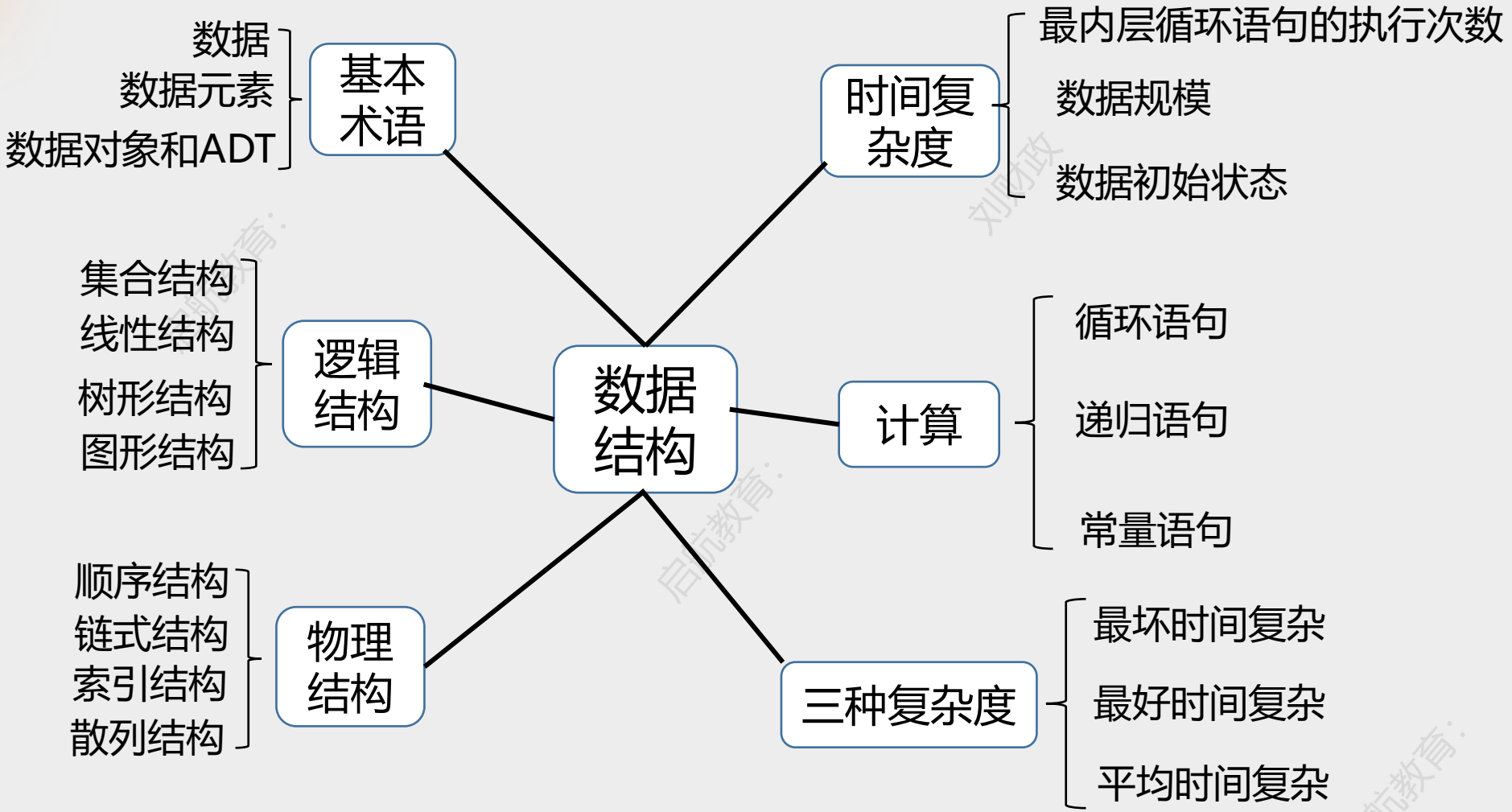
计算机考研系列书课包

玩转数据结构

主讲人 刘财政

第一讲　绪论

本讲知识结构


数据结构
基本术语
数据
数据元素
数据对象和ADT
时间复杂度
最内层循环语句的执行次数
数据规模
数据初始状态
逻辑结构
集合结构
线性结构
树形结构
图形结构
计算
循环语句
递归语句
常量语句
物理结构
顺序结构
链式结构
索引结构
散列结构
三种复杂度
最坏时间复杂
最好时间复杂
平均时间复杂

本讲内容

考点一：基本概念

考点二：逻辑结构 ★

考点三：物理结构 ★

考点四：算法的概念和评价

考点五：时间复杂度 ★★★

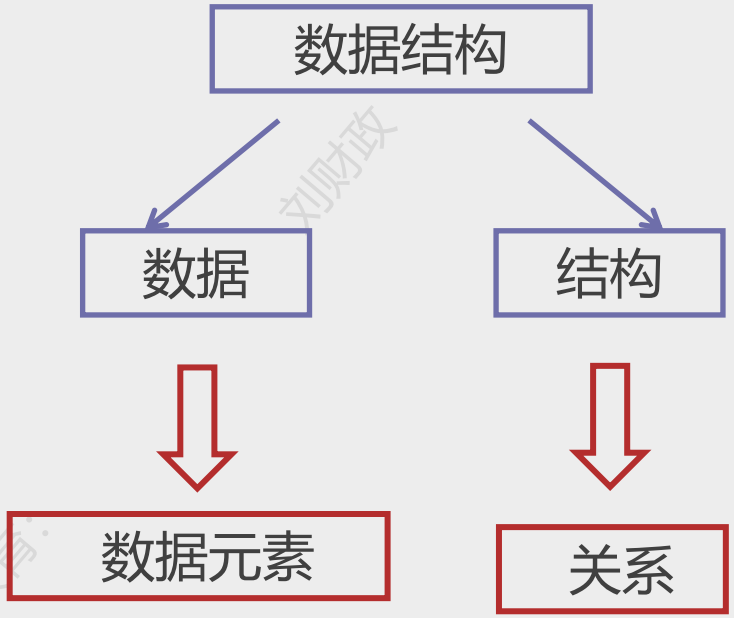
考点六：空间复杂度 ★★

考点框架

考点一：基本概念

考点一：基本概念

什么是数据结构呢？



什么是数据？什么是数据元素？

什么是结构？关系指的是什么？

考点一：基本概念

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

姓名	学历	年龄/岁	年收入/万元	房子	车子
张三	博士	31	60	有	无
王五	硕士	27	45	有	无
李四	硕士	26	40	无	有
张二	本科	24	25	无	无

考点一：基本概念

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

姓名	学历	年龄/岁	年收入/万元	房子	车子
张三	博士	31	60	有	无
王五	硕士	27	45	有	无
李四	硕士	26	40	无	有
张二	本科	24	25	无	无

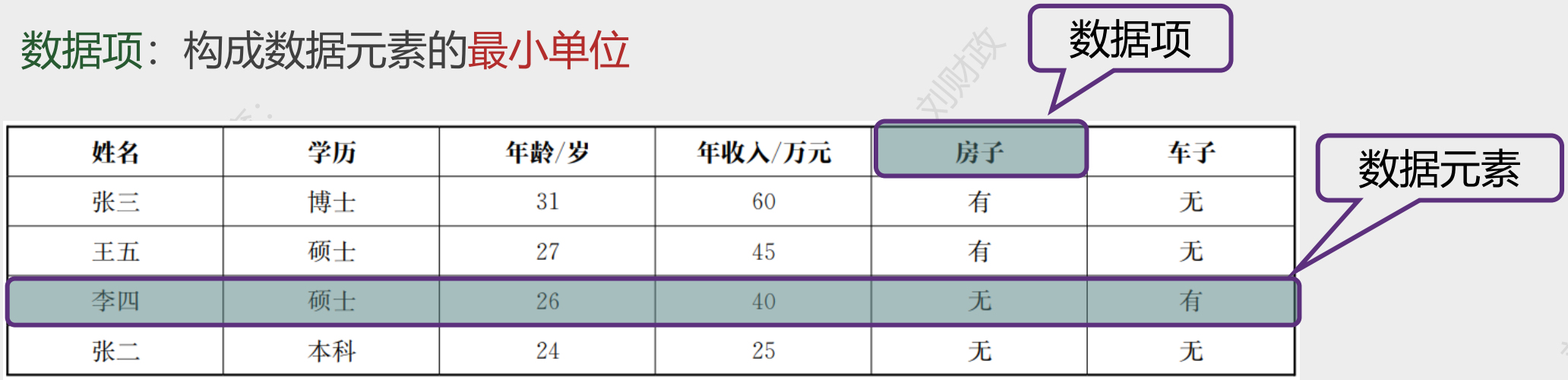
数据对象(Data Object)：是性质相同的数据元素的集合，是数据的一个子集。

如字符集合C={ ‘A’ ,’ B’ ,’ ‘C’ ,…} 。

考点一：基本概念

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

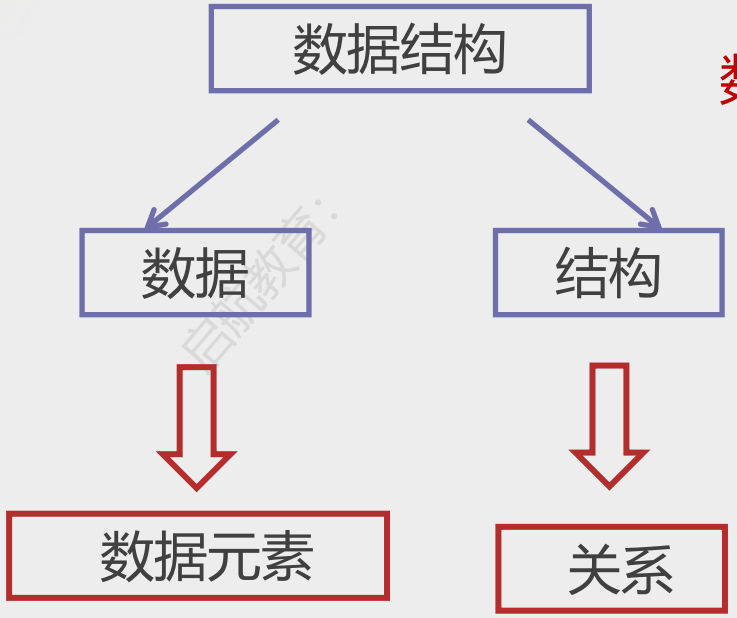


姓名	学历	年龄/岁	年收入/万元	房子	车子
张三	博士	31	60	有	无
王五	硕士	27	45	有	无
李四	硕士	26	40	无	有
张二	本科	24	25	无	无

通常情况下，数据元素具有相同个数和类型的数据项

数据元素（data element）是数据的基本单位，在计算机程序中通常作为一个整体考虑和处理。一个数据元素可由若干个**数据项**（data item）组成，数据项是数据不可分割的最小单位。

考点一：基本概念



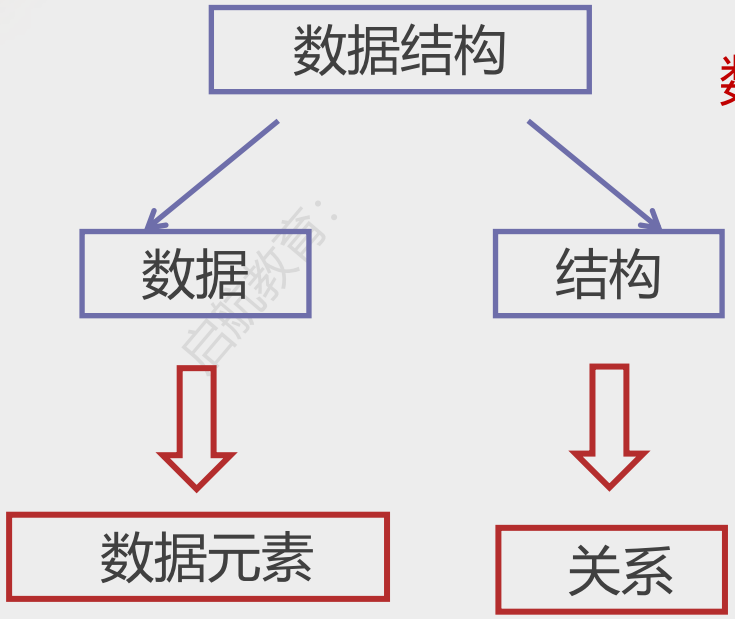
数据结构 = 数据对象 + 结构

相同性质的数据元素的集合

数据元素之间的关系构成结构

数据结构（data structure）即数据的组织形式，它是数据元素之间存在的一种或多种特定关系的数据元素集合。

考点一：基本概念



数据结构 = 数据对象 + 结构

相同性质的数据元素的集合

数据元素之间的关系构成结构

姓名	学历	年龄/岁	年收入/万元	房子	车子
张三	博士	31	60	有	无
王五	硕士	27	45	有	无
李四	硕士	26	40	无	有
张二	本科	24	25	无	无

考点一：基本概念

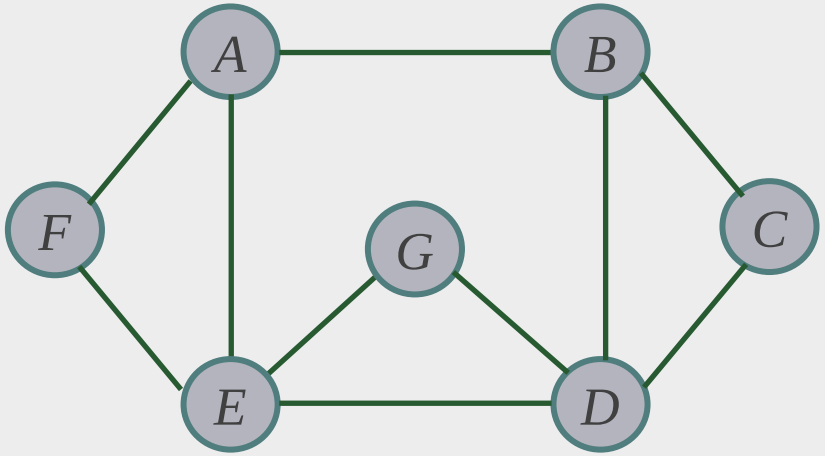
数据结构：相互之间存在一定**关系**的**数据元素**的集合

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：

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



Data_Structure = (D, R)

其中D = {*A*, *B*, *C*, *D*, *E*, *F*, *G*}

R = {R1}，R1 = {(*A*, *B*), (*A*, *E*), (*A*, *F*), (*B*, *C*), (*B*, *D*), (*C*, *D*), (*D*, *E*), (*D*, *G*), (*E*, *F*), (*E*, *G*)}

