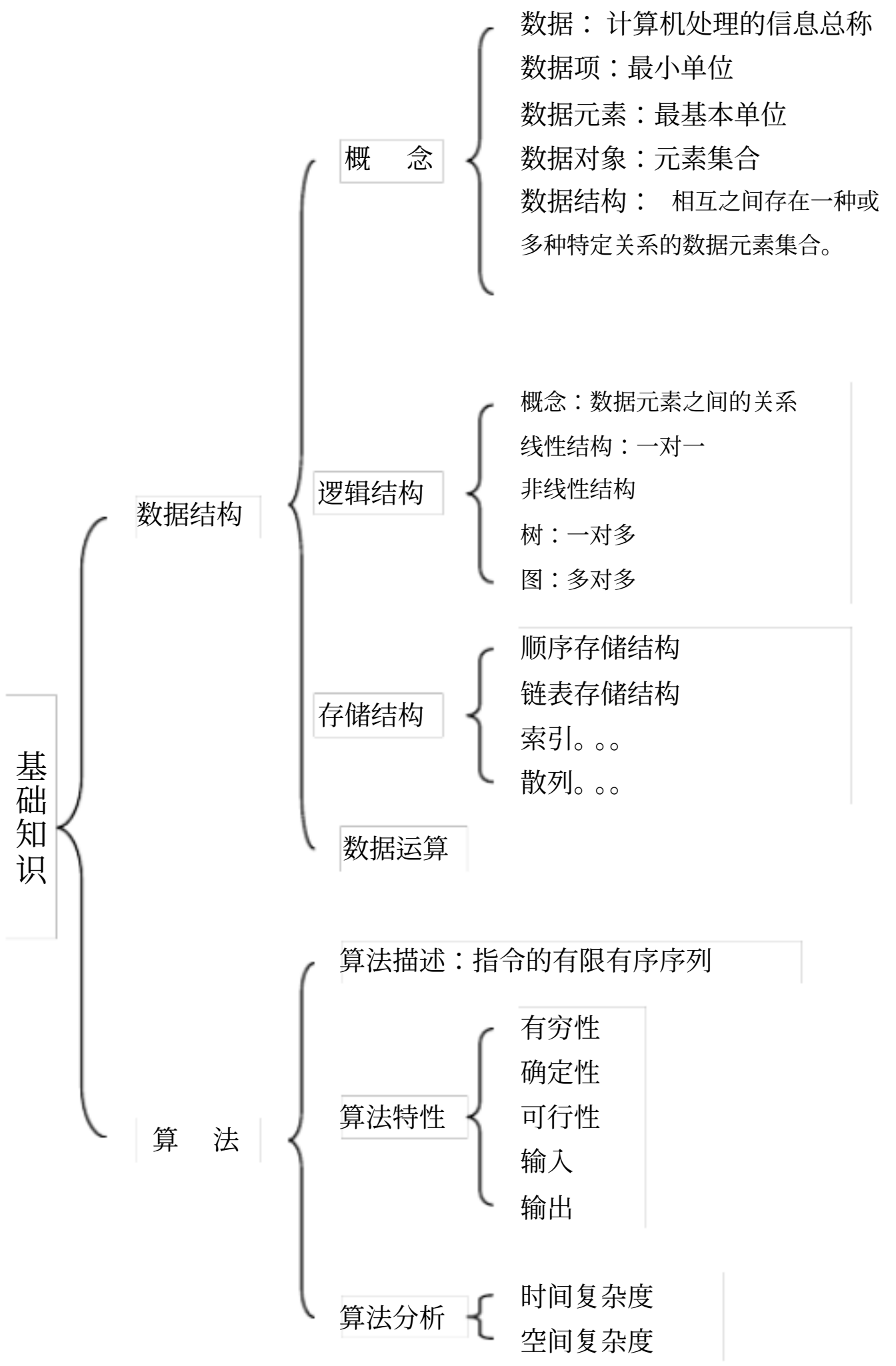
期末复习

第一章　绪论　复习



1、计算机算法必须具备输入、输出、可行性、确定性、有穷性 5 个特性。

2、算法分析的两个主要方面是空间复杂度和时间复杂度。

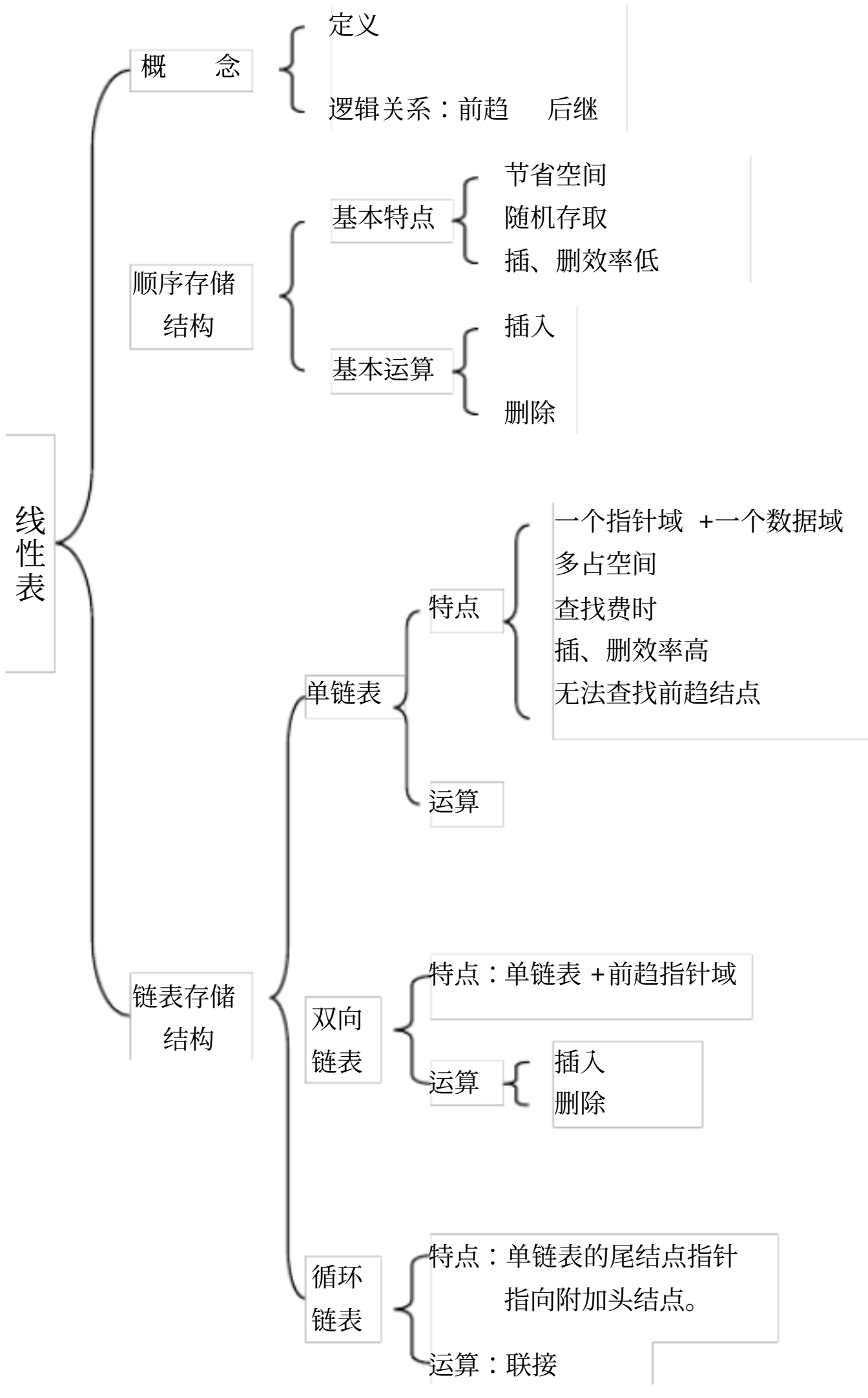
3、数据元素是数据的基本单位。

4、数据项是数据的最小单位。

5、数据结构是带结构的数据元素的集合。

6、数据的存储结构包括顺序、链接、散列和索引四种基本类型。

第二章　线性表　复习



1、在双链表中，每个结点有两个指针域，包括一个指向前驱结点的指针　　、一个指向后继结点的指针

2、线性表采用顺序存储，必须占用一片连续的存储单元

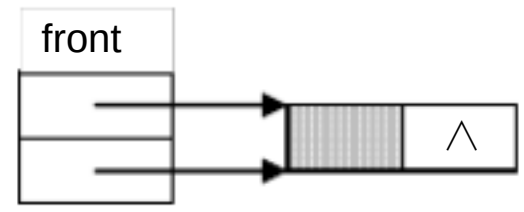
3、线性表采用链式存储，便于进行插入和删除操作

4、线性表采用顺序存储和链式存储优缺点比较。

5、简单算法

第三章　栈和队列　复习

- 栈
  - 栈的概念：在一端操作的线性表
  - 栈的特点：先进后出　　LIFO
  - 存储结构
  - 运算算法
    - 初始化
    - 进栈 push
    - 出栈 pop
- 队列
  - 队列概念：在两端操作的线性表
  - 队列特点：先进先出　　FIFO
  - 顺序队列
循环队列
    - 假溢出
    - 队空： front=rear
    - 队满： front=(rear+1)%MAXSIZE
  - 链队列　队空：



  - 基本运算
    - 顺序：
    - 链队：
      - 初始化
      - 判空
      - 进队
      - 出队
      - 取队首元素

