第一轮认证知识点

试题形式

- 选择题：共15题，每题2分，共计30分。
- 阅读程序：3个大题，每题有若干个判断题和选择题，判断题1.5分，选择题3分，共计40分。题目给出一段程序，根据程序理解，解答对应的问题。
- 完善程序：2个大题，每题有若干个选择题，选择题3分，共计30分。题目给出一段关于程序功能的文字说明，然后给出一段程序代码，在代码中略去了若干个语句或语句的一部分并在这些位置给出空格，要求考生根据程序的功能说明和代码的上下文，选择对应的语句。

二进制及其运算

二进制及其运算

- 十进制数
  - 计算特点：逢十进一
- 60进制：时分秒的换算
- 360进制：1周=360度
- 12进制：1打
- 二进制

二进制

- 二进制是计算机技术中广泛采用的一种数制。
- 只有“0”和“1”两个数码。
- 它的基数为2，进位规则是“逢二进一”，借位规则是“借一当二”。
- 二进制的加法：
  - 1+1=10
  - 10+1=11
  - 11+1=100
  - 1011+11=?
- 进制数的表示方法：
  - 用一个下表来表明——$(10)_{10}$、$(10)_{2}$、$(10)_{16}$

十进制转换为二进制

- 整数部分：除以二取余法

  89/2=44……1

  44/2=22……0

  22/2=11……0

  11/2=5……1

  5/2=2……1

  2/2=1……0

  1/2=0……1

- $(89)_{10}=(1011001)_2$

十进制转换为二进制

- 小数部分：乘以二取整法

  0.65*2=1.3　取1

  0.3*2=0.6　取0

  0.6*2=1.2　取1

  0.2*2=0.4　取0

  0.4*2=0.8　取0

  0.8*2=1.6　取1

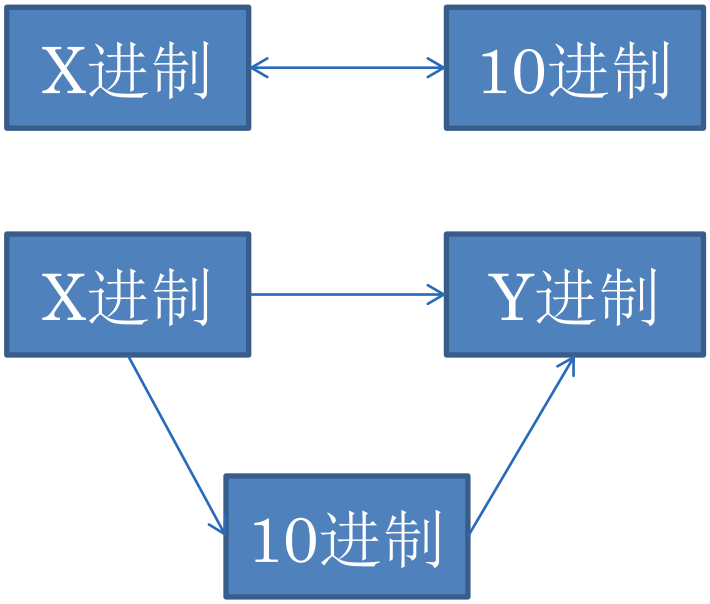
  0.6*2=1.2　取1

- $(0.65)_{10}=(0.1010011\ldots)_2$

二进制转十进制

- 按位权展开
- $(111.01)_{10}=(1*10^{2})+(1*10^{1})+(1*10^{0})+(0*10^{-1})+(1*10^{-2})$
- $(111.01)_{2}=(1*2^{2})+(1*2^{1})+(1*2^{0})+(0*2^{-1})+(1*2^{-2})$

数制的转换



数制的转换

- 二进制数转换成八进制数采用“三位一并法”，即从二进制数的小数点开始向两个方向以三位二进制数字分组，不足以零补足，用它的八进制等值代替这样的组。
- $(10110.0011)_2$

$=(010\ 110\ .\ 001\ 100)_2$

$=(26.14)_8$

- 反之亦然，八进制数转换成二进制数，即把每一个八进制数转换成3位的二进制数，就得到一个二进制数。
- $(37.416)_8$

$=(011\ 111\ .\ 100\ 001\ 110)_2$

数制的转换

- 二进制数转换成十六进制数类似于二进制与八进制的转换，但采用“四位一并法”。二进制数每4位对应1位十六进制数。
- $(01011110.10110010)_2$

$=(0101\ 1110\ .\ 1011\ 0010)_2$

$=(5E.B2)_{16}$

- 反之亦然，十六进制数转换成二进制数，即把每一个十六进制数转换成4位的二进制数，就得到一个二进制数。
- $(5DF)_{16}$

$=(0101\ 1101\ 1111)_2$

逻辑运算

- 运算：
  - 与：　and　∧　·
  - 或：　or　∨　+
  - 非：　not　¬
  - 异或：　xor
- 运算的优先级：非>与>或

“与”运算（“·”，“∧”，AND）

- 在逻辑问题中，如果决定某一事件发生的多个条件必须同时具备，事件才能发生，则这种因果关系称之“与”逻辑（并且）。
- “与”运算的运算法则为：
  - 0·0=0
  - 0·1=0
  - 1·0=0
  - 1·1=1

“或”运算（“+”，“V”，OR）

- 在逻辑问题中，如果决定某一事件是否发生的多个条件中，只要有一个或一个以上条件成立，事件便可发生，则这种因果关系称之为“或”逻辑。
- “与”运算的运算法则为：
  - 0·0=0
  - 0·1=1
  - 1·0=1
  - 1·1=1

“非”运算（“¬”）

- 在逻辑问题中，如果某一事件的发生取决于条件的否定，即事件与事件发生的条件之间构成矛盾，则这种因果关系称之为“非”逻辑。
- “非”运算的逻辑关系可表示为$F = \bar{A}$或者$F = \neg A$
- “非”运算的运算法则为：
  - $\bar{0} = 1$
  - $\bar{1} = 0$

例题

- 十进制小数13.375对应的二进制数是（　）。
  - A. 1101.011
  - B. 1011.011
  - C. 1101.101
  - D. 1010.01

例题

- 逻辑表达式（　）的值与变量A的真假无关。
  - A. (A∨B)∧¬A
  - B. (A∨B)∧¬B
  - C. (A∧B) ∨(¬A∧B)
  - D. (A∨B) ∧ ¬A ∧ B

例题

- 下列四个不同进制的数中，与其他三项数值上不相等的是（　）。
  - A. $(269)_{16}$
  - B. $(617)_{10}$
  - C. $(1151)_{8}$
  - D. $(1001101011)_{2}$

例题

- 二进制数11 1011 1001 0111和01 0110 1110 1011进行逻辑与运算的结果是（　）。
  A. 01 0010 1000 1011
  B. 01 0010 1001 0011
  C. 01 0010 1000 0001
  D. 01 0010 1000 0011

例题

- 设x=true，y=true，z=false，以下逻辑运算表达式值为真的是（　）。
  - A. (x∧y)∧z
  - B. x∧(z∨y)∧z
  - C. (x∧y)∨(z∨x)
  - D. (y∨z)∧x∧z

例题

- 二进制数1011转换成十进制数是（　）。
  - A. 10
  - B. 13
  - C. 11
  - D. 12

例题

- 二进制数101.11对应的十进制数是（　）。
  - A. 6.5
  - B. 5.5
  - C. 5.75
  - D. 5.25

例题

- 八进制数32.1对应的十进制数是（　）。
  - A. 24.125
  - B. 24.250
  - C. 26.125
  - D. 26.250

信息存储

信息存储单位

- 位（bit，缩写为b）：度量数据的最小单位，表示一位二进制信息。
- 字节（byte，缩写为B）：一个字节由八位二进制数字组成（1byte=8bit）。字节是信息存储中最常用的基本单位。
- 常用的单位有：
  - KB　　1KB=1024B
  - MB　　1MB=1024KB
  - GB　　1GB=1024MB
  - TB　　1TB=1024GB

例题

- 计算机存储数据的基本单位是（　）。
  A. bit
  B. Byte
  C. GB
  D. KB

例题

- 分辨率为800×600、16位色的位图，存储图像信息所需的空间为（　）。
  - A. 937.5KB
  - B. 4218.75KB
  - C. 4320KB
  - D. 2880KB

例题

- 分辨率为1600×900、16位色的位图，存储图像信息所需的空间为（　）。
  - A. 2812.5KB
  - B. 4218.75KB
  - C. 4320KB
  - D. 2880KB

