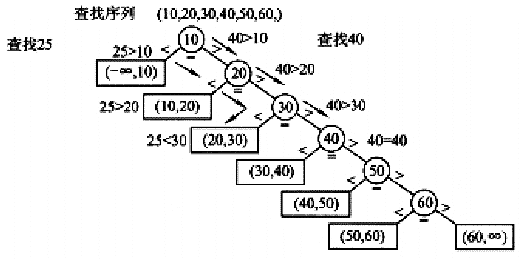
第七章

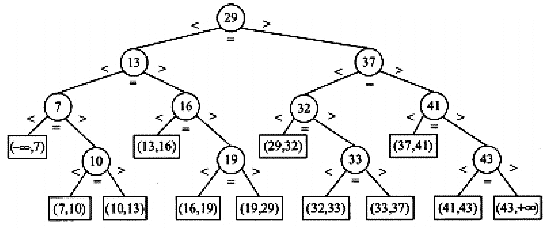
1. 查找的基本概念
2. 查找：在数据集合中寻找满足某种条件的数据元素的过程称为查找；
3. **查找表**（查找结构）：用于查找的数据集合称为查找表，由同一类型的数据元素（或记录）组成；查找表常用操作有4种：①查询某个特定的数据元素是否在查找表中；②检索满足条件的某个特定的数据元素的各种属性；③在査找表中插入一个数据元素；④从查找表中册除某个数据元素；
4. 静态查找表和动态查找表：①若一个査找表的操作只涉及上述操作①和②，则无须动态地修改查找表，此类査找表称为**静态查找表**；②需要动态地插入或删除的查找表称为**动态查找表**；③适合静态查找表的查找方法有顺序査找、折半查找、散列查找，适合动态查找表的查找方法有二叉排序树的查找、散列查找等；
5. 关键字：数据元素中唯一标识该元素的某个数据项的值；
6. 平均查找长度ASL：一次查找的长度是指需要比较的关键字次数，而平均查找长度则是所有查找过程中进行关键字的比较次数的平均值；

2.顺序查找

1） 定义：顺序查找（线性查找），主要用于在线性表中进行查找。顺序查找通常分为对一般的无序线性表的顺序査找和对按关键字有序的顺序表的顺序査找；

2） 一般线性表的顺序查找：①查找成功时平均查找长度：；②查找失败时：；

3） 有序表的顺序查找：①**查找判定树**：树中的圆形结点表示有序顺序表中存在的元素，矩形结点称为失败结点（n个结点的树有n+1个失败节点）；②有序表的顺序查找查找成功的平均查找长度和一般线性表相同；③查找失败时：；

4） 折半查找（二分查找）：①算法思想：将给定值key与表中中间位置的元素比较，相等则查找成功并返回该元素的存储位置，不等，则所需查找的元素只能在中间元素以外的前半部分或后半部分，递归查找；②折半查找判定树： i 若

则对任何一结点必有； ii 一定是平衡二叉排序树，只有最下面一层不满；

5） 分块查找（索引顺序查找）：①基本思想：将查找表分为若干子块，块内元素无序块间有序（第一个块中的最大关键字小于第二个块中的所有记录的关键字）；建立一个索引表，表中每个元素含有各块的最大关键字和各块中的第一个元素的地址，索引表按关键字有序排列；②分块查找过程： i 在索引表中用顺序查找或折半査找确定待查记录所在的块 ii 是在块内顺序查找；③分块查找的平均查找长度为素引查找和块内查找的平均长度之和，；④将长度为n的查找表均匀地分为b块，每块有s个记录，若，则平均查找长度取最小值；

3. B树和B+树

1） B树（多路平衡查找树）的概念：①B树的阶：B树中所有结点的孩子个数的最大值；②一棵m阶B树或为空树，或为满足如下特性的m叉树： i 树中每个结点至多有m棵子树，即至多含有m-1个关键字； ii 若根结点不是终端结点，则至少有两棵子树（绝对平衡）； iii 除根结点外的所有非叶结点至少有棵子树，即至少含有个关键字； iv 非叶子结点中的关键字严格递增，且指针Pi所指子树中所有节点必须位于(Ki,Ki+1)的区间内； v 所有的叶结点都出现在同一层次上，并且不带信息（带信息的末端结点称为终端节点）；

