CSP-J

初赛知识点汇编

目录

目录

第一章　计算机基础.....2
第 1 节　计算机常识.....2
第 2 节　计算机的基本构成.....4
第 3 节　计算机软件.....7
第 4 节　计算机网络和 Internet 的基本概念.....9
第 5 节　程序设计语言以及程序编译和运行的基本概念.....15
第 6 节　进制的基本概念与进制转换、字节与字.....16
第二章　算法.....23
第 1 节　算法的基本概念.....23
第 2 节　入门组基础算法理论.....24
第三章　数据结构.....28
第 1 节　线性数据结构.....28
第 2 节　非线性数据结构.....35
第四章　组合数学.....43

第一章 计算机基础

第1节 计算机常识

一、发展史

1．计算机发展代别划分：

代别	年代	逻辑（电子）元件
第一代	1946—1958	电子管
第二代	1959—1964	晶体管
第三代	1965—1970	集成电路
第四代	1971—至今	大规模、超大规模集成电路

2．第一台电子计算机

1946 年 2 月，在美国宾夕法尼亚大学诞生了世界上第一台电子计算机 ENIAC（Electronic Numerical Integrator and Computer），这台计算机占地 170 平方米，质量 30 吨，用了 18000 多个电子管，每秒能进行 5000 次加法运算。

3．冯·诺依曼理论

1944 年，美籍匈牙利数学家冯·诺依曼提出计算机基本结构和工作方式的设想，为计算机的诞生和发展提供了理论基础。时至今日，尽管计算机软硬件技术飞速发展，但计算机本身的体系结构并没有明显的突破，当今的计算机仍属于冯·诺依曼架构。

其理论要点如下：

计算机硬件设备由存储器、运算器、控制器、输入设备和输出设备 5 部分组成。

存储程序思想——把计算过程描述为由许多命令按一定顺序组成的程序，然后把程序和数据一起输入计算机，计算机对已存的程序和数据处理后，输出结果。

二、计算机的分类

根据计算机的性能指标，如机器规模的大小、运算速度的高低、主存储容量的大小、指令系统性能的强弱以及机器的价格等，可将计算机分为巨型机、大型机、中型机、小型机、微型机和工作站。

巨型机：具有很强的计算和处理数据的能力，主要特点表现为高速度和大容量，配有多种外部和外围设备及丰富的、高功能的软件系统。主要用来承担重大的科学研究、国防尖端技术和国民经济领域的大型计算课题及数据处理任务。如大范围天气预报，整理卫星照片，原子核物的探索，研究洲际导弹、宇宙飞船等。“天河一号”为我国首台千万亿次超级计算机。2010 年 9 月开始进行系统调试与测试，并分步提交用户使用。

大、中型机：大型机使用专用的处理器指令集、操作系统和应用软件，大量使用冗余等技术确保其安全性及稳定性，擅长非数值计算（数据处理），主要用于商业领域，如银行和电信。

小型机是指采用精简指令集处理器，性能和价格介于 PC 服务器和大型主机之间的一种高性能 64 位计算机。

小型机与普通服务器相比具有：

（1）高可靠性（Reliability）：计算机能够持续运转，从来不停机。

（2）可用性（Availability）：重要资源都有备份；能够检测到潜在要发生的问题，并且能够转移其上正在运行的任务到其他资源，以减少停机时间，保持生产的持续运转；具有实时在线维护和延迟性维护功能。

（3）高服务性（Serviceability）：能够实时在线诊断，精确定位根本问题所在，做到准确无误的快速修复。

微型机：通常作为个人计算机，由硬件系统和软件系统组成，是一种能独立运行，完成特定功能的设备。个人计算机不需要共享其他计算机的处理、磁盘和打印机等资源也可以独立工作。从台式机（或称台式计算机、桌面电脑）、笔记本电脑到上网本和平板电脑以及超级本等都属于个人计算机的范畴。

工作站是一种高端的通用微型计算机。它是为了单用户使用并提供比个人计算机更强大的性能，尤其是在图形处理能力，任务并行方面的能力。通常配有高分辨率的大屏、多屏显示器及容量很大的内存储器和外部存储器，并且具有极强的信息和高性能的图形、图像处理功能的计算机。另外，连接到服务器的终端机也可称为工作站。

三、计算机的应用

计算机的快速性、通用性、准确性和逻辑性等特点，使它不仅具有高速运算能力，而且还具有逻辑分析和逻辑判断能力。如今，计算机已渗透到人们生活和工作的各个层面中，主要体现在以下几个方面的运用。

1．科学计算

科学计算（或数值计算）是指利用计算机来完成科学研究和工程技术中提出的数学问题的计算。在现代科学技术工作中，科学计算问题是大量的和复杂的。利用计算机的高速计算、大存储容量和连续运算的能力，可以实现人工无法解决的各种科学计算问题。

2．信息处理

信息处理（数据处理）是指对各种数据进行收集、存储、整理、分类、统计、加工、利用、传播等一系列活动的统称。据统计，80%以上的计算机主要用于数据处理，这类工作量大、面宽，决定了计算机应用的主导方向。

3．自动控制

自动控制（过程控制）是利用计算机及时采集检测数据，按最优值迅速地对控制对象进行自动调节或自动控制。采用计算机进行自动控制，不仅可以大大提高控制的自动化水平，而且可以提高控制的及时性和准确性，提高产品质量及合格率。目前，计算机过程控制已在机械、冶金、石油、化工、纺织、水电、航天等部门得到了广泛的应用。

4．计算机辅助技术

计算机辅助技术是指利用计算机帮助人们进行各种设计、处理等过程，它包括计算机辅助设计（CAD）、计算机辅助制造（CAM）、计算机辅助教学（CAI）和计算机辅助测试（CAT）等。另外，计算机辅助技术还有辅助生产、辅助绘图和辅助排版等。

5．人工智能

人工智能（Artificial Intelligence，AI）又可称为智能模拟，是计算机模拟人类的智能活动，诸如感知、判断、理解、学习、问题求解和图像识别等。人工智能的研究目标是使计算机更好地模拟人的思维活动，那时的计算机将可以完成更复杂的控制任务。

6．网络应用

随着社会信息化的发展，通信业也发展迅速，计算机在通信领域的作用越来越大，特别是促进了计算机网络的迅速发展。目前，全球最大的网络（Internet，即国际互联网）已把全球的大多数计算机联系在一起。除此之外，计算机在信息高速公路、电子商务、娱乐和游戏等领域也得到了快速的发展。

第2节　计算机的基本构成

计算机系统由硬件和软件两部分组成。硬件系统是计算机的“躯干”，是物质基础。而软件系统则是建立在这个“躯干”上的“灵魂”。

- 计算机
  - 硬件系统
    - 主机
      - 中央处理器CPU
        - 控制器
        - 运算器
      - 内存
        - ROM(只读存储器)
        - RAM(读写存储器)
        - Cache(高速缓冲存储器)
    - 外设
      - 输入设备——键盘、鼠标、扫描仪、麦克风等
      - 输出设备——显示器、打印机、音响等
      - 外存——硬盘、软盘、光盘、闪存(优盘)等
  - 软件系统
    - 系统软件
      - 操作系统
      - 编译程序和解释程序
      - 数据库管理系统
    - 应用软件
      - 各种字处理系统
      - 各种软件包

一、计算机硬件

计算机硬件由五大部分组成：运算器、控制器、存储器、输入设备、输出设备。

1．中央处理器（CPU——Central Processing Unit）

由运算器、控制器和一些寄存器组成；

运算器进行各种算术运算和逻辑运算；

控制器是计算机的指挥系统；

CPU 的主要性能指标是主频和字长。

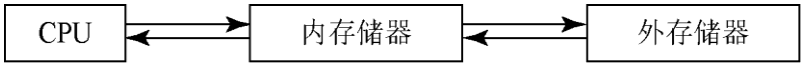
2．存储器

存储器的主要功能是用来保存各类程序的数据信息。

存储器可分为主存储器和辅助存储器两类。

①主存储器（也称为内存储器），属于主机的一部分。用于存放系统当前正在执行的数据和程序，属于临时存储器。

②辅助存储器（也称外存储器），属于外部设备。用于存放暂不用的数据和程序，属于永久存储器。存储器与 CPU 的关系表示：



（1）内存储器

内存又称为主存，它和 CPU 一起构成了计算机的主机部分，它存储的信息可以被 CPU

直接访问。内存由半导体存储器组成，存取速度较快，但一般容量较小。内存中含有很多的存储单元，每个单元可以存放 1 个 8 位的二进制数，即 1 个字节（Byte，简称“B”）。内存中的每个字节各有一个固定的编号，这个编号称为地址。CPU 在存取存储器中的数据时是按地址进行的。所谓存储器容量，即指存储器中所包含的字节数，通常用 KB、MB、GB、TB 和 PB 作为存储器容量单位。它们之间的关系为：

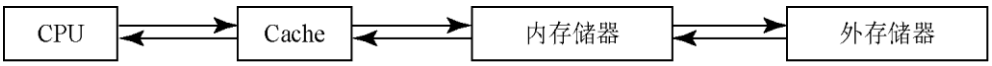
1 KB＝1024 B　1 MB＝1024 KB　1 GB＝1024 MB　1 TB＝1024 GB　1 PB＝1024 TB

内存储器通常可以分为随机存储器 RAM、只读存储器 ROM 和高速缓冲存储器 Cache 三种。

①RAM 是一种读写存储器，其内容可以随时根据需要读出，也可以随时重新写入新的信息。当电源电压去掉时，RAM 中保存的信息都将全部丢失。

②ROM 是一种内容只能读出而不能写入和修改的存储器，其存储的信息是在制作该存储器时就被写入的。在计算机运行过程中，ROM 中的信息只能被读出，而不能写入新的内容。计算机断电后，ROM 中的信息不会丢失。它主要用于检查计算机系统的配置情况并提供最基本的输入/输出（I/O）控制程序。

③由于 CPU 速度的不断提高，RAM 的速度很难满足高速 CPU 的要求，所以，在读/写系统内存时都要加入等待的时间，这对高速 CPU 来说是一种极大的浪费。Cache 是指在 CPU 与内存之间设置的一级或两级高速小容量存储器，称之为高速缓冲存储器，固化在主板上。在计算机工作时，系统先将数据由外存读入 RAM 中，再由 RAM 读入 Cache 中，然后 CPU 直接从 Cache 中取数据进行操作。



（2）外存储器

外存储器又称为辅助存储器，它的容量一般都比较大，而且大部分可以移动，便于在不同计算机之间进行信息交流。在微型计算机中，常用的外存有软盘、硬盘、闪存和光盘 4 种。

①软盘存储器

软盘存储器由软盘、软盘驱动器和软盘适配器三部分组成。软盘是活动的存储介质，软盘驱动器是读写装置，软盘适配器是软盘驱动器与主机连接的接口。软盘驱动器安装在主机箱内，软盘驱动器插槽暴露在主机箱的前面板上，可方便地插入或取出软盘。

②硬盘存储器

硬盘存储器是由电机和硬盘组成的，一般置于主机箱内。硬盘是涂有磁性材料的磁盘组件，用于存放数据。硬盘的机械转轴上串有若干个盘片，每个盘片的上下两面各有一个读/写磁头，与软盘磁头不同，硬盘的磁头不与磁盘表面接触，它们“飞”在离盘片面百万分之一英寸的“气垫”上。硬盘是一个非常精密的机械装置，磁道间只有百万分之几英寸的间隙，磁头传动装置必须把磁头快速而准确地移到指定的磁道上。

③闪存

闪存又名优盘，是在存储速度与容量上介于软盘与硬盘之间的一种外部存储器。

④光盘

光盘的存储介质不同于磁盘，它属于另一类存储器。由于光盘的容量大、存取速度较快、不易受干扰等特点，其应用越来越广泛。光盘根据其制造材料和记录信息方式的不同一般分为三类：只读光盘、一次写入型光盘和可擦写光盘。

3．输入设备

输入设备是外界向计算机传送信息的装置。在微型计算机系统中，最常用的输入设备是键盘和鼠标。此外，还有光电笔、数字化仪、图像扫描仪、触摸屏、麦克风、视频输入设备、条形码扫描器等，也可以用磁盘和磁带进行输入。

4．输出设备

输出设备的作用是将计算机中的数据信息传送到外部媒介，并转化成某种为人们所认识的表示形式。在微型计算机中，最常用的输出设备有显示器和打印机。此外，还有绘图仪等，也可以通过磁盘和磁带输出。

二、总线结构

按照总线上传输信息的不同，总线可以分为数据总线（DB）、地址总线（AB）和控制总线（CB）三种。

①数据总线：用来传送数据信息，它主要连接了 CPU 与各个部件，是它们之间交换信息的通路。数据总线是双向的，而具体的传送方向由 CPU 控制。

②地址总线：用来传送地址信息。CPU 通过地址总线中传送的地址信息访问存储器。通常地址总线是单向的。同时，地址总线的宽度决定可以访问的存储器容量大小，如 20 条地址总线可以控制 1 MB 的存储空间。

③控制总线：用来传送控制信号，以协调各部件之间的操作。控制信号包括 CPU 对内存储器和接口电路的读写控制信号、中断响应信号，也包括其他部件传送给 CPU 的信号，如中断申请信号、准备就绪信号等。

三、主要的性能指标

计算机的常用指标有：

1．字长

字长是指一台计算机所能处理的二进制代码的位数。计算机的字长直接影响它的精度、功能和速度。字长越长，能表示的数值范围就越大，计算出的结果的有效位数也就越多；字长越长，能表示的信息就越多，机器的功能就更强。目前常用的是 16 位、32 位、64 位字长。

2．运算速度

运算速度是指计算机每秒钟所能执行的指令条数，一般用 MIPS（Million of Instructions Per Second，即每秒百万条指令）为单位。由于不同类型的指令执行时间长短不同，因而运算速度的计算方法也不同。

3．主频

主频是指计算机 CPU 的时钟频率，它在很大程度上决定了计算机的运算速度。一般时钟频率越高，运算速度就越快。主频的单位一般是 MHz（兆赫）或 GHz（吉赫），如微处理器 Pentium4/2.0 GHz 的主频为 2*1000 MHz。

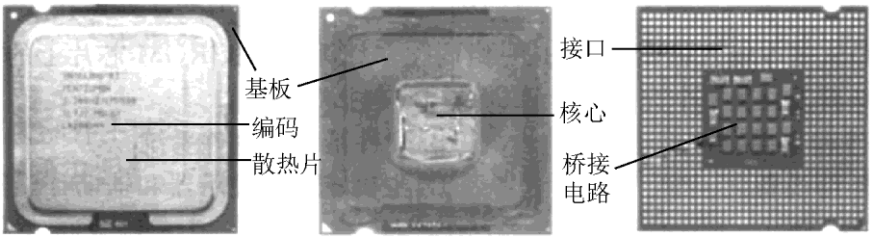
4．内存容量

内存容量是指内存储器中能够存储信息的总字节数，一般以 GB 为单位。内存容量反映内存储器存储数据的能力。目前计算机的内存容量有 2 GB、4 GB、8 GB 等。

四、CPU

CPU（中央处理单元）是微机的核心部件，是决定微机性能的关键部件。20 世纪 70 年代微型机的 CPU 问世，微型计算机的核心部件微处理器从 Intel 4004，80286，80386，80486 发展到 Pentium Ⅱ/Ⅲ和 Pentium 4，数位从 4 位、8 位、16 位、32 位发展到 64 位，主频从几

MHz 到今天的数 GHz 以上（1 GHz＝1000 MHz），CPU 芯片里集成的晶体管数由 2 万个跃升到 1000 万个以上。CPU 的发展和技术的进展直接推动了微型计算机的发展，也是微机各个发展阶段的主要标志。