例题

- 1MB等于（　）。
  - A. 1000字节
  - B. 1024字节
  - C. 1000×1000字节
  - D. 1024×1024字节

例题

- 一个32位整型变量占用（　）个字节。
  - A. 32
  - B. 128
  - C. 4
  - D. 8

例题

- 现有一张分辨率为2048*1024像素的32位真彩色图像。请问要存储这张图像，需要多大的存储空间（　）。
  - A. 4MB
  - B. 8MB
  - C. 32MB
  - D. 16MB

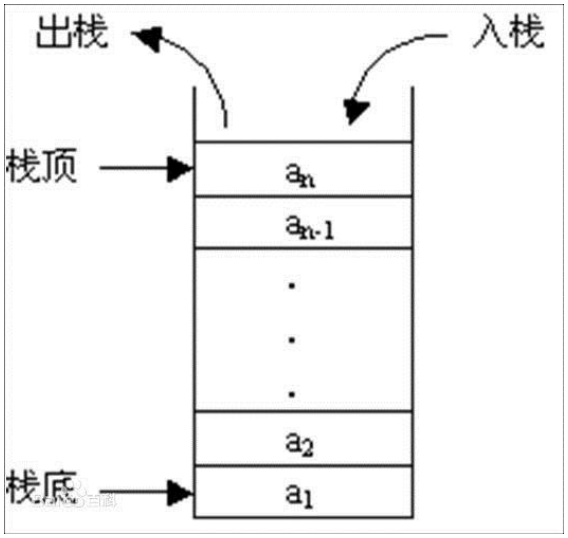
例题

- 目前主流的计算机存储数据最终都是转换成（　）数据进行存储。
  - A. 二进制
  - B. 十进制
  - C. 八进制
  - D. 十六进制

栈和队列、链表

栈

- 栈是限定仅能在表尾一端进行插入、删除操作的线性表。
- 栈的特点：先进后出（FILO）



第1个进栈的元素在栈底
最后一个进栈的元素在栈顶
第1个出栈的元素为栈顶元素
最后一个出栈的元素为栈底元素

队列

- 只允许在一端插入元素，另一端删除元素的线性表
- 允许插入的一端称为队尾（rear），允许删除的一端称为对头（front）
- 特点：先进先出（FIFO）

链表

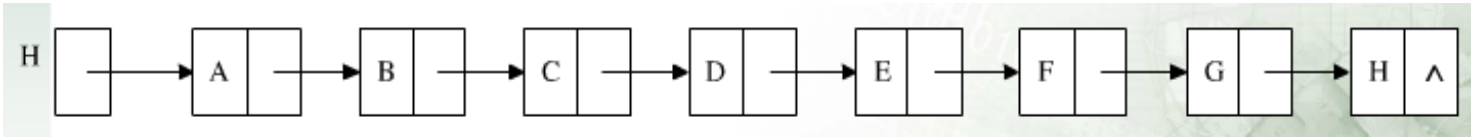
- 链表：用链式存储结构存储的线性表。
- 特点：用一组地址任意的存储单元存放线性表中的数据元素。

数据域	指针域

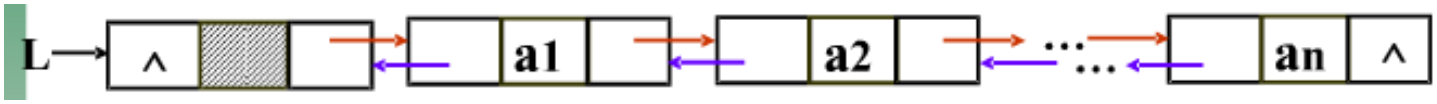
- 数据域：元素本身信息
- 指针域：指示直接后继（直接前驱）的存储位置，通过指针体现数据之间的次序关系。

链表

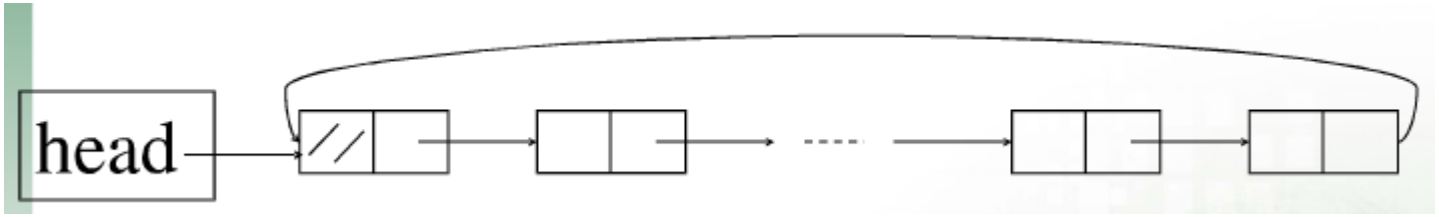
- 单链表：结点中只含一个指针域的链表称作单链表。



- 双链表：结点中有两个指针的链表称作双链表。



- 单循环链表：数据元素结构同单链表，将链表中最后一个结点的指针域指向表头。

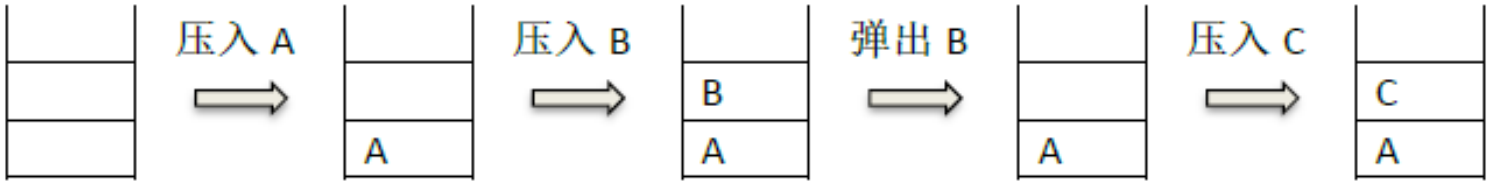


例题

- 链表**不具有**的特点是（　）。
  - A. 插入删除不需要移动元素
  - B. 可随机访问任一元素
  - C. 不必事先估计存储空间
  - D. 所需空间与线性表长度成正比

例题

- 下列图中所使用的数据结构式（　）。



A. 哈希表
B. 二叉树
C. 栈
D. 队列

例题

- 对于入栈顺序为a,b,c,d,e的序列，下列（　）不是合法的出栈序列。
  - A. a,b,c,d,e
  - B. e,d,c,b,a
  - C. b,a,c,d,e
  - D. c,d,a,e,b

例题

- 有6个元素，按照6、5、4、3、2、1的顺序进入栈S，请问下列哪个出栈序列是非法的（　）。
  - A. 5 4 3 6 1 2
  - B. 4 5 3 1 2 6
  - C. 3 4 6 5 2 1
  - D. 2 3 4 1 5 6

例题

- 链表和数组的区别包括（　）。
  - A. 数组不能排序，链表可以
  - B. 链表比数组能存储更多的信息
  - C. 数组大小固定，链表大小可动态调整
  - D. 以上均正确

例题

- 对假设栈S和队列Q的初始状态为空。存在e1~e6六个互不相同的数据，每个数据按照进栈S、出栈S、进队列Q、出队列Q的顺序操作，不同数据间的操作可能会交错。已知栈S中依次有数据e1、e2、e3、e4、e5和e6进栈，队列Q依次有数据e2、e4、e3、e6、e5和e1出队列。则栈S的容量至少是（　）个数据。

  A. 2
  B. 3
  C. 4
  D. 6

例题

- 以下对数据结构的表述不恰当的一项为：（　）。
  - A. 图的深度优先遍历算法常使用的数据结构为栈。
  - B. 栈的访问原则为后进先出，队列的访问原则是先进先出。
  - C. 队列常常被用于广度优先搜索算法。
  - D. 栈与队列存在本质不同，无法用栈实现队列。

例题

- 以下哪组操作完成在双向循环链表结点p之后插入结点s的效果（其中，next域为结点的直接后继，prev域为结点的直接前驱）：（　）。

A. p->next->prev=s; s->prev=p; p->next=s; s->next=p->next;

B. p->next->prev=s; p->next=s; s->prev=p; s->next=p->next;

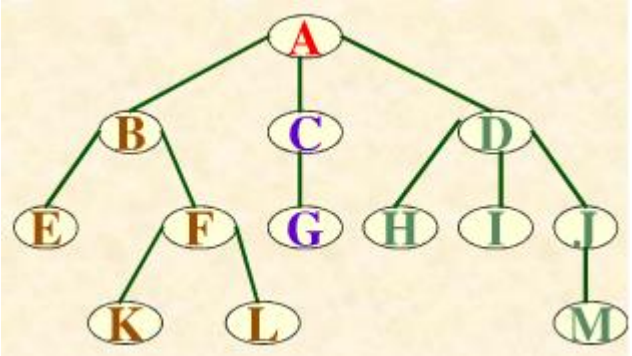
C. s->prev=p; s->next=p->next; p->next=s; p->next->prev=s;