考点一：基本概念

数据结构三要素：

逻辑结构

存储结构

数据运算

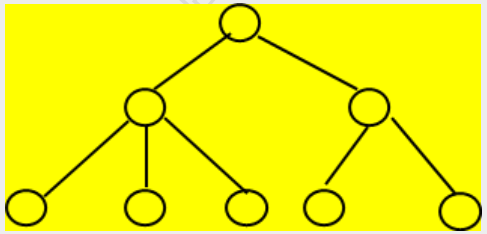
- 数据元素之间的逻辑关系 ⇨ 数据的逻辑结构。
- 数据元素及其关系在计算机存储器中的存储方式 ⇨ 数据的存储结构（或物理结构）。
- 施加在该数据上的操作 ⇨ 数据运算。

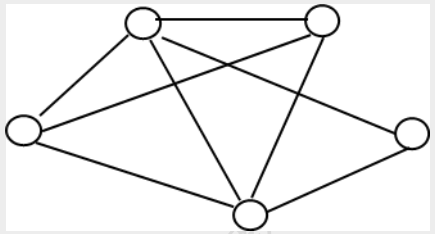
考点二：逻辑结构

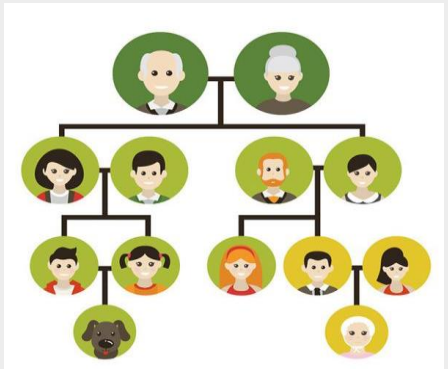
考点二：逻辑结构

逻辑结构

数据元素之间的关系可以是元素之间代表某种含义的自然关系，也可以是为处理问题方便而人为定义的关系，这种人为定义的 “关系”称为数据元素之间的逻辑关系，相应的结构称为逻辑结构/概念结构。



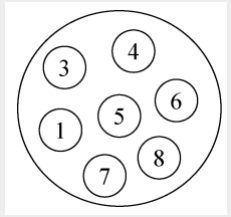




考点二：逻辑结构

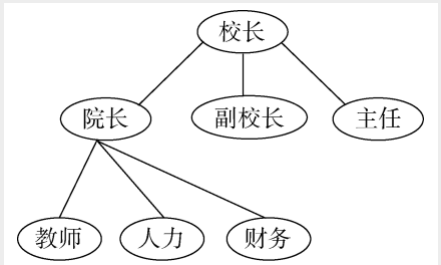
数据结构从逻辑上分为四类：

(1) 集合：数据元素之间没有关系

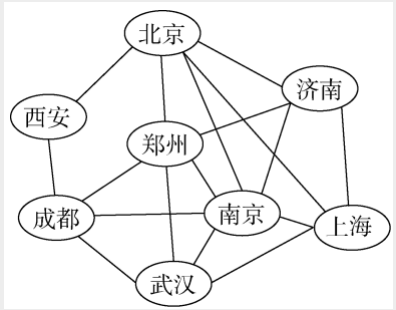


(2) 线性结构：数据元素之间是一对一的线性关系

(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



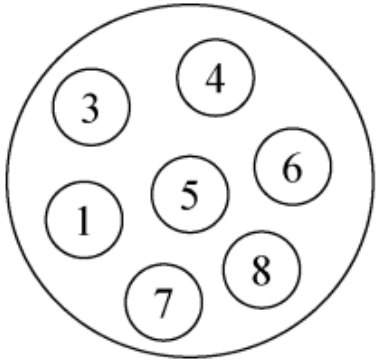
- 线性关系：线性结构
- 非线性关系：树结构和图结构

考点二：逻辑结构

（1）集合

元素之间关系：无。

特点：数据元素之间除了“属于同一个集合”的关系外，别无其他逻辑关系。是最松散的，不受任何制约的关系。



考点二：逻辑结构

（2）线性结构

元素之间关系：一对一。

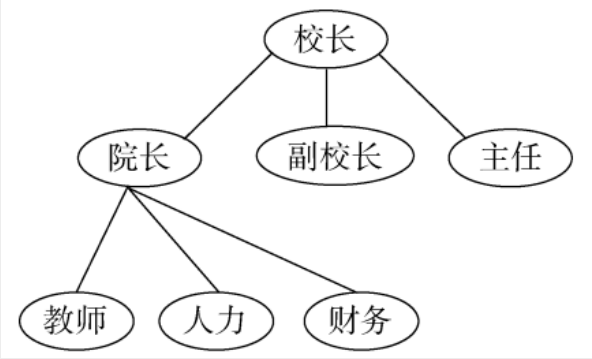
特点：开始元素和终端元素都是唯一的，除此之外，其余元素都有且仅有一个前驱元素和一个后继元素。

考点二：逻辑结构

（3）树形结构

元素之间关系：一对多。

特点：开始元素唯一，终端元素不唯一。除终端元素以外，每个元素有一个或多个后继元素；除开始元素外，每个元素有且仅有一个前驱元素。

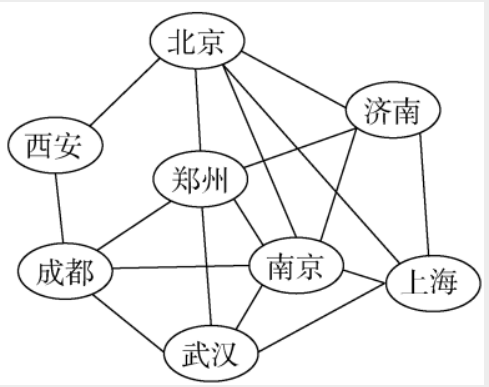


考点二：逻辑结构

(4) 图形结构

元素之间关系：多对多。

特点：所有元素都可能有多个前驱元素和多个后继元素。



考点二：逻辑结构

逻辑结构:（有时直接称为数据结构）

- 线性结构:　线性表、栈、队列、串(最多只有一一个直接前趋和一个直接后继
- 非线性结构: 树、图、多维数组、广义表

说明:

1、逻辑结构与数据元素本身的形式、内容无关

2、逻辑结构与数据元素的相对位置无关

3、逻辑结构与所含结点个数无关

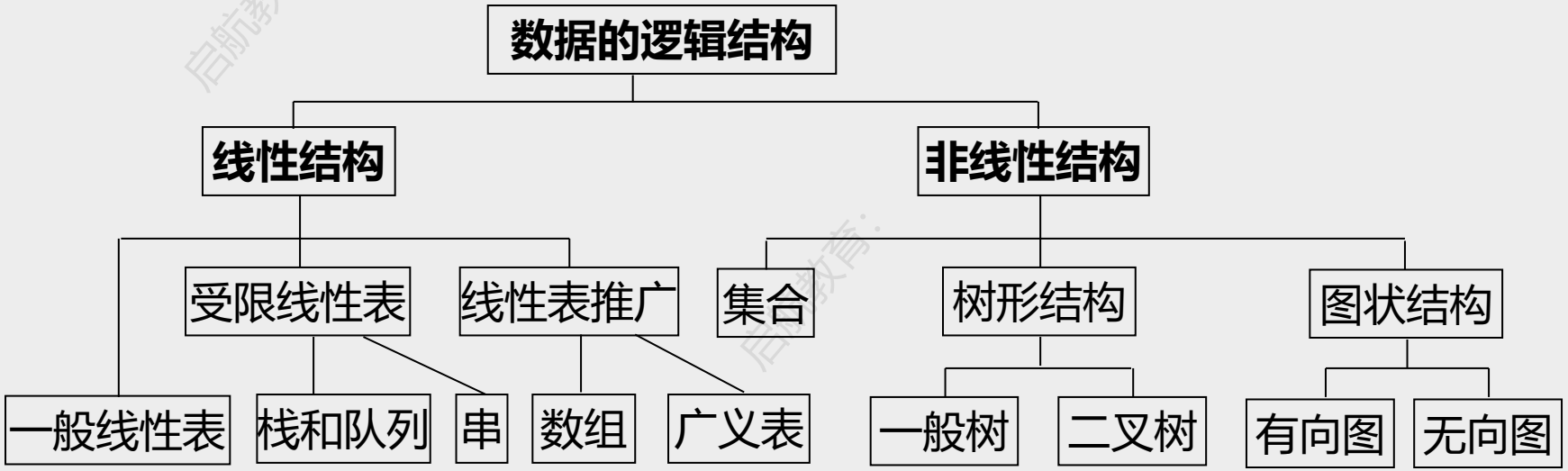
4、逻辑结构与计算机无关

【政哥点拨】

1．逻辑上通常可以将数据结构分为(　　)。

A．动态结构和静态结构　　B．顺序结构和链式结构

C．线性结构和非线性结构　　D．初等结构和组合结构



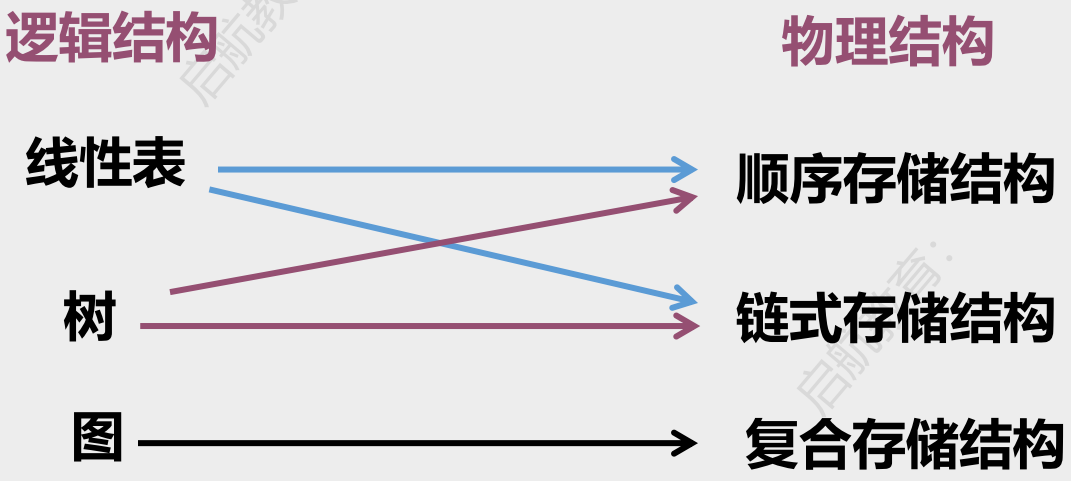
数据逻辑结构层次关系图

【政哥点拨】

1．逻辑上通常可以将数据结构分为(　　)。

A．动态结构和静态结构　　B．顺序结构和链式结构

C．线性结构和非线性结构　　D．初等结构和组合结构



逻辑结构与所采用的存储结构

【政哥点拨】

1．逻辑上通常可以将数据结构分为(　　)。

A．动态结构和静态结构　　B．顺序结构和链式结构

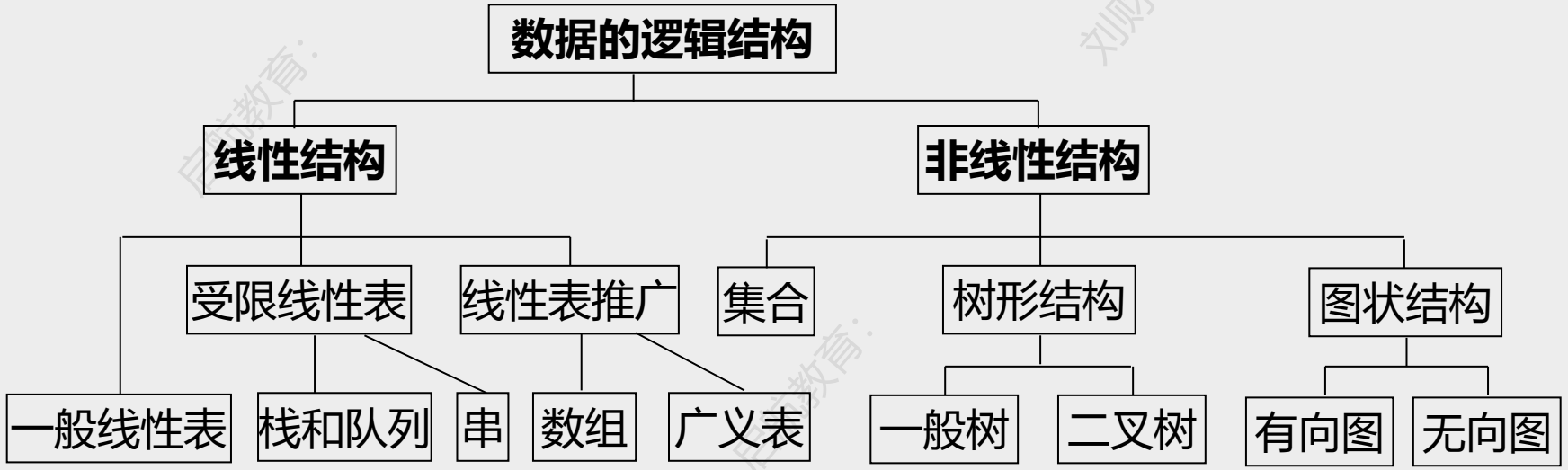
C．线性结构和非线性结构　　D．初等结构和组合结构

C 【解析】按照逻辑结构，可以把数据结构分为集合结构、线性结构、树形结构和图状结构，在分类上，集合结构、树形结构和图状结构统称为非线性结构，因此分为线性结构和非线性结构。B选项对应的是两个主要的存储结构，即顺序结构和链式结构。

【政哥点拨】

2．下列数据结构中，(　)是非线性数据结构。

A．栈　B．队列　C．二叉树　D．线性表



数据逻辑结构层次关系图

【政哥点拨】

2．下列数据结构中，（　）是非线性数据结构。

A．栈　B．队列　C．二叉树　D．线性表

C 【**解析**】按照逻辑结构，可以把数据结构分为集合结构、线性结构、树形结构和图状结构，其中线性结构包括线性表、栈、队列、广义表(广义线性结构)、串等；树形结构包括树、二叉树等；图形结构包括有向图和无向图、带权图和无权图等。

考点二：逻辑结构

什么是数据类型呢?

int a, b; ⇨ a = 10000000000000; 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