从原理上看，CPU 的内部结构分控制单元、逻辑单元、存储单元三部分。从组成器件上看，CPU 的内部是由成千上万个晶体管组成，晶体管实质上就是一双位开关：即“开”和“关”。

CPU 的主要性能指标包括时钟主频、字长、高速缓存容量、指令集合和动态处理技术、制造工艺、封装方式和工作电压等。

主频是指 CPU 的工作时钟频率，是 CPU 内核电路的实际运行频率。一般来说，主频越高，一个时钟周期里面完成的指令数也越多，速度也越快。主频的单位为兆赫兹（MHz）和吉赫兹（GHz）。我们通常所说的 2.8 GHz、3.0 GHz 就是指 CPU 的主频。

字长（word size）指的是微处理器 CPU 能够同时处理的二进制位数的个数。字长的大小取决于 ALU 中寄存器的容量和连接着这些寄存器的电路性能。例如，8 位字长的微处理器有 8 位的寄存器，每次能处理 8 位的数据，因此，被称为“8 位处理器”。有更大字长的处理器能够在每个处理器周期内处理更大的数据，因此，字长越长计算机性能越好。目前的个人计算机通常都带有 32 位或 64 位的处理器。

高速缓存（Cache）也称为“RAM 缓存”或“缓冲存储器”。它是一种具有很高速度的特殊内部存储器，与安装在主板上其他位置的内存相比，它能够使微处理器更快地获得数据。

字节和字长的区别：常用的英文字符用 8 位二进制就可以表示，所以通常就将 8 位称为一个字节，字节是一种存储容量单位。而字长是 CPU 处理能力的一种标准，字长的长度是不固定的，对于不同的 CPU 字长的长度也不一样。8 位的 CPU 一次只能处理一个字节，而 32 位的 CPU 一次就能处理 4 个字节。同理，字长为 64 位的 CPU 一次可以处理 8 个字节。

1971 年，英特尔公司推出了世界上第一款微处理器 4004，字长 4 位，是 4 位微处理器。

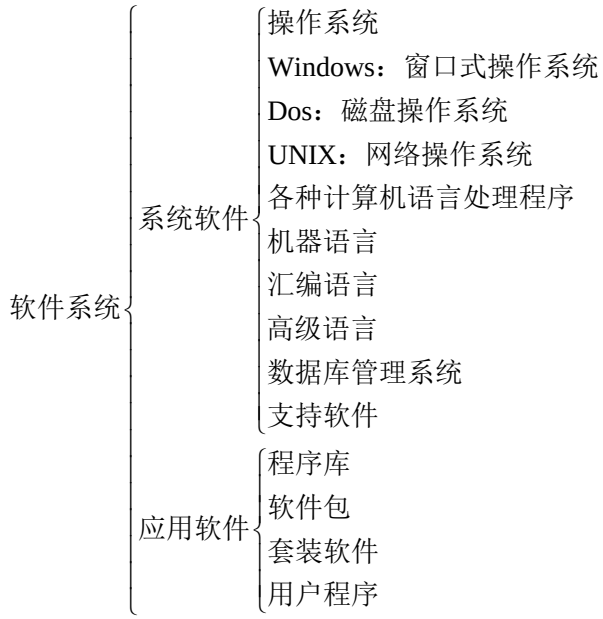
1978 年，英特尔公司生产的 8086 是第一个 16 位的微处理器。

1985 年，英特尔生产出 32 位字长处理器 80386。

目前市场上主流的 CPU 的字长几乎都达到了 64 位。

第3节　计算机软件

软件是计算机的灵魂。没有安装软件的计算机称为“裸机”，无法完成任何工作。硬件为软件提供运行平台。软件和硬件相互关联，两者之间可以相互转化、互为补充。计算机的软件分成系统软件和应用软件两大类。



一、系统软件

系统软件是指控制和协调计算机及外部设备，支持应用软件开发和运行的系统，是无需用户干预的各种程序的集合，主要功能是调度，监控和维护计算机系统；负责管理计算机系统中各种独立的硬件，使得它们可以协调工作。系统软件使得计算机使用者和其他软件将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。

常用的操作系统：

1．桌面操作系统从软件上可主要分为两大类，分别为类 Unix 操作系统和 Windows 操作系统。

Unix 和类 Unix 操作系统：Mac OS X，Linux 发行版（如 Debian．Ubuntu，Linux Mint，openSUSE，Fedora，Mandrake，Red Hat，Centos）；

微软公司 Windows 操作系统[5]：Windows 98，Windows 2000．Windows XP，Windows Vista，Windows 7，Windows 8，Windows 8.1，Windows 10 等。

2．服务器操作系统。

服务器操作系统主要集中在三大类：

Unix 系列：SUNSolaris，IBM—AIX，HP—UX，FreeBSD，OS X Server[6]等；

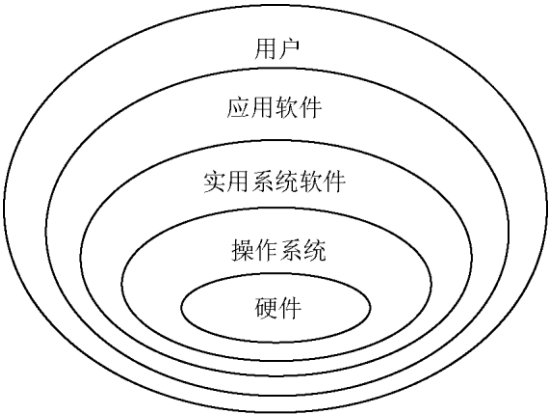
Linux 系列：Red Hat Linux，CentOS，Debian，UbuntuServer 等；

Windows 系列：Windows NT Server，Windows Server 2003，Windows Server 2008，Windows Server 2008 R2，windows server 2012，windows server technical 等。

二、应用软件

应用软件是用户为了解决各自应用领域里的具体任务而编写的各种应用程序和有关文档资料的统称。这类软件能解决特定问题。应用软件与系统软件的关系是；系统软件为应用软件提供基础和平台，没有系统软件应用的软件是无源之本．反过来应用软件又为系统软件服务。

常用的应用软件有以下几类：（1）字处理软件；（2）电子制表软件；（3）计算机辅助设计软件；（4）图形软件；（5）教育软件；（6）电子游戏软件。



三、计算机的指令

指令是一组二进制代码，它规定了由计算机执行的程序的一步操作。一条指令由操作码和操作数组成，前者规定指令要完成的操作，必不可少；后者是这个操作针对的对象，可以没有。

指令系统是一种计算机所能识别并可执行的全部指令的集合。例如，80386 的指令系统共有 123 种指令，可分为 9 类指令操作：数据传递、算术运算、逻辑运算、串操作、位操作、程序控制、高级语言指令、保护模式、处理器控制指令。

程序是计算机为了执行某种操作任务而将一条条指令按照一定的顺序排列起来的指令集。

第4节　计算机网络和 Internet 的基本概念

一、网络的定义

所谓计算机网络，就是利用通信线路和设备，把分布在不同地理位置上的多台计算机连接起来。

计算机网络是现代通信技术与计算机技术相结合的产物。

网络中计算机与计算机之间的通信依靠协议进行。协议是计算机收、发数据的规则。

TCP/IP：用于网络的一组通信协议。包括 IP（Internet Protocol）和 TCP（Transmission Control Protocol）

二、网络的发展

计算机网络的发展过程大致可以分为三个阶段：

远程终端联机阶段：主机一终端

计算机网络阶段：$\begin{cases}\text{计算机 — 计算机}\\ \text{Internet阶段：Internet}\end{cases}$

三、网络的主要功能

（1）资源共享；（2）信息传输；（3）分布处理；（4）综合信息服务。

四、网络的分类

按网络的地理范围进行分类：局域网（LAN）、城域网（MAN）、广域网（WAN）。

1．局域网（Local Area Network）

一般局限在1km 的范围内，局域网内传输速率较高，误码率低，结构简单、容易实现，具体标准是美国电气工程师协会制定的 IEEE802 系列标准。

2．城域网（Metropolitan Area Network）

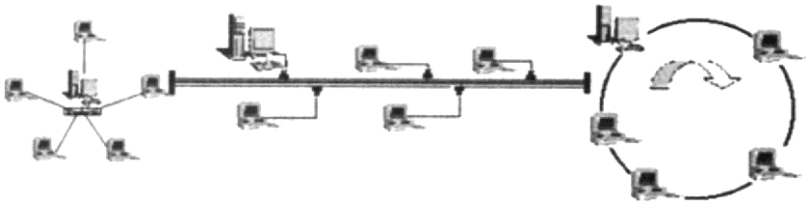
城域网的范围为几千米到几十千以内。

3．广域网（Wide Area Network）

广域网的范围在几十千米到几千千米以上。

注：MAN 和 WAN 一般都是由多个 LAN 构成的。

按网络的拓扑结构进行分类：星形、总线形、环形、树形、网状形。



星形　　总线形　　环形

1．星形

通信协议简单，对外围站点要求不高，单个站点故障不会影响全网，电路利用率低，连线费用大，网络性能依赖中央结点，每个站点需要一个专用链路。

2．总线形

结构简单，可靠性高；布线容易，连线总长度小于星形结构；总线任务重，易产生瓶颈问题；总线本身的故障对系统是毁灭性的。

3．环形

传输速率高，传输距离远；各节点的地位和作用相同；各节点传输信息的时间固定；容易实现分布式控制；站点的故障会形起整个网络的崩溃。

4．树形

分级结构，又称为分级的集中式网络。

5．网状形（不规则形）

计算机之间无规则地连接，一般广域网属于不规则形。

网络拓扑结构是指计算机网络节点和通信链路所组成的几何形状。

Internet 网是当今世界上规模最大、用户最多、影响最广泛的计算机互联网络。Internet 网上联有大大小小成千上万个不同拓扑结构的局域网、城域网和广域网。因此，Internet 网本身的拓扑只是一种虚拟拓扑结构，无固定形式。

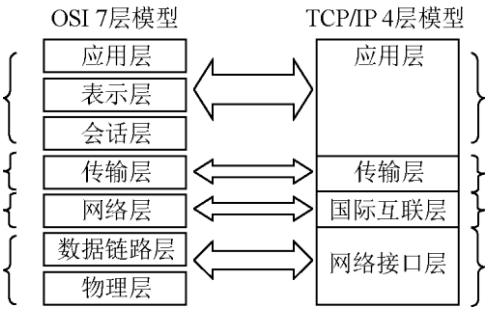
按采用的交换技术进行分类：电路交换、报文交换、分组交换。

五、网络的体系结构

国际标准化组织（International Standardization Organization，ISO）提出的开放式系统互联（Open System Interconnection，OSI）参考模型。它将数据从一个站点到达另一个站点的工作按层分割成七个不同的任务。

开放性是指任何遵循 OSI 标准的系统，只要物理上连接起来，它们之间都可以相互通信。OSI 参考模型并不是网络体系结构。OSI 只是描述了每一层的功能，并没有确定一个层的协议。而网络体系结构是网络层次结构和相关协议的集合。

TCP/IP模型与OSI模型的对应



六、IP 地址

在 TCP/IP 体系中，IP 地址是一个最重要的概念，什么叫 IP 地址?

所谓 IP 地址，是用于标识 Internet 网络上节点的 32 位地址（以后可能使用的 V6 版本是 128 位的，分 8 组，每组 16 位)。对于 Internet 网络上的每个节点，都必须指派一个唯一的地址，它由网络 ID 和唯一的主机 ID 组成。该地址通常用由句点分隔的八位字节的十进制数表示（例：192．168．7．27)。

Internet 的 IP 地址分成 A、B、C、D、E 五类，其中 A、B、C 为常用类，都由网络 ID 和主机 ID 两个部分组成，网络 ID 也叫作网络地址，标识大规模 TCP/IP 网际网络（由网络组成的网络）内的单个网段，连接到并共享访问同一网络的所有系统在其完整的 IP 地址内都有一个公用的网络 ID，这个 ID 也用于唯一地识别大规模的网际网络内部的每个网络；主机 ID，也叫作主机地址，识别每个网络内部的 TCP/IP 节点（工作站、服务器、路由器或其他 TCP/IP 设备)，每个设备的主机 ID 唯一地识别所在网络内的单个系统。



A类	126	1677214	1.0.01~126.255.255.254
B类	16384	65534	128.1.0.1~191.255.255.254
C类	209752	254	192.0.0.1~223.255.255.254

注:

①尽管 IP 地址的主机号的每个域的取值范围是 0~255，但主机 ID 所有域不能都为 0 或 255。例如：如果网络 ID 为 10，那么就不能把 10．0．0．0 和 10．255．255．255 两个 IP 地址分配给任何主机；如果网络 ID 为 192．114．3l，那么就不能把 192．114．31．0 和 192．114．31．255 两个 IP 地址分配给任何主机。

②专用网络寻址选项：如果某个连接仅存在于群集节点之间，并且不支持其他网络的客户机。可以为它配置私有 IP 网络地址，而不是企业的正式 IP 网络地址。经过 Internet 编号指派机构（IANA）协商同意，几个 IP 网络被保留下来以用作企业内部私有。这些保留的号码是:

10．0．0．0到10．255．255．255（A类）

172．16．0．0到172．31．255．255（B类）

192．168．0．0到192．168．255．255（C类）

IP地址的表示方法：采用点分十进制记法（Dotted Decimal Notation），即将32 bit的IP地址中的每8位用。等效的十进制表示，并每8位之间加上一个点，即4段二进制位组成，“．”隔开，每组数字的取值范围只能是0~255。

例如：10000001　00001011　00000011　00011111

129　　11　　3　　31

对应的IP地址为：129．11．3．31。

七、Internet概述

因特网（Internet）是一个建立在网络互联基础上的最大的、开放的全球性网络。因特网拥有数千万台计算机和上亿个用户，是全球信息资源的超大型集合体。

因特网起源于20世纪60年代中期，由美国国防部高级研究计划局（ARPA）资助的ARPANET，此后提出的TCP/IP协议为因特网的发展奠定了基础。

我国正式接入因特网是在1994年4月。当时为了发展国际科研合作的需要，中国科学院高能物理研究所和北京化工大学开通了到美国的因特网专线，并有千余科技界人士使用了因特网。

我国Internet的发展情况：

20世纪80年代末、90年代初才起步。

1989年我国第一个公用分组交换网CNPAC建成运行。

我国已陆续建成与Internet互联的四个全国范围的公用网络：中国公用计算机互联网（CHINANET）、中国金桥信息网（CHINAGBN）、中国教育和科研计算机网（CERNET）、中国科学技术网（CSTNET）。

八、域名和网址

1．网址

网址在因特网中，如果要从一台计算机访问网上另一台计算机，就必须知道对方的网址。这里所说的网址实际上指两个内涵，即IP地址、域名地址和URL。

IP地址（英语：Internet Protocol Address）是一种在Internet上的给主机编址的方式，也称为网络协议地址。常见的IP地址分为IPv4与IPv6两大类。

IPV4就是有4段数字，每一段最大不超过255。由于互联网的蓬勃发展，IP地址的需求量愈来愈大，使得IP地址的发放愈趋严格，各项资料显示全球IPv4地址可能在2005至2010年间全部发完（实际情况是在2011年2月3日IPv4地址分配完毕）

IPv6采用128位地址长度。在IPv6的设计过程中除了一劳永逸地解决了地址短缺问题以外，还考虑了在IPv4中解决不好的其他问题。

2．域名

32位二进制数IP地址对计算机来说是十分有效的，但记忆一组并无意义的且无任何特征的IP地址是困难的，为此，因特网引进了字符形式的IP地址，即域名。域名采用层次结构的基于“域”的命名方案，每一层由一个子域名组成，子域名间用“．”分隔。

