- 1.2 基本概念和术语
- 1.3 算法的描述与分析
- 思考题

1.1 学习数据结构的意义



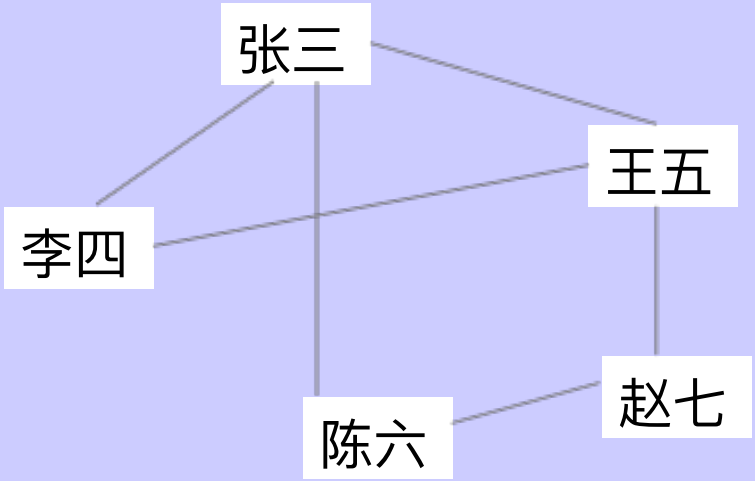
- 是为研究和解决如何有效地组织和处理非数值数据而产生的理论、技术和方法。
- 是一门综合性的专业课。

