前言　学习过程中的常见问题

问题 1：公共基础知识的考试情况是怎样的?

公共基础知识的考试方式为笔试，与 C 语言程序设计(C++语言程序设计、Java 语言程序设计、Visual Basic 语言程序设计、Visual Foxpro 数据库程序设计、Access 数据库程序设计和 Delphi 语言程序设计)的笔试部分合为一张试卷。

问题 2：公共基础知识的考试大纲

基本要求

1. 掌握算法的基本概念。
2. 掌握基本数据结构及其操作。
3. 掌握基本排序和查找算法。
4. 掌握逐步求精的结构化程序设计方法。
5. 掌握软件工程的基本方法，具有初步应用相关技术进行软件开发的能力。
6. 掌握数据库的基本知识，了解关系数据库的设计。

考试内容

一、基本数据结构与算法

1. 算法的基本概念；算法复杂度的概念和意义（时间复杂度与空间复杂度）。
2. 数据结构的定义；数据的逻辑结构与存储结构；数据结构的图形表示；线性结构与非线性结构的概念。
3. 线性表的定义；线性表的顺序存储结构及其插入与删除运算。
4. 栈和队列的定义；栈和队列的顺序存储结构及其基本运算。
5. 线性单链表、双向链表与循环链表的结构及其基本运算。
6. 树的基本概念；二叉树的定义及其存储结构；二叉树的前序、中序和后序遍历。
7. 顺序查找与二分法查找算法；基本排序算法（交换类排序，选择类排序，插入类排序）。

二、程序设计基础

1.程序设计方法与风格。
2.结构化程序设计。
3.面向对象的程序设计方法，对象，方法，属性及继承与多态性。

三、软件工程基础

1. 软件工程基本概念，软件生命周期概念，软件工具与软件开发环境。
2. 结构化分析方法，数据流图，数据字典，软件需求规格说明书。

3. 结构化设计方法，总体设计与详细设计。
4. 软件测试的方法，白盒测试与黑盒测试，测试用例设计，软件测试的实施，单元测试、集成测试和系统测试。
5. 程序的调试，静态调试与动态调试。

四、数据库设计基础

1. 数据库的基本概念：数据库，数据库管理系统，数据库系统。
2. 数据模型，实体联系模型及 E-R 图，从E-R 图导出关系数据模型。
3. 关系代数运算，包括集合运算及选择、投影、连接运算，数据库规范化理论。
4. 数据库设计方法和步骤：需求分析、概念设计、逻辑设计和物理设计的相关策略。

第 1 章 算法与数据结构

考点 1：算法、时间复杂度和空间复杂度

算法

算法，是指解题方案的准确而完整的描述。对于一个问题，如果可以通过一个计算机程序，在有限的存储空间内运行有限长的时间而得到正确的结果，则称这个问题是算法可解的。但算法不等于程序，也不等计算机方法。程序也可以作为算法的一种描述，但程序通常还要考虑程序运行时的环境限制等。程序的编制不可能优于算法的设计。

算法的基本特征

(1) 可行性：算法总是在某个特定的计算工具上执行的，因此算法在执行过程中常常要受到计算机工具的限制，使执行结果产生偏差。算清和数学计算公式是有差别的，在设计一个算法时，必须考虑到它的可行性，否则是不会得到满意结果的。

(2) 确定性：算法中每一步骤都必须有明确定义，不允许有模棱两可的解释，不允许有多义性；例如在某些特殊情况时，数学公式是正确的，但计算机就是无法操作。这一性质也反映了算法与数学公式的明显差别。

(3) 有穷性：是指算法必须能在有限的时间内做完，即算法必须能在执行有限个步骤后终止，包括合理的执行时间的含义。例如无理数 $\frac{1}{3}=0.33333...$ 的问题。

(4) 拥有足够的情报。算法执行的结果总是与输入的初始数据有关，不同的输入将会有不同的结果输出。当输入不够或输入错误时，算法本身也就无法执行或导致执行有错。因此在设计算法时要考虑到所有的各种可能情况。