其格式为：开头．主机名．主机类别．国家名（可以不要）。

如 www．ycwb．com．en：域名地址采用层次结构，一个域名一般有 3~5 个子段，中间用“．”隔开。

因特网上的域名由域名系统（Domain Name System，DNS）统一管理。DNS 是一个分布式数据库系统，由域名空间、域名服务器和地址转换请求程序三部分组成。有了 DNS，凡域名空间中有定义的域名都可以有效地转换为对应的 IP 地址，同样，IP 地址也可通过 DNS 转换成域名。在因特网上，域名和 IP 地址一样都是唯一的。

顶级域名有三类：

·国家顶级域名，如 cn（中国）、us（美国）、uk（英国）；

·国际顶级域名——int，国际性组织可在 int 下注册；

·通用顶级域名，如 com、net、edu、gov、…

IP 地址是纯数字化的，不便于人们记忆和识别。因此，用一串具有某种意义又便于记忆的文字来标识网络中的计算机是必要的，这便引入了域名的概念。

因特网中的计算机必须首先具有 IP 地址才能使用，域名则直观地标识出计算机的地理位置、所属机构等信息，对于 TCP/IP 来说，域名必须被映射为 IP 地址才能使用。在因特网中，专门用来管理域名与 IP 地址之间映射关系的计算机叫域名服务器（Domain Name Server，DNS）。

域名采用层次结构的命名方法，因特网的顶层域名（第一级域名）分为两大类，通用的（网络性质划分）和国家（地区）的，常用的见下表：

通用的		国家（地区）的	
edu	教育机构	cn	中国
gov	政府部门	hk	香港
net	网络组织	mo	澳门
com	商业组织	tw	台湾
org	非赢利性组织	jp	日本
mil	军事部门	sg	新加坡
…		…	

与一级域名类似，二级域名也有两类：一类表示网络性质（同一级）；一类表示国内的各省、直辖市、自治区等（如 fj，bj 分别表示福建和北京）。

根据需要还可以有第三级、第四级……每一级的域名都由英文字母和数字组成（随着技术的发展有可能可以用其他文字），最长不超过计划 63 个字符，不分大小写，层次低的域名写在左边，高的写在右边，之间用英文的点号分开，完整的域名不超过 255 个字符。

说明：域名的最左边（最底层）往往是一台主机的名字，通常用主机提供的服务表示，如 WWW、FTP、MAIL、BBS 等。因此，一个具体的域名也可以分成两个部分，即机器名．域名。如 WWW．TSINGHUA．EDU．CN 这个域名，TSINGHUA．EDU．CN 是域名，WWW 是主机名。如果主机名被省略，系统默认为 WWW。

九、因特网提供的服务

Internet 的服务有电子邮件、远程登录、文件传输、信息服务等。

1．万维网（WWW）

全球信息网，又称万维网（World Wide Web，WWW），是一个全球规模的信息服务系

统，由遍布于全世界的数以万计的 Web 站点组成。

万维网是瑞士日内瓦欧洲粒子实验室最先开发的一个分布式超媒体信息查询系统，目前它是因特网上最为先进、交互性能最好、应用最为广泛的信息检索工具，万维网包括各种各样的信息，如文本、声音、图像、视频等。万维网采用了“超文本”的技术，使得用户以通用而简单的办法就可获得因特网上的各种信息。

2．电子邮件（E-mail）

电子邮件地址格式为：收信人邮箱名@邮箱所在主机的域名。

例：winner01@2lcn．com。

电子邮件（Electronic mail，E-mail）是因特网上使用最广泛的一种服务。用户只要能与因特网连接，具有能收发电子邮件程序及个人的电子邮件地址，就可以与因特网上具有电子邮件地址的所有用户方便、快捷、经济地交换电子邮件。电子邮件可以在两个用户间交换，也可以向多个用户发送同一封邮件，或将收到的邮件转发给其他用户。电子邮件中除文本外，还可包含声音、图像、应用程序等各类计算机文件。此外，用户还可以以邮件方式在网上订阅电子杂志、获取所需文件、参与有关的公告和讨论组等。

3．文本传输协议（FTP）

文本传输协议（File Transfer Protocol，FTP）：用于在计算机间传输文件，如下载软件等。

FTP 是因特网上文件传输的基础，通常所说的 FTP 是基于该协议的一种服务。FTP 文本传输协议允许因特网上的用户将一台计算机上的文件传输到另一台上，几乎所有类型的文件，包括文本文件、二进制可执行文件、声音文件、图像文件、数据压缩文件等，都可以用 FTP 传送。

4．远程登录（Telnet）

远程登录（Telnet）：指通过 Internet 与其他主机连接。

Telnet 是远程登录服务的一个协议，该协议定义了远程登录用户与服务器交互的方式。允许用户在一台联网的计算机登录到一个远程分时系统时，像使用自己的计算机一样使用该远程系统。

十、浏览网页的相关概念

1．WWW 与 HTML

WWW 是因特网上集文本、声音、图像、视频等多媒体信息于一身的全球信息资源网络，是因特网上的重要组成部分。浏览器（Browser）是用户通向 WWW 的桥梁和获取 WWW 信息的窗口。通过浏览器，用户可以在浩瀚的因特网海洋中漫游，搜索和浏览自己感兴趣的所有信息。

WWW 的网页文件是用超文本标记语言 HTML（Hyper Text Markup Language）编写的，并在超文本传输协议 HTTP（Hyper Text Transmission Protocol）支持下运行的。超文本中不仅含有文本信息，还包括图形、声音、图像、视频等多媒体信息（故超文本又称超媒体），更重要的是超文本中隐含着指向其他超文本的链接，这种链接称为超链（Hyper Links）。

2．URL

简单地讲，URL（Uniform Resource Locator，统一资源定位器）就是因特网上的资源地址。每个 Web 页面，包括主页，都有一个唯一的地址，其格式为：协议名：//IP 地址或域名。

协议名表示所提供的服务，如 http：//www．online．sh．cn（上海热线）就是我们常用的万维网服务的 URL 地址。ftp：//pchome .net 就是因特网上文件传输服务的 URL 地址。

3．浏览器

WWW 浏览器是一个客户端的程序，其主要功能是使用户获取因特网上的各种资源。常用的浏览器有 Microsoft 的 Internet Explorer（IE）。

十一、电子邮件的相关概念

1．电子邮件概述

电子邮件（Electronic Mail，E-mail）是因特网上使用最广泛的一种服务。

2．电子邮件使用的协议

邮件服务器使用的协议有简单邮件传输协议（Simple Message Transfer Protocol，SMTP）、电子邮件扩展协议（Multipurpose Internet Mail Extensions，MIME）和 POP 协议（Post Office Protocol）。POP 服务需要一个邮件服务器来提供，用户必须在该邮件服务器上取得帐号才可使用这种服务。目前使用较普遍的 POP 协议为第三版，故又称为 POP3 协议。

3．信箱地址及其格式

使用电子邮件系统的用户首先要有一个电子邮件信箱，该信箱在因特网上有唯一的地址，以便识别。

像传统信件的信封有格式要求一样，电子邮件也有规范的地址格式。电子邮件的信箱地址由字符串组成，该字符串被字符“@”分成两部分（字符“@”在英语中可以读作“at”），前一部分为用户标识，可以使用该用户在该计算机上的登录名或其他标识，只要能够区分该计算机上的不同用户即可，如“zhangshan”；后一部分用户信箱所在的计算机的域名，如 ecust .edu .cn（华东理工大学网络中心邮件服务器主机域名）。像 zhangshan@ecust .edu .cn 就是一个电子邮件的地址。

电子邮件地址的一般格式为：〈用户标识〉@〈主机域名〉。

第5节　程序设计语言以及程序编译和运行的基本概念

程序就是一系列的操作步骤，计算机程序就是由人事先规定的计算机完成某项工作的姆作步骤。每一步骤的具体内容由计算机能够理解的指令来描述，这些指令告诉计算机“做什么”和“怎样做”。编写计算机程序所使用的语言称为程序设计语言。

通常分为三类；机器语言、汇编语言和高级语言。

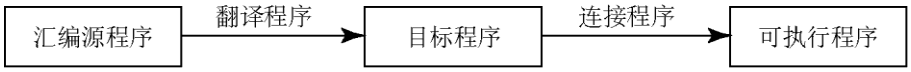
1．机器语言

计算机最早的语言处理程序是机器语言，它是计算机能直接识别的语言，而且速度快。机器语言是用二进制代码来编写计算机程序的，因此又称二进制语言。例如用机器语言来表示“8+4”这个算式，是一串二进制码“00001000 00000100 00000100”。机器语言书写困难、记忆复杂，一般很难掌握。

2．汇编语言

由于机器语言的缺陷，人们开始用助记符编写程序，用一些符号代替机器指令所产生的语言称为汇编语言。但是，用汇编语言编写的源程序不能被计算机直接识别，必须使用某种

特殊的软件将用汇编语言写的源程序翻译和连接成能被计算机直接识别的二进制代码。其示意图如图所示。



汇编源程序翻译连接过程

汇编语言虽然采用了助记符来编写程序，比机器语言简单，但是汇编语言仍属于低级语言，它与计算机的体系结构有关，在编写程序前要花费相当多的时间和精力去熟悉机器的结构。因此工作量大、繁琐，而且程序可移植性差。

3．高级语言

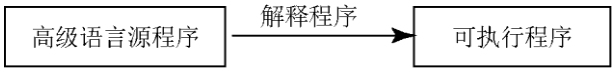
计算机并不能直接接受和执行用高级语言编写的源程序，源程序在输入计算机时，通过“翻译程序”翻译成机器语言形式的目标程序，计算机才能识别和执行。这种“翻译”通常有两种方式，即编译方式和解释方式。

编译方式是：编译方式的翻译工作由“编译程序”来完成，它是先将整个源程序都转换成二进制代码，生成目标程序，然后把目标程序连接成可执行的程序，以完成源程序要处理的运算并取得结果。



高级语言编译过程

解释方式是：源程序进入计算机时，解释程序边扫描边解释，对源程序的语句解释一条、执行一条，不产生目标程序。解释方式的翻译工作由“解释程序”来完成。



高级语言解释过程

编译性语言有 C/C++、Pascal/Object Pascal（DelPhi）等。

解释性语言有 ASP、PHP、Java、JavaScript、VBScriPt、Perl、Python、Ruby、MATLAB 等。

使用编译语言程序将整个源程序编译连接为可执行的文件，这种方式效率高、可靠性高、可移植性好。不过，当源程序修改后，必须重新编译。

面向对象语言借鉴了 20 世纪 50 年代的人工智能语言 LISP，引入了动态绑定的概念和交互式开发环境的思想；始于 20 世纪 60 年代的离散事件模拟语言 Simula67，引入了类的要领和继承。成形于 20 世纪 70 年代的 Smalitalk。

面向对象语言的发展有两个方向：一种是纯面向对象语言，如 Smalltalk、EIFFEL 等；另一种是混合型面向对象语言，即在过程式语言及其他语言中加入类、继承等成分，如 C++、Objectire-C 等。

第6节　进制的基本概念与进制转换、字节与字

一、进位计数制的基本概念

将数字符号按序排列成数位，并遵照某种由低位到高位的进位方式计数表示数值的方法，称作进位计数制。

1．十进制

十进制计数制由 0、1、2、3、4、5、6、7、8、9 共 10 个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满十就向高位进一，即“逢十进一”。

2．八进制

八进制计数制由 0、l、2、3、4、5、6、7 共 8 个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满八就向高位进一，即“逢八进一”。

3．二进制

二进制计数制由 0 和 1 共两个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满二就向高位进一，即“逢二进一”。

4．其他进制

在日常生活和工作中还会使用其他进制数。如：十二进制数、十六进制数、百进制数和千进制数等。无论哪种进制数，表示的方法都是类似的。如：十六进制数由 0、l、2、3、4、5、6、7、8、9、A、B、C、D、E 和 F 共 16 个符号组成，“逢十六进一”。不同的是，用 A、B、C、D、E 和 F 分别表示 10、1l、12、13、14 和 15 六个数字符号。

5．基数与权

某进制计数制允许选用的基本数字符号的个数称为基数。一般而言，J 进制数的基数为 J，可供选用的基本数字符号有 J 个，分别为 0 到 J—1，每个数位计满 J 就向高位进一，即“逢 J 进一”。

某进制计数制中各位数字符号所表示的数值表示该数字符号值乘以一个与数字符号所处位置有关的常数，该常数称为“位权”（简称“权”）。位权的大小是以基数为底、数字符号所处的位置的序号为指数的整数次幂。

十进制数允许使用十个基本数字符号，所以基数为 10，每位数字符号代表的位数的大小是以 10 为底，数字符号所处位置的序号为指数的整数次幂。

十进制数的百位、十位、个位和十分位的权分别为10^2、10^1、10^0、10^{-1}。故$(555.5)_{10}$可表$(555.5)_{10}=5*10^2+5*10^1+5*10^0+5*10^{-1}$。J 进制数相邻两位数相差 J 倍，若小数点向左移 n 位，则整个数值就缩小J^n。反之，小数点向右移 n 位，数值就放大J^n。

如下给出了任意进制数$(K^2$、K^1、K^0、K^{-1}、$K^{-2})$，当 J 分别为 2、8、10 和 16 时各位权值对照。

进位制 J \ 数位	K^2	K^1	K^0	小数点位置	K^{-1}	K^{-2}
J＝2	$2^2=4$	$2^1=2$	$2^0=1$		$2^{-1}=0.5$	$2^{-2}=0.25$
J＝8	$8^2=64$	$8^1=8$	$8^0=1$		$8^{-1}=0.125$	$8^{-2}=0.015625$
J＝10	$10^2=100$	$10^1=10$	$10^0=1$		$10^{-1}=0.1$	$10^{-2}=0.01$
J＝16	$16^2=256$	$16^1=16$	$16^0=1$		$16^{-1}=0.0625$	$16^{-2}=0.00390625$

二、数制之间的转换

计算机内部使用的数字符号只有“0”和“1”两个。也就是说，计算机内部使用的是二进制数，所有的数值数据和非数值数据，都是由“0”和“1”这两个数字符号加以组合而成的，我们称之为“二进制代码”。

计算机只用二进制的两个数码“0”和“1”来实现算术和逻辑运算，而人们仍然用十进

制的形式向计算机中输入原始数据，并让计算机也用十进制形式显示和打印运算结果。所以，必须有一种自动转换方法，即让数据输入计算机后，将十进制转换成对应的二进制数，并在处理完毕后，再自动将二进制结果转换为十进制数。

为了表达方便起见，常在数字后加一缩写字母后缀作为不同进制数的标识。各种进制数的后缀字母分别为：

B：二进制数。　　O：八进制数。

D：十进制数。　　H：十六进制数。

对于十进制数，通常不加后缀，也即十进制数后的字母 D 可省略。

1．二进制与十进制的转换

（1）二进制转十进制

方法：“按权展开求和”。

例：$(1011.01)_2=(1*2^3+0*2^2+1*2^1+1*2^0+0*2^{-1}+1*2^{-2})_{10}$

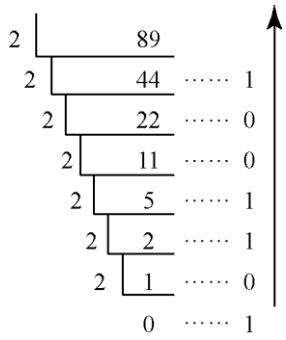
$=(8+0+2+1+0+0.25)_{10}$

$=(11.25)_{10}$

（2）十进制转二进制

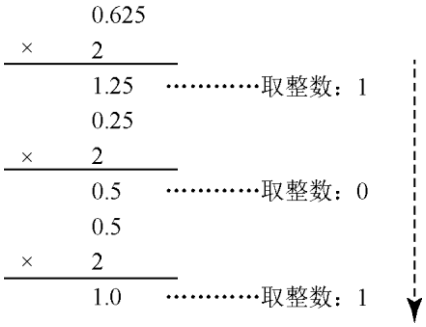
十进制整数转二进制数：“除以 2 取余，逆序输出”。

例：$(89)_{10}=(1011001)_2$



十进制小数转二进制数：“乘以 2 取整，顺序输出”。

例：$(0.625)_{10}=(0.101)_2$



2．八进制与二进制的转换

例：将八进制的 37.416 转换成二进制数。

3　7　.　4　1　6

011　111　.　100　001　110

即：$(37.416)_8=(1111.10000111)_2$。

例：将二进制的 10110.0011 转换成八进制。

$\underset{2}{\underline{010}}\ \underset{6}{\underline{101}}.\underset{1}{\underline{001}}\ \underset{4}{\underline{100}}$