考点二：逻辑结构

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结构和基本操作

考点二：逻辑结构

抽象数据类型（ADT）指的是从求解问题的数学模型中抽象出来的数据逻辑结构和运算（抽象运算），而不考虑计算机的具体实现。

抽象数据类型 = 逻辑结构 + 抽象运算

例如，定义复数抽象数据类型Complex

一个复数的形式：e_1+e_2i 或（e_1，e_2）

ADT Complex
{
数据对象：
$D=\{ e_1, e_2 \mid e_1, e_2$均为实数 $\}$
数据关系：
$R=\{<e_1, e_2> \mid e_1$是复数的实部，e_2 是复数的虚部 $\}$

考点二：逻辑结构

基本运算：

AssignComplex(&z，v1，v2)：构造复数Z。

DestroyComplex(&z)：复数z被销毁。

GetReal(z，&real)：返回复数z的实部值。

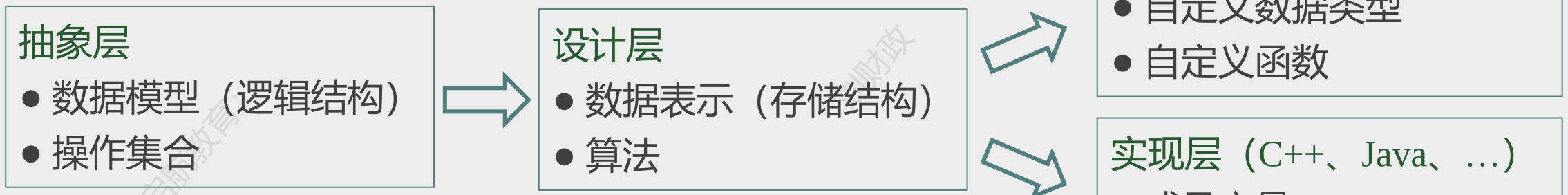
GetImag(z，&Imag)：返回复数z的虚部值。

Add(z1，z2，&sum)：返回两个复数z1、z2的和。

} ADT Complex

运算功能描述

考点二：逻辑结构



（a）定义——ADT定义

（b）设计——数据结构设计

（c）实现——程序语言实现

考点二：逻辑结构

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

考点二：逻辑结构

抽象数据类型：定义功能 ADT

线性结构　　树形结构　　图状结构

编程实现该数据结构

牛刀小试

1．在数据结构中，与所使用的计算机无关的是数据的(　)结构。

A．逻辑　　B．存储　　C．逻辑和存储　　D．物理

数据元素之间的关系可以是元素之间代表某种含义的自然关系，也可以是为处理问题方便而人为定义的关系，这种人为定义的“关系”称为数据元素之间的逻辑关系，相应的结构称为逻辑结构/概念结构。

牛刀小试

1．在数据结构中，与所使用的计算机无关的是数据的(　)结构。

A．逻辑　　B．存储　　C．逻辑和存储　　D．物理

A　【解析】本题考查逻辑结构独立于具体物理设备的性质。存储结构可能会因不同的计算机而存在差异，但是逻辑结构与具体的物理设备无关。

牛刀小试

2．线性结构是数据元素之间存在的一种(　　)。

A．一对多关系　　B．多对多关系

C．多对一关系　　D．一对一关系

(1) 集合：数据元素之间没有关系

(2) 线性结构：数据元素之间是一对一的线性关系

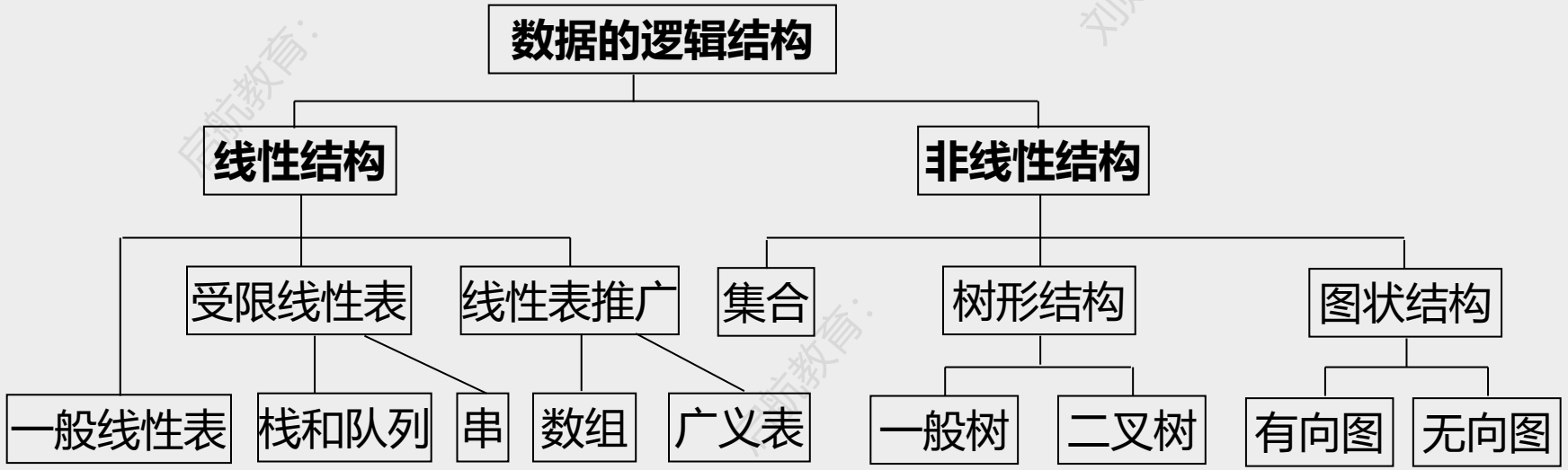
(3) 树结构：数据元素之间是一对多的层次关系

(4) 图结构：数据元素之间是多对多的任意关系

牛刀小试

3．以下哪个数据结构是非线性数据结构？（　）

A．树　B．字符串　C．队列　D．栈



数据逻辑结构层次关系图

牛刀小试

3．以下哪个数据结构是非线性数据结构？（　）

A．树　B．字符串　C．队列　D．栈

A　**【解析】**本题考查常见的线性结构和非线性结构。线性表属于线性结构，栈和队列也属于线性结构，都满足除第一个结点之外每个结点都有且仅有一个前驱结点，除最后一个结点之外每个结点都有且仅有一个后继结点。树是一种非线性结构，结点之间存在着一对多的关系，每个结点可能有0～2个后继结点。

牛刀小试

4．【南京邮电大学2014】如果数据结构中每个元素的前驱和后继的数目都不限，即元素之间存在M对N(M：N)的联系，且M和N可能均大于1，则该结构为四种基本逻辑结构中的(　)结构。

A．线性　　B．集合　　C．树形　　D．图状

(1) 集合：数据元素之间没有关系

(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

D【解析】元素之间存在M对N(M:N)的联系，并且M和N可能均大于1，那么就表示元素之间存在多对多的关系，因此采用图状结构来表示。

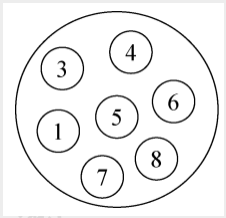
牛刀小试

5．【昆明理工大学2021】从逻辑关系来看，数据元素的直接前驱为0个或1个的数据结构只能是(　)。

A．线性结构　B．树形结构

C．线性结构和树形结构　D．线性结构和图状结构

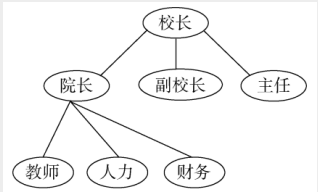
数据结构从逻辑上分为四类：



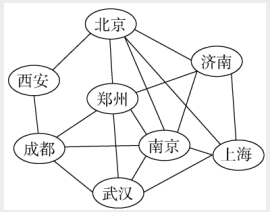
(1) 集合：数据元素之间没有关系

(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



牛刀小试

5．【昆明理工大学2021】从逻辑关系来看，数据元素的直接前驱为0个或1个的数据结构只能是(　)。

A．线性结构　B．树形结构

C．线性结构和树形结构　D．线性结构和图状结构

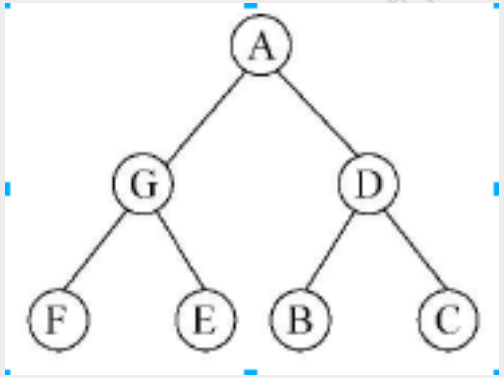
C【解析】逻辑结构中线性结构表示的是一对一关系，树形结构表示的是一对多关系，图状结构表示的是多对多关系。如果数据元素的直接前驱为0个或1个，那么表示的是线性结构或者树形结构，

牛刀小试

6．【西南交通大学2020】如果某数据结构的数据元素的集合为S={A,B,C,D,E,F,G}，数据元素之间的关系为R={ < A,D > , < A,G > , < D,B > , < D,C > , < G,E > , < G,F > }，则该数据结构最有可能是一种(　)。