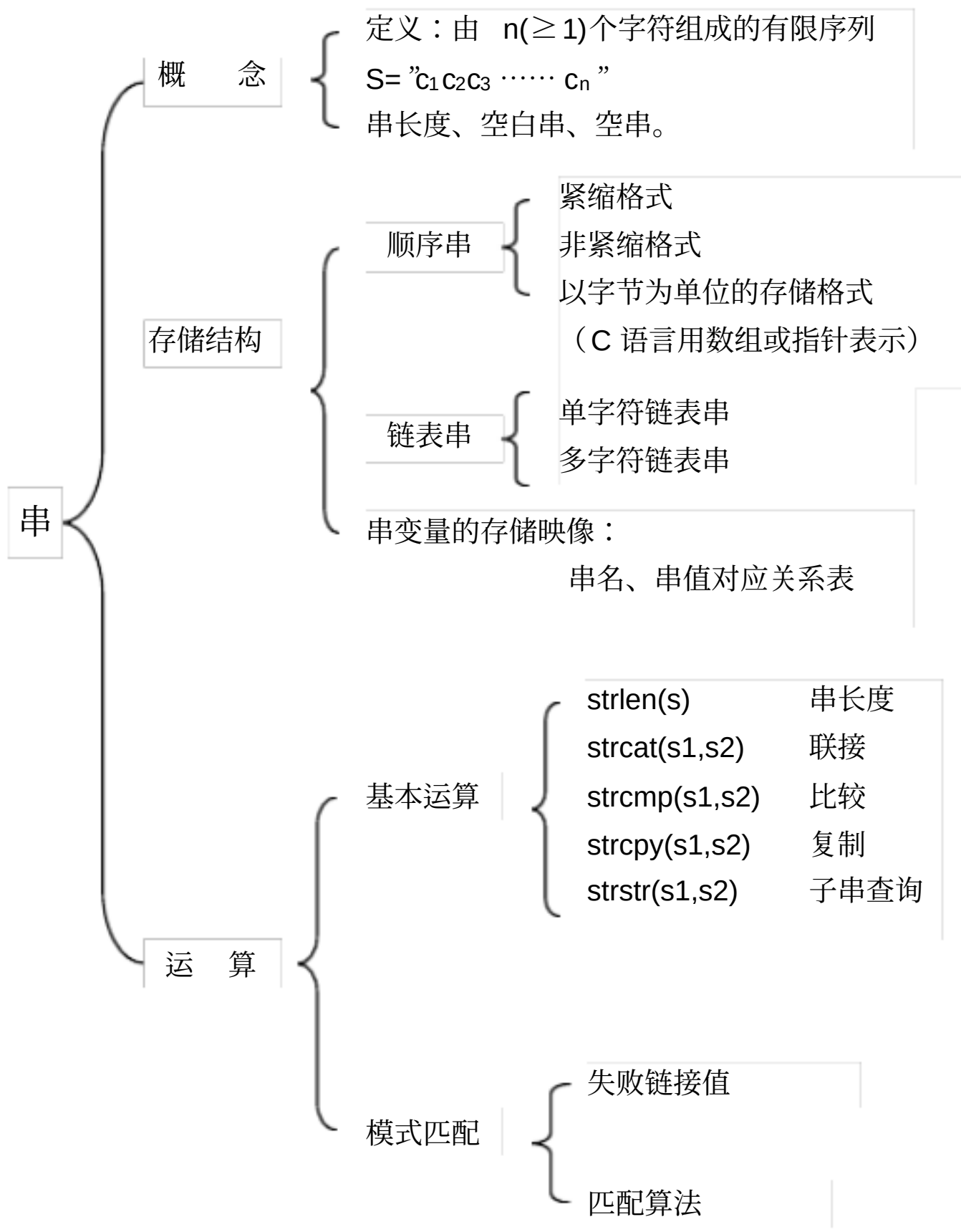
1、 栈和队列的异同点。

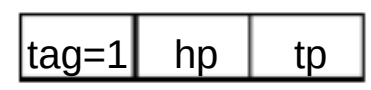
2、 栈和队列的基本运算

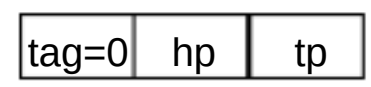
3、 出栈和出队

4、 基本运算

第四章　串　复习


串
概　念
定义：由 n(≥1)个字符组成的有限序列
S= ”c1 c2c3 …… cn ”
串长度、空白串、空串。
存储结构
顺序串
紧缩格式
非紧缩格式
以字节为单位的存储格式
（C 语言用数组或指针表示）
链表串
单字符链表串
多字符链表串
串变量的存储映像：
串名、串值对应关系表
运　算
基本运算
strlen(s) 串长度
strcat(s1,s2) 联接
strcmp(s1,s2) 比较
strcpy(s1,s2) 复制
strstr(s1,s2) 子串查询
模式匹配
失败链接值
匹配算法

- 数组
  - 顺序存储方式
    - 行优先顺序存放
    - 列优先顺序存放
    - C 语言数组：行优先
    - 下标从 [0] 开始，公式变化
  - 压缩存储方式
    - 特殊矩阵
      - 对称矩阵
      - 三角矩阵
      - 对角矩阵
    - 稀疏矩阵
      - 三元组存储：三元组 m n t
      - 链表存储：十字链表
  - 应　用
    - 表达式
    - 程序调用
- 广义表
  - 定义： n(≥0)个元素的有限序列
  - 概念：长度、深度、原子、子表
  - 运　算
    - 表头： Head(A)= a_1
    - 表尾： Tail(A)=(a_2,a_3, ⋯ ,a_n)
  - 存储结构 (链表)
    - 表结点 

    - 原子结点 

第六章　树　复习

- 树
  - 概　　念
    - 定义：递归定义，不为空
    - 双亲、孩子、叶子、兄弟、祖先
    - 树深、结点的度、有序树、无序树
  - 二叉树
    - 二叉树定义：树中结点的度≤　2　有序树
      可为空树（ n=0）
    - 性质
      1. 第 i 层至多有　2^{i-1} 个结点。
      2. 数深为　k 的二叉树，　至多有　2^k-1 个结点。
      3. $n_0=n_2+1$
      4. n　个　结　点　的　二　叉　树　树　深　为　$\lfloor \log_2 n/2 \rfloor +1$
      5. 双亲结点为　i，做孩子结点的编号为 2i，有孩子　2i+1。
    - 存储方式
      - 顺序：满、完全二叉树
      - 链表：二叉、三叉链表
  - 二叉树的遍历
    - 先根遍历序列
    - 中根遍历序列
    - 后根遍历序列