即：$(10110.011)_2=(26.14)_8$。

3．十六进制与二进制的转换

例：将十六进制数 5DF.9 转换成二进制。

5　　D　　F　.　9

0101　1101　1111　.　1001

即：$(5DF.9)_{16}=(10111011111.1001)_2$。

例：将二进制数 1100001.111 转换成十六进制。

0110　0001　.1110

6　　1　　.　E

即：$(1100001.111)_2=(61.E)_{16}$。

4．把一个八进制转换成十进制

把这个八进制的最后一位乘上8^0，倒数第二位乘上8^1，……，一直到最高位乘上8^n，然后将各项乘积相加，结果即为它的十进制表达式。

例：把八进制 36 转换为十进制。

$(36)_8=3*8^1+6*8^0=24+6=(30)_{10}$

5．把一个十六进制转换成十进制

把这个十六进制的最后一位乘上16^0，倒数第二位乘上16^1，……，一直到最高位乘上16^n，然后将各项乘积相加，结果即为它的十进制表达式。

例：把十六制 1E 转换为十进制。

$(1E)_{16}=1*16^1+14*16^0=16+14=(30)_{10}$

三、数的原码、补码和反码表示

（1）机器数与真值

在计算机中，表示数值的数字符号只有 0 和 1 两个数码，我们规定最高位为符号位，并用 0 表示正数符号，用 1 表示负数符号。这样，机器中的数值和符号全“数码化”了。为简化机器中数据的运算操作，人们采用了原码、补码、反码及移码等几种方法对数值位和符号位统一进行编码。为区别起见，我们将数在机器中的这些编码表示称为机器数（如 10000001），而将原来一般书写表示的数称为机器数的真值（如 –0000001）。

（2）原码表示法

原码表示法是一种简单的机器数表示法，即符号和数值表示法。设 x 为真值，则[x]原为机器数表示。

例；设 $x=1100110$，则$[x]_{原}=01100110$；

$x=-1100111$，则$[x]_{原}=11100111$。

（3）反码表示法

正数的反码就是真值本身；负数的反码，只须对符号位以外各位按位“求反”（0 变 1，1 变 0）即可。

例；设 $x=1100110$，则$[x]_{反}=01100110$；

$x=-1100111$，则$[x]_{反}=10011000$。

（4）补码表示法

负数用补码表示时，可以把减法转化成加法。正数的补码就是真值本身；负数的补码是符号位为 1，数值各位取反（0 变为 1、1 变为 0），最低位加 1。

例：设$x=1100110$，则$[x]_{补}=01100110$；

$x=-1100111$，则$[x]_{补}=10011001$。

从上面关于原码、反码、补码的定义可知：一个正数的原码、反码、补码的表示形式相同，符号位为 0，数值位是真值本身；一个负数的原码、反码、补码的符号位都为 1，数值位原码是真值本身，反码是各位取反，补码是各位取反，最低位加 1。真值 0 的原码和反码表示不唯一，而补码表示是唯一的，即

$[+0]_{原}=000\cdots0$，$[-0]_{原}=100\cdots0$；

$[+0]_{反}=000\cdots0$，$[-0]_{反}=111\cdots1$；

$[+0]_{补}=[-0]_{补}=000\cdots0$。

注：不同编码表示的整数的范围是这样的（以 N 位二进制位）：

原码：$0\sim2^{n}-1$（无符号），$-2^{n-1}-1\sim2^{n-1}-1$（有符号）；

反码：$-2^{n-1}-1\sim2^{n-1}-1$（不存在无符号的情况）；

补码：$-2^{n-1}\sim2^{n-1}-1$（不存在无符号的情况）。

大家很明显看出，补码表示的范围最大。现以 8 位二进制位为例说明如下：

原码：$00000000\sim11111111$，即 0~255（无符号），

$11111111\sim01111111$，即$-127\sim+127$（有符号）；

反码：$10000000-01111111$，即

$-127\sim+127$（因为10000000的值为-127，11111111的值为-0）；

补码：$10000000-01111111$，即

$-128\sim+127$（因为10000000的值为-128，11111111的值为-1）；

如果没说具体编码形式，则计算机中 N 位二进制无符号数的范围是$0\sim2^{n}-1$；有符号数的范围是$-2^{n-1}\sim2^{n-1}-1$。

以 8 位有符号整数为例子：

原码：若 X 为正数，则最高位（符号位）为 0，其余按照二进制数排列；

若 X 为负数，则最高位为 1，后面和正数原码一样。

例：$+7$：00000111，-7：10000111；

$+0$：00000000，-0：10000000。

反码：若 X 为正数，则反码与原码相同；

若 X 为负数，则将原码除符号位取反。

例：-7：11111000。

补码：反码$+1$。

例：-7：1111001。

四、编码基本概念

1．编码

计算机要处理的数据除了数值数据以外，还有各类符号、图形、图像和声音等非数值数据。而计算机只能识别两个数字。要使计算机能处理这些信息，首先必须将各类信息转换成“0”和“1”表示的代码，这一过程称为编码。

2．数据

能被计算机接受和处理的符号的集合都称为数据。

3．比特

比特（Bit：——二进制数位）是指 1 位二进制的数码（即 0 或 1）。比特是计算机中表示信息的数据编码中的最小单位。

4．字节

字节表示被处理的一组连续的二进制数字。通常用 8 位二进制数字表示一个字节，即一个字节由 8 个比特组成。

字节是存储器系统的最小存取单位。

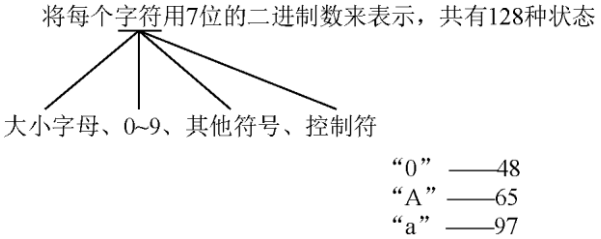
五、字符的表示

字符是人与计算机交互过程中不可缺少的重要信息。要使计算机能处理、存储字符信息，首先必须用二进制“0”和“1”代码对字符进行编码。

下面以西文字符和汉字字符为例，介绍常用的编码标准。

六、ASCII 编码

ASCII 编码是由美国国家标准委员会制定的一种包括数字、字母、通用符号和控制符号在内的字符编码集，全称为美国国家信息交换标准代码（American Standard Code for Information Interchange）。ASCII 码是一种 7 位二进制编码，能表示 $2^7=128$ 种国际上最通用的西文字符，是目前计算机中，特别是微型计算机中使用最普通的字符编码集。



ASCII 编码包括 4 类最常用的字符。

①数字“0”~“9”。ASCII 编码的值分别为 $0110000\,\mathrm{B}\sim0111001\,\mathrm{B}$，对应十六进制数为 $30\,\mathrm{H}\sim39\,\mathrm{H}$。

②26 个英文字母。大写字母“A”~“Z”的 ASCII 编码值为 $41\,\mathrm{H}\sim5\mathrm{AH}$，小写字母“a”~“z”的 ASCII 编码值为 $61\,\mathrm{H}\sim7\mathrm{AH}$。

③通用符号。如“+”“－”“=”“*”和“/”等共 32 个。

④控制符号。如空格符合回车符等共 34 个。

ASCII 码是一种 7 位编码，它存储时必须占全一个字节，也即占用 8 位：b7、b6、b5、b4、b3、b2、b1、b0，其中 b7 恒为 0，其余几位为 ASCII 码值。

人们可以通过键盘输入和显示器显示不同的字符，但在计算机中，所有信息都是用二进

制代码表示的。 n 位二进制代码能表示 2^n 个不同的字符，这些字符的不同组合就可表示不同的信息。为使计算机使用的数据能共享和传递，必须对字符进行统一的编码。ASCII 码（美国标准信息交换码）是使用最广泛的一种编码。ASCII 码由基本的 ASCII 码和扩充的 ASCII 码组成。在ASCII码中，把二进制位最高位为0的数字都称为基本的ASCII码，其范围是 $0 \sim 127$；把二进制位最高位为 1 的数字都称为扩展的 ASCII 码．其范围是 $128 \sim 255$ 。

七、汉字信息编码

1．汉字交换码

汉字交换码是指不同的具有汉字处理功能的计算机系统之间在交换汉字信息时所使用的代码标准。自国家标准 GB2312—80 公布以来，我国一直延用该标准所规定的国标码作为统一的汉字信息交换码（GB5007—85 图形字符代码）。

GB2312—80 标准包括了 6763 个汉字，按其使用频度分为一级汉字 3755 个和二级汉字 3008 个。一级汉字按拼音排序，二级汉字按部首排序。该标准还包括标点符号、数种西文字母、图形、数码等符号 682 个。

区位码的区码和位码均采用从 01 到 94 的十进制，国标码采用十六进制的 21H 到 73H（数字后加 H 表示其为十六进制数）。区位码和国标码的换算关系是：区码和位码分别加上十进制数 32。如“国”字在表中的 25 行 90 列，其区位码为 2590，国标码是 397AH。

2．字形存储码

字形存储码是指供计算机输出汉字（显示或打印）用的二进制信息．也称字模。通常，采用的是数字化点阵字模。


16×16表示

一般的点阵规模有 $16*16$ 、$24*24$ 等，每一个点在存储器中用一个二进制位（bit）存储。在 $16*16$ 的点阵中，需 $8*32$ bit 的存储空间，每 8 bit 为 1 字节。所以，需 32 字节的存储空间。在相同点阵中，不管其笔画繁简，每个汉字所占的字节数相等。

为了节省存储空间，普遍采用字形数据压缩技术。所谓矢量汉字，是指用矢量方法将汉字点阵字模进行压缩后得到的汉字字形的数字化信息。

第二章 算法

第 1 节 算法的基本概念

1．算法

算法是对特定问题求解步骤的一种描述，它是指令（规则）的有限序列，其中每一条指令表示一个或多个操作。简单地说，算法就是解决问题的操作步骤。

一个算法必须满足以下五个重要的特征。

（1）有穷性

对于任意一组合法的输入值，算法的每个操作步骤都能在有限的时间内完成，这包括合理的执行时间的含义。如果一个算法执行耗费的时间太长，即使最终得出了结果，也是没有意义的。

（2）确定性

算法中的每一步都必须有明确的定义，不允许有歧义性和多义性。确定性使算法的执行者或者阅读者能够明确其含义及如何执行，并且在任何条件下，算法都只有一条执行路径。

（3）输入

一个算法应该有 0 个或多个输入，以刻画运算对象的初始情况。所谓 0 个输入。是指有的算法表面上可以没有输入，实际上已被嵌入算法之中。

（4）输出

一个算法应该有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

（5）可行性

一个算法必须遵循特定条件下的解题规则，算法描述的每一个操作都应该是特定的解题规则中允许使用的、可执行的，并可以通过执行有限次来实现。

2．算法的复杂度

同一个问题可用不同的算法来解决，而一个算法质量的优劣将影响算法乃至程序的效率。算法分析的目的在于选择合适的算法和改进算法。一个算法的评价主要从时间复杂度和空间复杂度来考虑。

（1）时间复杂度

算法的时间复杂度（Time Complexity）是指算法所需要的计算工作量，用算法所执行的基本运算次数来度量。

常见的时间复杂度有：常数阶 $O(1)$，对数阶 $O(\log_2 n)$，线性阶 $O(n)$，线性对数阶 $O(n\log_2 n)$，平方阶 $O(n^2)$，立方阶 $O(n^3)$，……，指数阶 $O(2^n)$。随着问题规模 n 的不断增大，上述时间复杂度不断增大，算法的执行效率将越低。

（2）空间复杂度

算法的空间复杂度（Space Complexity）是指执行这个算法所需要的内存空间。算法执行期间所需的存储空间主要包括三部分：输入数据所占的存储空间、程序本身所占的空间和

算法执行过程中时所需的存储空间。

3．算法的基本结构

算法的结构不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。一般一个算法可以用顺序、选择和循环三种基本结构组合而成。

（1）顺序结构

顺序结构是最简单、最常用的算法结构，语句与语句之间、框与框之间按从上到下的顺序进行。

（2）选择结构

选择结构是先根据条件做出判断，再决定执行哪一种操作的算法结构。它必须包含判断框。当条件 P 成立（或称为真）时执行 A，否则执行 B，不可能两者同时执行，但 A 或 B 两个框中可以有一个是空的，即不执行任何操作。

（3）循环结构

在一些算法中，经常会出现从某处开始，按照一定条件，反复执行某一处理步骤的情况，这就是循环结构。反复执行的处理步骤为循环体，它可以细分为两类：当型循环结构和直到型循环结构。

第 2 节 入门组基础算法理论

1．高精度计算

利用计算机进行数值计算，有时会遇到这样的问题：有些计算要求精度高，希望计算的数的位数可达几十位甚至几百位，虽然计算机的计算精度不断提高了，但因受到硬件的限制，往往达不到实际问题所要求的精度。我们可以利用程序设计的方法去实现这样的高精度计算。主要涉及高精度的加、减、乘、除运算，实现的时候采用逐位加、减、乘、除即可。

常见问题有“汉诺塔的步数”“国王的米粒”等。

2．穷举算法

所谓穷举法，即枚举法，指的是从可能的解的集合中一一枚举各元素，用题目给定的检验条件判定哪些是无用的、哪些是有用的。能使命题成立，即为其解。

有些问题可以用循环语句和条件语句直接求解，有些问题用循环求解时循环次数太多，无法编写程序，则需要用到回溯、递归、分治等方法。

常见问题有“百钱买百鸡”“素数判断”等。

3．数据排序

排序就是将杂乱无章的数据元素，通过一定的方法按关键字顺序排列的过程。

排序的方法很多，下面根据初赛的要求，简单介绍各种常见排序算法在一些方面的比较，尤其是时间复杂度和稳定性两个方面。稳定性，指在原序列中相同元素的相对位置与排好序的新序列中相同元素的相对位置是否相同。若相同，则该算法是稳定的，否则不稳定。

时间复杂性比较：

插入排序、冒泡排序、选择排序的时间复杂性为 $O(n^2)$；

其他非线性排序的时间复杂性为 $O(n\log n)$；

线性排序的时间复杂性为O(n)。

稳定性比较:

插入排序、冒泡排序、二叉树排序、归并排序及其他线性排序是稳定的;

选择排序、希尔排序、快速排序、堆排序是不稳定的;

辅助空间的比较:

线性排序、归并排序的辅助空间为O(n)，其他排序的辅助空间为O(1)。

其他比较:

①插入、冒泡排序的速度较慢。但参加排序的序列局部或整体有序时，这种排序能达到较快的速度。反而在这种情况下，快速排序慢了，时间复杂度会达到其上限。当数据为随机数据时，快速排序远远快于插入、冒泡、选择排序，时间复杂度接近其下限。