A．图状结构　　B．循环结构　　C．线性结构　　D．树形结构

D【解析】根据题意，我们构建出其结构，如图所示，从图中可以看出其是树形结构。



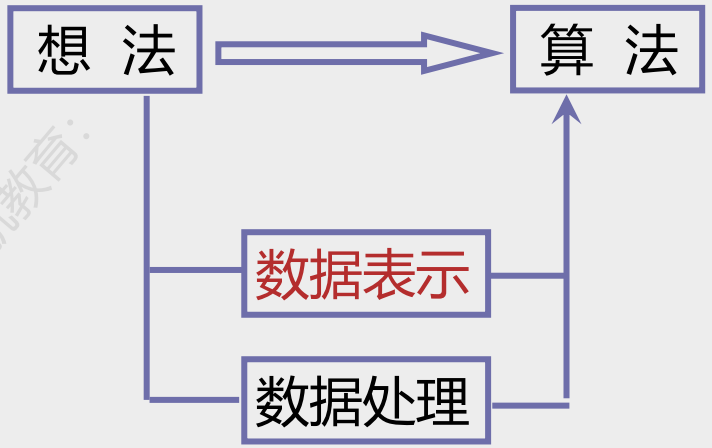
考点三：物理结构

考点三：物理结构

数据结构：相互之间存在一定**关系**的**数据元素**的集合

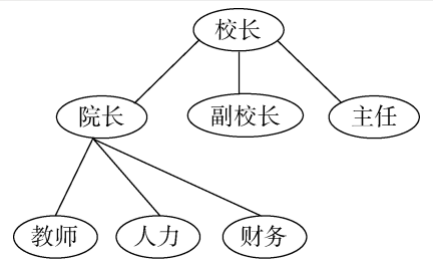
数据的存储（物理）结构：数据及其逻辑结构在**计算机**中的表示

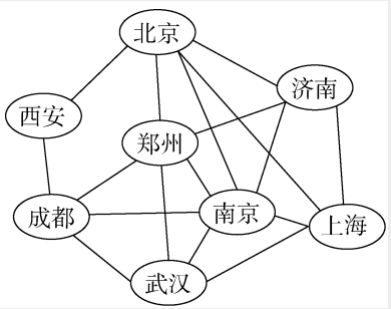
计算机语言如何进行内存分配？



存储结构实质上是内存分配，在具体实现时依赖于计算机语言

考点三：物理结构





在存储程序的计算机中，数据和指令都是以二进制形式存储在存储器中的。从存储器存储的内容来看两者并无区别，都是由0和1组成的二进制序列。

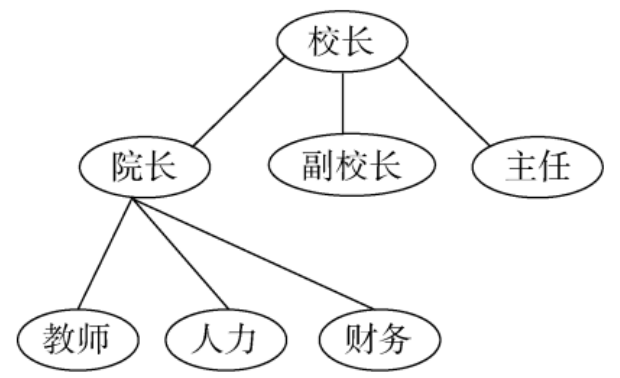
考点三：物理结构

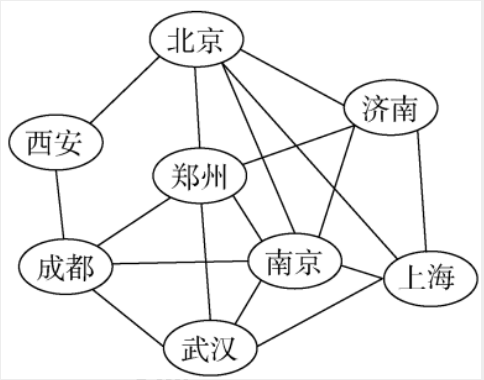
数据的存储（物理）结构：数据及其逻辑结构在**计算机**中的表示

数据元素

逻辑关系

内存





数据元素

- 数据元素
- 也就是各个属性值（数据项）

逻辑结构

- 线性结构，集合结构
- 图型结构，树型结构

考点三：物理结构

物理结构有四种基本类型：

① 顺序结构

② 链式结构

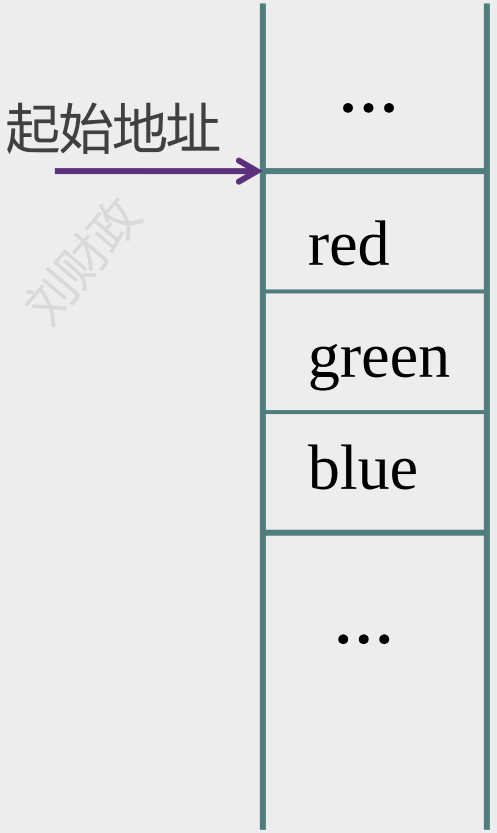
③ 索引结构

④ 散列结构

考点三：物理结构

（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

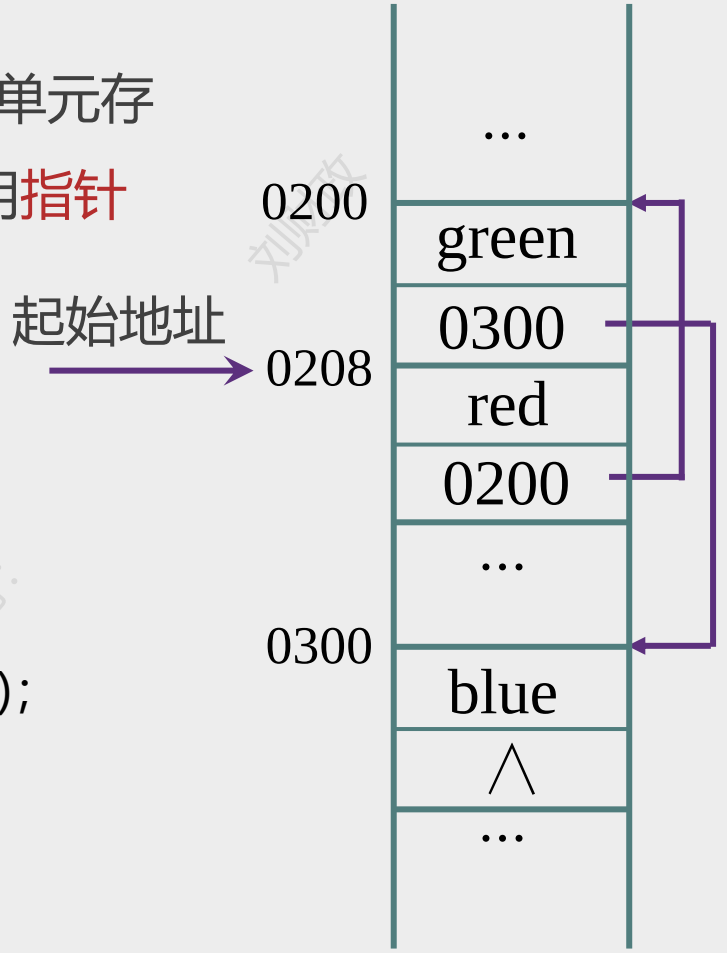
- 在C语言中，通常用一维数组表示顺序存储结构；
- 数据元素存放的地址是连续的；
- 随机存取表中元素；
- 插入和删除操作需要移动元素；



存储结构实质上是内存分配，在具体实现时依赖于计算机语言

考点三：物理结构

（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



- 用结构体类型表示链式存储结构；
- 比顺序存储结构的存储密度小；
- 逻辑上相邻的节点物理上不必相邻；
- 插入、删除灵活 (只要改变节点中的指针)；
- 查找结点时链式存储要比顺序存储慢。
- 每个结点是由数据域和指针域组成；

存储结构实质上是内存分配，在具体实现时依赖于计算机语言

考点三：物理结构

(3)索引结构：除建立存储结点信息外，还建立附加的索引表来标识结点的地址。索引表由若干索引项组成。

索引	数据
1	3.0
2	2.3
3	3.5
4	5.0
5	-8.5
6	11.0

- 索引存储结构是用结点的索引号来确定结点存储地址；
- 优点是检索速度快；
- 缺点增加了附加的索引表,会占用较多的存储空间；

存储结构实质上是内存分配，在具体实现时依赖于计算机语言

考点三：物理结构

(4)散列存储方法:散列存储，又称hash存储，是一种试图将数据元素的存储位置与关键码之间建立确定对应关系的查找技术。

- x(数据元素) $\xrightarrow{f}$ y（存储位置）
- 散列法存储的基本思想是：由节点的关键码值决定节点的存储地址。

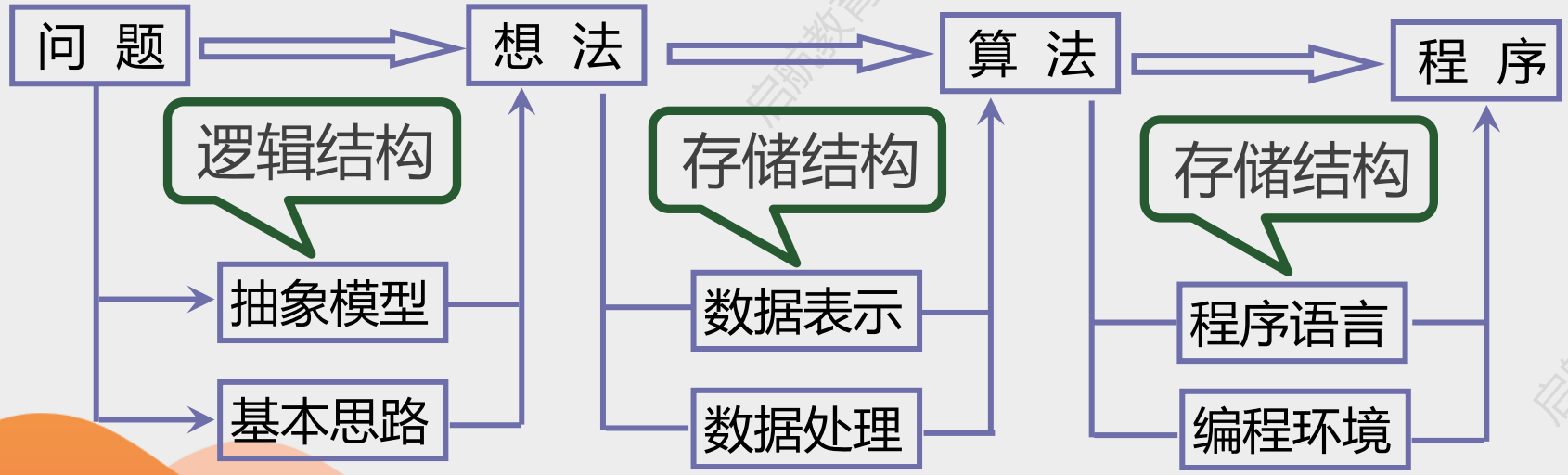
散列技术除了可以用于查找外，还可以用于存储。

存储结构实质上是内存分配，在具体实现时依赖于计算机语言

考点三：物理结构

逻辑结构和存储结构的关系

- 数据的逻辑结构属于用户视图，是面向问题的
- 数据的存储结构属于具体的实现视图，是面向计算机的
- 数据的逻辑结构反映了数据本身的构成方式
- 数据的存储结构反映了数据在内存的存储表示



考点三：物理结构

- 增加(Create)
- 检索(Retrieve)
- 更新(Update)
- 删除(Delete)
- 。。。

【政哥点拨】

1．【江苏大学2019，中国石油大学2019】在设计存储结构时，通常不仅要存储各数据元素的值，还要存储(　)。

A．数据的处理方法　　B．数据元素的类型

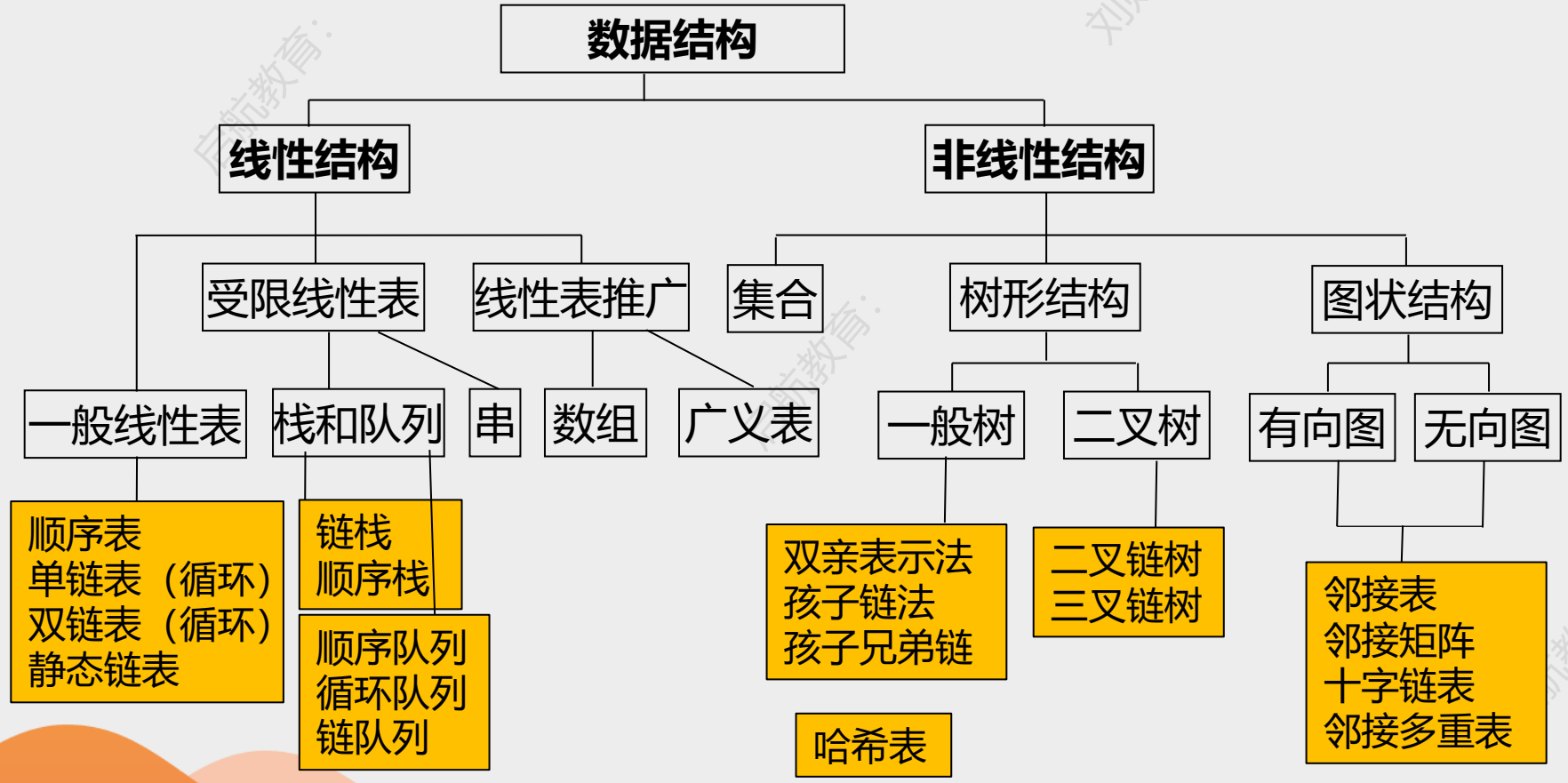
C．数据元素之间的关系　D．数据的存储方法

C 【**解析**】冯·诺依曼计算机告诉我们，指令和数据都是以二进制的形式存放在内存中的。因此，在设计时，物理结构必须包括数据元素的存储和元素之间的关系的表示，并且物理结构必须体现逻辑结构的逻辑关系是设计的关键。因此，在设计存储结构时，通常不仅要存储各数据元素的值，还要存储数据元素之间的关系。

【政哥点拨】

2．【烟台大学2019】以下与数据的存储结构无关的术语是(　　)。

A．循环队列　B．链表　C．哈希表　D．栈



【政哥点拨】

2．【烟台大学2019】以下与数据的存储结构无关的术语是(　　)。

A．循环队列　B．链表　C．哈希表　D．栈

D **【解析】** 本题比较综合，这里总结一下，大家先记住，逻辑结构包括线性表、栈、队列、树、二叉树、有向图、无向图、带权图、无权图、哈夫曼树、完全二叉树和满二叉树。