    → 已知先根、中根序列画树；已知后根、中根序列画树；
  - 线索二叉树
    - 先根线索
    - 中根线索
    - 后根线索

    → 线索树的画法
  - 树、森林
    - 树、森林与二叉树的相互转换
    - 树、森林的遍历
  - 树的应用
    - 二叉排序树
      - 左　中　右
      - 小　中　大
    - 哈夫曼树
      - 哈夫曼树的画法
      - 编码：左　0 右 1

1、三个结点可以组成　2 种不同形态的树。

2、一个稀疏矩阵　Am*n采用三元组形式表示，若完成了其的转置运算要经过哪几步　　：

矩阵的行、列数值互换　　、矩阵元素所在行列值互换、元素在矩阵中排列的位置）重新排列

3、若二叉树中每一层结点的个数都达到了最大，则称为一棵满二叉树。

4、树最适合用来表示现有元素之间具有分支层次关系的数据

5、哈夫曼树是带权路径长度最小的二叉树。

6、以下那些项为用十字链表表示的稀疏矩阵元素结点信息元素所在行和列　　、元素的值　　、指向该元素所在行的下一个元素的指针　　、指向该元素所在列的下一个元素的指针。

7、一个广义表可以为其它广义表所共享。

8、广义表可以是一个多层次的结构。

9、压缩存储的三角矩阵和对称矩阵的存储空间相同。

10、广义表中的元素类型可以不相同。

11、两个稀疏矩阵的和仍为稀疏矩阵。

12、二叉树的先序遍历序列中，任意一个结点均处在其孩子结点的前面。

13、对于一棵具有　n 个节点的树，该树中所有节点的度数之和为　　n-1 。

14、树和森林的遍历中有中序遍历。

15、二叉树用链式存储时，空链域数多于非空链域数。

16、由森林转换成二叉树，其根节点的右子树总是空的。

17、哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。

18、当一棵具有　n 个叶子结点的二叉树的　　WPL值为最小时，称其树为　　Huffman 树，且其二叉树的形状必是唯一的。 x

19、某二叉树的先序遍历序列和中序遍历序列相同的二叉树为空树或任一结点均无左孩子的非空二叉树。

20、某二叉树的先序遍历序列和后序遍历序列相同的二叉树为空树或仅有一个结点的非空二叉树。

21、某二叉树的后序遍历序列和中序遍历序列相同的二叉树为空树或任一结点均无左孩子的非空二叉树。　　x

22、某二叉树的先序遍历序列和后序遍历序列相反的二叉树为高度等于结点数的二叉树。

满二叉树就是除叶子结点外的任何结点均有两个孩子结点，且所有的叶子结点都在同一层上的二叉树。

23、用一维数组存放二叉树时，总是以前序遍历存储结点，这是错误的说法

24、在度为　k 的树中，至少有一个度为　　k 的结点。

25、在非空完全二叉树中，只有最下面一层的结点为叶结点。

26、在完全二叉树中，没有左孩子的结点一定是叶子结点。

27、　　特殊矩阵主要形式有对称矩阵、上三角矩阵、下三角矩阵、对角矩阵

28、在结点数目一定的前提下，各种形态的二叉树中，完全二叉树具有最小深度。

29、在所有深度相同的二叉树中，满二叉树具有最大结点数目。

30、给定一组权值，构造出来的哈夫曼树是不惟一的。

31、哈夫曼树中不存在度为　　1 的结点。