  涉及计算机软件、硬件以及数学等

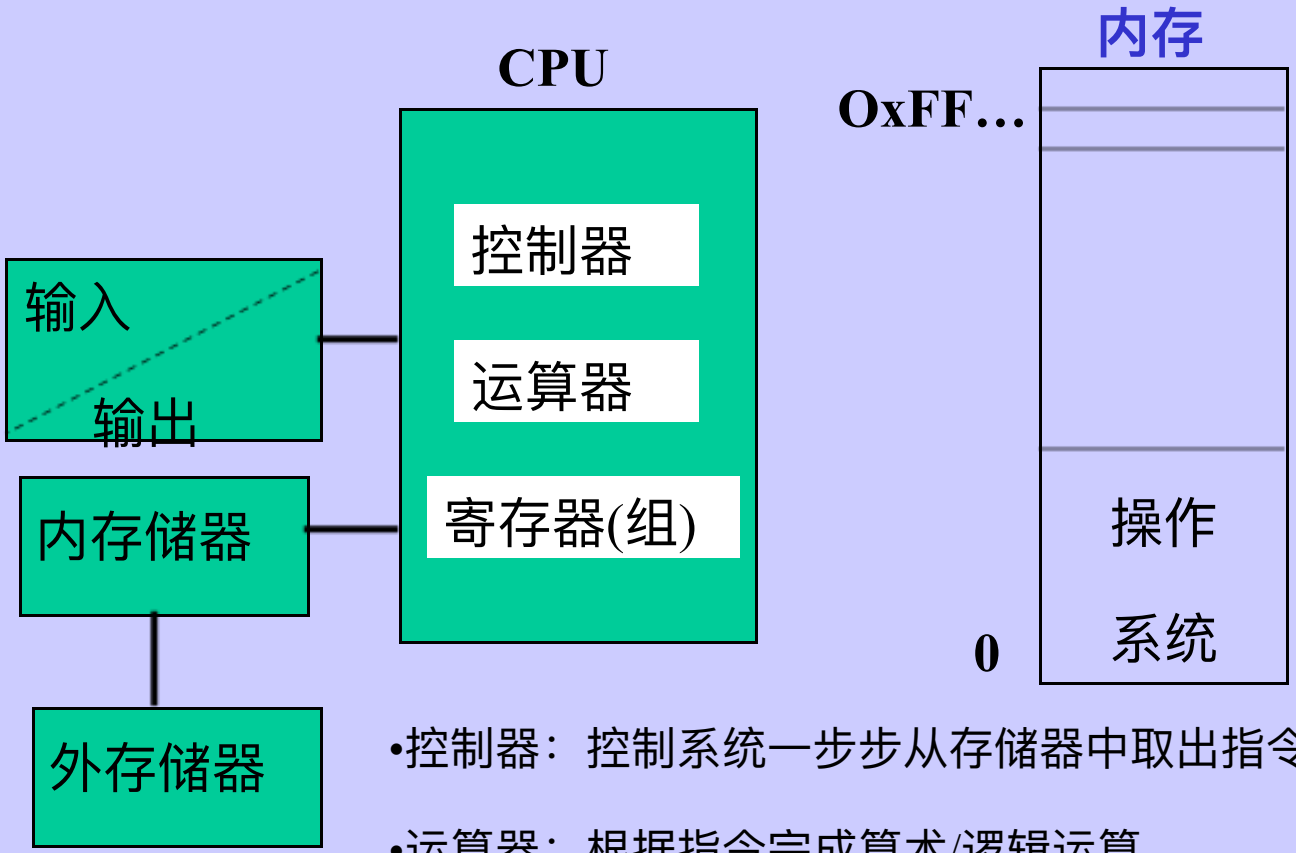
- 数据结构在软件开发中的地位：