总之，算法是一组严谨地定义运算顺序的规则，并且每一个规则都是有效的，是明确的，此顺序将在有限的次数下终止。

一个算法通常由两种基本要素组成，一是对数据对象的运算和操作:

(1) 算术运算：加、减、乘、除等；

(2) 逻辑运算：与、或、非等；

(3) 关系运算：大于、小于、等于、不等于等；

(4) 数据传输：赋值、输入和输出等。

二是算法的控制结构，一个算法一般都可以用顺序、选择和循环三种基本控制结构组合而成。

三是算法设计的基本方法，包括：列举法、归纳法、递推、递归、减半递推技术和回溯法。

算法的复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。一个算法的优劣将影响到算法乃至程序的效率。算法分析的目的在于选择合适算法和改进算法。

(1) 算法的时间复杂度：是指执行算法所需要的计算工作量，可以执行算法的过程中所需要的基本运算的执行次数来度量。分析算法工作量的方法有：平均性态、最坏情况复杂性。

(2) 算法的空间复杂度：是指执行这个算法所需要的内存空间。主要包括：算法程序所占的空间、输入的初始数据所占的空间、算法执行过程中所需要的额外空间。

考点精选题目

1. 下列叙述中正确的是_____。

A) 算法就是程序

B) 设计算法时只需要考虑数据结构的设计

C) 设计算法时只需要考虑结果的可靠性

D) 以上三种说法都不对

2. 算法的时间复杂度是指_____。

A) 算法的执行时间

B) 算法所处理的数据量

C) 算法程序中的语句或指令条数

D) 算法在执行过程中所需要的基本运算次数

3. 算法的空间复杂度是指_____。

A) 算法在执行过程中所需要的计算机存储空间

B) 算法所处理的数据量

C) 算法程序中的语句或指令条数

D) 算法在执行过程中所需要的临时工作单元数

考点 2：数据结构、线性结构和非线性结构

数据结构

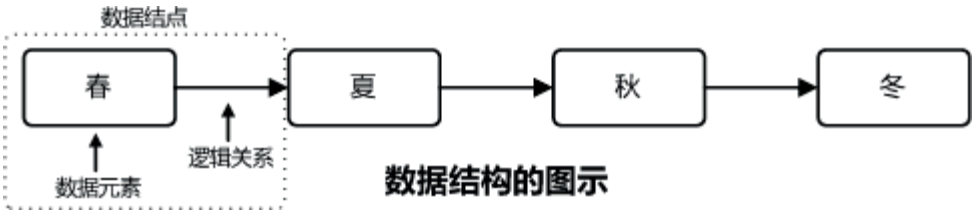
利用计算机进行数据处理是计算机应用的一个重要领域。大量的数据元素在计算机中如何组织，以便提高数据处理的效率，并且节省计算机的存储空间，这是进行数据处理的关键问题。数据结构作为一门计算机学科，主要研究以下三个方面的问题：

(1) 数据集合中各数据元素之间所固有的逻辑关系，即数据的逻辑结构；

(2) 在对数据进行处理时，各数据元素在计算机中的存储关系，即数据的存储结构；

(3) 对各种数据结构进行的运算。

研究以上问题的主要目的是为了提高数据处理的效率（一是提高数据处理的速度，二是尽是节省数据处理过程所占的存储空间。）数据结构示例的如下 图 2-1 所示：



数据结构的图示

在数据处理领域中，每一个需要处理的对象都可以抽象成数据元素，简称为元素。一般情况下，在具有相同特征的数据元素集合中，各个数据元素之间存在有某种关系，这种关系可以用前后件关系来描述。

数据的逻辑结构

数据的逻辑结构：是指反映数据元素之间逻辑关系的数据结构。数据结构是指反映数据元素之间关系的数据元素集合的表示，其实质是带有前后件关系的数据元素的集合。因此，数据结构包含：

(1) 表示数据元素的信息；

(2) 表示各数据元素之间的前后件关系。

数据的存储结构

数据的存储结构：是指数据的逻辑结构在计算机存储空间中的存放形式，也称数据的物理结构。在数据处理时，被处理的各种数据元素总是被存放在计算机的存储空间中，其存放的位置关系与它们的逻辑关系不一定是相同的，而且一般也不可能相同。一般来说，一种数据逻辑结构根据需要可以表示成多种存储结构，常用的数据的存储结构有顺序、链接、索引等。而采用不同的存储结构，其数据处理效率是不同的。