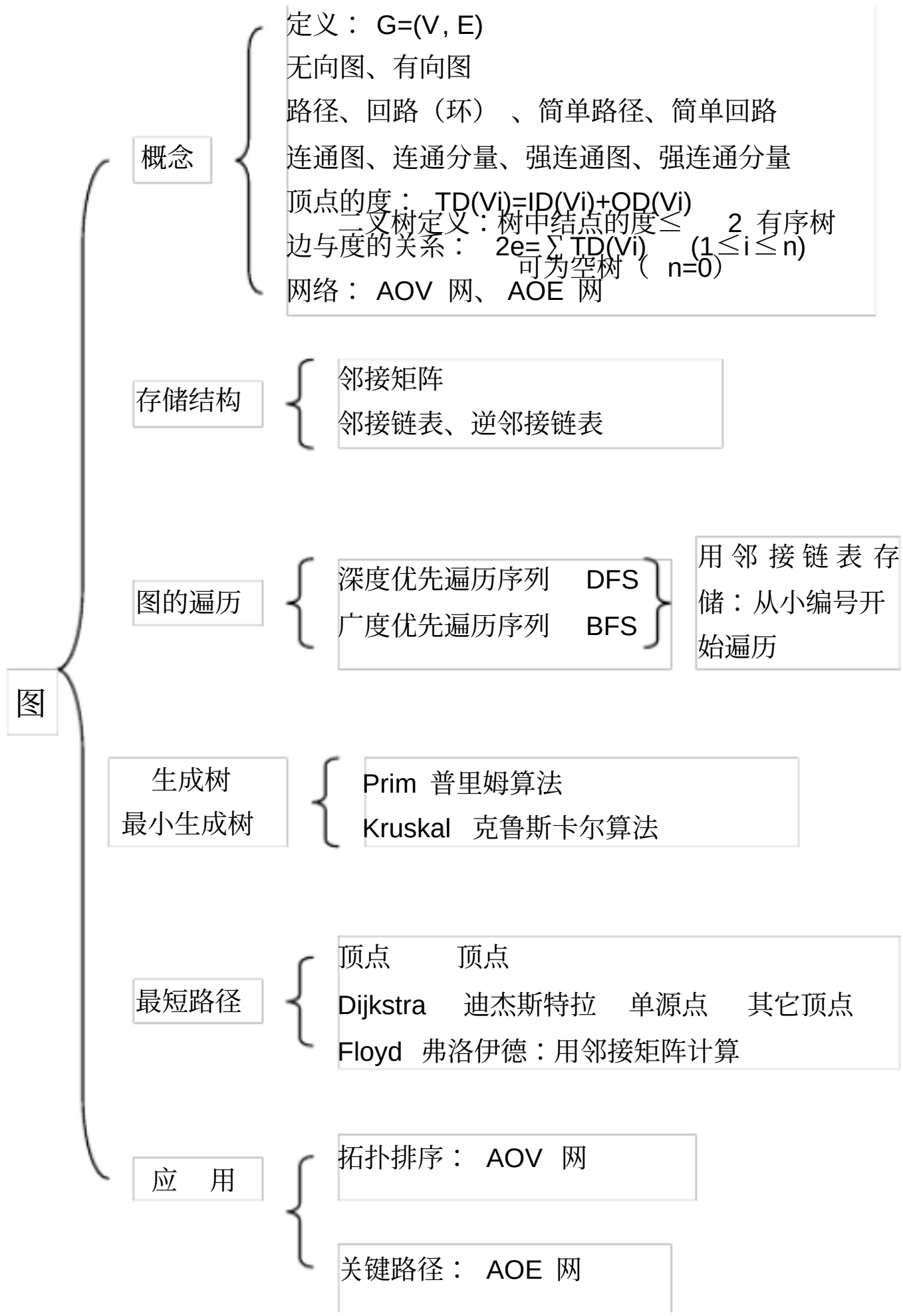
32、线索二叉树中的每个结点通常包含有　　5 个数据成员。

33、判断两个串相等的充分必要条件有两个　　：两个串的长度相等　；两个串上对应位置的字符相同

34、下列哪些是广义表的特性　：层次性　　、共享性　　、递归性

35、稀疏矩阵元素的三元组表示的项　　：元素所在行　　、元素所在列　　、元素的值

第七章　图　复习



1、强连通分量是有向图的极大连通子图。连通分量指的是无向图中的极大连通子图。

2、在一个图中，所有顶点的度数之和等于图的边数的 2 倍。

3、最短路径的生成斯算法可用迪杰斯特拉算法。

4、有向图的邻接表表示适用于求顶点的出度，逆邻接链表适用于求顶点的入度。

5、最小生成树只能是带权连通图的运算。

6、一个有向无环图的拓扑排序的序列是不唯一的。

7、一个图的邻接矩阵表示法是惟一的。一个图的邻接表表示法是不惟一的。

8、若一个有向图的邻接矩阵中对角线以下元素均为零， 则该图的拓扑有序序列必定存在。 若一个有向图中存在回路，则该图的拓扑有序序列不存在。

9、用邻接矩阵法存储一个图所需的存储单元数目与图的边数无关，与顶点数有关。

10、有 n 个顶点的无向图 ， 采用邻接矩阵表示 ，图中的边数等于邻接矩阵中非零元素之和的一半。

11、邻接表中边结点数目为奇数的图一定是有向图。

12、不同的求最小生成树的方法最后得到的生成树是不一定相同的。

13、在一个图中，所有顶点的度数之和等于图的边数的 2 倍。

14、在一个有向图的拓朴序列中，若顶点 a 在顶点 b 之前，则图中必有一条弧 <a,b>，这是不正确的说法。

15、关键路径是从源点到汇点的最长路径。任意 AOE网的关键路径是不唯一的。

16、有向图中一个顶点的度应该是它的出度与入度之和。

17、 n 个顶点的无向图最多有 n(n-1)/2 条边， n 个顶点的有向图最多有 n(n-1) 条边。

18、在无向图中，若顶点 i 到顶点 j 有路径，则这两个顶点之间是连通的。

19、连通图的最小生成树是不惟一的。

20、邻接矩阵主要用来表示顶点之间的关系。 若表示某图的邻接矩阵不是对称矩阵， 则该图一定是有向图。

21、若表示某图的邻接矩阵中出现了全零行或者全零列，则该图一定是非连通图或非强连通图

22、无向图的邻接表中边结点数目一定为偶数。

23、最短路径一定是简单路径。

24、不能对强连通图进行拓扑排序。

25、存储无向图的邻接矩阵是对称的，因此可以只存储邻接矩阵的下（上）三角部分。

26、在 AOE网络中可以有多条关键路径。

27、用边表示活动的网络（ AOE网）的关键路径是指从源点到终点的路径长度最长的路径。

28、对一个连通图进行一次深度优先搜索可以遍访图中的所有顶点。

29、图中各个顶点的编号是人为的，不是它本身固有的，因此可以根据需要进行改变。

30、连通图的广度优先搜索中一般要采用队列来暂存刚访问过的顶点

31、图的深度优先搜索中一般要采用栈来暂存刚访问过的结点

32、有向图的遍历可采用广度优先搜索方法

33、在 AOE网中，减小任一关键活动上的权值后，整个工期也就相应减少，是错误的。