物理结构包括顺序表、单链表、双链表、静态链表、链栈、顺序栈、顺序队列、循环队列、链队列、二叉链树、三叉链树、邻接表、邻接矩阵和哈希表等。

牛刀小试

1．【南京理工大学2018】数据结构在计算机内存中的表示是指(　　)。

A．数据的存储结构　　B．数据结构

C．数据的逻辑结构　　D．数据元素之间的关系

A　**【解析】**　本题考查数据结构的物理结构的定义。数据的存储结构是指数据结构在计算机内部的实际存储表示。

牛刀小试

2．下列文件的物理结构中，不利于文件长度动态增长的文件物理结构是(　)。

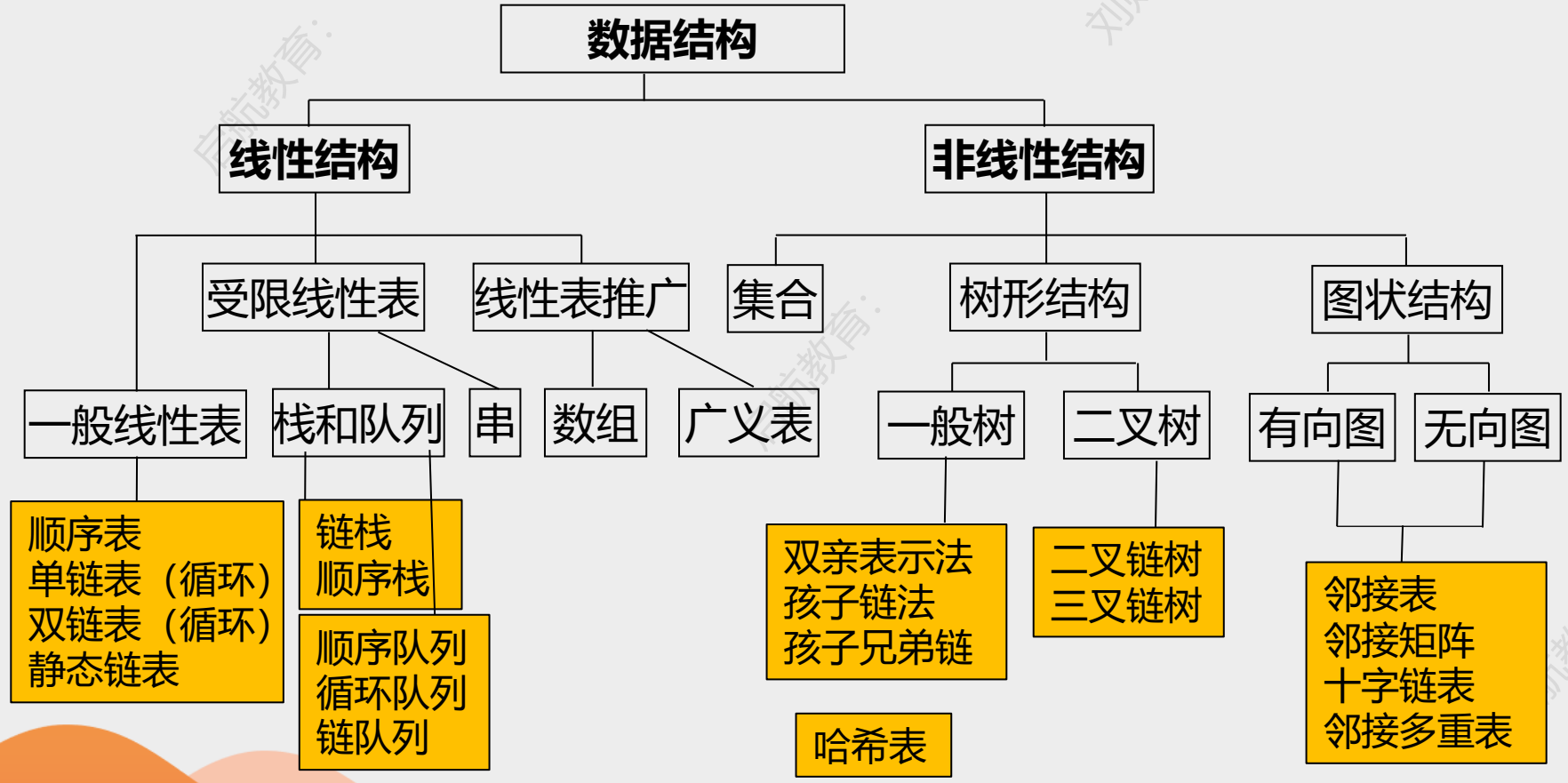
A．顺序结构　B．链接结构　C．索引结构　D．Hash结构

A【解析】本题考查顺序存储结构不利于文件长度动态增长的性质。顺序结构，即顺序表，需要预先估计存储空间的大小，不利于文件长度的动态增长。

牛刀小试

3．以下属于逻辑结构的是(　)。

A．顺序表　B．哈希表　C．有序表　D．单链表



牛刀小试

4．在设计连续存储时，存储单元的地址(　　)。

A．一定连续　B．一定不连续

C．不一定连续　D．部分连续，部分不连续

A 【**解析**】顺序存储结构需要通过相对位置来体现逻辑结构，其存储需要占用一个连续空间，存储地址一定是连续的。链式结构、索引结构、散列结构的存储位置可以连续，也可以不连续。

牛刀小试

5．【扬州大学2016】数据在计算机存储器内表示时，根据结点的关键字直接计算出该结点的存储地址，这种方法称为(　)。

A．索引存储方法　B．顺序存储方法

C．链式存储方法　D．散列存储方法

D【解析】将数据元素的存储位置与关键码之间建立确定对应关系的查找技术。散列法存储的基本思想是由结点的关键码值决定结点的存储地址。

牛刀小试

6．【昆明理工大学2021】数据的四种基本存储结构是指(　　)。

A．顺序存储结构、索引存储结构、直接存储结构、倒排存储结构

B．顺序存储结构、索引存储结构、链式存储结构、散列存储结构

C．顺序存储结构、非顺序存储结构、指针存储结构、树形存储结构

D．顺序存储结构、链式存储结构、树形存储结构、图形存储结构

B【解析】物理结构有四种基本类型：

① 顺序结构

② 链式结构

③ 索引结构

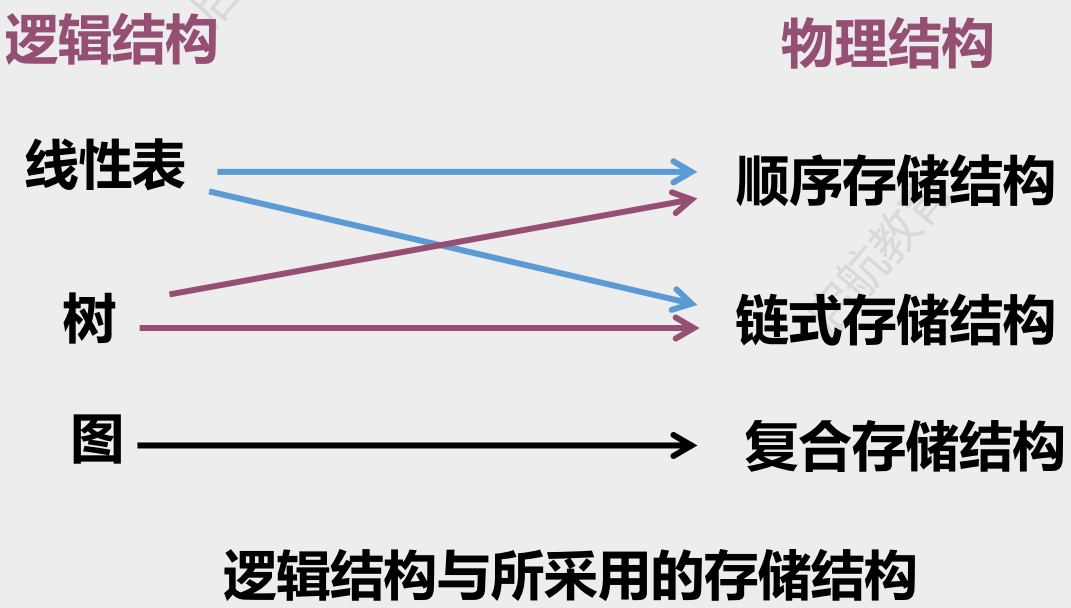
④ 散列结构

牛刀小试

7．【重庆理工大学2017】数据元素之间的存储结构，除了链式存储结构，另外一种存储结构是(　)。

A．线性存储结构　B．树形存储结构

C．顺序存储结构　D．图形存储结构



逻辑结构与所采用的存储结构

C　**【解析】**　物理结构包括四种，分别是顺序存储结构、索引存储结构、链式存储结构、散列存储结构，通常使用时，我们将存储结构分为链式存储结构和顺序存储结构，因为这两种存储结构最常用。

牛刀小试

8．【长沙理工大学2019】每个结点只含有一个数据元素，所有存储结点相继存放在一个连续的存储区里，这种存储结构称为(　)结构。

A．顺序存储　B．链式存储　C．索引存储　D．散列存储

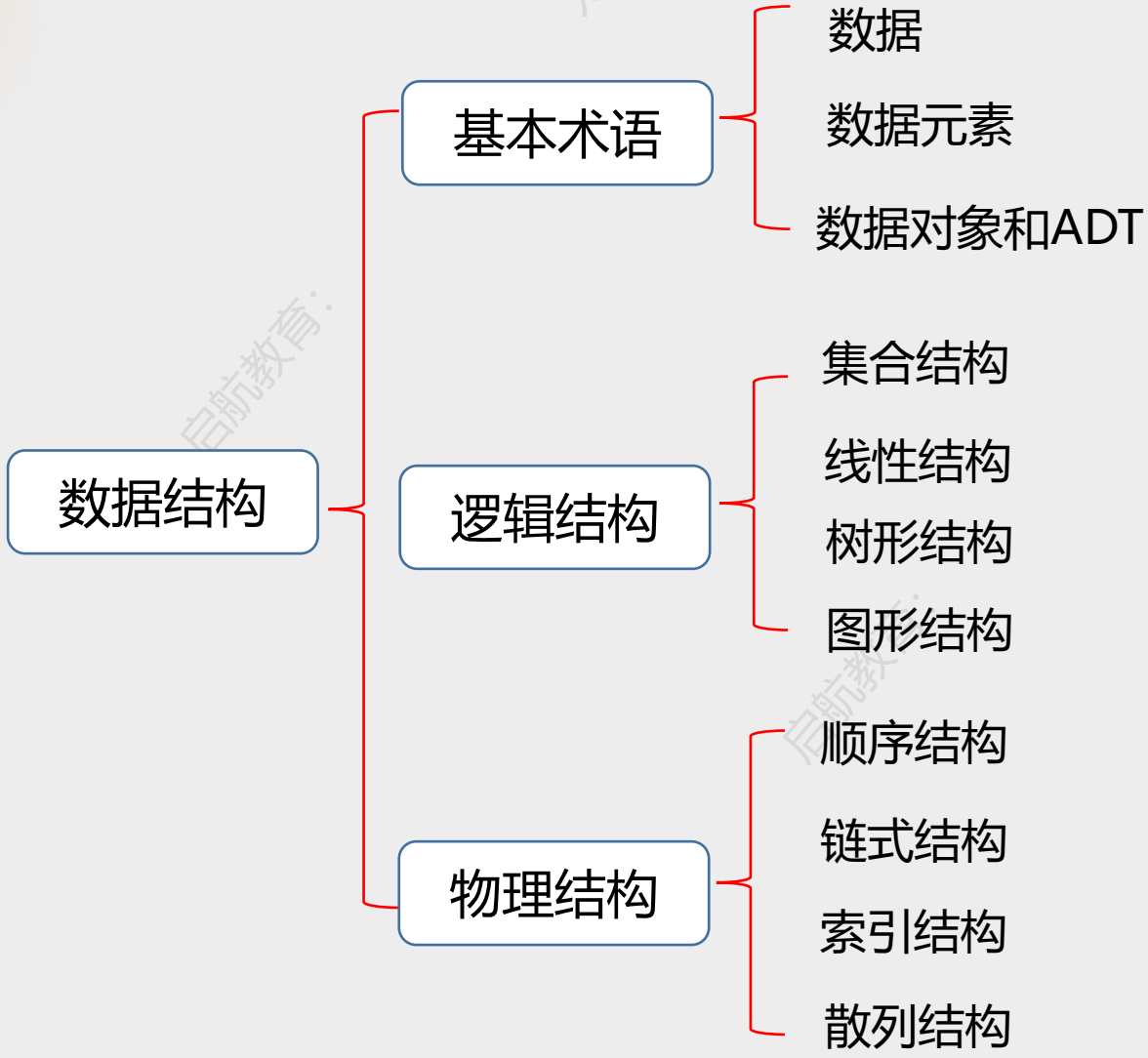
A【解析】物理结构有四种基本类型：

① 顺序结构：通过位置相邻来表示逻辑关系

② 链式结构：通过指针来表示逻辑关系

③ 索引结构：通过索引表来表示逻辑关系

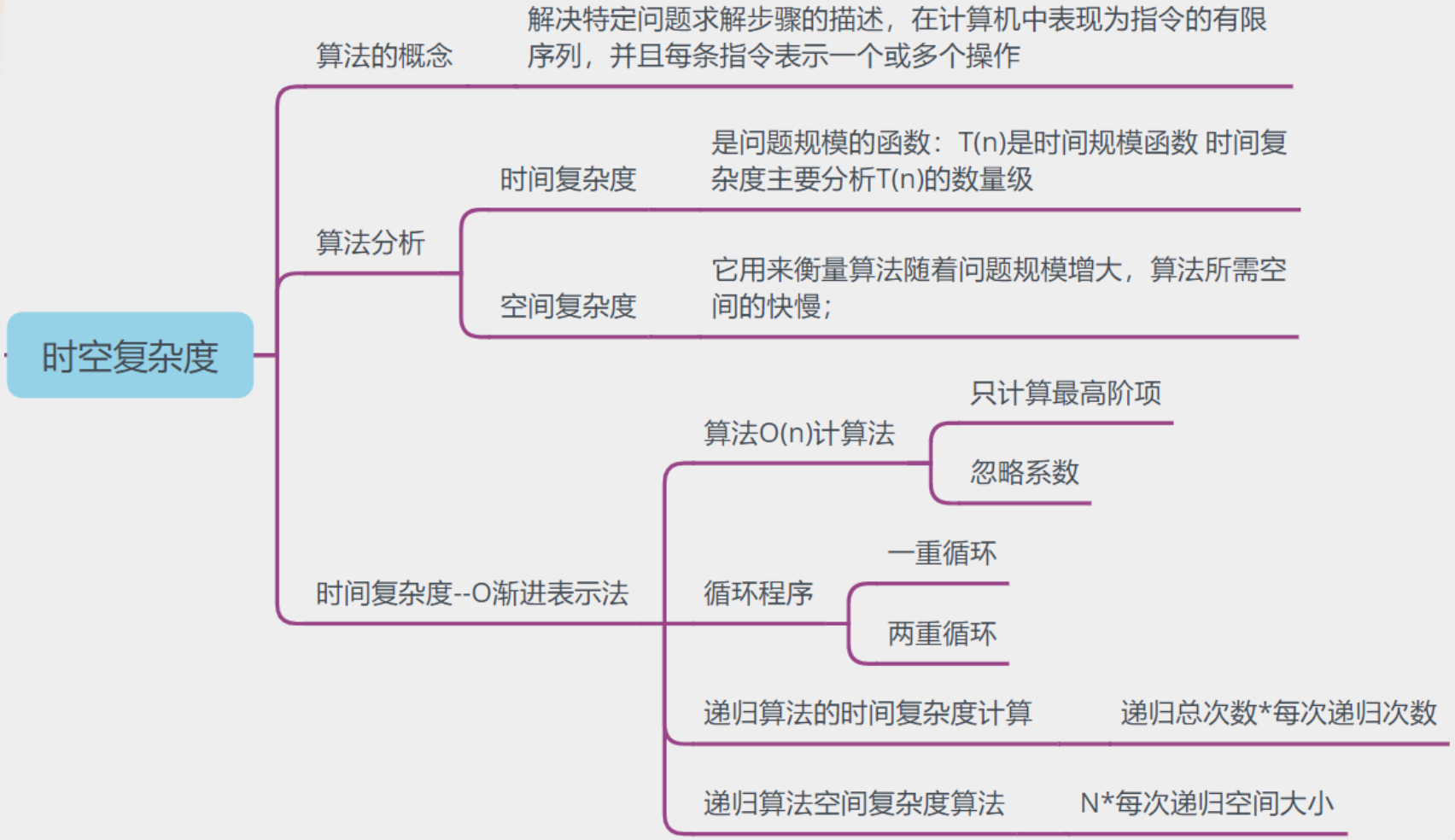
④ 散列结构：通过哈希函数来表示逻辑关系


数据结构
基本术语
数据
数据元素
数据对象和ADT
逻辑结构
集合结构
线性结构
树形结构
图形结构
物理结构
顺序结构
链式结构
索引结构
散列结构

谢谢大家

考点四：算法的概念和评价

考点框架


时空复杂度
算法的概念
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
算法分析
时间复杂度
是问题规模的函数：T(n)是时间规模函数 时间复杂度主要分析T(n)的数量级
空间复杂度
它用来衡量算法随着问题规模增大，算法所需空间的快慢；
时间复杂度--O渐进表示法
算法O(n)计算法
只计算最高阶项
忽略系数
循环程序
一重循环
两重循环
递归算法的时间复杂度计算
递归总次数*每次递归次数
递归算法空间复杂度算法
N*每次递归空间大小