2） B树的高度：B树的高度不包括不含信息的叶节点层； 对任一棵包含n(n≥1)个关键字、高度为h、阶数为m的B树：①关键字个数应满足：，其中是每个结点最多有多少关键字，是每层最多有多少个节点，因此有；②关键字个数为n的B树，叶结点即查找不成功的结点为n+1；③除根结点外的每个非终端结点至少有棵子树，则第三层至少有个结点…第h+1层（叶结点层）至少有个结点，从而；

3） B树的查找：①B树的查找包含两个基本操作i在B树中找结点（在磁盘中，将结点信息读入内存）ii在结点内找关键字（在内存中，顺序查找）；②查找到某个结点后，在结点有序表中查找，未找到则按照对应的指针信息到所指的子树中去查找，查找到叶节点则失败；

4） B树的插入：①定位：找到插入关键字的最底层的某个非叶子结点；②插入结点后，若导致原结点关键字数超过上限，则从中间位置（）将关键字分为两部分，左部分在原结点中，右部分放到新结点中，中间位置插入原结点的父结点（原指针的右侧）；③若其父结点的关键字个数也超过了上限，则继续进行这种分裂操作，直至这个过程传到根结点为止；

5） B树的删除：①若被删除关键字茌终端节点，则直接删除该关键字；②若被除关键字在非终端节点，则用直接前驱或直接后继来替代被删除的关键字（直接前驱：当前关键字左侧指针所指子树中“最右下”的元素；直接后继：当前关键字右侧指针所指子树中“最左下”的元素）；③当右兄弟很宽裕时，用当前结点的后继、后继的后继来填补空缺；当左兄弟很宽裕时，用当前结点的前驱、前驱的前驱来填补空缺；④兄弟不够借，则将关键字删除后与左（或右）兄弟结点及双亲结点中的关键字进行合并；在合并过程中，双亲结点中的关键字个数会减1，若双亲结点为根节点且关键字个数减少至0，则直接将根结点除，合并后的新结点成为根；若双亲结点不是根结点，且-1后不满足要求，向上重复上述步骤直到满足要求；

6） B+树的概念：①每个分支结点最多有m棵子树；②非叶根结点至少有两棵子树，其他每个分支结点至少有棵子树；③结点的子树个数与关键字个数相等；④所有叶结点包含全部关键字及指向相应记录的指针，叶结点中关键字按顺序排列，并且相邻叶结点按顺序相互链接（可顺序查找）；⑤所有分支结点（可视为索引的索引）中仅包含它的各个子结点（即下一级的索引块）中关键字的最大值及指向其子结点的指针；

7） B树与B+树的区别：①B+树中，具有n个关键字的结点只含有n棵子树；B树中，具有n个关键字的结点含有n+1棵子树；②B+树中结点关键字个数n的范围是 (根结点)；B树中结点关键字个数n的范围是 (根结点)；③B+树中仅叶结点包含信息，所有非叶结点起索引作用；④B+树中叶结点包含了全部关键字；B树中叶结点和分支结点包含的关键字是不重复的；

4. 散列表

1） 散列表的基本概念：①散列函数：一个把查找表中的关键字映射成该关键字对应的地址的函数；②冲突：散列函数把两个或两个以上的不同关键字映射到同一地址；③同义词：发生冲突的不同关键字称为同义词；④散列表：根据关键字而直接进行访问的数椐结构（散列表建立了关键字和存储地址之间的一种直接映射关系）；

2） 散列查找：①计算散列地址Addr=Hash(key)；②检查Addr是否有记录，无记录返回查找失败，有记录比较是否相等，相等返回查找成功；③用给定处理冲突方法计算下一个散列地址，把Addr置为此地址，转为②；

3） 查找效率：①査找效率取決于三个因素：散列函数、处理冲突的方法、装填因子（）；②散列表的平均查找长度依赖于散列表的装填因子；③越大，装填的记录越“满”，发生冲突的可能性越大；

4） 散列函数：①除留余数法：散列表表长为m，取一个不大于m但最接近或等于m的质数p，，简单常用；②直接定址法：，计算最简单，且不会产生冲突，适合关键字的分布基本连续的情况；③数字分析法：选取数码分布较为均匀的若干位作为散列地址，适合于已知的关键字集合（如手机号后4位）；④平方取中法：取关键字的平方值的中间几位作为散列地址，这种方法得到的散列地址与关键字的每位都有关系，适用于关键字的每位取值都不够均匀或均小于散列地址所需的位数（如身份证号）；

5） 冲突处理：①开放定址法：线性探测法、平方探测法、再散列法、伪随机序列法；②拉链法；