线性结构与非线性结构

如果在一个数据结构中一个数据元素都没有，则称为空的数据结构。根据数据结构中各数据元素之间的前后件关系的复杂程度，分为线性结构与非线性结构。

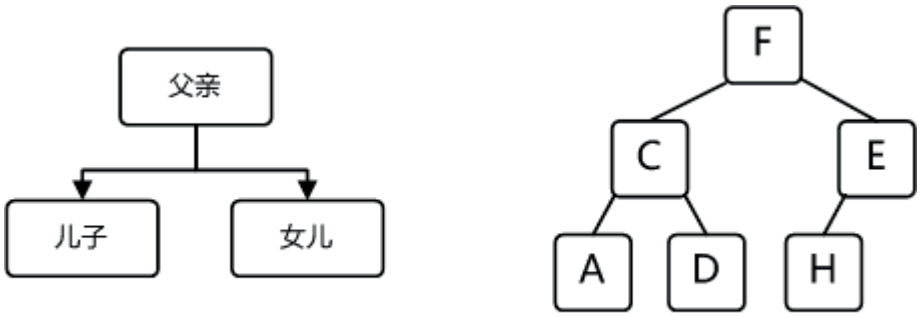
如果一个非空的数据结构满足下列两个条件：

(1) 有且只有一个根结点；
(2) 每一个结点最多有一个前件，也最多有一个后件；

则称该数据结构为线性结构。线性表是一个典型的线性结构。常见的有：线性表，栈，队列，循环队列和线性链表等。线性结构如下 **图 2-2** 所示：

线性结构

不满足线性结构条件的数据结构，就是非线性结构。常见的有：数组、广义表、树、二叉树和图都是非线性结构。非线性结构如下 **图 2-3** 所示：



非线性结构

线性结构和非线性结构都可以是空的数据结构。空的数据结构空间是属于线性或非线性结构，视具体情况决定。如果是按线性结构规则来处理就是线性结构，否则就是非线性结构。

考点精选题目

1. 数据结构分为线性结构与非线性结构，带链的栈属于_____。

2. 下列叙述中正确的是_____。

A) 有一个以上根节点的数据结构不一定是非线性结构

B) 只有一个根节点的数据结构不一定是线性结构

C) 循环链表是非线性结构

D) 双向链表是非线性结构

3. 下列数据结构中，属于非线性结构的是_____。

A) 循环队列

B) 带链队列

C) 二叉树

D) 带链栈

4. 下列叙述正确的是_____。

A) 程序执行的效率与数据的存储结构密切相关

B) 程序执行的效率只取决于程序的控制结构

C) 程序执行的效率只取决于所处理的数据量

D) 以上三种说法都不对

5. 下列叙述中正确的是_____。

A) 数据的逻辑结构与存储结构必定是一一对应的

B) 由于计算机存储空间是向量式的存储结构，因此，数据的存储结构一定是线性结构

C) 程序设计语言中的数组一般是顺序存储结构，因此，利用数组只能处理线性结构

D) 三种说法都不对

6. 下列叙述中正确的是_____。

A) 一个逻辑数据结构只能有一种存储结构

B) 数据的逻辑结构属于线性结构，存储结构属于非线性结构

C) 一个逻辑数据结构可以有多种存储结构，且各种存储结构不影响数据处理的效率

D) 一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率

考点 3：线性表、线性链表、双向链表与循环链表

线性表

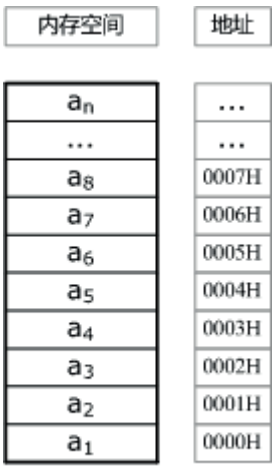
在计算机内，线性表的存储结构有两种：顺序存储（称为线性表）和链式存储（线性链表）。线性表的链式存储结构所需要的存储空间一般要多于顺序存储结构。线性表由一组数据元素构成，数据元素的位置只取决于自己的序号，元素之间的相对位置是线性的。在复杂线性表中，由若干项数据元素组成的数据元素称为记录，而由多个记录构成的线性表又称为文件。非空线性表的结构特征：