考点四：算法的概念和评价

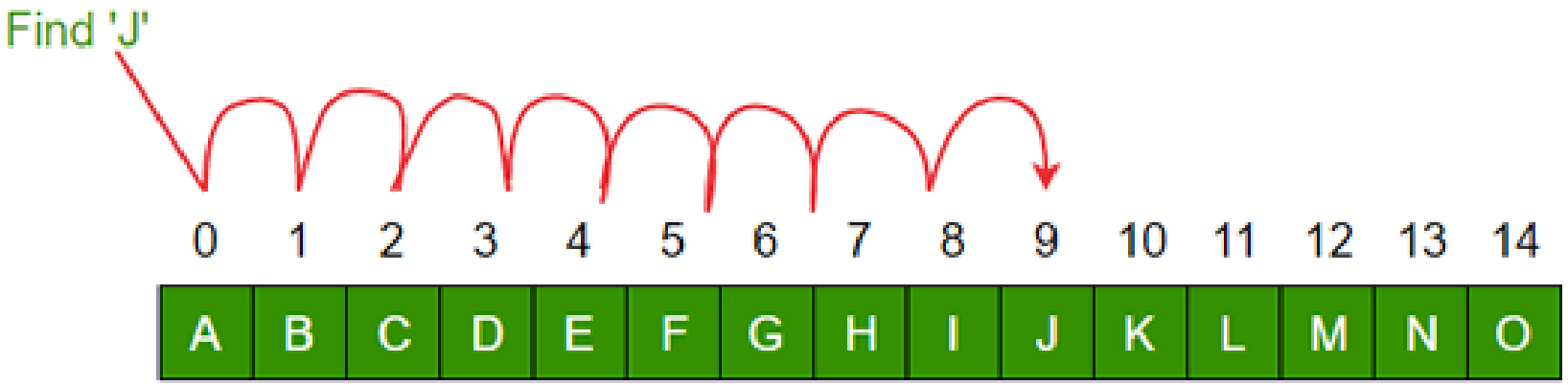
算法(Algorithm)：是对特定问题求解方法(步骤)的一种描述，是指令的有限序列，其中每一条指令表示一个或多个操作。

算法可以有三种表示形式：

- 伪代码
- 自然语言
- 流程图

考点四：算法的概念和评价

在一群数中，查找特定值



考点四：算法的概念和评价

输入 → 算法 → 输出

算法描述的一般格式

```
返回值  算法对应的函数名(形参列表)
{  //临时变量的定义
   //实现由输入参数到输出参数的操作
          …
}
```

函数体

- 返回值：通常为bool类型，表示算法是否成功执行。
- 形参列表：由输入型参数和输出型参数构成。

算法输入　　算法输出

考点四：算法的概念和评价

算法的五个重要的特性

(1) 有穷性：在有穷步之后结束，算法能够停机。

(2) 确定性：无二义性。

(3) 可行性：可通过基本运算有限次执行来实现，
也就是算法中每一个动作能够被机械地执行。

(4) 输入：0个或多个输入

(5) 输出：1个或多个输出

} 表示存在数据处理

考点四：算法的概念和评价

算法和程序的区别

不一定满足有穷性

- 程序是指使用某种计算机语言对一个算法的具体实现，即具体要怎么做。
- 算法侧重于对解决问题的方法描述，即要做什么。

一定满足有穷性

- 算法用计算机语言描述 ⇨ 程序

考点四：算法的概念和评价

什么是算法分析？

算法设计：面对一个问题，如何设计一个有效的算法

算法分析：对已设计的算法，如何评价或判断其优劣

考点四：算法的概念和评价

分析算法占用的资源

- CPU时间 → 时间性能分析
- 内存空间 → 空间性能分析

算法分析目的：分析算法的时空效率以便改进算法性能。

考点四：算法的概念和评价

效率指的是算法执行的时间

存储量需求指算法执行过程中所需要的最大存储空间

考点五：时间复杂度

考点五：时间复杂度

算法执行时间需通过依据该算法编制的程序在计算机上运行所消耗的时间来度量。其方法通常有两种：

事后统计：计算机内部进行执行时间和实际占用空间的统计。

问题：必须先运行依据算法编制的程序；依赖软硬件环境，容易掩盖算法本身的优劣；没有实际价值。

考点五：时间复杂度

算法执行时间需通过依据该算法编制的程序在计算机上运行所消耗的时间来度量。其方法通常有两种：

事后统计：计算机内部进行执行时间和实际占用空间的统计。

- 编写程序的语言不同
- 执行程序的环境不同
- 其他因素

所以不能用绝对执行时间进行比较。

考点五：时间复杂度

算法执行时间需通过依据该算法编制的程序在计算机上运行所消耗的时间来度量。其方法通常有两种：

事前分析：求出该算法的一个时间界限函数。

与此相关的因素有：

- 依据算法选用何种策略；
- 程序设计的语言；
- 编译程序所产生的机器代码的质量；
- 机器执行指令的速度；
- 问题的规模；
- 数据的初始状态有关

考点五：时间复杂度

一个算法是由控制结构（顺序、分支和循环三种）和原操作构成的。

算法的基本构成：

控制语句1 原操作　控制语句2 原操作　…　控制语句n 原操作

- 顺序结构：按照所述顺序处理
- 分支结构：根据判断条件改变执行流程
- 循环结构：当条件成立时，反复执行给定的处理操作

指固有数据类型的操作，如+、-、*、/、++和--等

- 算法执行时间取决于两者的综合效果。

考点五：时间复杂度

看一个简单例子

```
void fun(int a[], int n)
{   int i;
    for (i=0;i<n;i++)
        a[i]=2*i;
    for (i=0;i<n;i++)
        printf("%d",  a[i]);
    printf("\n");
}
```

原操作

考点五：时间复杂度

算法执行时间需通过依据该算法编制的程序在计算机上运行所消耗的时间来度量。其方法通常有两种：

事前分析：求出该算法的一个时间界限函数。

- 求出算法所有原操作的执行次数（也称为频度），它是问题规模n的函数，用T(n)表示。
- 算法执行时间 = 原操作所需的时间×T(n)。所以T(n)与算法的执行时间成正比。
- 用T(n)表示算法的执行时间。

考点五：时间复杂度

若$T(n) = a_m n^m + a_{m-1} n^{m-1} + \ldots + a_1 n + a_0$是一个$m$次多项式，则$T(n) = O(n^m)$

关注增长率——忽略所有低次幂和最高次幂的系数

考点五：时间复杂度

算法执行时间需通过依据该算法编制的程序在计算机上运行所消耗的时间来度量。其方法通常有两种：

事前分析：求出该算法的一个时间界限函数。

- 算法中基本操作重复执行的次数是问题规模n的某个函数，其时间量度记作 T(n)=O(f(n))，称作算法的渐近时间复杂度(Asymptotic Time complexity)，简称时间复杂度。
- 一般地，常用最深层循环内的语句中的原操作的执行频度(重复执行的次数)来表示。

考点五：时间复杂度

常量阶

```
{++x;  s=0 ;}
```

将x自增看成是基本操作，则语句频度为 1，时间复杂度为O(1) 。

将s=0也看成是基本操作，则语句频度为 2，时间复杂度仍为O(1)。

考点五：时间复杂度

线性阶

```
for(i=1; i<=n; ++i) {
    ++x;  ①
    s+=x ; ②
}
```

i = 1 ①②
i = 2 ①②
i = 3 ①②
i = n ①②

n次

语句频度为：2n，其时间复杂度为：O(n)，即为线性阶。

考点五：时间复杂度

平方阶

```
for(i=1; i<=n; ++i){
    for(j=1; j<=n; ++j) {
        ++x;
        s+=x ;
    }
}
```

①

i = 1 ①

i = 2 ①

i = 3 ①

i = n ①

n次

2n

语句频度为：$2n^2$，其时间复杂度为：$O(n^2)$，即为平方阶。

考点五：时间复杂度

三次方阶

两个n阶方阵的乘法：给定a[][],b[][],c[][]

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

i = 1

j = 1　　j = 2　　j = n

k = 1　　k = 1　　k = 1

k = 2　　k = 2　　k = 2

k = 3　　k = 3　　k = 3

k = n　　k = n　　k = n

考点五：时间复杂度

三次方阶

两个n阶方阵的乘法：给定a[][],b[][],c[][]

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

i = 2

j = 1　　j = 2　　j = n

k = 1　　k = 1　　k = 1

k = 2　　k = 2　　k = 2

k = 3　　k = 3　　k = 3

k = n　　k = n　　k = n

考点五：时间复杂度

三次方阶

两个n阶方阵的乘法：给定a[][],b[][],c[][]

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

由于是一个三重循环，每个循环从1到n，则总次数为： $n\times n\times n=n^3$ 时间复杂度为$T(n)=O(n^3)$

i = n

j = 1　　j = 2　　j = n

k = 1　　k = 1　　k = 1

k = 2　　k = 2　　k = 2

k = 3　　k = 3　　k = 3

k = n　　k = n　　k = n

考点五：时间复杂度

对数阶

下列程序段的时间复杂度是（）。

```
count=1;
while（count < n）
    count *= 2;
```

A．$O(\log_2 n)$　　B．$O(n)$　　C．$O(n\log_2 n)$　　D．$O(n^2)$

i = 1　count = count * 2

i = 2　count = count * 2 * 2 = count * 2^2

i = 3　count = count * 2 * 2 * 2 = count * 2^3

i = k　count = count * 2 *2…*2* 2 = count * 2^k<n

k个2

count初值是1，得到 2^k<n

=> $O(\log_2 n)$

考点五：时间复杂度

对数阶

下列程序段的时间复杂度是（ ）。

```
count=0;
for(k=1;k<=n;k*=2)
    count++;
```

A.$O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

i = 1　k = k * 2

i = 2　k = k * 2 * 2 = k * 2^2

i = 3　k = k * 2 * 2 * 2 = k * 2^3

i = x　k = k * 2 *2…*2* 2 = k * 2^x <= n

x个2

k初值是1，得到 $2^x <= n$ => $O(\log_2 n)$

考点五：时间复杂度

递归

求整数n(n≥0)阶乘的算法如下，其时间复杂度是（ ）。

```
int fact(int n) {
    if (n<=1)    return 1;
    return   n * fact(n-1);
}
```

递归：就是在运行的过程中不断地调用自己

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

考点五：时间复杂度

递归

求整数n(n≥0)阶乘的算法如下，其时间复杂度是（ ）。

```
int fact(int n) {
    if (n<=1)    return 1;
    return   n * fact(n-1);
}
```

递归：就是在运行的过程中不断地调用自己

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

考点五：时间复杂度

递归

求整数n(n≥0)阶乘的算法如下，其时间复杂度是（ ）。

```
①
int fact(int n) {
    if (n<=1)② return 1;
    return   n * fact(n-1);
                      ③
}
```

递归三要素
- ①递归初始条件
- ②递归终止条件
- ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

考点五：时间复杂度

递归

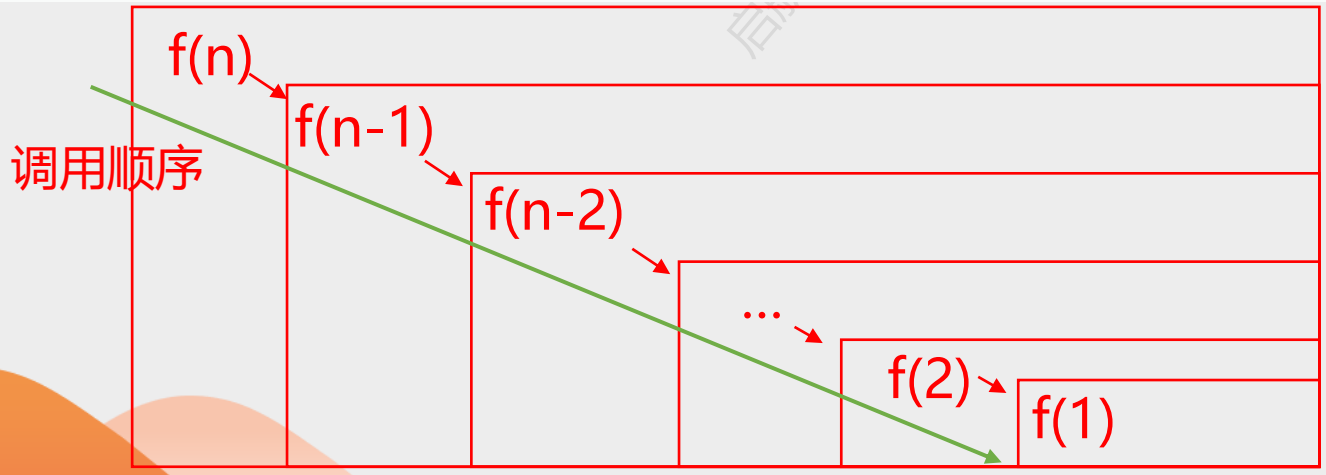
求整数n(n≥0)阶乘的算法如下，其时间复杂度是（ ）。

```
int fact(int n) {
    if (n<=1)    return 1;
    return   n * fact(n-1);
}
```

①递归初始条件
②递归终止条件
③递归转移条件

递归三要素

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$



考点五：时间复杂度

递归

求整数n(n≥0)阶乘的算法如下，其时间复杂度是（ ）。

```
int fact(int n) {
    if (n<=1)    return 1;
    return   n * fact(n-1);
}
```

①

③

递归三要素
- ①递归初始条件
- ②递归终止条件
- ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

n $\xrightarrow{n=n-1}$ 1；共执行了n次，所以时间复杂度是O(n)