  系统分析　系统设计　系统实现　系统维护

[例] 查找某人的社会关系



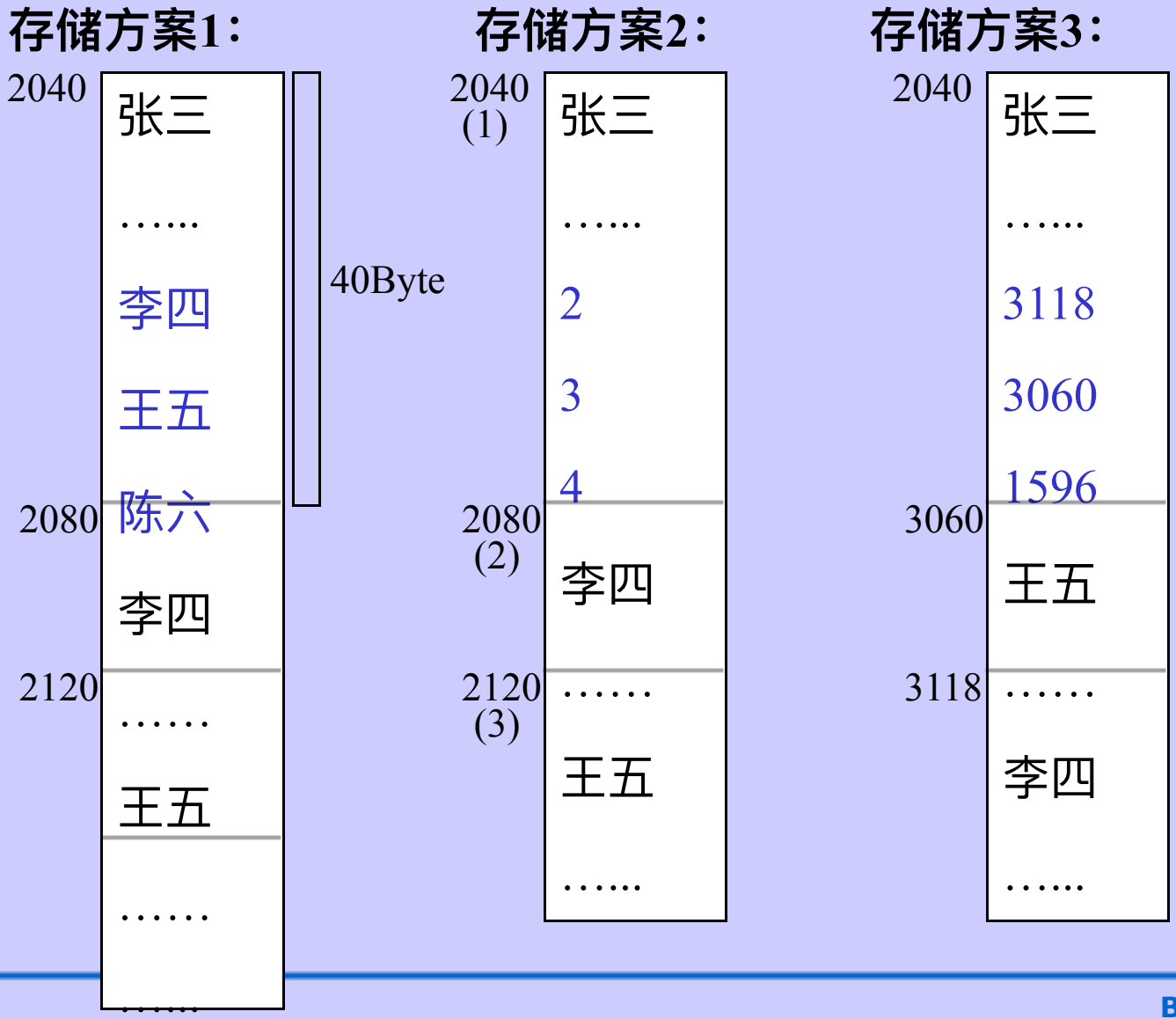
计算机中如何表示/存储和操作?