34、 AOE网工程工期为关键路径上的权之和

35、在关键路径上的活动都是关键活动，而关键活动也必须在关键路径上

36、任何一个关键活动提前完成，不一定将使整个工程提前完成

37、求关键路径是以拓扑排序为基础的

38、一个事件的最早开始时间同以该事件为尾的弧的活动最早开始时间相同

39、关键活动一定位于关键路径上

40、在 AOV网中弧表示优先关系，是一种有向无环图， AOV网拓扑排序的结果不惟一确定

41、最小生成树可以用普里姆、克鲁斯卡尔算法。 、

42、表示图的存储结构有（邻接矩阵、邻接表 、邻接多重表 ）。

43、图的深度优先搜索算法类似于二叉树的（前序遍历 ）。

44、具有 n 个顶点、 e 条边的无向图采用邻接矩阵存储方法。则邻接矩阵的大小为（ n×n） 。

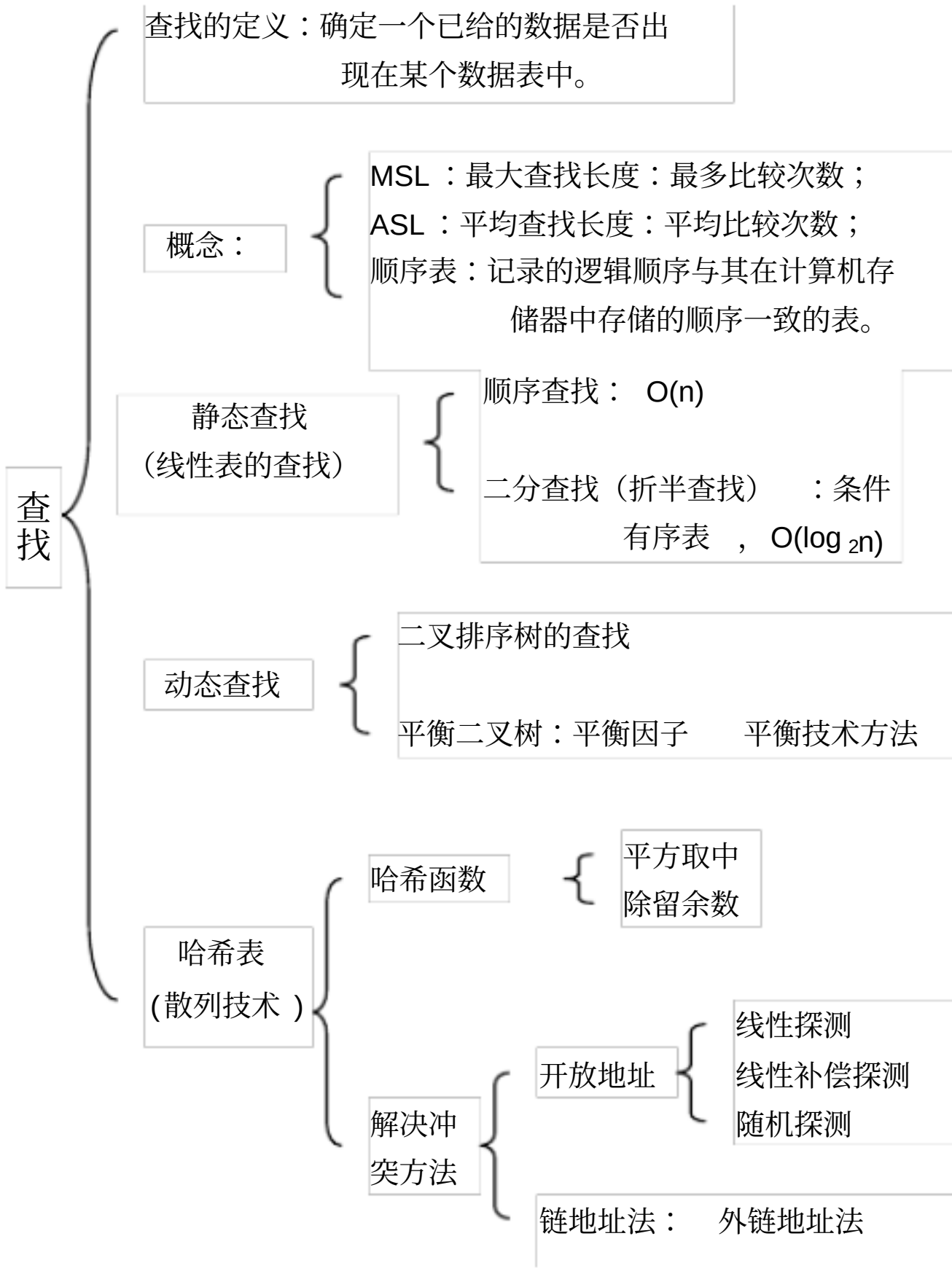
45、图的广度优先搜索算法类似于二叉树的 按层次遍历

46、一个连通图的生成树是包含图中所有顶点的一个极小连通子图

47、任何一个连通图的生成树一棵或多棵

48 邻接表中边结点数目为偶数的图可能是无向图，也可能是有向图。

第八章　查找　复习



1、常用处理冲突的两类方法是开放地址法和拉链法。

2、如在查找的同时对表修改，则相应的表称动态查找表。

3、二叉平衡树又称　AVL树。

4、二分查找法要求待查表的关键字的值必须有序。

5、哈希法是一种将关键字转换为存储地址的存储方法。

6、对有序表而言，采用二分查找总比采用顺序查找法速度快。

7、一般来说，用哈希函数得到的地址，冲突不可能避免，只能尽可能减少。

8、在二叉排序树中插入新结点时，　不必移动其他结点，　仅需改动某个结点的指针，　使它由空变为非空即可。

9、只要按值有序排列的线性表采用顺序存储结构就可以采用折半查找方法。

10、分块查找过程是首先查找索引表，然后再到相应的块中具体查找记录。

11、在散列文件中进行查找不涉及关键字的比较。

12、散列冲突是指同一个关键字对应了多个不同的散列地址。

13、在采用链地址法处理冲突的散列表中，散列函数值相同的关键字是链接在同一个链表中的。

14、装载因子是散列表的一个重要参数，它反映了散列表的装满程度。

15、散列表的查找效率主要取决于处理冲突方法　　、　散列函数　。

16、二分查找要求结点有序、顺序存储

17、散列函数的构造方法有直接定址法　　、数字分析法　　、平方取中法　　、除留余数法等几种。

18、按存储器不同，可以将排序方法分为内部排序、外部排序。

19、对于折半搜索所对应的判定树，它既是一棵二叉搜索树、　　理想平衡树的树　.

20、散列查找是由键值（　　散列函数值）确定散列表中的位置，并进行存储或查找。

21、采用二分查找长度为　n 的线性表时，每个元素的平均查找长度为（　　O(log2n)　　）。

22、采用顺序查找长度为　n 的线性表时，每个元素的平均查找长度为（　　（ n+1）/2　　）。

23、通常把查找过程中对关键字需要执行的（　　ASL　）作为衡量一个查找算法效率优劣的标准。

24、在表长是　N的顺序表中，实施顺序查找，在查找不成功时，与关键字比较的次数（　　N+1　）。

25、如果要求一个线性表既能较快的查找，又能适应动态变化的要求，则采用（分块　　）查找方法。