考点五：时间复杂度

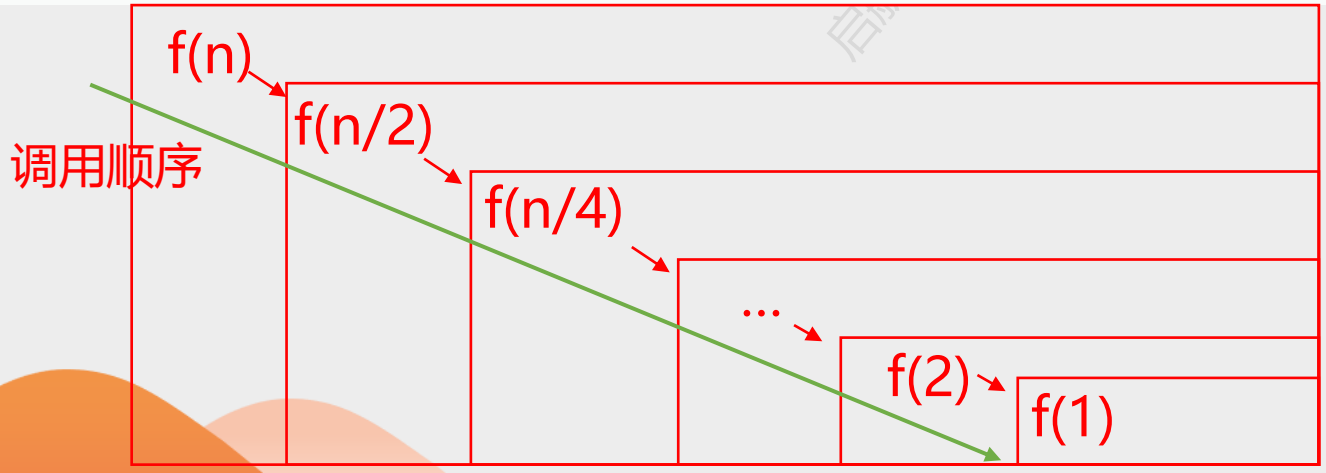
递归

给定下面算法如下，其时间复杂度是（ ）。

```
①
int fact(int n) {
    if (n<=1)②  return 1;
    return   1 + fact(n/2);
                    ③
}
```

递归三要素
- ①递归初始条件
- ②递归终止条件
- ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$



考点五：时间复杂度

递归

给定下面算法如下，其时间复杂度是（ ）。

```
①
int fact(int n) {
    if (n<=1)② return 1;
    return   1+ fact(n/2);
                    ③
}
```

递归三要素
- ①递归初始条件
- ②递归终止条件
- ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

n $\xrightarrow{n=n/2}$ 1 共执行了$\log_2 n$次，所以时间复杂度是$O(\log_2 n)$

考点五：时间复杂度

常见的时间复杂度：

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

多项式时间，易解问题 | 指数时间，难解问题

时间复杂度是在不同数量级的层面上比较算法

考点五：时间复杂度

时间复杂度的三种情形

- 最好情况：不能代表算法的效率，当出现概率较大时分析
- 最坏情况：最坏能坏到什么程度，实时系统需要分析
- 平均情况：已知输入数据分布情况，通常假设等概率分布

如果算法的时间代价与输入数据有关，则需要分析最好情况、最坏情况、平均情况

【政哥点拨】

1、设 n 是描述问题规模的非负整数，下面程序片段的时间复杂度是_____。

```
x=2;
while（x<n/2）
    x=2*x;
```

A．$O(\log_2 n)$　　B．$O(n)$　　C．$O(n\log_2 n)$　　D．$O(n^2)$

i = 1　x = x * 2

i = 2　x = x * 2 * 2 = x * 2^2

i = 3　x = x * 2 * 2 * 2 = x * 2^3

i = k　x = x * 2 *2…*2* 2 = x * 2^k<n/2

k个2

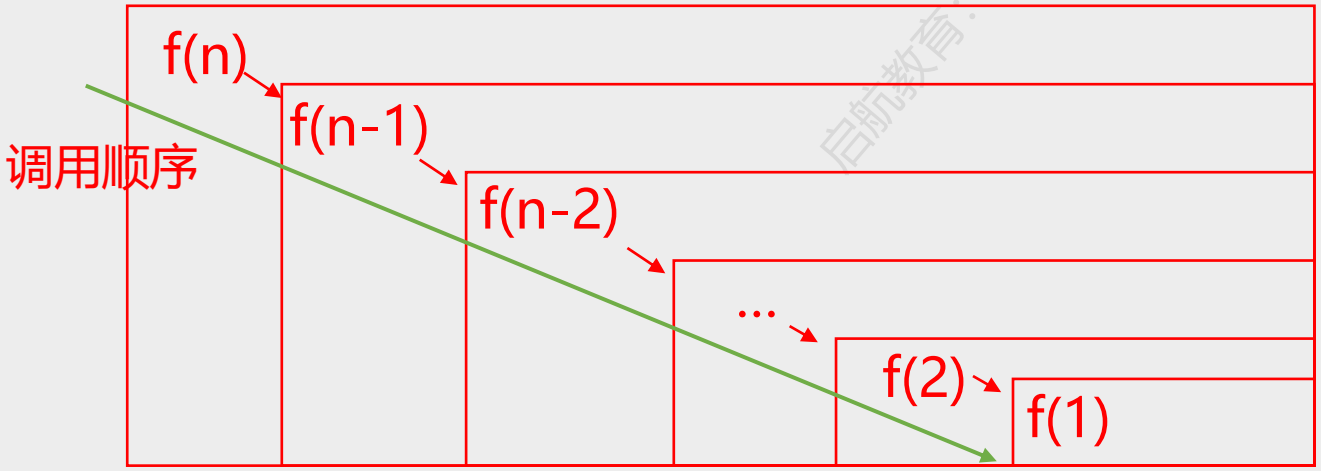
x初值是2，得到 2^{k+2}<n　=> $O(\log_2 n)$

【政哥点拨】

2、求整数n(n≥0)阶乘的算法如下，其时间复杂度是_____。

```
int fact(int n) {
  if (n<=1) return 1;
  return n*fact(n-1);
}
```

A. $O(\log_2 n)$　　B. $O(n)$　　C. $(n\log_2 n)$　　D. $O(n^2)$



n $\xrightarrow{n=n-1}$ 1；共执行了n次，所以时间复杂度是O(n)

【政哥点拨】

3、下列函数的时间复杂度是_____。

```
int func (int n){
  int i=0, sum=0;
  while(sum< n)  sum  +=  ++ i;
  return i ;
}
```

A. $O(log_2n)$　　B. $O(n^{1/2})$　　C. $O(n)$　　D. $O(nlog_2n)$

B【**解析**】本题中最内层循环语句是“sum+=++i；”。

执行1次，i=0+1；sum=0+1；

执行2次，i=1+1；sum=1+2；

以此类推，执行K次，K=i+1，sum=1+2+...+K，于是sum=(K+1)K/2，得到时间复杂度是$O(n^{1/2})$，即$T(n)=O(n^{1/2})$。

牛刀小试

1. 下面程序段的时间复杂度是（ ）。

```
x=0;
for(i=0; i<n; i++)
    for(j=i; j<n; j++)
        x++;
```

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$

D 【解析】本题考查给定循环语句的时间复杂度分析方法。我们常说的分析算法的时间复杂度，就是分析算法的规模n的函数f(n)。

本题中存在着两层for循环,当i=1时,内层循环执行n次,

当i=2时内层循环执行n-1次……,

由分析可知,总共执行了近$n^2/2$次,

故而时间复杂度为$O(n^2)$。需要注意的是，时间复杂度只考虑最高阶，不用考虑低阶。

牛刀小试

2、下面程序段的时间复杂度是（ ）。

```
j=0;    s=0;
while(s<n) {
    j++;      s=s+j;
}
```

A. $O(\sqrt{n})$　　B. $O(\sqrt{2n})$　　C. O(n)　　D. $O(n^2)$

A【解析】算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数。本题中最内层循环语句是j++;和s = s +j;

执行1次，j = 1, s = 1

执行2次，j = 2, s = 1 + 2

依次类推，

执行K次，j= K，(s = 1 + 2 +... + K = (K+1) * K)/2)，于是 (K+1) * K)/2 < n; 得到时间复杂度是开根号，所以选A。

牛刀小试

3、算法的时间复杂度取决于（ ）

A．问题的规模

B．待处理数据的初态

C．A和B

C【解析】算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数，也就是等价于问题的规模；同时在很多算法中，数据的初始状态也影响算法的时间复杂度。例如大家比较熟悉的冒泡排序，当数据有序时，时间复杂度是O（n），当数据逆序是时间复杂度是O（n^2），因此影响时间复杂度的是数据规模和数据的初始状态。

牛刀小试

4、下面程序段的时间复杂度是()。

```
for(i = 0; i< m; i++ )
for(j = 0;j<n;j++)
    a[i][j]=i* j;
```

A.O(m^2)　　B.O(n^2)　　C.O(m×n)　　D.O(m十n)

C 【解析】算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数。本题中存在着两层for循环, 第一层循环执行m次，第二层循环执行n次，于是执行了m*n次。所以时间复杂度是O(m*n)。

牛刀小试

5、执行下面程序段时,执行S语句的**次数**为()。

```
for( int i= 1;i< = n;i++ )
    for( int j= 1 ; j<= i;j++)
        S++;
```

A. n^2　　B. $n^2/2$　　C. n(n+1)　　D. n(n+1)/2

D【解析】本题中存在着两层for循环,当i=1时,内层循环执行1次,当i=2时内层循环执行2次……,由分析可知,总共执行了(1+2+3+…+n)次,故而执行的次数是（n+1）* n/ 2。需要注意的是，语句的执行次数，而不是时间复杂度。

牛刀小试

6、算法分析的两个主要方面是()。

A．空间复杂性和时间复杂性

B. 正确性和简明性

C .可读性和文档性

D. 数据复杂性和程序复杂性

A【解析】算法分析的主要任务是时间复杂度和空间复杂度分析。

牛刀小试

7、下面算法的时间复杂度是()。

```
int suanfa3 (int n) {
    int i =1,s = 1;
    while(s <n) s + = ++i;
    return i ;
}
```

A. O(n)　　B. $O(2^n)$　　C. $O(\log_2 n)$　　D．$O(\sqrt{n})$

D【解析】算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数。本题中最内层循环语句是s + = ++i;

执行1次，i = 2, s = 1 + 2

执行2次，i = 3, s = 1 + 2 + 3； 依次类推，

执行K次，j= K + 1，(s = 1 + 2 +… + (K+1) = (K+1) * (K+2))/2)，于是 (K+1) * (K+2))/2 < n; 得到时间复杂度是开根号，所以选D。

牛刀小试

8、下面程序段的时间复杂度为（）。

(n >1)

sum =1;

for (i = 0 ; sum <n;i ++)

sum + = 1;

A. O(n)　　B. O(2^n)　　C. O($\log_2 n$)　　D．O($\sqrt{n}$)

A【解析】此题具有极强的迷惑性，根据定义，算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数。本题中最内层循环语句是sum += 1

执行1次，i = 0, sum = 1

执行2次，i = 1, sum = 1 + 1　依次类推，

执行K次，j= K-1，sum= 1+1+1..+1 于是sum < n; 得到时间复杂度是O(n)，所以选A。

牛刀小试

9、下面程序段的时间复杂度为_。

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

A. O(n)　　B. $O(2^n)$　　C. $O(\log_3 n)$　　D．$O(\sqrt{n})$

C【解析】根据定义，算法的时间复杂度的定义是执行时间等价于最内层循环语句的执行次数。本题中最内层循环语句是i = i * 3

执行1次，i = 1, sum = 1* 3

执行2次，i = 2, sum = 1 * 3 * 3

依次类推，

执行K次，j= K，sum= 1*3*3...*3, 一共乘了K个3，　于是sum = 3^K< n; 得到时间复杂度是$O(\log_3 n)$，所以选C。

牛刀小试

10、在下列算法中，“x =x *2”的**执行次数**是()。

```
int suanfa1 (int n){
   int i,j,x = 1;
   for (i = 0 ; i < n; i++)
      for(j = i; j< n; j ++ )
         x = x* 2;
    return x;
}
```

A. n(n+1)/2　　B. $n\log_2 n$　　C. n^2　　D. n(n -1)/2

A【解析】我们常说的分析算法的时间复杂度,就是分析算法的规模n的函数f(n)。本题中存在着两层for循环,当i=0时,内层循环执行n次,当i=1时内层循环执行n-1次……,由分析可知,总共执行了(1+2+3+…+n)次,故而执行的次数是（n+1）* n/ 2）次,故而时间复杂度为O(n^2)。需要注意的是，时间复杂度只考虑最高阶，不用考虑低阶。但是执行次数需要考虑所有的值。

考点六：空间复杂度分析

考点六：空间复杂度分析

空间复杂度：用于量度一个算法运行过程中临时占用的存储空间大小。

一般也作为问题规模n的函数，采用数量级形式描述，记作：

$S(n) = O(g(n))$

- 程序代码
- 执行数据
- **辅助空间/临时变量**

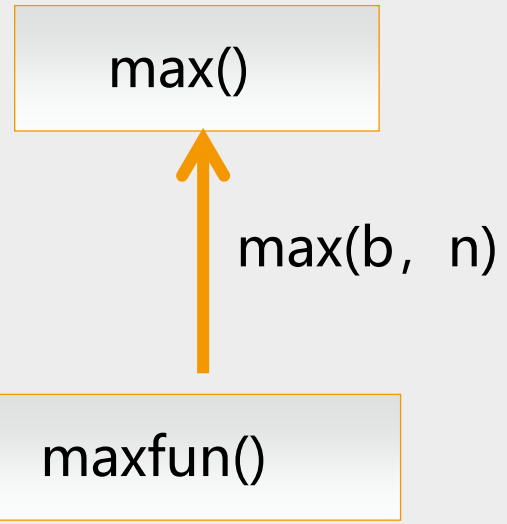
考点六：空间复杂度分析

为什么空间复杂度分析只考虑临时占用的存储空间?

```
int max(int a[]，int n)
{  int i，maxi=0;
   for (i=0;i< n;i++)
        if (a[i]>a[maxi])
            maxi=i;
   return a[maxi];
}
```

如果max函数中再考虑形参*a*的空间，
就重复累计了执行整个算法所需的空间。

max算法的空间复杂度为O(1)




```
void maxfun()
{  int b[]={1，2，3，4，5}，n=5;
      printf("Max=%d\n"，max(b，n));
}
```

maxfun算法中为*b*数组分配了相应的内存空间，其空间复杂度为O(*n*)

考点六：空间复杂度分析

```
int fun(int n)
{  int i，j，k，s;
   s=0;
   for (i=0;i< n;i++)
      for(j=0;j<=i;j++)
        for (k=0;k<=j;k++)
            s++;
   return(s);
}
```