•控制器：控制系统一步步从存储器中取出指令、译码

•运算器：根据指令完成算术/逻辑运算

•寄存器：保持程序运行状态、存储当前指令信息及下一条指令地址等


存储方案1：
2040
张三
……
李四
王五
陈六
2080
李四
2120
……
王五
……
……
40Byte
存储方案2：
2040
(1)
张三
……
2
3
4
2080
(2)
李四
2120
(3)
……
王五
……
存储方案3：
2040
张三
……
3118
3060
1596
3060
王五
3118
……
李四
……

数据结构的作用范畴

- 抽象数据对象的数学模型（逻辑结构）

  例：图状结构

- 明确操作（运算的定义）

  例：查找、插入、……

- 在存储结构上映射数据（存储结构）

  例：顺序存储

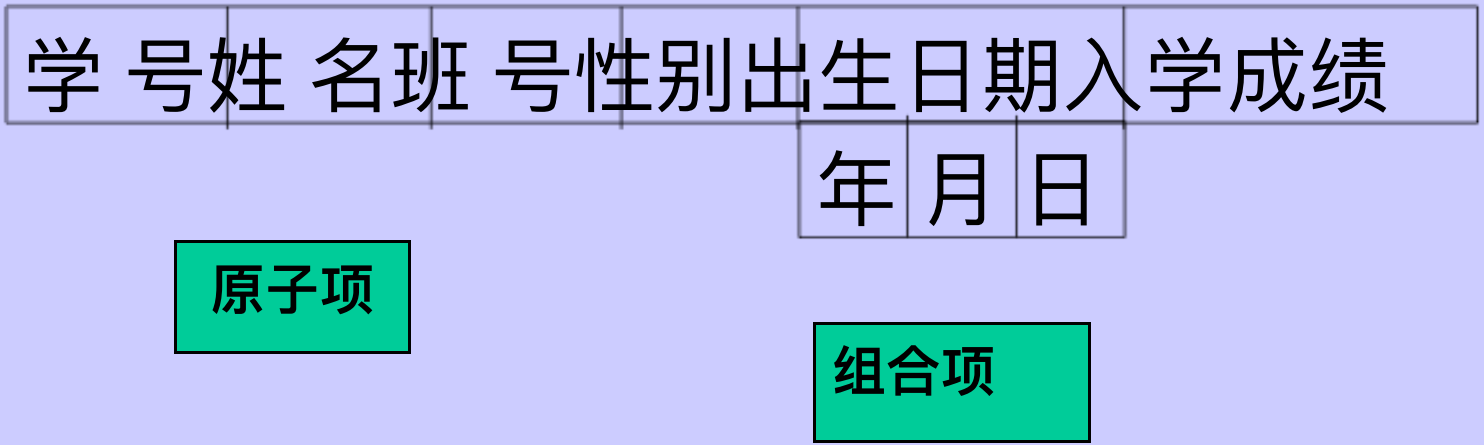
- 实现操作

1.2 基本概念和术语

数据 被计算机加工处理的对象。

数据元素（**记录**、**表目**） 数据的基本单位，是数据集合中的一个个体。

一个数据元素可由若干个**数据项**组成。



数据对象 是性质相同的数据元素的集合，是数据的一个子集。

学号	姓名	班号	性别	出生日期	入学成绩
001	刘影	01	女	19840417	623
002	李恒	01	男	19831211	679
003	陈诚	02	男	19840910	638
…	…	…	…	…	…

数据元素

数据结构 具有结构的数据元素的集合。它包括数据元素的**逻辑结构**、**存储结构**和相适应的**运算**。

逻辑结构 数据元素之间的逻辑关系，与计算机无关。

可用一个二元组表示：

Data_Structure = (D,R)