26、线性表必须是（　　用向量存储的有序表），才能进行二分查找。

27、设有　100 个元素，用折半查找法进行查找时，最小比较次数是（　　1　　）。

27、（ 散列查找）不是利用查找表中数据元素的关系进行查找的方法。

28、采用顺序查找方法查找长度为　n 的线性表时，每个元素的平均查找长度为（　　(n+1)/2　　）

29、哈希表长　m＝14, 哈希函数　H(key)＝key％11。表中已有　4 个结点：addr(15)=4　addr(38)=5　addr(61)=6　addr(84)=7，　其余地址为空　　如果用二次探测再散列处理冲突，关键字为　　49 的结点的地址是　(9　).

30、采用分块查找时，若线性表中共有　　625 个元素，查找每个元素的概率相同，假设采用顺序查找来确定结点所在的块时，每块应分　(　25)　个结点最佳。

31、查找表是以（　　集合　）为查找结构的。

32、在表长是　N的顺序表中，实施顺序查找，在查找不成功时，与关键字比较的次数（　　N+1　）。

33、对于长度为　n 的顺序存储的有序表，若采用折半查找，则对所有元素的查找长度中最大的为　(log2(n+1)　)　　的值向上取整。

34、对于长度为　n 的顺序存储的有序表，若采用折半查找，则对所有元素的查找长度中最大的为　　(log2n)　的值的向下取整加　1。

35、对于长度为　9 的顺序存储的有序表，若采用折半查找在等概率情况下查找成功的平均查找长度为　(25)　除以 9。

36、对于长度为　18 的顺序存储的有序表，若采用折半查找，则查找第　　15 个元素的查找长度为　(4　)　　。

37、在一棵高度为　h 的具有　n 个元素的二叉查找树中，查找一个元素的最大查找长度为　　(h+1　)　　。

38、向具有　n 个结点的二叉查找树中插入一个元素的时间复杂度大致为　　(O(log2n))　　。

39、从具有　n 个结点的二叉查找树中查找一个元素时，在等概率情况下进行成功查找的时间复杂度大致为　(O(log2n))　。

40、向一棵　AVL树插入元素时，可能引起对最小不平衡子树的调整过程，此调整分为　　(4)　种旋转类型。

41、散列函数应该有这样的性质，即函数值应当以（同等）概率取其值域范围内的每一个值。

42、散列地址空间为　0　m-1，k 为表项的关键码，散列函数采用除留余数法，即　　Hash(k)=k%p。为了减少发生冲突的频率，一般取　p 为（小于　m的最大质数）。

43、解决散列法中出现的冲突问题常采用的方法是（　　线性探查法、双散列法、开散列法）　。

44、采用线性探查法解决冲突时所产生的一系列后继散列地址（可以大于或小于原散列地址　　）。

45、 200 个元素采用二分查找时，最大的比较次数是（　　8　　）。

46、为了有效利用散列查找技术，需要解决问题是找一个好的散列函数、设计有效的解决冲突的方法

47、散列表的地址区间为　0　一 16　，散列函数为　Hl (K) = K % 17　　，采用线性探测法解决冲突，将关键字序列　26 , 25 , 72 , 38 , 8 , 18 , 59　　依次存储到散列表中。　，元素　59　存放在散列表中的地址为（ 11　）。