②当n较小时，对稳定性不作要求时宜用选择排序，对稳定性有要求时宜用插入或冒泡排序。

③若待排序的记录的关键字在一个明显有限范围内且空间允许时用桶排序。

④当n较大时，关键字元素比较随机且对稳定性没要求，宜用快速排序。

⑤当n较大时，关键字元素可能出现本身是有序的且对稳定性有要求时，在空间允许的情况下，宜用归并排序。

常见问题有“合并果子”“逆序对问题”等。

排序算法	平均时间复杂度	最好情况	最坏情况	空间复杂度	排序方式	稳定性
冒泡排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	In-place	稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	In-place	不稳定
插入排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	In-place	稳定
希尔排序	$O(nlogn)$	$O(nlog^2n)$	$O(nlog^2n)$	$O(1)$	In-place	不稳定
归并排序	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(n)$	Out-place	稳定
快速排序	$O(nlogn)$	$O(nlogn)$	$O(n^2)$	$O(logn)$	In-place	不稳定
堆排序	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(1)$	In-place	不稳定
计数排序	$O(n+k)$	$O(n+k)$	$O(n+k)$	$O(k)$	Out-place	稳定
桶排序	$O(n+k)$	$O(n+k)$	$O(n^2)$	$O(n+k)$	Out-place	稳定
基数排序	$O(n\times k)$	$O(n\times k)$	$O(n\times k)$	$O(n+k)$	Out-place	稳定

4．递推算法

递推法是一种重要的数学方法，在数学的各个领域中都有广泛的运用，也是计算机用于

数值计算的一个重要算法。这种算法特点是：一个问题的求解需一系列的计算，在已知条件和所求问题之间总存在着某种相互联系的关系，在计算时，如果可以找到前后过程之间的数量关系（即递推式），那么从问题出发逐步推到已知条件。此种方法叫逆推。无论顺推还是逆推，其关键是要找到递推式。这种处理问题的方法能使复杂运算化为若干步重复的简单运算，充分发挥出计算机擅长于重复处理的特点。递推算法的首要问题是得到相邻的数据项间的关系（即递推关系）。递推算法避开了求通项公式的麻烦，把一个复杂问题的求解，分解成了连续的若干步简单运算。一般说来，可以将递推算法看成是一种特殊的迭代算法。

常见问题有“斐波那契数列”“过河卒”等。

5．递归算法

递归程序设计是 C++语言程序设计中的一种重要方法，它使许多复杂的问题变得简单，容易解决了。递归特点是：函数或过程调用它自己本身。其中直接调用自己称为直接递归，而将 A 调用 B，B 再调用 A 的递归叫作间接递归。

常见问题有“汉诺塔问题”“Ackerman 函数”等。

6．搜索与回溯算法

搜索与回溯是计算机解题中常用的算法，很多问题无法根据某种确定的计算法则来求解，可以利用搜索与回溯的技术求解。回溯是搜索算法中的一种控制策略。它的基本思想是：为了求得问题的解，先选择某一种可能情况向前探索，在探索过程中，一旦发现原来的选择是错误的，就退回一步重新选择，继续向前探索，如此反复进行，直至得到解或证明无解。

如迷宫问题：进入迷宫后，先随意选择一个前进方向，一步步向前试探前进，如果碰到死胡同。说明前进方向已无路可走，这时，首先看其他方向是否还有路可走，如果有路可走，则沿该方向再向前试探；如果已无路可走，则返回一步，再看其他方向是否还有路可走；如果有路可走，则沿该方向再向前试探。按此原则不断搜索回溯再搜索，直到找到新的出路或从原路返回入口处无解为止。

常见问题有“八皇后问题”“骑士游历问题”等。

7．贪心算法

贪心算法是指在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的仅是在某种意义上的局部最优解。

贪心算法没有固定的算法框架，算法设计的关键是贪心策略的选择。必须注意的是，贪心算法不是对所有问题都能得到整体最优解，选择的贪心策略必须具备无后效性。即某个状态以后的过程不会影响以前的状态，只与当前状态有关。所以，对所采用的贪心策略，一定要仔细分析其是否满足无后效性。

常见问题有“删数问题”“排队打水问题”等。

8．分治算法

分治就是指分而治之，即将较大规模的问题分解成几个较小规模的问题，通过对较小规模问题的求解达到对整个问题的求解。当我们将问题分解成两个较小问题求解时的分治方法称之为二分法。

常见问题有“归并排序”“比赛日程安排”等。

9．广度优先搜索

广度优先算法的核心思想是：从初始节点开始，应用算符生成第一层节点，检查目标节

点是否在这些后继节点中，若没有，再用产生式规则将所有第一层的节点逐一扩展，得到第二层节点，并逐一检查第二层节点中是否包含目标节点。若没有，再用算符逐一扩展第二层的所有节点……如此依次扩展，检查下去，直到发现目标节点为止。这种搜索的次序体现沿层次向横向扩展的趋势，所以称之为广度优先搜索。

常见问题有“分油问题”“八数码问题”等。

10．动态规划

动态规划程序设计是解决多阶段决策过程最优化问题的一种途径、一种方法，而不是一种特殊算法。由于各种问题的性质不同，确定最优解的条件也互不相同，因而动态规划的设计方法对不同的问题，有各具特色的解题方法，而不存在一种万能的动态规划算法，可以解决各类最优化问题。需要满足“最优子结构”和“无后效性”的两项基本条件。实现时主要分成几个步骤：划分阶段、确定状态和状态变量、确定决策并写出状态转移方程、寻找边界条件、程序设计实现。大致可以分为以下几类：线性、区间、背包形、树形等。

常见问题有“最长不下降序”“最长公共子序列”等。

第三章 数据结构

第1节 线性数据结构

1.1 数组

1.1.1 数组定义

数组用一组连续内存空间来存储相同类型的线性数据，不支持动态扩容。数组在定义时要指定大小。

在 C++中，数组定义个数如下：

类型 数组名[元素个数];

其中类型代表数组中存储的变量类型，元素个数代表数组中元素的个数。元素个数必须为整数或者常量表达式。

例如 **int a[8];**

该数组定义了一个名字为 a、大小为 8 的整型数组，相当于定义了 8 个整型变量。

因为数组元素类型相同，存储空间连续，所以可以通过索引(数组下标)来计算出某个元素的地址。数组下标是从 **0** 到**元素个数-1** 的整数。注意，数组元素的首个元素下标为 0。

在 C++中，数组元素的引用格式如下：

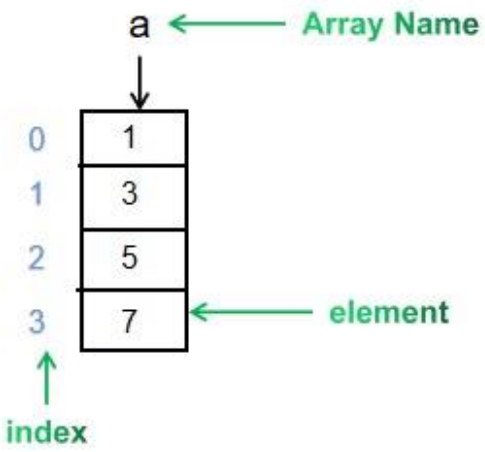
数组名[下标]

数组能通过下标随机访问元素，读取和修改效率很高。但是数组存储空间是连续的，插入和删除数组元素效率非常低，所以数组定义需要事先估计存储空间。

比如上面的 a 数组在内存中的示意如下：

内存大小	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节
变量	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

在 C++语言中，数组名表示的是首个数组元素的地址，也可以对数组名加上一个位移 n 得到第 n 个元素的地址。通过下标访问数组元素和通过数组元素地址访问效果是一样的。例如 a[0]相当于*a, a[1]相当于*(a+1)。



1.1.2 数组初始化

数组的单个元素赋值与变量赋值方式相同。数组也可以在定义时初始化。

C++中，数组初始化的格式为：

类型 数组名[元素个数] = {初值 0,初值 1……};

例如：

```
int b[10] = {10, 9, 8, 7, 6, 5, 4, 3, 2};
```

其中所赋值的元素个数小于等于数组元素个数，系统会从下标为 0 的元素开始依次赋值，到大括号内的最后一个值赋值结束。比如上述 b 数组中 `b[0]=10; b[1]=9; b[8]=2; b[9]=0;`

C++程序中，可以将数组作为全局变量在主函数外定义，系统会将所有元素默认初始化为 0。

在 C++中也可以将 memset 函数(需包含 cstring 头文件)将数组统一初始化为一个值。

memset(数组名,0,sizeof(数组名));

因为 memset 函数是按照字节赋值的，所以尽量避免将数组初始化为 0 或-1 之外的其他值，除非每个字节的内容相同。如用 0x3f 将数组初始化为最大值，或-0x3f 将所有元素赋值为最小值。假定数组名为 a，

memset(数组名,0x3f,sizeof(数组名)); //初始化为最大值

memset(数组名,-0x3f,sizeof(数组名));//初始化为最小值

1.1.3 数组的操作

数组元素在内存上是连续的，可随机访问某个元素，但是需要事先预估数组容量，插入、删除某个元素需要 O(n)。

1.1.4 二维数组

二维数组本质上是以数组作为元素的数组，即“数组的数组”。

在 C++中，二维数组的定义格式如下：

类型 数组名[行数][列数];

例如 int a[2][4]。a[0][0]代表二维数组第一个元素。

该二维数组的表示形式如下表：

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]

实际上，二维数组在内存中是使用一维形式存储的，因此上述数组在内存中的示意如下：

内存大小	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节
变量	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[1][0]	a[1][1]	a[1][2]	a[1][3]

在初始化二维数组时，由于表示形式和内存形式的区别，通常有两种初始化方式：

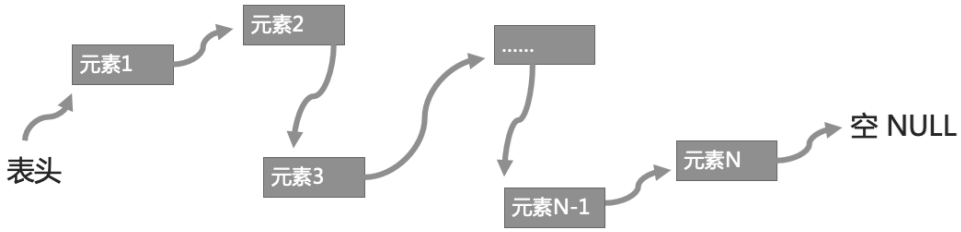
```
int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
  int a[2][3] = {1, 2, 3, 4, 5, 6};
```

两种方式得到的数组是完全相同的。其中 `a[0][0]=1; a[1][0]=4; a[1][2]=6;`

1.2 链表

1.2.1 链表定义

链表是由一系列结点组成的线性结构。每个结点包括两部分：一个存储数据元素的数据域，另一个是存储下一个结点地址的指针域。数组的线性顺序是由数组下标决定的，而链表的线性是由各个节点里的指针决定的。



链表元素的存储不连续，不需要预先估计容量，也无法随机访问。链表除了存储数据元素外还要额外空间存储下一个结点的地址，存储空间的利用率不如数组。但是相对数组来讲，链表的插入和删除效率非常高。

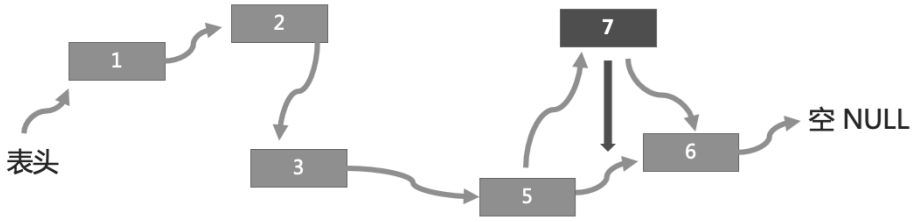
特殊情况下也可以在节点中存储两个指针，一个指向前一个元素，一个指向后一个元素。这种链表叫双向链表。

1.2.2 链表访问

链表的缺点是不能直接访问元素，比如我们想访问链表的第 4 个元素，我们从表头开始，需要先访问 1、2、3 号元素，然后依次推导到第 4 个元素，效率很低。最差的情况下，访问一个 n 个对象的时间为 O(n)。

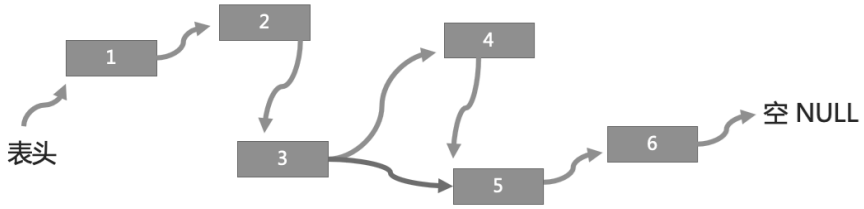
1.2.3 链表插入

链表的插入非常容易，我们可以将节点的地址指向插入位置之后的节点。然后要插入位置上一个元素的指针指向新插入的节点。如果我们想插入一个元素，比如在 5 和 6 之间插入 7 号元素，我们需要将 5 号元素的边指向 7 号元素，然后 7 号元素指向 6 号元素，就实现了元素的插入。



1.2.3 链表删除

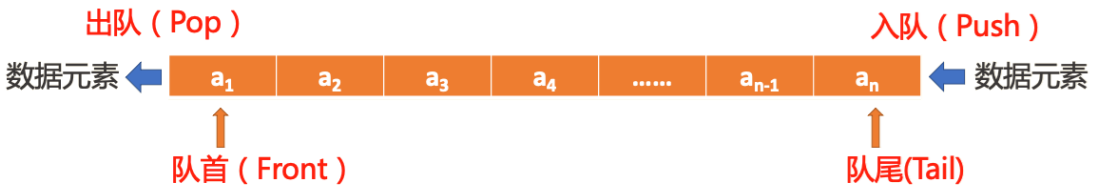
链表的删除也非常容易。我们可以将要删除前一个节点的指针指向要删除元素的后一个节点，然后将节点删除。比如我们需要删除 4 号元素，只需要将 3 号元素的边从指向 4 号改成指向 5 号，我们就不能在链表中找到 4 号元素了。



1.3 队列

1.3.1 队列定义

队列是限定一端插入，另一端进行读取删除的特殊线性结构。队列结构像排队买票，排在前面的先买到离开，新来的人排在队伍末尾。我们也通常把队列的删除和插入叫做出队和入队。出队的一端叫队首(或队头)，入队的一端叫队尾。由于总是先入队的元素先出队，我们也把队列这种结构叫做先进先出（FIFO，First In First Out）表。



1.3.2 队列操作

下面我们主要讲究采用数组 q[n+1]作为队列的底层结构。数组大小就是队列的最大容量。在队列运算时，需要设置两个指针：

head：队头指针，指向实际队头元素所在位置

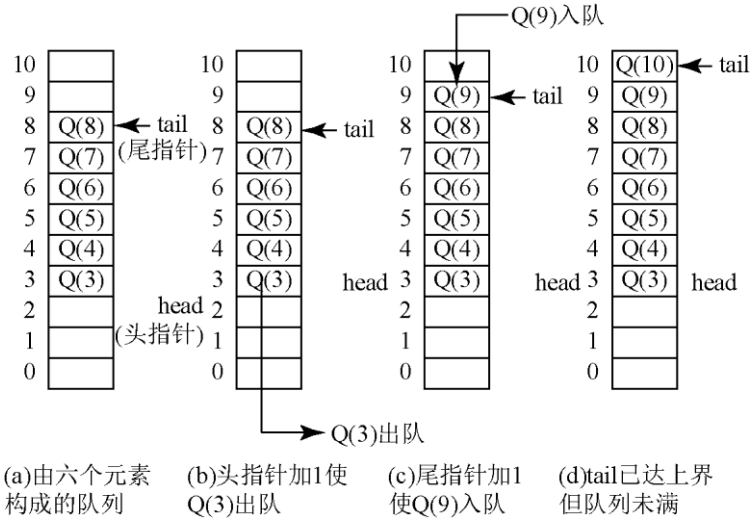
tail：队尾指针，指向实际队尾元素的下一个位置。

一般将两个指针初始值设置为 0，队列为空。队列中的元素个数是 tail - head，当 head 和 tail 相等时队列为空。

队列出队入队要分别移动队首和队尾指针。

出队：x = q[head++];