D. s->next=p->next; p->next->prev=s; s->prev=p; p->next=s;

树

树

- 数据对象D：D是具有相同特性的数据元素的集合。
- 数据关系R：
  - 若D为空集，则称为空树。
  - 否则：
    1. 在D中存在唯一的称为根的数据元素root；
    2. 当n>1时，其余结点可分为m(m>0) 个互补相交的有限集$T_1,T_2,\ldots,T_m$，其中每一棵子集本身又是一棵符合本定义的树，称为根root的子树。



树

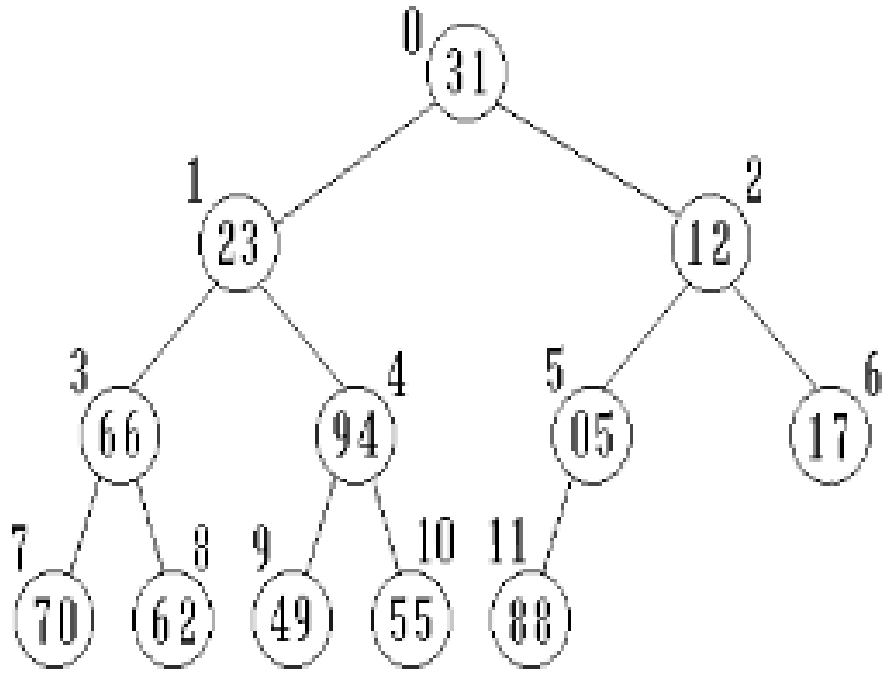
- 树型结构：
  - 根结点（无前驱）
  - 多个叶子结点（无后继）
  - 其他数据元素（一个前驱、多个后继）
- 基本术语
  - 结点：数据元素+若干指向子树的分支
  - 结点的度：分支的个数
  - 树的度：树中多有结点的度的最大值
  - 叶子节点：度为零的结点
  - 分支结点：度大于零的结点

树

- （从根到结点的）路径：由从根到该结点所经分支和结点构成
- 孩子结点、双亲结点
  兄弟结点、堂兄弟
  祖先结点、子孙结点
- 结点的层次：假设根结点的层次为1，第l层的结点的子树根结点的层次为l+1
- 树的深度：树中叶子结点所在的最大层次
- 森林：是m(m≥0)棵互补相交的树的集合

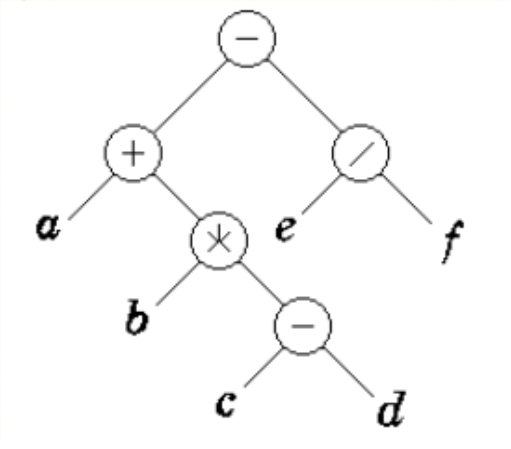
二叉树

- 二叉树或为空树，或是由一个根结点加上两棵分别称为左子树和右子树的、互不交的二叉树组成。
- 二叉树是一棵度数<=2的有序数。



二叉树的遍历

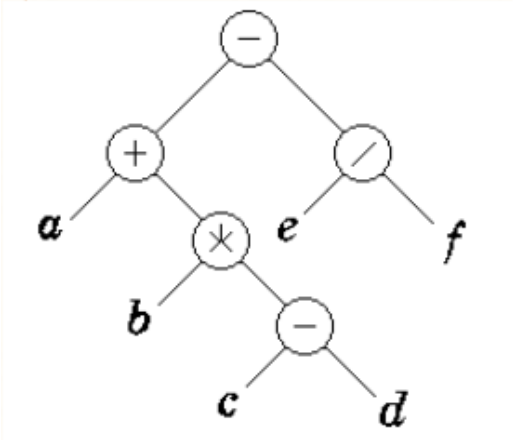
- 先序遍历算法：
  - 若二叉树为空树，则空操作；否则，
  1. 访问根结点；
  2. 先序遍历左子树；
  3. 先序遍历右子树。



  - -+a*b-cd/ef

二叉树的遍历

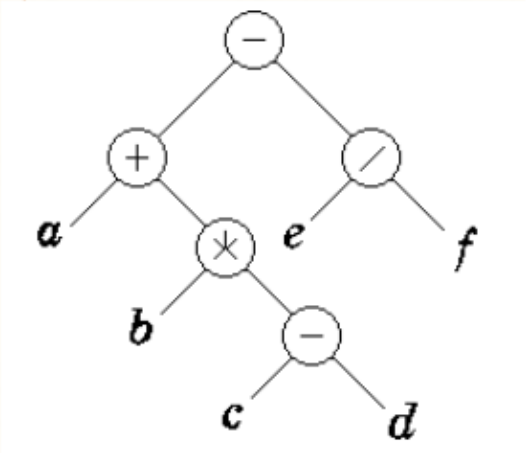
- 中序遍历算法：
  - 若二叉树为空树，则空操作；否则，
  1. 中序遍历左子树；
  2. 访问根结点；
  3. 中序遍历右子树。



  - a+b*c-d-e/f

二叉树的遍历

- 后序遍历算法：
  - 若二叉树为空树，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。



  - abcd-*+ef/-

哈夫曼树与哈夫曼编码

- 树的带权路径长度定义为：
  树中所有叶子结点的带全路径长度之和
  $WPL(T)=\sum w_k l_k$
- 在所有含n个叶子结点、并带相同权值的二叉树中，必存在一棵其带全路径长度取最小值的树，称为“哈夫曼树”。

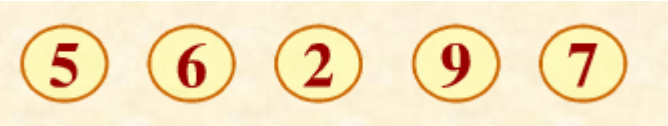
哈夫曼树与哈弗曼编码

① 根据给定的n个权值$\{w_1,w_2,\ldots,w_n\}$，构造n棵二叉树的集合$F=\{T_1,T_2,\ldots,T_n\}$，其中每棵二叉树中只含一个带权值为w_i的根结点，其左、右子树为空树；