D—数据元素的有穷集合，R—D上关系的有穷集合。

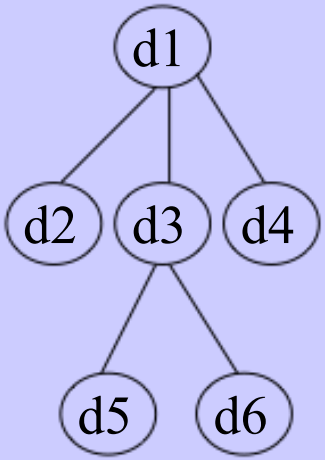
[例] 设有数据结构 B = (D,R) ，

其中 D= {d1, d2, d3, d4, d5, d6}，

R={r}， r={<d1,d2>, <d1,d3>,

<d1,d4>, <d3,d5>, <d3,d6>}，

试画出其逻辑结构图。



四种基本的逻辑结构

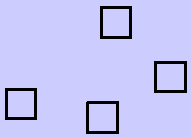
（以班级学生关系为例）

(1)**集合结构** 数据元素除了“属于同一集合”的联系之外，没有其它的关系。

(2)**线性结构** 数据元素之间存在一对一的关系。

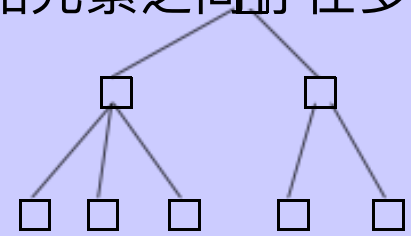
(3)**树型结构** 数据元素之间存在一对多的关系。

(4)**图状结构**或**网状结构** 数据元素之间存在多对多的关系。

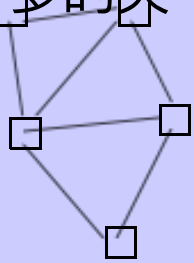


成员关系

长幼关系



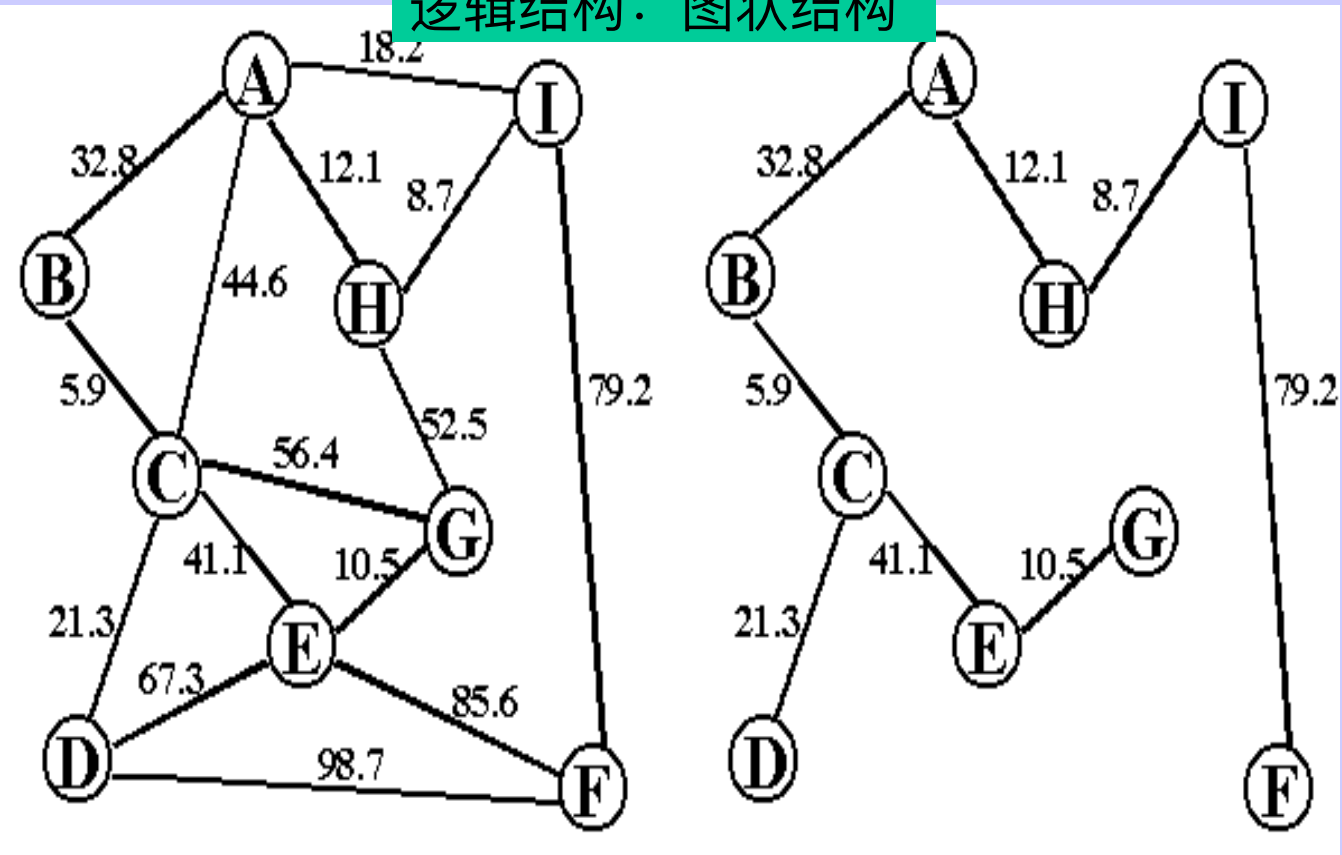
管理关系



朋友关系

[例] 铺设城市通信管线，使总投资最少?

逻辑结构：图状结构



存储结构（物理结构）：指数据的逻辑结构在计算机存储器中的映象表示。

- **数据元素的映象**

  用二进制位(bit)的位串表示数据元素。

  每个数据元素的映象称为**结点**

  每个数据项的映象称为**数据域**

- **关系的映象**

  两种基本方法及其组合使用。

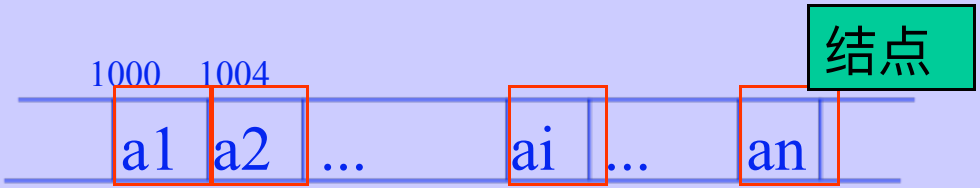
  - 顺序映象：以相对的存储位置表示关系
  - 链式映象：以附加信息(指针)表示关系