入队：q[tail++] = x;



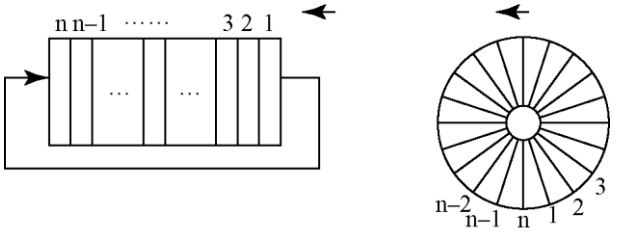
1.3.3 循环队列

设数组长度为 n:

当 front = 0，tail = n 时,再有元素入队则发生溢出，这种溢出成为真溢出

当 front ≠ 0，tail = n 时，再有元素入队则发生溢出，这时候因为队列的前面还有位置，所以称这种溢出为假溢出。

为了解决假溢出问题，我们可以将数组看成一个首尾相接的环形区域。当存放到 n 地址后，下一个地址就翻转为 1。在结构上采用这种技巧存储的队列就称为循环队列。



循环队

对于循环队列，出队和入队操作与普通队列稍有不同:

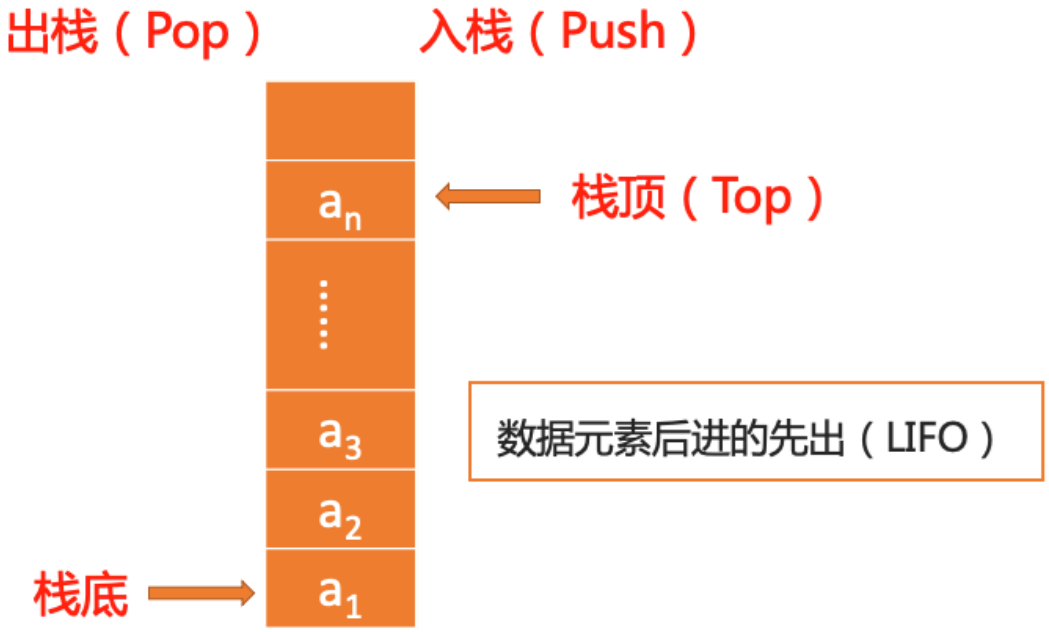
出队：x = q[front]; front = (front + 1) % n;

入队：q[tail] = x; tail = (tail + 1) % n;

1.4 栈

1.4.1 栈定义

栈是只能从一端插入和删除的特殊线性结构。允许插入删除的一端叫做栈顶，因此栈顶是动态变化的，固定的一端叫做栈底。栈的插入操作也被称为压入(push)，删除操作也被称为出栈或弹出(pop)。栈结构的特性可以餐馆里摞盘子的栈。盘子从栈中弹出的次序刚好同他们压入的次序相反。



1.4.2 栈与递归

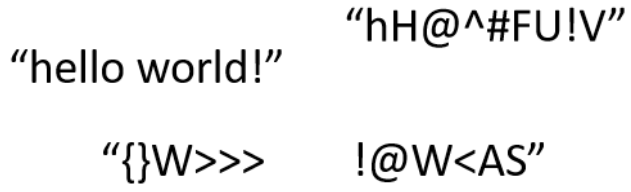
在包括 C++在内的高级语言程序中，函数与被调用函数之间的链接和信息交换必须通过栈进行。当在一个函数的运行期间调用另一个函数时，在运行该被调用函数之前，需要保存所有的参数、返回地址等信息。保存这些信息的结构就是栈。递归函数的调用类似于多层函数的嵌套调用，只是调用单位和被调用单位是同一个函数而已。

递归算法的效率一般较低。对于简单的递归问题，一般用循环结构算法替代。

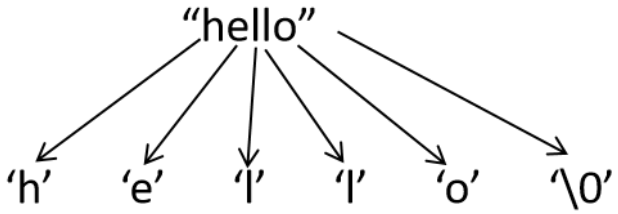
1.5 字符串

1.5.1 字符串相关定义

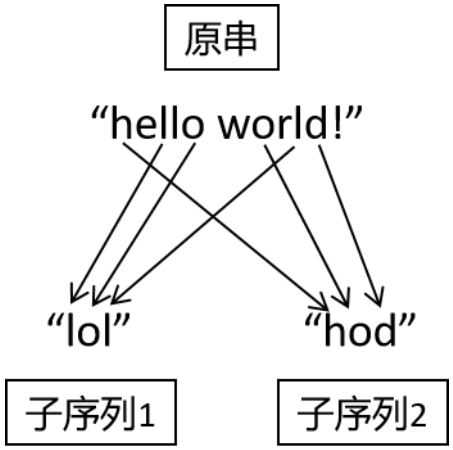
（1）**字符串**：字符串是由数字、字母、下划线等组成的一串字符，它是有顺序性的（从左到右）。



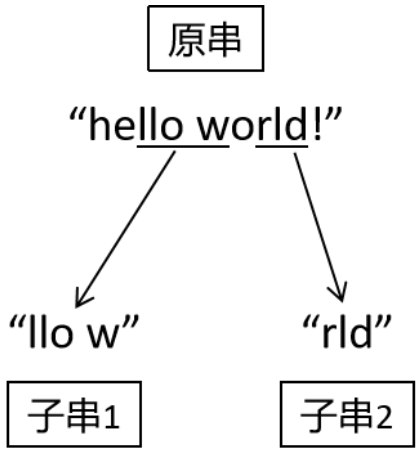
（2）**结尾符**：判断字符串结尾的符号，用 ASCII 码值 0 （写作‘\0’）表示，如果字符串没有结尾符，那么将无法判断字符串何时结束。



（3）**子序列**：通过去除某些元素但不破坏余下元素的相对位置（在前或在后）所而形成的序列。



（4）**子串**：串中任意个连续的字符组成的子序列称为该串的子串。



1.5.2 字符串与字典序

字典序是基于字母顺序排列的单词按字母顺序排列的方法，在字典序的规则下，字符串之间拥有一个大小关系，比较方法如下：

1、 对齐两个字符串 A, B 的起始位置。

2、 从左到右依次考虑每一个对应位置上的字符 A[i], B[i] 的 ACSII 码：

2.1 如果 A[i] < B[i]，那么 A < B。

2.2 如果 A[i] > B[i]，那么 A > B。

2.3 如果 A[i] = B[i]，那么继续考虑下一个位置。

3、如果仍旧没有得到结果，那么 A = B。

1.5.3 字符数组

字符数组起源于 C 语言，并在 C++中继续得到支持。字符串实际上是使用 空字符 '\0' 终止的一维字符数组。因此，一个以空字符结尾的字符串，包含了组成字符串的字符。

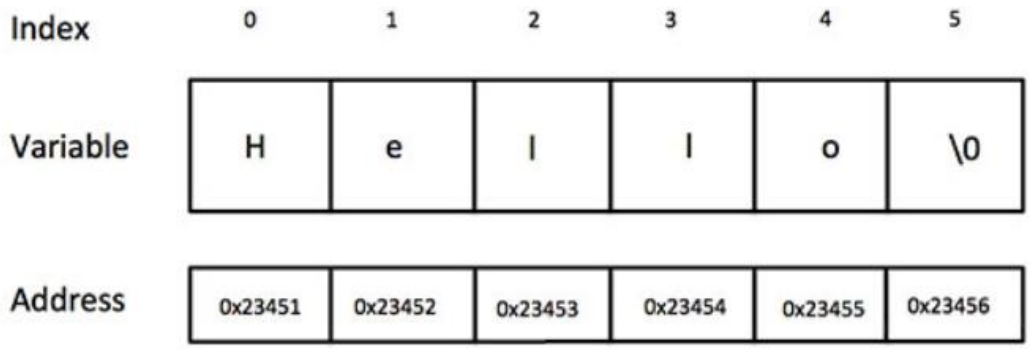
下面的声明和初始化创建了一个 “Hello” 字符串。由于在数组的末尾存储了空字符，所以字符数组的大小比单词 “Hello” 的字符数多一个。

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

其实，并不需要把空字符放在字符串常量的末尾。C++ 编译器会在初始化数组时，自动把 ‘\0’ 放在字符串的末尾。所以也可以利用下面的形式进行初始化：

```
char greeting[] = "Hello";
```

以下是 C/C++ 中定义的字符串的内存表示：



字符数组处理字符串的常用操作

- 字符串输入输出： scanf("%s",s); cin>>s; gets(s); printf("%s",s); cout<<s; puts(s);
- 字符串遍历： for(int i = 0;s[i] != '\0';++i) ...
- 字符串字符修改： s[i] = 'x';

- 字符串拼接： strcat(s1,s2);
- 字符串拷贝： strcpy(s1,s2);
- 字符串子串查找： char *s1 = strstr(s,w);
- 获取字符串长度： int l = strlen(s);
- 字符串比较： if(strcmp(s1,s2) == ?)...

1.5.4 String 类

String 相较于字符数组在处理字符串上功能更加强大方便（当然效率会偏慢），一般在遇到以字符串为单位（不改动字符串内容）或需要更强大灵活的字符串处理功能时，会考虑使用 String 来代替字符数组。

String 类处理字符串的常用操作

- 字符串输入输出： cin>>s; getline(cin,s); cout<<s;
- 字符串遍历： for(int i = 0;s[i] != '\0';++i) ...
- 字符串字符修改： s[i] = 'x';
- 字符串拼接： s = s1 + s2;
- 字符串插入： s1.insert(2,s2);
- 字符串删除： s.erase(2,1);
- 字符串替换： s1.replace(2,3,s2);
- 字符串拷贝： s1 = s2;
- 字符串子串查找： if(s1.find(s2) != s1.npos)...
- 获取字符串长度： int l = s.length();
- 字符串反转： reverse(s.begin(),s.end());
- 字符串比较： if(s1 < s2)...
- 字符串交换： swap(s1,s2);

第2节　非线性数据结构

1．　树

前两章学习的栈和队列属于线性结构。在这种结构中，数据元素的逻辑位置之间呈线性关系，每一个数据元素通常只有一个前件（除第一个元素外）和一个后件（除最后一个元素外）。在实际生活中，可以用线性结构描述数据元素之间逻辑关系的问题是很广泛的，但也有很多问题不能依靠线性结构来解决，例如家谱、行政组织机构等都是非线性的数据结构。其中树就是一种非线性的数据结构。

一、树的定义

一棵树是由 $n(n>0)$ 个元素组成的有限集合，其中：

（1）每个元素称为结点（node）；

（2）有一个特定的结点，称为根结点或树根（root）；

（3）除根结点外，其余结点能分成 $m(m>=0)$ 个互不相交的有限集合 T_0，T_1，T_2，……

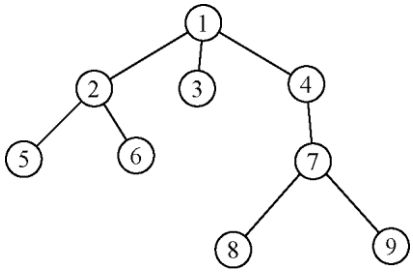
T_{m-1}。其中的每个子集又都是一棵树，这些集合称为这棵树的子树。

如下图是一棵典型的树:

二、树的基本概念

A．树是递归定义的;

B .一棵树中至少有 1 个结点。这个结点就是根结点，它没有前驱，其余每个结点都有唯一的一个前驱结点。每个结点可以有 0 或多个后继结点。因此，树虽然是非线性结构，但也是有序结构。至于前驱后继结点是哪个，还要看树的遍历方法，我们将在后面讨论;



C．一个结点的子树个数，称为这个结点的度（degree，结点 1 的度为 3，结点 3 的度为 0)；度为 0 的结点称为叶结点（树叶 leaf，如结点 3、5、6、8、9)；度不为 0 的结点称为分支结点（如结点 1、2、4、7)；根以外的分支结点又称为内部结点（结点 2、4、7)；树中各结点的度的最大值称为这棵树的度（这棵树的度为 3。

D．在用图形表示的树型结构中，对两个用线段（称为树枝）连接的相关联的结点，称上端结点为下端结点的父结点，称下端结点为上端结点的子结点。称同一个父结点的多个子结点为兄弟结点。如结点 l 是结点 2、3、4 的父结点，结点 2、3、4 是结点 1 的子结点，它们又是兄弟结点，同时结点 2 又是结点 5、6 的父结点。称从根结点到某个子结点所经过的所有结点为这个子结点的祖先。如结点 l、4、7 是结点 8 的祖先。称以某个结点为根的子树中的任一结点都是该结点的子孙。如结点 7、8、9 都是结点 4 的子孙。

E．定义一棵树的根结点的层次（level）为 1，其他结点的层次等于它的父结点层次加 1。如结点 2、3、4 的层次为 2，结点 5、6、7 的层次为 3，结点 8、9 的层次为 4。一棵树中所有的结点的层次的最大值称为树的深度（depth)。如这棵树的深度为 4。

F．对于树中任意两个不同的结点，如果从一个结点出发。自上而下沿着树中连着结点的线段能到达另一结点，称它们之间存在着一条路径。可用路径所经过的结点序列表示路径，路径的长度等于路径上的结点个数减 1。如上图中，结点 1 和结点 8 之间存在着一条路径，并可用（1、4、7、8）表示这条路径，该条路径的长度为 3。注意，不同子树上的结点之间不存在路径，从根结点出发，到树中的其余结点一定存在着一条路径。

G．森林（forest）是 $m(m >= 0)$ 棵互不相交的树的集合。

三、树的遍历

在应用树结构解决问题时，往往要求按照某种次序获得树中全部结点的信息，这种操作叫作树的遍历。遍历的方法有多种，常用的有;

A．先序（根）遍历：先访问根结点，再从左到右按照先序思想遍历各棵子树。

如上图先序遍历的结果为：125634789;

B．后序（根）遍历：先从左到右遍历各棵子树，再访问根结点。

如上图后序遍历的结果为：562389741;