② 在F中选取其根结点的权值为最小的两个二叉树，分别作为左、右子树构造一棵新的二叉树，并置这棵新的二叉树根结点的权值为其左、右子树根结点的权值之和；

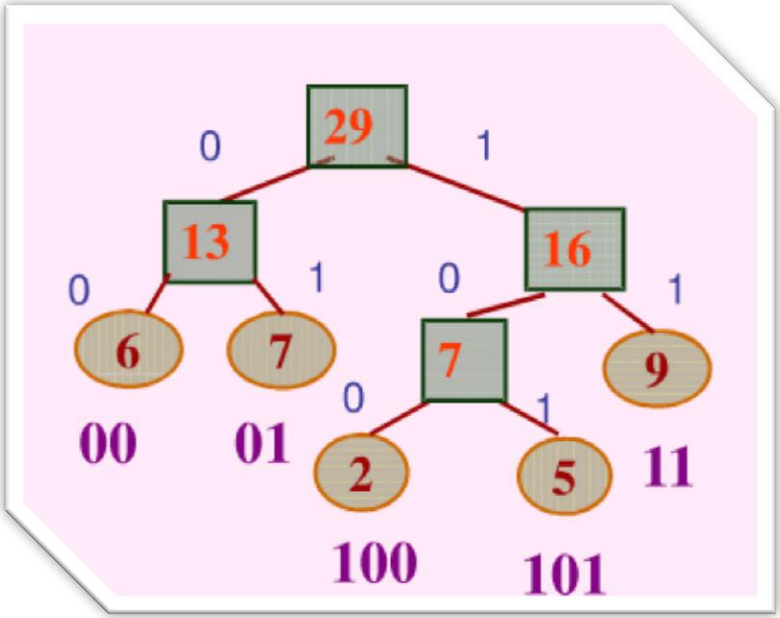
③ 从F中删去这两棵树，同时加入刚生成的新树；

④ 重复②和③两步，直至F中只含一棵树为止。



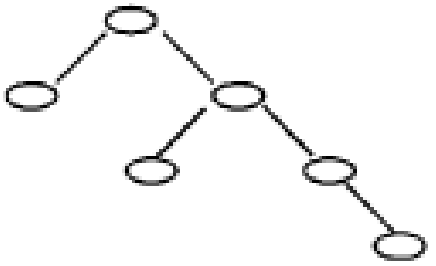
哈夫曼树与哈夫曼编码

- 假设有8个符号
  000 001 010 011 100 101 110 111
- 有时候，每个字符出现的频率不一样，采用变长编码，使得出现频率多的编码短，频率低的编码长。
  - 建立哈夫曼树
  - 对边编号
  - 求出叶子结点的路径
  - 得到字符编码
- 任何一个字符的编码都不是同一字符集中另一个字符的编码的前缀。



例题

- 一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1，若某结点的下标为i，则其左孩子位于下标2i处、右孩子位于下标2i+1处），则该数组的最大下标至少为（　）。



A. 6
B. 10
C. 15
D. 12

例题

- 假设一棵二叉树的后序遍历序列为DGJHEBIFCA，中序遍历序列为DBGEHJACIF，则其前序遍历序列为（　）。
  A. ABCDEFGHIJ
  B. ABDEGHJCFI
  C. ABDEGJHEFI
  D. ABDEGHJFIC

例题

- 独根树的高度为1。具有61个结点的完全二叉树的高度为（　）。
  - A. 7
  - B. 5
  - C. 8
  - D. 6

例题

- 如果一棵二叉树只有根结点，那么这棵二叉树高度为1。请问高度为5的完全二叉树有（　）种不同的形态？
  - A. 16
  - B. 15
  - C. 17
  - D. 32

例题

- 表达式a*(b+c)*d的后缀表达式为（　），其中“*”和“+”是运算符。
  - A. **a+bcd
  - B. abc+*d*
  - C. abc+d**
  - D. *a*+bcd

例题

- 对表达式a+(b-c)*d的前缀表达式为（　），其中+、-、*是运算符。
  A. *+a-bcd
  B. +a*bcd
  C. abc-d*+
  D. abc-+d

例题

- 假设字母表{a，b，c，d，e}在字符串出现的频率分别为10%，15%，30%，16%，29%。若使用哈弗曼编码房室对字母进行不定长的二进制编码，字母d的编码长度为（　）位。
  - A. 1
  - B. 2
  - C. 2或3
  - D. 3

例题

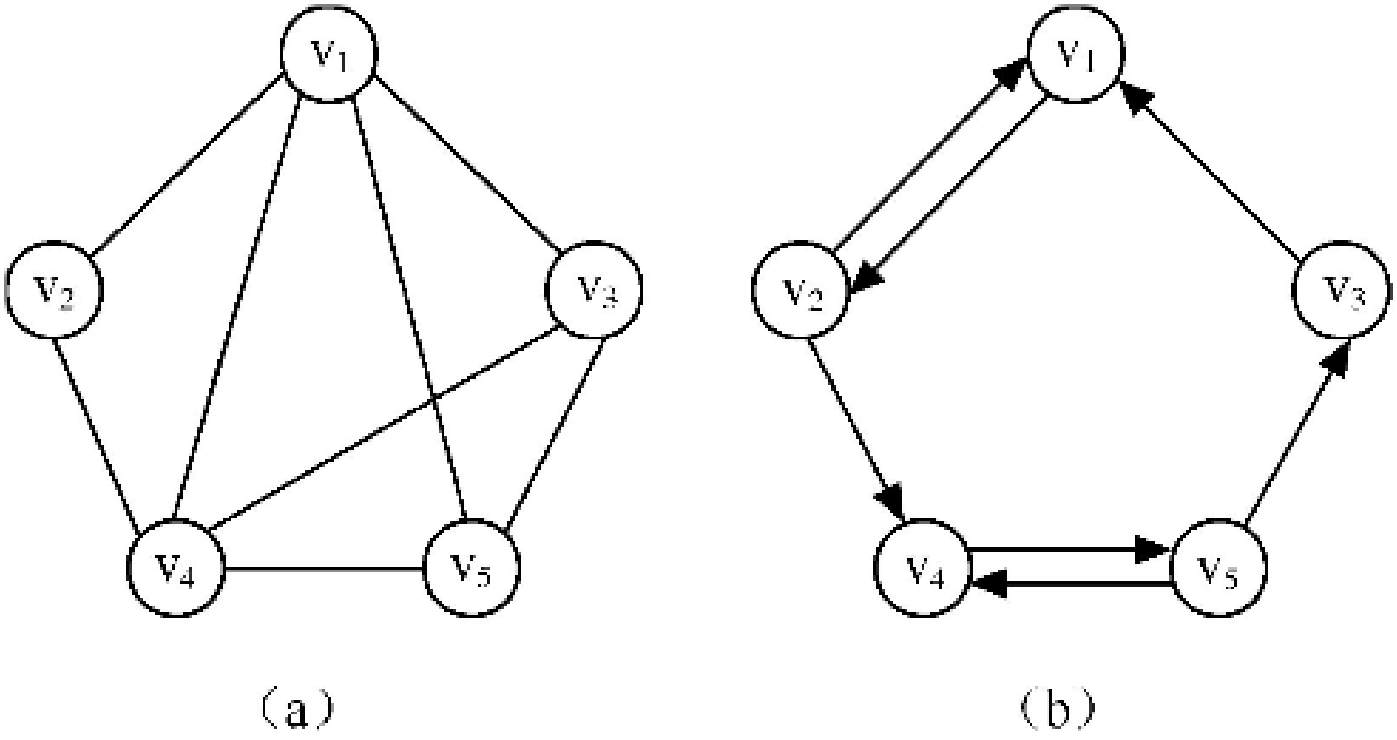
- 一棵有n个结点的完全二叉树用数组进行存储与表示，已知根结点存储在数组的1个位置。若存储在数组第9个位置的结点存在兄弟结点和两个子结点，则它的兄弟结点和右子结点的位置分别是（　）。
  - A. 8、18
  - B. 10、18
  - C. 8、19
  - D. 10、19

图

图

- 在讲述图时，习惯把数据元素统称为是顶点。
- “图”是图状结构的简称，它由一个非空的顶点集合V和一个描述顶点之间邻接关系的边集合E组成，E中每条边连接的两个顶点都必须属于集合V。于是，一个图可以记为：G=(V,E)
- 对于一个图G来说，边的集合E可以是空的。若v_i、v_j是V集合中的两个顶点，且有边连接，那么就用记号(v_i,v_j)表示顶点v_i到顶点v_j之间的边。当图中的边不带有方向时，称该图为“无向图”。
- 若图中的边是有向的，这样的图被称为“有向图”。常称有向图中的边为“弧”，把有向图中从顶点v_i到顶点v_j的弧记为$< v_i,v_j >$。