临时占用的存储空间：函数体内分配的空间

算法中临时分配的变量个数与问题规模*n*无关，所以空间复杂度均为O(1)。

若一个算法的空间复杂度为O(1)，则称此算法为原地工作或就地工作算法。

考点六：空间复杂度分析

求整数n(n≥0)阶乘的算法如下，其空间复杂度是（ ）。

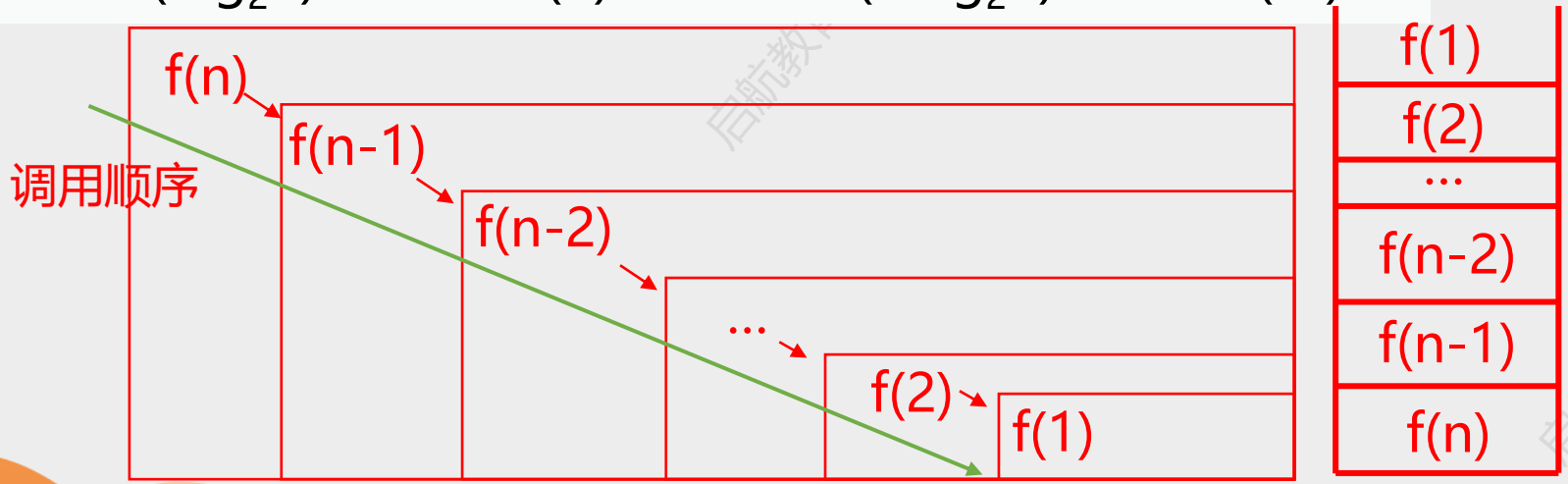
```
int fact(int n) {
    if (n<=1)    return 1;
    return   n * fact(n-1);
}
```

①

③

递归三要素
- ①递归初始条件
- ②递归终止条件
- ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$



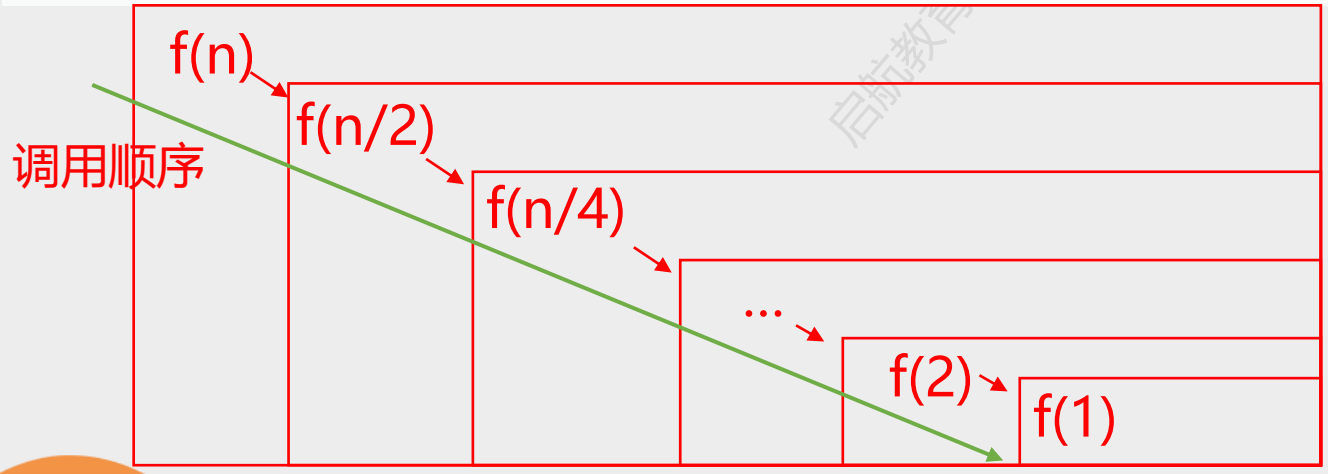
考点六：空间复杂度分析

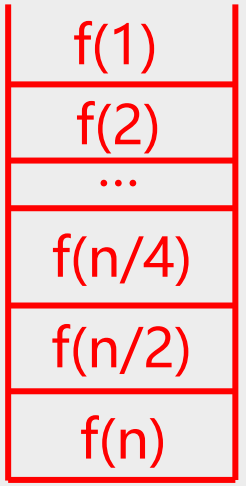
给定下面算法如下，其空间复杂度是（ ）。

```
①
int fact(int n) {
    if (n<=1)②  return 1;
    return   1 +  fact(n/2);
                  ③
}
```

递归三要素：①递归初始条件 ②递归终止条件 ③递归转移条件

A. $O(\log_2 n)$　　B. $O(n)$　　C. $O(n\log_2 n)$　　D. $O(n^2)$





【政哥点拨】

2、下列函数的空间复杂度是_____。

```
int func (int n){
    int i=0, sum=0;
    while(sum< n) sum  +=  ++ i;
    return i ;
}
```

A. $O(\log_2 n)$　　B. $O(n^{1/2})$　　C. $O(n)$　　D. $O(1)$

D **【解析】** 空间复杂度是算法运行过程中所需临时变量的存储空间大小，此题中只有i、sum、n三个临时变量，空间复杂度是O(1)。

【政哥点拨】

3、下面程序段的空间复杂度是()。

```
for(i = 0; i< m; i++ )
    for(j = 0;j<n;j++)
            a[i][j]=i* j;
```

A.$O(m^2)$　　B.$O(n^2)$　　C.$O(m\times n)$　　D.$O(m+n)$

C 【解析】 空间复杂度是算法运行过程中所需临时变量的存储空间大小，此题中只有i、j、m、n四个临时变量，以及a[m][n]数组，空间复杂度是$O(m\times n)$。

牛刀小试

1．某算法的空间复杂度为O(1)，则(　　)。

A．该算法执行不需要任何辅助空间

B．该算法执行所需辅助空间大小与问题规模n无关

C．该算法执行不需要任何空间

D．该算法执行所需空间大小与问题规模n无关

B 【**解析**】算法的空间复杂度的定义是执行时所需要的辅助空间的量度，而空间复杂度为O(1)表示的是该算法执行所需辅助空间大小与问题规模n无关，是个常数(就地空间)，所以选择B项。注意选项D和选项B的差别。

牛刀小试

2．下面说法错误的是(　)。

①算法就地空间的含义是指不需要任何额外的辅助空间；

②在相同的规模n下，复杂度O(n)的算法在时间上总是优于复杂度O(2n)的算法；

③所谓时间复杂度，是指最坏情况下，估算算法执行时间的一个上界；

④同一个算法，实现语言的级别越高，执行效率就越高。

A．①　　B．①②　　C．①④　　D．③

C 【**解析**】算法的空间复杂度的定义是执行时所需要的辅助空间的量度。就地空间指的是执行所需辅助空间大小与问题规模n无关，是个常数，所以①错误；时间复杂度的计量只考虑最高阶，所以②正确；③是时间复杂度的定义，正确；同一个算法，实现语言级别越高，往往效率和性能越低，如机器语言通常快于C语言，C语言通常比Java快，所以④错误。综上，选择C项。

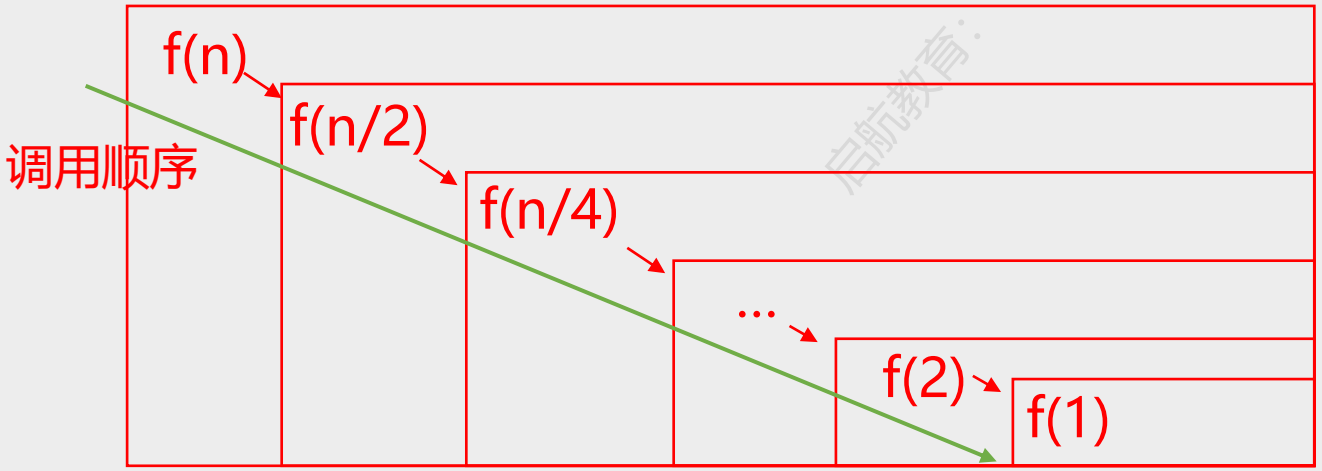
谢谢大家

真题实战

1、【重庆邮电大学 2021】设N是描述问题规模的非负整数，下列程序段的时间复杂度是()。

```
static int fun(int n) {
  if (n == 1) return 0;
  return 1 + fun(n /2);
}
```

A、$O(\log_2 n)$　　B、O (n)　　C、$O (n\log_2 n)$　　D、$O (n^{1/2})$



n $\xrightarrow{n=n/2}$ 1 共执行了$\log_2 n$次，所以时间复杂度是$O(\log_2 n)$

2、【华东师范大学　2013】如果do(IT)这个算法的复杂度为n，n为描述问题规模的非负整数(n≥0)，下面程序片段的时间复杂度是(　　)。

```
int x=2;
while(x≤n){
    do(IT);
    x=x*2;
}
```

A．O(n)　　B．O ($\log_2 n$)　　C．O($n\log_2 n$)　　D．O(n^2)

C 【解析】本题中最内层循环语句是x=2*x和do(IT)；并且二者的执行频度一样，我们假设一共执行执行K次，

执行1次，x = x * 2

执行2次，x = x * 2 * 2；依次类推，

执行K次，j= K，x = x * 2 * 2 *…* 2,一共乘了K个2，于是sum = 2^K < n; 得到x=2*x和do(IT)的执行频度是$\log_2 n$，而do(IT)；的时间复杂度是n，根据乘法原理，得到时间复杂度是O($n\log_2 n$) 。

我们将do(IT)看成黑匣子，不用关心他的执行细节。

3、已知两个长度分别为 m 和 n 的升序链表，若将它们合并为一个长度为 m+n 的降序链表，则最坏情况下的时间复杂度是_____。

A. O(n)　　B. O(m*n)　　C. O(min(m, n))　　D. O(max(m, n))

D【解析】m、n 是两个升序链表，长度分别为 m 和 n。两个升序链表合并，两两比较表中元素，每比较一次确定一个元素的位置。当一个链表比较结束后，将另一个链表的剩余元素插入即可。考虑两种极端情况：

（1）最好的情况：其中一个表的元素的值比另外一个的表的元素都大，此时比较次数最小，等于元素比较小的表的长度；

（2）最坏的情况：两个链表中的元素依次进行比较，比较的次数最多是两个表的长度和。最坏的情况是两个链表中的元素依次进行比较，直到两个链表都到表尾，即每个元素都经过比较。

但是不管哪种情况，其时间复杂度为 O(m + n) ≅ O(max(m, n))。

4、下列程序段的时间复杂度是_____。

```
count=0;
for(k=1;k<=n;k*=2)
  for(j=1;j<=n;j++)
    count++;
```

A．$O(\log_2 n)$　　B．$O(n)$　　C．$O(n\log_2 n)$　　D．$O(n^2)$

C【**解析**】外层循环条件为 k<=n，增量定义为 k*=2，可知循环次数为 $2^k<=n$，即 $k<=\log_2 n$。内层循环条件j<=n 与外层循环的变量无关，每次循环j 自增 1，每次内层循环都执行 n 次。对于嵌套循环，该段程序的时间复杂度 T(n) = $T_1(n)T_2(n) = O(n)O(\log_2 n) = O(n\log_2 n)$，选 C。

5、【哈尔滨工业大学 2017】设n是描述问题规模的非负整数，下面程序片段的时间复杂度是()。

```
int x=n*n;
while(x>=1){
    x=x/2;
}
```

A、O (log_2n)　　B、O (n)　　C、O ($nlog_2n$)　　D、O ($n^{1/2}$)

B 【解析】 本算法的最内层语句是：x=x/2；每执行一次，n除以2，假设执行的次数是K次， 则 $n^2/2^K=1$， 所以$K=2*log_2n$，故时间复杂度为 $O(log_2n)$。

6、设 n 是描述问题规模的非负整数，下列程序段的时间复杂度是（）

```
x=0;
while(n>=(x+1)*（x+1))
   x=x+1;
```

A. O(log n)　　B. $O(n^{1/2})$　　C. O(n)　　D. $O(n^2)$

B【解析】本算法的最内层循环语句是：x=x+1；假设第k次循环终止，则第k次执行时，$(x+k)^2 > n$,x的初始值为0, 第k次判断时，x=k,即$k^2>n$, $k>\sqrt{n}$, 因此该程序段的时间复杂度为$O(\sqrt{n})$。因此选B。

谢谢大家