(1) 且只有一个根结点a_1，它无前件；

(2) 有且只有一个终端结点a_n，它无后件；

(3) 除根结点与终端结点外，其他所有结点有且只有一个前件，也有且只有一个后件。

线性表中结点的个数 n 称为线性表的长度，当 n=0 时，称为空表。线性表如下 **图 2-4** 所示：



线性表

线性表的顺序存储结构，具有下面两个基本特点：

(1) 线性表中所有元素所占的存储空间是连续的；

(2) 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

第 i 个元素 a_i 在计算机存储空间中的存储地址为：

$$ADR(a_i) = ADR(a_1) + (i - 1) * k$$

链式存储

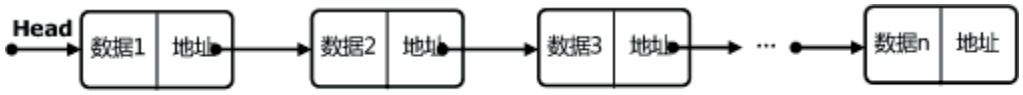
对于小的线性表或长度固定的线性表，采用顺序存储结构。在程序设计语言中，通常定义一个一维数组来表示线性表的存储空间。对于大的线性表，特别是是元素变化频繁的线性表不宜采用顺序存储结构，要用链式存储结构。

链式存储结构中，每个数据存储在一个存储单元中，这个存储单元称为结点。在链式存储方式中，要求每个结点由两部分组成：一部分用于存放数据元素值，称为数据域；另一部分

用于存放指针，称为指针域。其中指针用于指向该结点的前一个或后一个结点(即前件或后件)。在链式存储结构中，存储数据结构的存储空间可以不连续，各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据元素之间的逻辑关系是由指针域来确定的。链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

线性链表

线性链表：是指线性表的链式存储结构。线性链表如下 **图 2-5** 所示：



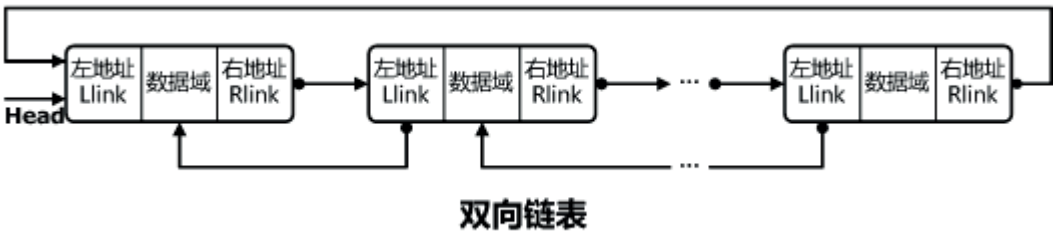
线性链表

一般来说，在线性链表中，各数据结点的存储序号是不连续的，并且各结点在存储空间中的位置关系与逻辑关系也不一致。各数据元素之间的前后件关系是由各结点的指针域来指示，指向线性链表中第一个结点的指针 HEAD 称为头指针，HEAD=NULL(或 0)称为空表。

线性链表的基本运算：查找、插入、删除。

双向链表

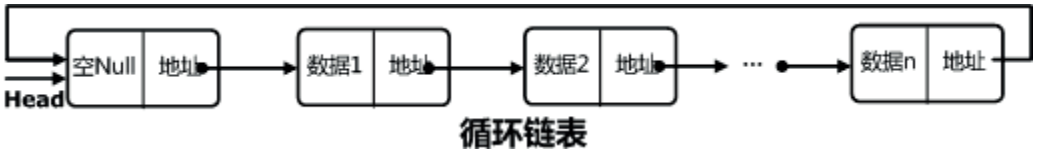
双向链表：在每一个结点中包含有两个指针，分别是左指针(Llink 指向前件结点)和右指针(Rlink 指向后件结点)。双向链表如下 **图 2-6** 所示：