图



顶点的度、入度、出度

- 无向图中，若顶点v_i和v_j之间有一条边(v_i, v_j)存在，则表明顶点v_i和v_j互为邻接点，简称v_i与v_j相邻接。所谓顶点v_i的“度”，是指与它相邻接的顶点的个数，记为$D(v_i)$。
- 在有向图中，以顶点v_i为弧尾的个数，称为顶点v_i的“出度”，记为$OD(v_i)$；以顶点v_i为弧头的个数，称为顶点v_i的“入度”，记为$ID(v_i)$。这时，一个顶点v_i的度是指它的入度与出度之和，即$D(v_i)=OD(v_i)+ID(v_i)$。

路径、路径长度

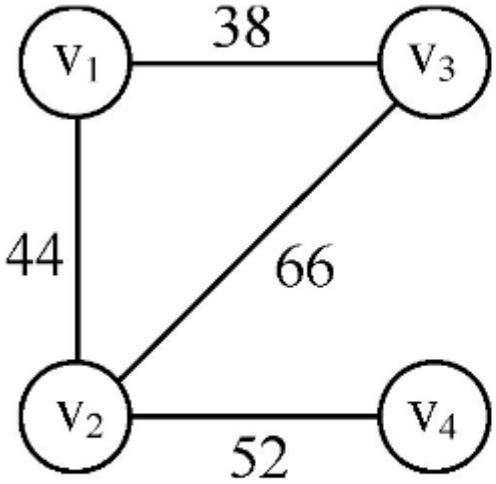
- 在无向图G中，所谓从顶点v_i到顶点v_j的一条“路径”，是指在顶点v_i与顶点v_j之间存在有一个弧的序列：$(v_i,v_{i1}),(v_{i1},v_{i2}),\ldots,(v_{im},v_{ij})$。其中顶点$v_i$、$v_{i1}$、$v_{i2}$、…、$v_{im}$、$v_j$属于G的顶点集合V，边$(v_i,v_{i1}),(v_{i1},v_{i2}),\ldots,(v_{im},v_{ij})$属于G的边集合E。
- 对于有向图G，所谓从顶点v_i到顶点v_j的一条“路径”，是指在顶点v_i与顶点v_j之间存在有一个边的序列：$<v_i,v_{i1}>,<v_{i1},v_{i2}>,\ldots,<v_{im},v_{ij}>$。其中顶点$v_i$、$v_{i1}$、$v_{i2}$、…、$v_{im}$、$v_j$属于G的顶点集合V，弧$<v_i,v_{i1}>,<v_{i1},v_{i2}>,\ldots,<v_{im},v_{ij}>$属于G的弧的集合E。
- 路径“长度”，是指在这条路径上拥有的边的个数。

连通、连通图

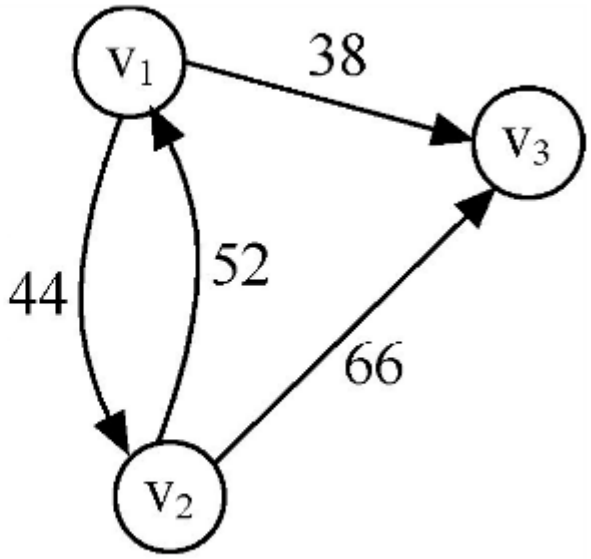
- 在无向图中，若从顶点v_i到顶点v_j之间有路径存在，则称v_i与v_j是“连通”的。如果无向图G中任意一对顶点之间都是连通的，则称该图G为“连通图”，否则是非连通图。
- 强连通图是只一个有向图中任意两点v_i、v_j间存在v_i到v_j及v_j到v_i的路径的图。

边的权

- 有时，可以给图的边或弧依附上某种数值，这种与图的边或弧相关的数值被称为“权”。



（a）



（b）

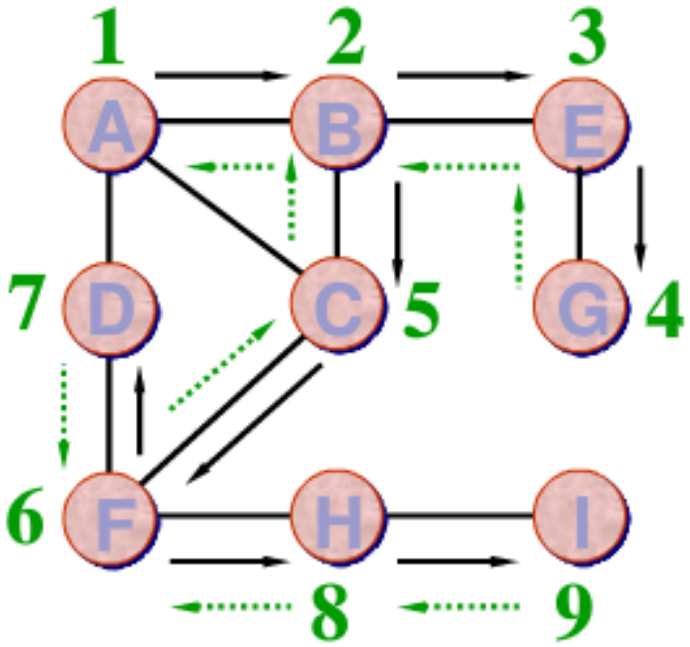
图的遍历

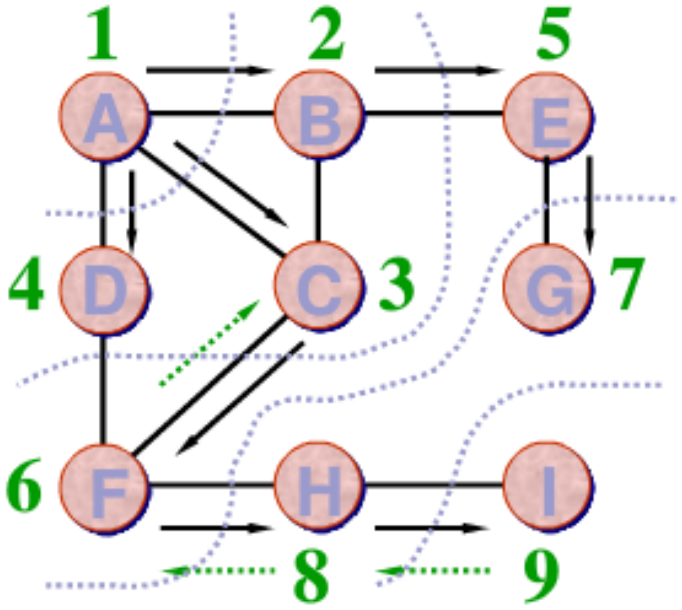
- 深度优先搜索（DFS）是指按照深度方向搜索，它类似于树的先根遍历。
- 深度优先搜索的基本思想是：
  ① 从图中某个顶点v_i出发，首先访问v_i。
  ② 找出顶点v_i的第一个未被访问的邻接点v_j，后访问顶点v_j。找出顶点v_j的第一个未被访问的邻接点v_{j1}，…，重复本步骤，直到顶点v_{j1}的所有的邻接点都被访问为止。
  ③ 返回前一个访问过的且仍未被访问的邻接点的顶点，找出并访问该顶点的下一个未被访问的邻接点，然后执行步骤②。否则行步骤④
  ④ 若此时图中还有顶点未被访问，则另选图中一个未被访问的顶点作为起始点，重复上述搜索过程，直至图中所有顶点均被访问过为止。

图的遍历

- 广度优先搜索（BFS）是指按照广度方向搜索，和深度优先搜索不同的是：广度优先搜索先访问完所有的邻接点，再去寻找与邻接点相邻的下一层的其他顶点，它类似于树的层次遍历。
- 广度优先搜索的基本思想是：
  ① 从图中某个顶点v_0出发，首先访问v_0。
  ② 依次访问v_0的各个未被访问的邻接点。
  ③ 分别从这些邻接点出发，依次访问它们的各个未被访问的邻接点。访问时保证：如果v_i和v_k为当前结点，且v_i在v_k之前被访问，则v_i的所有未被访问的邻接点应在v_k的所有未被访问的邻接点之前访问。重复步骤③，直到所有结点均没有未被访问的邻接点为止。
  ④ 重复上述过程，直至所有顶点均被访问过为止。