C．层次遍历：接层次从小到大逐个访问，同一层次按照从左到右的次序。

如上图层次遍历的结果为：123456789;

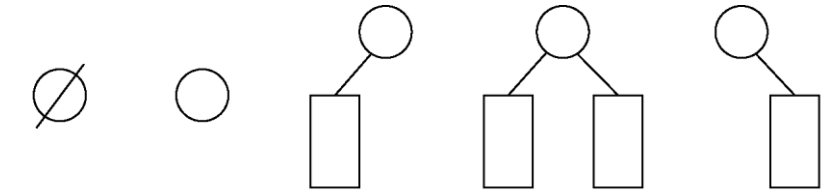
D．叶结点遍历：有时把所有的数据信息都存放在叶结点中，而其余结点都是用来表示

数据之间的某种分支或层次关系，这种情况就用这种方法。

如上图按照这个思想访问的结果为：56389;

四、二叉树基本概念

二叉树（binary tree，简写成 BT）是一种特殊的树型结构，它的度数为 2 的树。即二叉树的每个结点最多有两个子结点。每个结点的子结点分别称为左孩子、右孩子，它的两棵子树分别称为左子树、右子树。二叉树有 5 种基本形态:



前面引入的树的术语也基本适用于二叉树。但二叉树与树也有很多不同，如：首先二叉树的每个结点至多只能有两个子结点，二叉树可以为空，二叉树一定是有序的，通过它的左、右子树关系体现出来。

五、二叉树的性质

【性质 1】在二叉树的第 i 层上最多有 2^{i-1} 个结点 $(i>=1)$ 。

证明：很简单，用归纳法：当 $i=1$ 时，$2^{i-1}=1$ 显然成立；现在假设第 $i-1$ 层时命题成立，即第 $i-1$ 层上最多有 2^{i-2} 个结点。由于二叉树的每个结点的度最多为 2，故在第 i 层上的最大结点数为第 $i-1$ 层的 2 倍，即 $2*2^{i-2}=2^{i-1}$ 。

【性质 2】深度为 k 的二叉树至多有 2^k-1 个结点 $(k>=1)$ 。

证明：在具有相同深度的二叉树中。仅当每一层都含有最大结点数时，其树中结点数最多。因此利用性质 1 可得，深度为 k 的二叉树的结点数至多为:

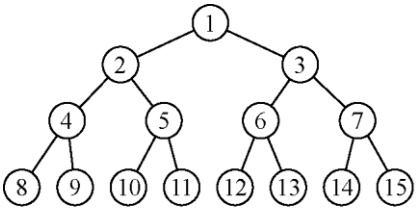
$$2^0+2^1+\cdots+2^{k-1}=2^k-1$$

故命题正确。

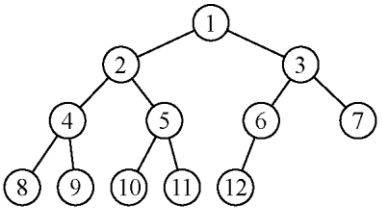
特别：一棵深度为 k 且有 2^k-1 个结点的二叉树称为满二叉树。如下图 A 为深度为 4 的满二叉树，这种树的特点是每层上的结点数都是最大结点数。

可以对满二叉树的结点进行连续编号，约定编号从根结点起，自上而下，从左到右，由此引出完全二叉树的定义，深度为 k，有 n 个结点的二叉树当且仅当其每一个结点都与深度为 k 的满二叉树中编号从 1 到 n 的结点一一对应时，称为完全二叉树。，

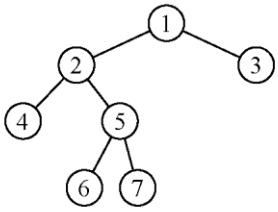
下图 B 就是一个深度为 4，结点数为 12 的完全二叉树。它有如下特征：叶结点只可能在层次最大的两层上出现；对任一结点，若其右分支下的子孙的最大层次为 m，则在其左分支下的子孙的最大层次必为 m 或 m+1。下图 C、D 不是完全二叉树，请大家思考为什么?



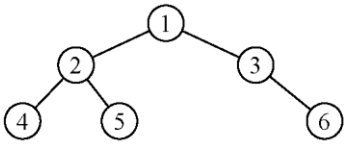
图A



图B



图C



图D

【性质 3】对任意一棵二叉树，如果其叶结点数为 n_0，度为 2 的结点数为 n_2，则一定满足：$n_0 = n_2 + 1$。

证明：因为二叉树中所有结点的度数均不大于 2，所以结点总数（记为 n）应等于 0 度结点数 n_0、1度结点 n_1 和 2 度结点数 n_2 之和：

$$n = n_0 + n_1 + n_2 \cdots\cdots \text{（式子 1）}$$

另一方面，1 度结点有一个孩子，2 度结点有两个孩子，故二叉树中孩子结点总数是：

$$n_1 + 2n_2$$

树中只有根结点不是任何结点的孩子，故二叉树中的结点总数又可表示为：

$$n = n_1 + 2n_2 + 1 \cdots\cdots \text{（式子 2）}$$

由式子 1 和式子 2 得到：

$$n_0 = n_2 + 1$$

【性质 4】具有 n 个结点的完全二叉树的深度为 $\text{floor}(\log_2^n) + 1$

证明：假设深度为 k，根据完全二叉树的定义，前面 $k-1$ 层一定是满的，所以 $n > 2^{k-1} - 1$。但 n 又要满足 $n <= 2^k - 1$。所以，$2^{k-1} - 1 < n <= 2^k - 1$。变换一下为 $2^{k-1} <= n < 2^k$。

以 2 为底取对数得到：$k - 1 <= \log_2^n < k$。而 k 是整数，所以 $k = \text{floor}(\log_2^n) + 1$。

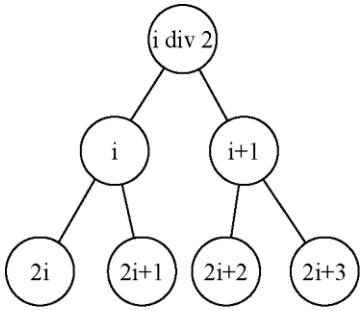
【性质 5】对于一棵 n 个结点的完全二叉树。对任一个结点（编号为 i），有：

①如果 $i = 1$，则结点 i 为根。无父结点；如果 $i > 1$，则其父结点编号为 $i/2$。

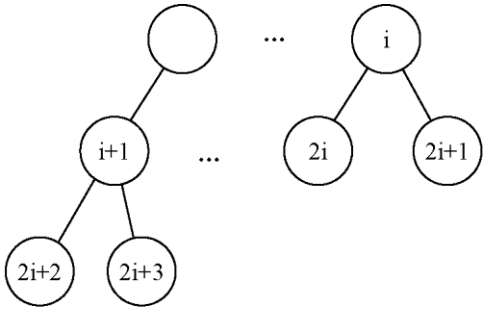
如果 $2 * i > n$，则结点 i 无左孩子（当然也无右孩子，为什么？即结点 i 为叶结点）；否则左孩子编号为 $2 * i$。

②如果 $2 * i + 1 > n$，则结点 i 无右孩子；否则右孩子编号为 $2 * i + 1$。

证明：略，我们只要验证一下即可。总结如下图所示：



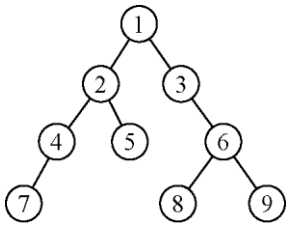
结点i和i+1在同一层上



结点i和i+1不在同一层上

六、遍历二叉树

在二叉树的应用中，常常要求在树中查找具有某种特征的结点，或者对全部结点逐一进行某种处理。这就是二叉树的遍历问题。所谓二叉树的遍历是指按一定的规律和次序访问树中的各个结点，而且每个结点仅被访问一次。“访问”的含义很广，可以是对结点作各种处理，如输出结点的信息等。遍历一般按照从左到右的顺序，共有 3 种遍历方法：先（根）序遍历，中（根）序遍历，后（根）序遍历。



（一）先序遍历的操作定义如下：

若二叉树为空，则空操作，否则

①访问根结点

②先序遍历左子树

③先序遍历右子树

先序遍历右图二叉树的结果为：124753689

（二）中序遍历的操作定义如下：

若二叉树为空，则空操作，否则

①中序遍历左子树

②访问根结点

③中序遍历右子树

中序遍历上图二叉树的结果为：742513869

（三）后序遍历的操作定义如下：

若二叉树为空，则空操作，否则

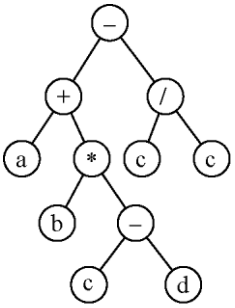
①后序遍历左子树

②后序遍历右子树

③访问根结点

后序遍历上图结果为：745289631

关于前面讲的表达式树，我们可以分别用先序、中序、后序的遍历方法得出完全不同的遍历结果，如对于右图二叉树的遍历结果如下，它们正好对应着表达式的 3 种表示方法。



表达式(a+b*(c–d)–e/f)的二叉树

$-+a*b-cd/ef$ （前缀表示、波兰式）

$a+b*c-d-e/f$ （中缀表示）

$abcd-*+ef/-$ （后缀表示、逆波兰式）

结论：已知前序序列和中序序列可以确定出二叉树；

已知中序序列和后序序列也可以确定出二叉树；

但，已知前序序列和后序序列却不可以确定出二叉树；为什么？

2.图

图（Graph）是一种复杂的非线性结构。在人工智能、工程、数学、物理、化学、生物和计算机科学等领域中，图结构有着广泛的应用。奥林匹克信息学竞赛的许多试题，亦需要用图来描述数据元素间的联系。

图 G 由两个集合 V 和 E 组成，记为：$G=(V，E)$，其中：V 是顶点的有穷非空集合，E 是 V 中顶点偶对（称为边）的有穷集。通常，也将图 G 的顶点集和边集分别记为 V（G）和 E（G）。E（G）可以是空集。若 E（G）为空，则图 G 只有顶点而没有边。

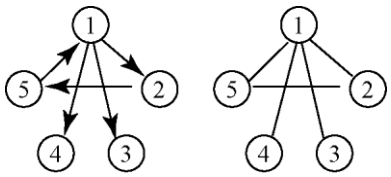
一、什么是图?

很简单，点用边连起来就叫做图，严格意义上讲，图是一种数据结构，定义为：graph=（V，E）。V 是一个非空有限集合，代表顶点（结点），E 代表边的集合。

二、图的一些定义和概念

1．有向图：图的边有方向，只能按箭头方向从一点到另一点。(a）就是一个有向图。

2．无向图：图的边没有方向，可以双向。(b）就是一个无向图。



3．结点的度：无向图中与结点相连的边的数目，称为结点的度。

4．结点的入度：在有向图中，以这个结点为终点的有向边的数目。

5．结点的出度：在有向图中，以这个结点为起点的有向边的数目。

6．权值：边的“费用”，可以形象地理解为边的长度。

7．连通：如果图中结点 U，V 之间存在一条从 U 通过若干条边、点到达 V 的通路，则称 U、V 是连通的。

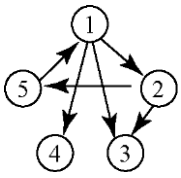
8．回路：起点和终点相同的路径，称为回路，或“环”。

9．完全图：一个 n 阶的完全无向图含有 $n*(n-1)/2$ 条边；一个 n 阶的完全有向图含有 $n*(n-1)$ 条边；

稠密图：一个边数接近完全图的图。

稀疏图：一个边数远远少于完全图的图。

10．强连通分量：有向图中任意两点都连通的最大子图。右图中，1—2—5 构成一个强连通分量。特殊地，单个点也算一个强连通分量，所以右图有三个强连通分量：l—2—5，4，3。



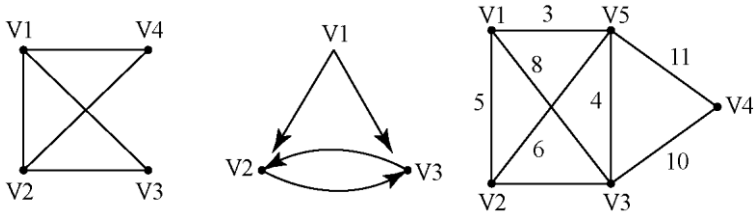
三、图的存储结构

二维数组邻接矩阵存储

定义 int　G[101][101];

G[i][j]的值，表示从点 i 到点 j 的边的权值，定义如下：

$$G[i][j]\begin{cases}1\text{或权值} & \text{当}v_i\text{与}v_j\text{之间有边或弧时，取值为1或权值}\\ 0\text{或}\infty & \text{当}v_i\text{与}v_j\text{之间无边或弧时，取值为0或}\infty\text{（无穷大）}\end{cases}$$



上图中的 3 个图对应的邻接矩阵分别如下：

$$G(A)=\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&0\\1&1&0&0\end{bmatrix}\quad G(B)=\begin{bmatrix}0&1&1\\0&0&1\\0&1&0\end{bmatrix}\quad G(C)=\begin{bmatrix}\infty&5&8&\infty&3\\5&\infty&2&\infty&6\\8&2&\infty&10&4\\\infty&\infty&10&\infty&11\\3&6&4&11&\infty\end{bmatrix}$$

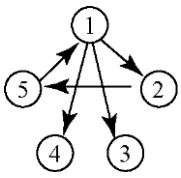
四、深度优先与广度优先遍历

从图中某一顶点出发系统地访问图中所有顶点，使每个顶点恰好被访问一次，这种运算操作被称为图的遍历。为了避免重复访问某个顶点，可以设一个标志数组 visited[i]，未访问时值为 false，访问一次后就改为 true。

图的遍历分为深度优先遍历和广度优先遍历两种方法，两者的时间效率都是 O（n*n）。

1．深度优先遍历

深度优先遍历与深搜 dfs 相似，从一个点 A 出发，将这个点标为已访问 visited[i]=true，然后再访问所有与之相连，且未被访问过的点。当 A 的所有邻接点都被访问过后，再退回到 A 的上一个点（假设是 B），再从 B 的另一个未被访问的邻接点出发，继续遍历。



例如对右边的这个有向图深度优先遍历，假定先从 1 出发，程序以如下顺序遍历：

$1\to 2\to 5$，然后退回到 2，退回到 1。

从 l 开始再访问未被访问过的点 3，3 没有未访问的邻接点，退回 1。

再从 1 开始访问未被访问过的点 4，再退回 1。

起点 l 的所有邻接点都已访问，遍历结束。

2．广度优先遍历

广度优先遍历并不常用，从编程复杂度的角度考虑，通常采用的是深度优先遍历。

广度优先遍历和广搜 bfs 相似。因此使用广度优先遍历一张图并不需要掌握什么新的知识，在原有的广度优先搜索的基础上，做一点小小的修改，就成了广度优先遍历算法。

五、一笔画问题

如果一个图存在一笔画，则一笔画的路径叫做欧拉路。如果最后又回到起点，那这个路径叫做欧拉回路。

我们定义奇点是指跟这个点相连的边数目有奇数个的点。对于能够一笔画的图，我们有以下两个定理。

定理 1：存在欧拉路的条件：图是连通的，有且只有 2 个奇点。

定理 2：存在欧拉回路的条件：图是连通的，有 0 个奇点。

两个定理的正确性是显而易见的，既然每条边都要经过一次，那么对于欧拉路，除了起点和终点外，每个点如果进入了一次，显然一定要出去一次，显然是偶点。对于欧拉回路。每个点进入和出去次数一定都是相等的，显然没有奇点。