在不同的编程环境下，存储结构有不同的描述方式。

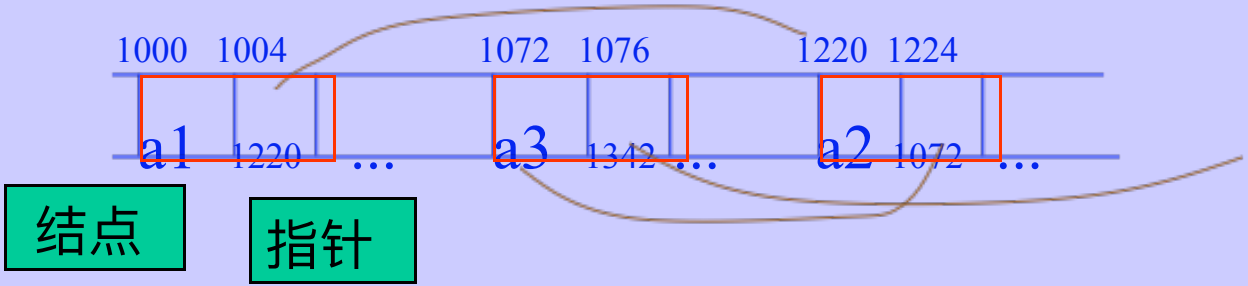
用高级程序语言编程时，通常可用其提供的数据类型描述。

数据存储方式的四种常用结构

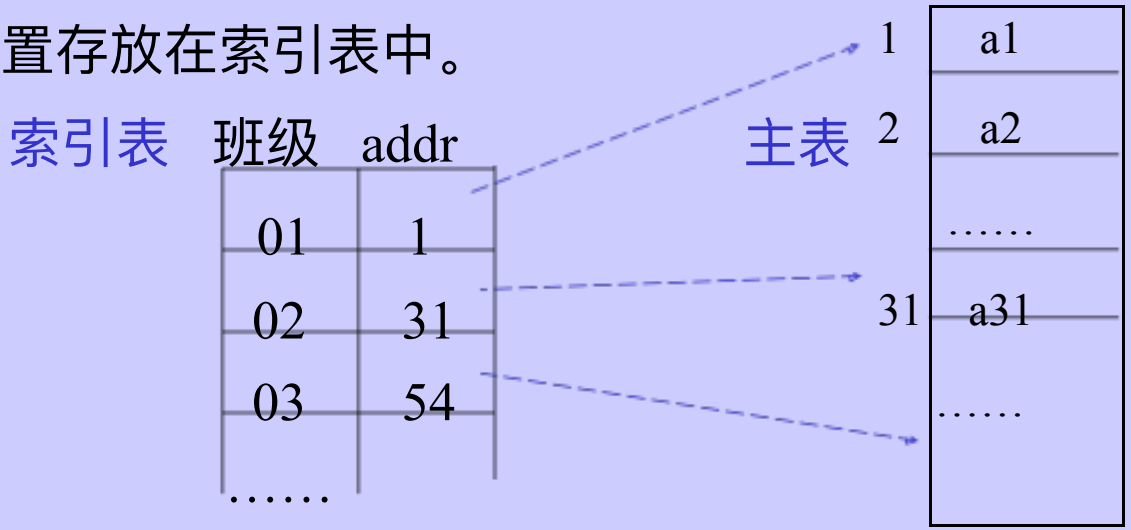
(1)**顺序存储**：数据元素依次放在连续的存储单元中。



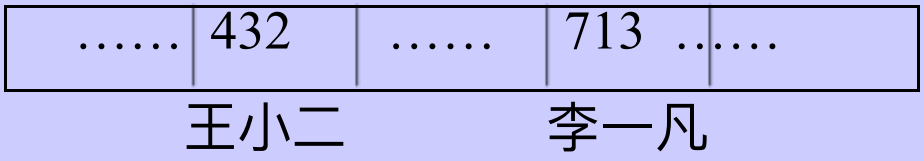
(2)**链式存储**：在存储结点中增加若干指针域，记录后继或者相关结点的地址（指针）。



(3)**索引存储**：将数据元素分为若干子表，子表的开始位置存放在索引表中。



(4)**散列存储**：根据数据元素的关键字值，由散列函数计算出存储地址。LOC(ai)=H(key)



运算（操作）：在数据逻辑结构上定义的一组数据被使用的方式，其具体实现要在存储结构上进行。

几种常用的运算有：

(1)建立数据结构
(2)清除数据结构
(3)插入数据元素
(4)删除数据元素
(5)排序
(6)检索*
(7)更新
(8)判空和判满*
(9)求长*

*操作为**引用型操作**，即数据值不发生变化；其它为**加工型操作**。

1.3 算法的描述和分析

1.3.1 算法的概念

建立在数据结构基础上的、求解问题的一系列确切的步骤。

一个算法必须满足以下五个重要特性

- 有穷性：对任何合法输入执行有穷步后能结束。
- 确定性：每条指令必须有确切的含义。
- 可行性：算法的每一条指令均能执行。
- 输入：有零个或多个输入。
- 输出：有一个或多个输出。