图的遍历





例题

- 由四个不同的点构成的简单无向连通图的个数是（　）。
  - A. 32
  - B. 35
  - C. 38
  - D. 41

例题

- 由四个没有区别的点构成的简单无向连通图的个数是（　）。
  - A. 6
  - B. 7
  - C. 8
  - D. 9

例题

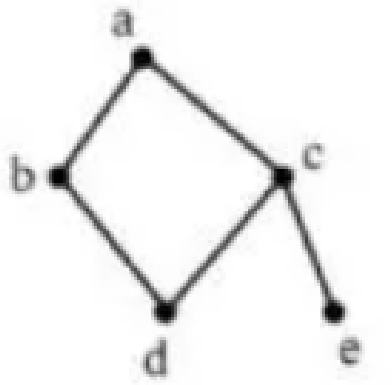
- 有10个顶点的无向图**至少**应该有（　）条边才能确保是一个连通图。
  - A. 10
  - B. 12
  - C. 9
  - D. 11

例题

- 对于有n个顶点、m条边的无向连通图（m>n），需要删掉（　）条边才能变其成为一棵树。
  - A. n-1
  - B. m-n
  - C. m-n-1
  - D. m-n+1

例题

- 以a为起点，对右边的无向图进行深度优先遍历，则b、c、d、e四个点中有可能作为最后一个遍历到的点的个数为（　）。



A. 1
B. 2
C. 3
D. 4

例题

- 考虑由N个顶点构成的有向连通图，采用邻接矩阵的数据结构表示时，该矩阵中至少存在（　）个非零元素。
  - A. N-1
  - B. N
  - C. N+1
  - D. N^2

程序、算法

排序算法

- 选择排序基本思想：

  对待排序的序列进行n-1遍处理：
  第1遍处理是从a[1],a[2],…,a[n]中选择最小的放在a[1]位置；
  第2遍处理是从a[2],a[3],…,a[n]中选择最小的放在a[2]位置；
  ……
  第i遍处理是将a[i],a[i+1],…,a[n]中最小的数与a[i]交换位置，这样经过第i遍处理后，a[i]是所有的数中第i小。即前i个数已经排好序了。
  n-1遍处理后，剩下的最后一个一定是最大的，不需要再处理了。

排序算法

- 冒泡排序的基本思想：

  将待排序的数据看作竖排的一列“气泡”，小的数据比较轻，从而要上浮。
  共进行n-1遍处理，每一遍处理，就是从底向上检查序列，如果相邻的两个数据顺序不对，即轻（小）的在下面，就交换他们的位置。
  第一遍处理完后，“最轻”的就浮到上面。
  第二遍处理完后，“次轻”的就浮到上面。
  共需要n-1遍处理即完成排序。

排序算法

- 插入排序的基本思想：

  经过i-1遍处理后，a[1],a[2],…,a[i-1]已排好序。
  第i遍处理是将元素a[i]插入到a[1],a[2],…,a[i-1]的适当的位置，从而使得a[1],a[2],…,a[i-1],a[i]又是拍好的序列。
  每次将一个待排序的数据元素，插入到前面已经排好序的数列中的适当位置，使数列依然有序；知道待排序数据元素全部插入完为止。

排序算法

- 快速排序的基本思想：

  把待排序的n个元素放到一个数组中，先取数组中的某一个元素作为一个基准元素，把这个元素放到数组中合适的位置，同时对其他元素进行调整，使得在这个基准元素的右边的所有元素都比这个基准元素大，而基准元素左边的元素都比它小。也就是说，这个基准元素当前所在的位置就是排序后的最终位置。然后再对基准元素的前后两个区间分别进行快速排序，知道每个区间为空或者只有一个元素时，整个快速排序结束。

排序算法

- 基数排序的基本思想：

  透过键值的部份咨询，将要排序的元素分配至某些“桶”中，藉以达到排序的作用。

  - 73,22,93,43,55,14,28,65,39,81
  - 首先根据个位数的数值，在走访数值时将它们分配至编号0到9的桶子中：
    ( )(81)(22)(73,93,43)(14)(55,65)( )( )(28)(39)
  - 接下来将这些桶子中的数值重新串接起来，成为以下的数列：
    81,22,73,93,43,14,55,65,28,39
    接着再进行一次分配，这次是根据十位数来分配：
    ( )(14)(22,28)(39)(43)(55)(65)(73)(81)(93)
  - 接下来将这些桶子中的数值重新串接起来。

排序算法

- 归并排序的基本思想：

  该算法是采用分治法的一个非常典型的应用。将已有序的子序列合并，得到完全有序的序列；即先使每个子序列有序，再使子序列段间有序。若将两个有序表合并成一个有序表，称为二路归并。

  - 初始状态：6,202,100,301,38,8,1
  - 第一次归并后：(6,202)(100,301)(8,38)(1)
  - 第二次归并后：(6,100,202,301)(1,8,38)
  - 第三次归并后：(1,6,8,38,100,202,301)

时间复杂度

- 一般情况下，算法的基本操作重复执行的次数是模块n的某一个函数f(n)，因此，算法的时间复杂度记作T(n)=O(f(n))
- 在计算时间复杂度的时候，先找出算法的基本操作，然后根据相应的各语句确定它的执行次数，再找出T(n)的同数量级（它的同数量级有一下：1，$\log_2 n$，n，$n\log_2 n$，n^2，n^3，2^n，$n!$），找出后f(n)=该数量级，若T(n)/f(n)求极限可得到一常数c，则时间复杂度T(n)=O(f(n))

空间复杂度

- 一个程序的空间复杂度是指运行完一个程序所需内存的大小。利用程序的空间复杂度，可以对程序的运行所需要的内存多少有个预先估计。一个程序执行时除了需要存储空间和存储本身所使用的指令、常数、变量和输入数据外，还需要一些对数据进行操作的工作单元和存储一些为现实计算所需信息的辅助空间。程序执行时所需存储空间包括以下两部分。
  - 固定部分。这部分空间的大小与输入/输出的数据的个数多少、数值无关。主要包括指令空间（即代码空间）、数据空间（常量、简单变量）等所占的空间。
  - 可变空间，这部分空间的主要包括动态分配的空间，以及递归栈所需要的空间等。

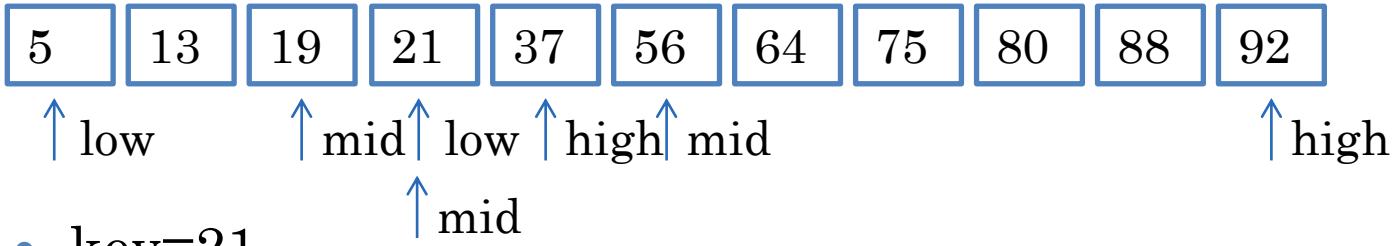
排序算法的比较

- 原地排序：不申请多余的空间来进行的排序，就是在原来的排序数据中比较和交换的排序。
- 稳定排序：当有两个相等记录的关键字R和S，且在原本的列表中R出现在S之前，在排序过的列表中R将会是在S之前。

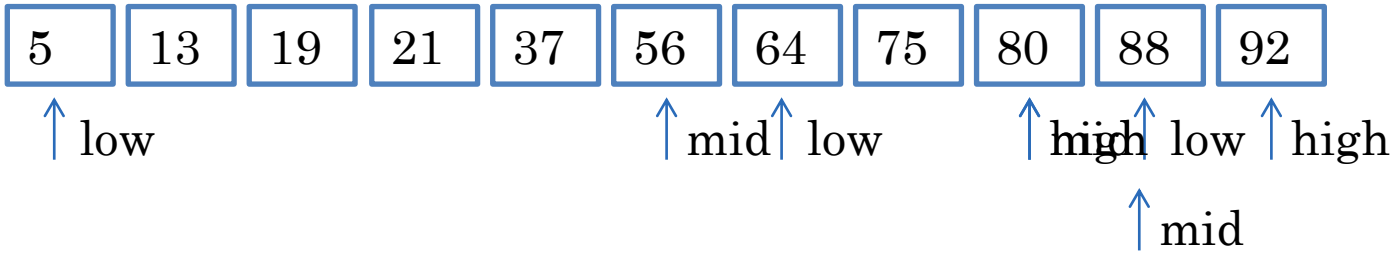
	时间复杂度	原地排序	稳定排序
选择排序	$O(n^2)$	是	不稳定
冒泡排序	$O(n^2)$	是	稳定
插入排序	$O(n^2)$	是	稳定
快速排序	O(n logn)	不是	不稳定
基数排序	O(n)	不是	稳定
归并排序	O(n logn)	不是	稳定