求欧拉路的算法很简单，使用深度优先遍历即可。

根据一笔画的两个定理，如果寻找欧拉回路，对任意一个点执行深度优先遍历；找欧拉路，则对一个奇点执行 dfs，时间复杂度为 O（m+n），m 为边数，n 是点数。

第四章 组合数学

一、加法原理与乘法原理

1．分类计数原理：做一件事，完成它有 n 类办法，在第一类办法中有 m_1 种不同的方法，在第二类办法中有 m_2 种方法，……，在第 n 类办法中有 m_n 种不同的方法．那么完成这件事共有 $N=m_1+m_2+\cdots+m_n$ 种不同的方法．又称加法原理．

2．分步计数原理：做一件事，完成它需要分成 n 个子步骤，做第一个步骤有 m_1 种不同的方法，做第二个步骤有 m_2 种不同方法，……，做第 n 个步骤有 m_n 种不同的方法．那么完成这件事共有 $N=m_1\times m_2\times\cdots\times m_n$ 种不同的方法．又称乘法原理．

二、排列组合的基本概念

1．排列：一般地，从 n 个不同的元素中任取 $m(m\leqslant n)$ 个元素，按照一定的顺序排成一列，叫做从 n 个不同元素中取出 m 个元素的一个排列．（其中被取的对象叫做元素）

排列数：从 n 个不同的元素中取出 $m(m\leqslant n)$ 个元素的所有排列的个数，叫做从 n 个不同元素中取出 m 个元素的排列数，用符号 A_n^m 表示．

排列数公式：$\mathrm{A}_n^m=n(n-1)(n-2)\cdots(n-m+1)$，$m$，$n\in\mathbf{N}_+$，并且 $m\leqslant n$．

全排列：一般地，n 个不同元素全部取出的一个排列，叫做 n 个不同元素的一个全排列．

n 的阶乘：正整数由 1 到 n 的连乘积，叫作 n 的阶乘，用 $n!$ 表示．规定：$0!=1$．

2．组合：一般地，从 n 个不同元素中，任意取出 m $(m\leqslant n)$ 个元素并成一组，叫做从 n 个元素中任取 m 个元素的一个组合．

组合数：从 n 个不同元素中，任意取出 m $(m\leqslant n)$ 个元素的所有组合的个数，叫做从 n 个不同元素中，任意取出 m 个元素的组合数，用符号 C_n^m 表示．

组合数公式：$\mathrm{C}_n^m=\dfrac{n(n-1)(n-2)\cdots(n-m+1)}{m!}=\dfrac{n!}{m!(n-m)!}$，$m$，$n\in\mathbf{N}_+$，并且 $m\leqslant n$．

组合数的两个性质：性质 1：$\mathrm{C}_n^m=\mathrm{C}_n^{n-m}$；性质 2：$\mathrm{C}_{n+1}^m=\mathrm{C}_n^m+\mathrm{C}_n^{m-1}$．（规定 $\mathrm{C}_n^0=1$）

【例1】 ⑴ 现有 3 辆公交车、3 位司机和 3 位售票员，每辆车上需配 1 位司机和 1 位售票员，问车

辆、司机、售票员搭配方案一共有多少种?

⑵（2008 天津 16）

有 4 张分别标有数字 1，2，3，4 的红色卡片和 4 张分别标有数字 1，2，3，4 的蓝色卡片，从这 8 张卡片中取出 4 张卡片排成一行．如果取出的 4 张卡片所标数字之和等于 10，则不同的排法共有_____种（用数字作答）．

⑶ 在 $\angle AOB$ 的边 OA 上有 A_1，A_2，A_3，A_4 四点，OB 边上有 B_1，B_2，B_3，B_4，B_5 共 9 个点，连

结线段 $A_iB_j(1\leqslant i\leqslant 4,1\leqslant j\leqslant 5)$，如果其中两条线段不相交，则称之为一对“和睦线”，和睦线的对数共有：（　）

A．60　　B．80　　C．120　　D．160

⑷ 某幢楼从二楼到三楼的楼梯共 11 级，上楼可以一步上一级，也可以一步上两级，若规

定从二楼到三楼用 7 步走完，则上楼梯的方法有_____种．

⑸（重庆市西南师大附中 2011 年 3 月高三第六次月考理科）

25 人排成 5×5 的方阵，从中选出 3 人，要求任意 2 人既不同行也不同列，则

不同的选法有（　　）

A．60种　　B．100种　　C．300种　　D．600种

【解析】⑴ 分两步完成，第一步，把3名司机安排到3辆车中，有$A_3^3=6$种安排方法；

第二步把3名售票员安排到3辆车中，有$A_3^3=6$种排法．故搭配方案共有$A_3^3\cdot A_3^3=36$种．

点评：许多复杂的排列问题，不可能一步就能完成，而应分解开来考虑：即经适当地分类或分步之后，应用分类计数原理、分步计数原理去解决．在分类或分步时，要尽量把整个事件的安排过程考虑清楚，防止分类或分步的混乱．

⑵ 432；

数字之和为10的情况有4，4，1，1、3，3，2，2、4，3，2，1．

所以共有$A_4^4+A_4^4+2^4A_4^4=18A_4^4=432$种不同排法（先选后排）．

⑶ A；

OA上任意两点与OB上任两点恰好确定一对和睦线，共$C_4^2C_5^2=60$对，选A．

⑷ 35；

从二楼到三楼用7步走完，共走11级，则必有4步每步走两级，其余3步每步1级，因此共有$C_7^4=35$种方法．

⑸ D；

从5行5列中选3行3列，不同行也不同列，$C_5^3A_5^3=600$

三、排列组合的经典方法

解排列组合问题，首先要用好两个计数原理和排列组合的定义，即首先弄清是分类还是分步，是排列还是组合，同时要掌握一些常见类型的排列组合问题的解法：

①特殊元素、特殊位置优先法

元素优先法：先考虑有限制条件的元素的要求，再考虑其他元素；

位置优先法：先考虑有限制条件的位置的要求，再考虑其他位置；

②排除法：从总体中排除不符合条件的方法数，这是一种间接解题的方法．

③捆绑法：某些元素必相邻的排列，可以先将相邻的元素“捆成一个”元素，与其它元素进行排列，然后再给那“一捆元素”内部排列．

④插空法：某些元素不相邻的排列，可以先排其它元素，再让不相邻的元素插空．

⑤隔板法：n个相同元素，分成$m(m\leqslant n)$组，每组至少一个的分组问题——把n个元素排成一排，从$n-1$个空中选$m-1$个空，各插一个隔板，有C_{n-1}^{m-1}．

对于较复杂的排列组合问题，常需要分类讨论或分步计算，一定要做到分类明确，层次清楚，不重不漏．对于抽出部分元素进行排列的问题一般是先选后排，以防出现重复．分几排的问题可以转化为直排问题处理．

【例2】 6个人站成一排：

⑴ 其中甲、乙两人必须相邻有多少种不同的排法？

⑵ 其中甲、乙两人不相邻有多少种不同的排法？

⑶ 其中甲、乙两人不站排头和排尾有多少种不同的排法？

⑷ 其中甲不站排头，且乙不站排尾有多少种不同的排法？

【解析】⑴（捆绑法）

因为甲、乙两人必须相邻，可视甲、乙在一起为一个元素与其他4人有A_5^5种排法，而甲、乙又有A_2^2种排法，根据分步计数原理共有$A_2^2A_5^5=240$种排法．

⑵（插空法）

甲、乙两人外的其余4人有A_4^4种排法，要使甲、乙不相邻只有排在他们的空档位置，有A_5^2种排法，所以共有$A_4^4A_5^2=480$种排法；

（间接法）用总的排法减去相邻的排法，即$A_6^6-A_2^2A_5^5=480$种排法.

⑶（位置分析法）

甲、乙两人不站排头和排尾，则这两个位置可从其余4人中选2人来站有A_4^2种排法，剩下的4人有A_4^4种排法，共有$A_4^2A_4^4=288$种排法；

（元素分析法）

甲、乙两人不站排头和排尾，故可以用中间四个位置中选2个站甲、乙，有A_4^2种排法，其它4人站在余下的4个位置上，有A_4^4种排法，共有$A_4^2A_4^4=288$种排法；

（间接法）

六人全排有A_6^6种排法，除去甲站排头、甲站排尾、乙站排头和乙站排尾的$4A_5^5$种，补上重复减去的甲、乙都在排头排尾的$A_2^2A_4^4$种排法，共有排法$A_6^6-4A_5^5+A_2^2A_4^4=288$种.

⑷ 甲站排头有A_5^5种排法，乙站排尾有A_5^5种排法，但两种情况都包含了“甲站排头，乙站排尾”的情况，有A_4^4种排法，故共有$A_6^6-2A_5^5+A_4^4=504$种排法.

【例3】 ⑴ 用数字2，3组成四位数，且数字2，3至少都出现一次，这样的四位数共有_____个（用数字作答）

⑵将甲、乙、丙、丁四名学生分到三个不同的班，每个班至少分到一名学生，且甲、乙两名学生不能分到同一个班，则不同分法的种数为（　　）

A．18　　B．24　　C．30　　D．36

【解析】排除法（间接法）

⑴ 个数为$2^4-2=14$.

⑵ C；

用间接法解答：四名学生中选两名学生分在一个班的种数是C_4^2，顺序有A_3^3种，而甲乙被分在同一个班的有A_3^3种，所以种数是$C_4^2A_3^3-A_3^3=30$.

【例4】 ⑴从6人中选4人分别到巴黎、伦敦、悉尼、莫斯科四个城市游览，要求每个城市有一人游览，每人只游览一个城市，且这6人中，甲、乙两人不去巴黎游览，则不同的选择方案共有____种（用数字作答）.

⑵ 给定数字0、1、2、3、5、9，每个数字最多用一次，

①可能组成多少个四位数?　　②可能组成多少个四位奇数?

③可能组成多少个四位偶数?　　④可能组成多少个自然数?

【解析】特殊位置优先法.

⑴ 因为甲、乙不去巴黎，故从其余4人选1人去巴黎有C_4^1种方法，再从剩余5人中选3

人去其余3市，有A_5^3种方法，所以共有方案$C_4^1A_5^3=240$种，故选B.

⑵ 注意0不能放在首位，还要注意个位数字，方法多种多样，利用特殊优先法，即特殊的元素，特殊的位置优先考虑.

①法一（位置分析法）

从“位置”考虑，由于0不能放在首位，因此首位数字只能有A_5^1种取法，其余3个数位可以从余下的5个数字（包括0）中任取3个排列，所以可以组成

$A_5^1A_5^3=300$ 个四位数；

法二（元素分析法）

从“元素”考虑，组成的四位数可以按有无数字 0 分成两类，有数字 0 的先排 0 的位置，有 $A_3^1A_5^3$ 个，无数字 0 的有 A_5^4 个，所以共组成 $A_3^1A_5^3+A_5^4=300$ 个四位数；

法三（排除法）

从 6 个元素中取 4 个元素的所有排列中，减去 0 在首位上的排列数即为所求，所以共有 $A_6^4-A_1^1A_5^3=300$ 个四位数；

②个位数字必须是奇数有 A_4^1 种排法，由于 0 不能放在首位，因此首位数字只能有 A_4^1 种取法，其余两个数位的排法有 A_4^2，所以共有 $A_4^1A_4^1A_4^2=192$ 个四位奇数；

③法一：由(1)(2)知共有 $300-192=108$ 个四位偶数；

法二：从“位置”考虑，按个位数字是否为 0 分成两种情况，0 在个位时，有 $A_1^1A_5^3$ 个四位偶数；2 在个位时，有 $A_1^1A_4^1A_4^2$ 个四位偶数，所以共有 $A_1^1A_5^3+A_1^1A_4^1A_4^2=108$ 个四位偶数；

④一位数：有 $A_6^1=6$ 个；两位数：有 $A_5^1A_5^1=25$ 个；三位数：有 $A_5^1A_5^2=100$ 个；四位数：有 $A_5^1A_5^3=300$ 个；五位数：有 $A_5^1A_5^4=600$ 个；六位数：有 $A_5^1A_5^5=600$ 个；

所以共有 $6+25+100+300+600+600=1631$ 个自然数.

点评：解有条件限制的排列问题思路：①正确选择原理；②处理好特殊元素和特殊位置，先让特殊元素占位，或特殊位置选元素；③再考虑其余元素或其余位置；④数字的排列问题，0 不能排在首位.

【例5】 (1)三对夫妇去上海世博会参观，在中国馆前拍照留念，6 人排成一排，每对夫妇必须相邻，不同的排法种数为（　）

a)

A．6　　B．24　　C．48　　D．72

(2) 7 名同学排队照相.

①若分成两排照，前排 3 人，后排 4 人，有多少种不同的排法?

②若排成两排照，前排 3 人，后排 4 人，但其中甲必须在前排，乙必须在后排，有多少种不同的排法?

③若排成一排照，甲、乙、丙三人必须相邻，有多少种不同的排法?

【解析】(1) C；

捆绑法，$3!\times2\times2\times2=48$

(2) 分析：①可分两步完成；第一步，从 7 人中选出 3 人排在前排，有 A_7^3 种排法；第二步，

剩下的 4 人排在后排，有 A_4^4 种排法，故一共有 $A_7^3\cdot A_4^4=A_7^7$ 种排法．事实上排两排与排成一排一样，只不过把第 4～7 个位子看成第二排而已，排法总数都是 A_7^7，相当于 7 个人的全排列.

① $A_7^3\cdot A_4^4=A_7^7=5040$ 种.

②第一步安排甲，有 A_3^1 种排法；第二步安排乙，有 A_4^1 种排法；第三步余下的 5 人排在剩下的 5 个位置上，有 A_5^5 种排法，由分步计数原理得，符合要求的排法共有 $A_3^1\cdot A_4^1\cdot A_5^5=1440$ 种.

③第一步，将甲、乙、丙视为一个元素，和其余 4 个元素排成一排，即看成 5 个元素的全排列问题，有 A_5^5 种排法；第二步，甲、乙、丙三人内部全排列，有 A_3^3 种排法．由分步计数原理得，共有 $A_5^5\cdot A_3^3=720$ 种排法.

【例6】 ⑴用1到8组成没有重复数字的八位数，要求1与2相邻，3与4相邻，5与6相邻，而7与8不相邻，这样的八位数共有____个（用数字作答）.

⑵ 马路上有编号为l，2，3，……，10 十个路灯，为节约用电又看清路面，可以把其中的三只灯关掉，但不能同时关掉相邻的两只或三只，在两端的灯也不能关掉的情况下，求满足条件的关灯方法共有____种．（用数字作答）

⑶在1，2，3，4，5，6，7的任一排列a_1，a_2，a_3，a_4，a_5，a_6，a_7中，使相邻两数都互质的排列方式种数共有（　　）

A．576　　B．720　　C．864　　D．1152

【解析】⑴ 此题是捆绑法和插空法的应用问题.

把相邻的两个数捆成一捆，分成四个空，然后再将7与8插进空中有A_4^2种插法；而相邻的三捆都有A_2^2种排法，在它们之间又有A_3^3种排序方法．故这样的八位数共有：$A_2^2A_2^2A_2^2A_3^3A_4^2=576$（个）.

⑵ 关掉的灯不能相邻，也不能在两端．又因为灯之间没有区别，因而问题为在7盏亮着的

灯形成的不包含两端的6个空中选出3个空放置熄灭的灯．有$C_6^3=20$种.

⑶ C；插空法

先让数字1，3，5，7作全排列，有A_4^4种，再排数字6，由于数字6不与3相邻，在排好的排列中，除3的左、右2个空隙，还有3个空隙可排数字6，故数字6有3种排法，最后排数字2，4，在剩下的4个空隙中排上2，4，有A_4^2种排法，共有$A_4^4\times3\times A_4^2=864$种.