双向链表

循环链表

循环链表：为了克服单向链表在对空表和第一个结点处理时必须单独考虑，使空表和非空表的运算不统一的问题，可以采用循环链表的结构。循环链表如下 **图 2-7** 所示：



循环链表

循环链表有以下两个特点： (1)增加了一个表头结点，其数据域为空，指针域指向第一个元

素的结点。循环链表的头指针 HEAD 指向表头结点。 (2)最后一个结点的指针域不是空，而是指向表头结点。

由于在循环链表中设置了一个表头结点，因此在任何情况下，循环链表中至少有一个结点存在，从而使空表和非空表的运算统一。

考点精选题目

1. 下列关于线性链表的叙述中，正确的是_____。

   A) 各数据结点的存储空间可以不连续，但它们的存储顺序与逻辑顺序必须一致

   B) 各数据结点的存储顺序与逻辑顺序可以不一致，但它们的存储空间必须连续

   C) 进行插入与删除时，不需要移动表中的元素

   D) 选项 D

2. 下列叙述中正确的是_____。

   A) 线性表的链式存储结构与顺序存储结构所需要的存储空间是相同的

   B) 线性表的链式存储结构所需要的存储空间一般要多于顺序存储结构

   C) 线性表的链式存储结构所需要的存储空间一般要少于顺序存储结构

   D) 上述三种说法都不对

3. 下列叙述中正确的是_____。

   A) 顺序存储结构的存储一定是连续的，链式存储结构的存储空间不一定是连续的

   B) 顺序存储结构只针对线性结构，链式存储结构只针对非线性结构

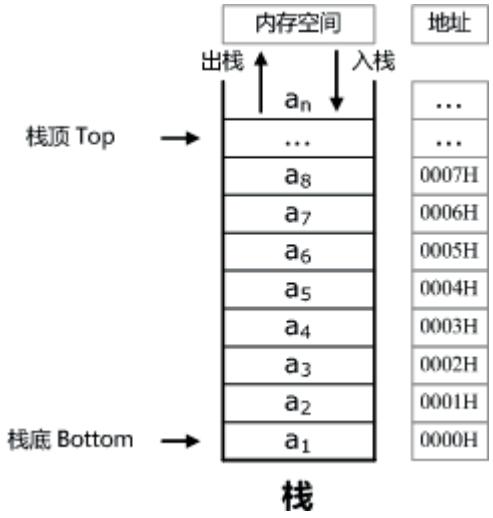
   C) 顺序存储结构能存储有序表，链式存储结构不能存储有序表

   D) 链式存储结构比顺序存储结构节省存储空间

考点 4：栈、队列和循环队列

栈

栈（Stack）又称堆栈。栈如下 **图 2-8** 所示：



(1) 栈是一种运算受限的线性表，其限制是仅允许在表的一端进行插入和删除运算。允许插入和删除的一端称为栈顶，不允许插入和删除的一端称为栈底。

(2) 由于栈的插入和删除运算仅在栈顶一端进行，后进栈的元素必定先出栈，所以又把栈称为后进先出表（Last In First Out，简称 LIFO）；先进栈的元素必定后出栈，所以又把栈称为先进后出表（First In Last Out，简称 FILO）。

(3) 栈的基本运算

①入栈：是指在栈顶位置插入一个新元素。操作步骤：首先将栈顶指针 Top 加 1，然后将新元素插入到栈顶指针指向的位置。

②退栈：是指取出栈顶元素并赋值给一个指定的变量。操作步骤：首先将栈顶元素赋值一个指定的变量，然后将栈顶指针 Top 减 1。

③读栈顶元素：是指取出栈顶元素并赋值给一个指定的变量。这个操作不删除栈顶元素。

队列

队列（Queue）简称队。队列如下 **图 2-9** 所示：

(1)队列是一种运算受限的线性表，其限制是仅允许在表的一端进行插入操作，而在表的另一端进行删除操作。我们把允许插入的一端称作队尾（rear），允许删除的一端称作队首（front）。