二分查找

- 折半查找（二分查找）的查找过程是：先确定待查记录所在的范围，然后逐步缩小范围直到找到或找不到该记录为止。



  - key=21



  - key=85

- 时间复杂度：$O(\log n)$

枚举法

- 在研究问题时，把所有可能发生的情况一一列举加以研究的方法叫做枚举法（也叫穷举法）。
- 特点：有条理、不重复、不遗漏
- 例：有一个三位数的各位数字都不是0，且各位数字之和是6，这样的三位数共有多少个？

贪心算法

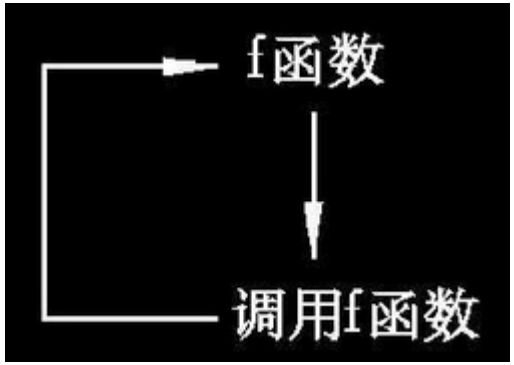
- 贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的仅是在某种意义上的局部最优解。
- 贪心算法不是对所有问题都能得到整体最优解，但对范围相当广泛的许多问题他能产生整体最优解或者整体最优解的近似解。
- Prim算法、Dijkstra算法、哈夫曼树

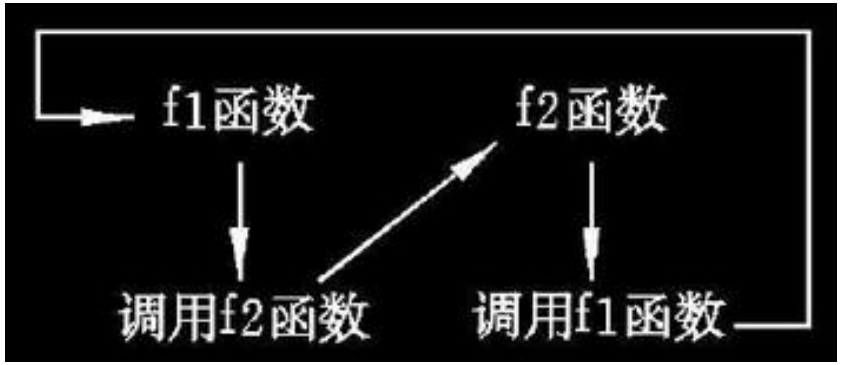
分治法

- 分治法的基本思想是将一个规模为n的问题分解为k个规模较小的子问题，这些子问题互相独立且与原问题相同。
- 对k个子问题分别求解。如果子问题的规模仍然不够小，则再划分为k个子问题，如此递归的进行下去，知道问题规模足够小，很容易求出其解为止。
- 将求出的小规模的问题的解合并为一个更大规模的问题的解，自底向上逐步求出原来问题的解。
- 分治法的设计思想是，将一个难以直接解决的大问题，分割成一些规模较小的相同问题，以便各个击破，分而治之。
- 例：给定数组A[1..n]，求数组最大最小值。

递归

- 在调用一个函数的过程中又出现直接或间接地调用该函数本身，称为函数的递归调用。





- 例：有5个人坐在一起，问第5个人多少岁，他说比第4个人大2岁。问第4个人岁数，他说他比第3个人大2岁。问第3个人，又说比第2个人大2岁。问第2个人，说比第1个人大2岁。最后问第1个人，他说是10岁。请问第5个人多大。

例题

- 给定一个含N个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要N-1次比较操作。则最坏情况下，在该数组中同时找最大与最小的数至少需要（　）次比较操作。
（⌈　⌉表示向上取整，⌊　⌋表示向下取整）

A. $\lceil 3N/2 \rceil - 2$

B. $\lfloor 3N/2 \rfloor - 2$

C. $2\,N - 2$

D. $2\,N - 4$

例题

- 下面的故事与（　）算法有着异曲同工之妙。
  从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：“从前有座山山里有座庙，庙里有个老和尚给小和尚讲故事：‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事……’”
  A. 枚举
  B. 递归
  C. 贪心
  D. 分治

例题

- 若f[0]=0,f[1]=1,f[n+1]=(f[n]+f[n-1])/2，则随着i的增大，f[i]将接近于（　）。
  - A. 1/2
  - B. 2/3
  - C. (√5-1)/2
  - D. 1

例题

- 若有如下程序段，其中s、a、b、c均定义为整型变量，且a、c均已赋值（c大于0）

  s=a;

  for(b=1; b<=c; b++) s=s-1;

  则与上述程序段功能等价的赋值语句是（　）。

A. s=a-c;
B. s=a-b;
C. s=s-c;
D. s=b-c;

例题

- 设有100个已排好序的数据元素，采用折半查找时，最大比较次数为（　）。
  - A. 7
  - B. 10
  - C. 6
  - D. 8

例题

- 设A是n个实数的数组，考虑下面的递归算法：

```
XYZ(A[1..n])
1. if n=1 then return A[1]
2. else temp←XYZ(A[1..n-1])
3.       if temp<A[n]
4.       then return temp
5.       else return A[n]
```

请问算法XYZ的输出时什么？（　）

A. A数组的平均
B. A数组的最小值
C. A数组的最大值
D. A数组的中值

例题

- 以比较作为基本运算，在N个数中找出最大数，最坏情况下所需要的最少的比较次数为（　）。
  - A. N^2
  - B. N
  - C. N-1
  - D. N+1

例题

- 在数据压缩编码中的哈弗曼编码方法，在本质上是一种（　）的策略。
  - A. 枚举
  - B. 贪心
  - C. 递归
  - D. 动态规划

例题

- 考虑如下递归算法

```
solve(n)
        if n<=1 return 1
        else if n>=5 return n*solve(n-2)
        else return n*solve(n-1)
```

则调用solve(7)得到的返回结果为（　）。

A. 105
B. 840
C. 210
D. 420

例题

- 运行以下代码片段的行为是（　）。

```
int x=101;
int y=201;
int *p=&x;
int *q=&y;
p=q;
```

A. 将x值赋为201
B. 将y的值赋为101
C. 将q指向x的地址
D. 将p指向y的地址

例题

- 以下排序算法的常见实现中，哪个选项的说法是错误的：（　）。
  - A. 冒泡排序算法是稳定的
  - B. 简单选择排序是稳定的
  - C. 简单插入排序是稳定的
  - D. 归并排序算法是稳定的

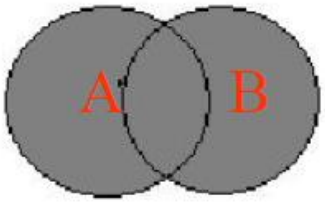
例题

- 以下对递归方法的描述中，正确的是：（　）。
  - A. 递归式允许使用多组参数调用函数的编程技术
  - B. 递归式通过调用自身来求解问题的编程技术
  - C. 递归式面向对象和数据而不是功能和逻辑的编程语言模型
  - D. 递归是将用某种高级语言转换为机器代码的编程技术

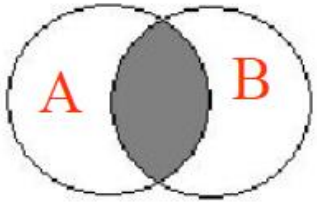
数学知识

集合及其运算

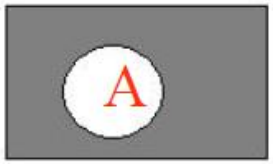
- 并：$\cup$
- 交：$\cap$
- 补：$\overline{\ }$
- 差：-