评价算法优劣的基本标准

- 正确性
- 可读性
- 健壮性
- 高效性

算法效率的度量

- 时间复杂度
- 空间复杂度

1.3.2 算法的描述

包括顺序、判定和重复三种基本控制结构和自然语言成分

选择描述工具的原则

不依赖于具体计算机与具体程序设计语言的一种形式化语言，可用于描述或表达算法思想。

本课程采用 C 语言

特点 它描述的算法自然易懂，具有较好的可移植性。

1.3.3 算法分析 ——算法效率的度量

1）时间复杂度

算法的消耗时间：算法中每条语句执行时间之和。

时间复杂度：算法中各语句的频度之和T(n)。

频度—语句的执行次数；

n—问题的规模，一般为数据的输入量

渐近时间复杂度：当问题的规模n趋于无穷大时，

T(n)的数量级(阶)。记为T(n)=O(f(n))。

O的严格含义— 存在正的常数c和n0，使得当n≥ n0时，

$0 \le T(n) \le c*f(n)$

实际中，将渐近时间复杂度简称为时间复杂度，用以描述算法的时间特性。

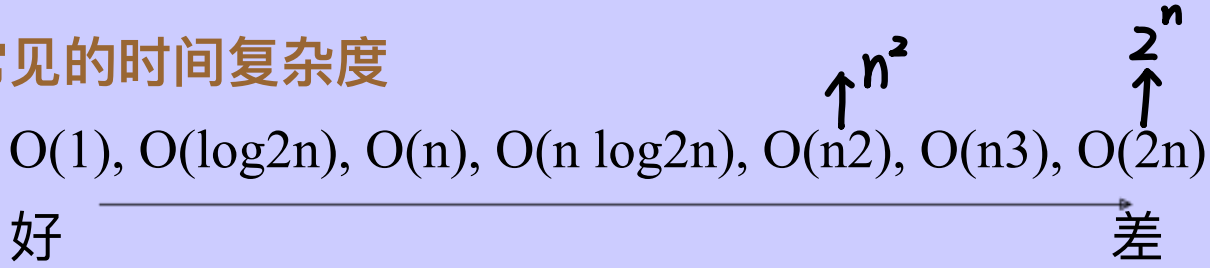
时间复杂度的求法

- 计算T(n)
- 从T(n)中推断f(n)

影响算法时间复杂度的因素

- 问题的规模
- 输入实例的初态

常见的时间复杂度



[例1] 交换i和j的内容

```
(1)  temp=i;
(2)  i=j;
(3)  j=temp;
```

解：T(n)=3 记作T(n)= O(1)

[例2] n×n矩阵相乘的算法语句

```
for ( i=0; i<n; i++)
    for (j=1;j<n;j++)
    {
      c[i][j]=0;
        for (k=1;k<n;k++)
            c[i][j]=c[i][ j]+a[i][k]*b[k][ j];
    }
```

一句

0 √

i=n-1 √

i=n 也进来判断

j=1

j=n-1

j=n 也进来但

n+1 √

n(n+1)√

n2 n^2

n2(n+1) $n^2(n+1)$

n3 n^3

行和列乘

原位置到末位置的 内存拷贝

memcpy(A,B,size(int)*n);

字节

解：语句频度 T(n)=2 n3 +3 n2 +2n+1

当n≥n0=1时，有T(n)≤8n3 ，即c=8，由此取f(n)= n3

则T(n)=O(n3)　最内层语句决定

算法中存在循环时，通常由嵌套层数最深的循环语句的

[例3] 在数组A[n]查找给定值K

```
(1)  i=n;
(2)  while ((i>=1) && A[i]!=K)
(3)      i=i-1;
(4)  return(i);
```

从n到0　且　!=　不是简单语句要进到函数里换算

↑没找到，n+1种可能情况～

最好：1次

最坏：n 或者 n+1次

解：最好情况的时间复杂度 T(n)=O(1)

最坏情况的时间复杂度 T(n)=O(n)

平均时间复杂度：所有可能的输入实例以等概率出现的情况

n+1种情况　所有情况.　终止位置不同次数不同

数据变化可能性不一样　要考察平均时间复杂度

$$T(n)=(1+2+...+n)/n=O(n)$$

算法与数据状态有关时，需讨论不同情况

eg. 按从小到大顺序排序.

3	9	5	2	4

2 ↓ 放在第1个5"3"交换位置

3

2）空间复杂度

算法的存储空间

- 输入数据所占空间X
- 程序本身所占空间X
- 辅助变量所占空间

} 不予考虑.

系统为其生成的辅助空间

空间复杂度

$S(n)=O(f(n))$

算法需要多少/那些存储空间

表示随着问题规模 n 趋于无穷的增大，算法运行所需存储量的增长率与 f(n) 的增长率相同。

存储密度

d=数据本身存储量/实际所占存储量

需要的 ①输入数据占的空间 ②③

$d=\frac{①}{②+③}$

eg. $\frac{n}{n+1}$

思考题

1. 设n为正整数，试确定下列各程序段中语句的频度和时间复杂度：

1）
```
i=1;  k=0;
while (i<=n-1)
    k=k+10*i;  i=i+1;
```

2）
```
i=1; x=0;
while (i<n)
 {
  x=x+1;
  i=2*i;
 }
```

2. 设n是**偶数**，试计算运行下列程序段后m的值，并给出该程序段的时间复杂度。

```
m=0;
for  (i=1; i<=n; i++)
 for (j=2*i; j<=n; j++)
     m=m+1;
```