48、具有　4　层结点的二叉平衡树中结点个数至少有（　　7　）。

49、散列查找是由键值（散列函数值　　）确定散列表中的位置，并进行存储或查找。

第九章　排序　复习

排序的定义：　把一批杂乱无章的数据序列重新排列成有序序列的过程。

- 内部排序
  - 插入排序
    - 直接插入排序：　$O(n^2)$　稳定
    - 折半插入排序：　$O(n^2)$　稳定
    - 希尔排序：　$O(n^{1.3})$　不稳定
  - 交换排序
    - 冒泡排序：　$O(n^2)$　稳定
    - 快速排序：　$O(n\log_2 n)$　不稳定
  - 选择排序
    - 简单选择排序：　$O(n^2)$　稳定
    - 堆排序：　$O(n\log_2 n)$　不稳定
  - 归并排序：　二路归并排序：　$O(n\log_2 n)$　稳定

1、直接插入排序的方法要求被排序的数据必须是顺序存储。

2、监视哨的作用是在查找循环条件中用来监视下标变量是否越界。

3、具有相同的关键字的纪录之间的相对次序保持不变的方法是稳定的，否则是不稳定的。

4、堆排序是一种选择排序。

5、每次从无序表中取出一个元素，把它插入到有序表中的适当位置，此中插入的方法叫做插入排序。

6、对于 n 个记录的集合进行冒泡排序，在最坏的情况下需要的时间为　　O(n*n) 。

7、堆是一个键值序列　{k1,k2, …, kn},　对 i=1,2, …,|_n/2_|,　满足 (ki ≤k2i 且 ki ≤k2i+1(2i+1 ≤n))

8、在顺序表　(3, 6, 8, 10, 12, 15, 16, 18, 21, 25, 30)　　中，用折半法查找关键码值　11，所需的关键码比较次数为　(4)

9、当从一个小根堆（最小堆）中删除一个元素时，需要把堆尾元素填补到堆顶位置，然后再按条件把它逐层向下调整，直到调整到合适位置为止。

10、平衡的二叉排序树的任何子树都是平衡的二叉排序树。

11、一棵　m　阶 B　树中每个结点最多有　m-1　个关键码，最少有　　m/2　-1　个关键码。

12、在索引顺序结构的搜索中，对索引表既可以采取顺序搜索，也可以采用折半搜索。

13、堆排序是一种不稳定的排序算法。

14、快速排序算法在每一趟排序中都能找到一个元素放在其最终位置上。

15、对　n　个记录的进行快速排序，所需要的平均时间是　　O (nlog2n)　。

16、具有相同的关键字的记录之间的相对次序保持不变的方法是稳定的，否则是不稳定的。

17、直接选择排序是一种不稳定的排序方法。

18、当输入序列已经基本有序时，起泡排序需要比较关键码的次数，比快速排序还要少。

19、简单选择排序、树形选择排序、堆排序属于选择排序。

20、平均时间复杂度是　O(logn)　的是堆排序、快速排序　　、归并排序　。

21、最坏情况下时间复杂度是　　O(n2) 的是快速排序　　、简单排序　。

22、排序方法稳定的是归并排序　　、计数排序　　、基数排序

23、下列排序方法不稳定的是堆排序　　、希尔排序　　、快速排序

24、堆排序结构分为大顶堆　　、小顶堆。

25、采用顺序查找方法查找长度为　　n 的线性表时，每个元素的平均查找长度为　((n+1)/2).

26、在所有的排序方法中，关键字的比较次数与记录的初始排列次序无关的是（　　选择排序）。

27、目前已比较为基础的内部排序中，其比较次数与待排序纪录的初始排列状态无关的是二分插入排序　　。

28、内部排序与外部排序的区别不在于（是否在内存中排序　　）。

29、排序时扫描待排序记录序列，顺次比较相邻的两个元素的大小，逆序时就交换位置。这是冒泡排序方法的基本思想　.

30、直接选择排序　　是不稳定的排序方法。稳定的排序方法直接插入排序、冒泡排序、归并排序

31、评价排序算法好坏的标准主要是执行时间和所辅助时间。

32、快速排序在（被排序的数据完全无序）的情况下最易发挥其长处。

33、插入排序、快速排序、选择排序、归并排序排序方法中，要求内存量最大是归并排序　　。

34、希尔排序、冒泡排序、选择排序、插入排序排序方法中，平均查找长度最小的是插入排序　　。

35、每次从无序表中取出一个元素，把它插入到有序表中的适当位置，此种排序方法叫做插入排序。

36、直接插入排序的方法要求被排序的数据（必须是顺序　　）存储。

37、对待排序的元素序列进行划分，将其分为左、右两个子序列，再对两个子序列施加同样的排序操作，直到子序列为空或只剩一个元素为止。这样的排序方法是　　(　　快速排序　)　。

38、一个对象序列的排序码为　{46, 79,　56, 38, 40,　84}，采用快速排序（以位于最左位置的对象为基准）所得到的第一次划分结果为　(　{40, 38, 46, 79, 56, 84}　)　　。

39、用快速排序法对包含　n　个关键字的序列进行排序，最坏情况下的排序时间复杂度为　　(0(n2)　　)。