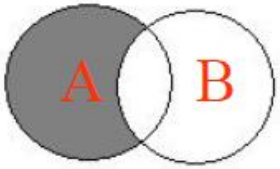
$A \cup B$



$A \cap B$



$\overline{A}$



A-B

容斥原理

- 先不考虑重叠的情况，把包含于某内容中的所有对象的数目先计算出来，然后再把计数时重复计算的数目排斥出去，使得计算的结果既无遗漏又无重复。
- 对有限集合S，用|S|表示S的元素个数。
- 容斥原理的第一形式：设A、B是有限集合，则
  $|A \cup B| = |A| + |B| - |A \cap B|$
- 容斥原理的第二形式：设A、B、C是有限集合，则
  $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$

排列与组合

- 排列：从n个不同元素中，任取m个元素，按照一定的顺序排成一列。

  $$p_n^m = n(n-1)(n-2)\dots(n-m+1) = \frac{n!}{(n-m)!}$$

- 全排列问题：n个不同的元素排成一排，排列方法有

  $$p_n^n = n(n-1)(n-2)\dots(1) = n!$$

- 组合：从n个不同元素中，任取m个元素，并成一组。

  $$C_n^m = \frac{p_n^m}{p_m^m} = \frac{n(n-1)(n-2)\dots(n-m+1)}{m!} = \frac{n!}{m!(n-m)!}$$

- 排列与组合的区别与联系：与顺序有关的为排列问题，与顺序无关的为组合问题。

例题

- 319和377的最大公约数是（　）。
  - A. 27
  - B. 33
  - C. 29
  - D. 31

例题

- 新学期开学了，小胖想减肥，健身教练给小胖制定了两个训练方案。方案一：每次连续跑3公里可以消耗300千卡（耗时半小时）；方案二：每次连续跑5公里可以消耗600千卡（耗时1小时）。小胖每周周一到周四能抽出半小时跑步，周五到周日能抽出一小时跑步。另外，教练建议小胖每周最多跑21公里，否则会损伤膝盖。请问如果小胖想严格执行教练的训练方案，并且不想损伤膝盖，每周最多通过跑步消耗多少千卡？（　）。

  A. 3000
  B. 2500
  C. 2400
  D. 2520

例题

- 一副纸牌除掉大小王有52张牌，四种花色，每种花色13张。假设从这52张牌中随机抽取13张纸牌，则至少（　）张牌的花色一致。

  A. 4
  B. 2
  C. 3
  D. 5

例题

- 一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6，其他数字颠倒过来都不构成数字。类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。请问这个城市最多有多少个车牌颠倒过来恰好还是原来的车牌（　）。

  A. 60
  B. 125
  C. 75
  D. 100

例题

- 五个小朋友并排占城一列，其中有两个小朋友是双胞胎，如果要求这两个双胞胎必须相邻，则有（　）种不同排列方法？
  - A. 24
  - B. 36
  - C. 72
  - D. 48

例题

- 10个二好学生名额分配到7个班级，每个班级至少有一个名额，一共有（　）种不同的分配方案。
  - A. 56
  - B. 84
  - C. 72
  - D. 504

例题

- 有五副不同颜色的手套（共10只手套，每副手套左右手各1只），一次性从中取6只手套，请问恰好能配成两副手套的不同取法有（　）种。
  - A. 30
  - B. 150
  - C. 180
  - D. 120

例题

- 6个人，两个人组一队，总共组成三队，不区分队伍的编号。不同的组队情况有（　）种。
  - A. 10
  - B. 15
  - C. 30
  - D. 20

例题

- 由1,1,2,2,3这五个数字组成不同的三位数有（　）种。
  - A. 18
  - B. 15
  - C. 12
  - D. 24

例题

- 有四个人要从A点坐一条船过河到B点，船一开始在A点。该船一次最多可坐两个人。已知这四个人中每个人独自坐船的过河时间分别为1,2,4,8，且两个人坐船的过河时间为两人独自过河时间的较大者。则最短（　）时间可以让四个人都过河到B点（包括从B点把船开回A点的时间）。

  A. 14
  B. 15
  C. 16
  D. 17

例题

- 一个字符串中任意个连续的字符组成的子序列称为该字符串的子串，则字符串abcab有（　）个内容互不相同的子串。
  - A. 12
  - B. 13
  - C. 14
  - D. 15

其他

例题

- IPv4协议使用32位地址，随着其不断被分配，地址资源日趋枯竭。因此，它正逐渐被使用（　）位地址的IPv6协议所取代。
  - A. 40
  - B. 48
  - C. 64
  - D. 128

例题

- 通常在搜索引擎中，对某个关键词加上双引号表示（　）。
  - A. 排除关键词，不显示任何包含该关键词的结果
  - B. 将关键词分解，在搜索结果中必须包含其中的一部分
  - C. 精确搜索，只显示包含整个关键词的结果
  - D. 站内搜索，只显示关键词所指向网站的内容

例题

- 中国的国家顶级域名是（　）。
  - A. .cn
  - B. .ch
  - C. .chn
  - D. .china

例题

- 1948年，（　）将热力学中的熵引入信息通信领域，标志着信息论研究的开端。
  - A. 冯·诺依曼（John von Neumann）
  - B. 图灵（Alan Turing）
  - C. 欧拉（Lenhard Euler）
  - D. 克劳德·香农（Claude Shannon）

冯·诺依曼是20世纪最重要的数学家之一，在现代计算机、博弈论、核武器和生化武器等领域内的科学全才之一，被后人称为“计算机之父”和“博弈论之父）”。

图灵是英国数学家、逻辑学家，被称为计算机科学之父，人工智能之父。图灵对于人工智能的发展有诸多贡献，此外，图灵提出的著名的图灵机模型为现代计算机的逻辑工作方式奠定了基础。

欧拉是18世纪数学界最杰出的人物之一，他不但为数学界作出贡献，更把整个数学推至物理的领域。

例题

- （　）是一种通用的字符编码，它为世界上绝大部分语言设定了统一并且唯一的二进制码，以满足跨语言、跨平台的文本交换。目前它已经收录了超过十万个不同字符。
  A. ASCII
  B. Unicode
  C. GBK 2312
  D. BIG 5

例题

- 下列协议中与电子邮件无关的是（　）。
  A. POP3
  B. SMTP
  C. WTO
  D. IMAP

例题

- 计算机应用的最早领域是（　）。
  - A. 数值计算
  - B. 人工智能
  - C. 机器人
  - D. 过程控制

例题

- 下列不属于面向对象程序设计语言的是（　）。
  - A. C
  - B. C++
  - C. Java
  - D. C#

例题

- NOI的中文意思是（　）。
  - A. 中国信息学联赛
  - B. 全国青少年信息学奥林匹克竞赛
  - C. 中国青少年信息学奥林匹克竞赛
  - D. 中国计算机协会

例题

- 从（　）年开始，NOIP竞赛奖不再支持Pascal语言。
  - A. 2020
  - B. 2021
  - C. 2022
  - D. 2023

例题

- 以下和计算机领域密切相关的奖项是（　）。
  - A. 奥斯卡奖
  - B. 图灵奖
  - C. 诺贝尔奖
  - D. 普利策奖

例题

- 中国的国家顶级域名是（　）。
  - A. .cn
  - B. .ch
  - C. .chn
  - D. .china

例题

- 以下哪个奖项是计算机科学领域的最高奖（　）。
  - A. 图灵奖
  - B. 鲁班奖
  - C. 诺贝尔奖
  - D. 普利策奖

例题

- 在内存储器中每个存储单元都被赋予一个唯一的序号，称为（　）。
  - A. 下标
  - B. 地址
  - C. 序号
  - D. 编号

例题

- 编译器的主要功能是（　）。
  - A. 将源程序翻译成机器指令代码
  - B. 将一种高级语言翻译成另一种高级语言
  - C. 将源程序重新组合
  - D. 将低级语言翻译成高级语言

例题

- 一下不属于面向对象程序设计语言的是（　）。
  - A. C++
  - B. Python
  - C. Java
  - D. C

例题

- 以下奖项与计算机领域最相关的是（　）。
  - A. 奥斯卡奖
  - B. 图灵奖
  - C. 诺贝尔奖
  - D. 普利策奖

例题

- 以下哪种功能没有涉及C++语言的面向对象特征支持：（　）。
  - A. C++中调用printf函数
  - B. C++中调用用户定义的类成员函数
  - C. C++中构造一个class或struct
  - D. C++中构造来源于同一基类的多个派生类