```
概论 ⊙
            线性表 •
            有限制的线性表 ®
            串 ⋅
            树 ☉
            图 0
                              ASL成功=(n+1)/2
                       顺序查找 ⊙
                             ASL失败=n+1
                              有序线性表ASL失败=n/2 + n/(n+1)
                       折半查找 ○ ASL成功=log2 (n+1)-1
                              块有序,块内无序
                      分块查找 ⊙
                                    块内和块间都顺序查找 ○ 根号(n)+1
                             B树中所有结点的孩子结点数的最大值称为B树的阶
                                一棵m阶B树,树中每个结点至多有m棵子树,至多有m-1个关键
                           定义 ⊖
                                若根结点不是终端结点,则至少有2棵子树
                                除根结点外的所有非叶子结点至少有[m/2]上界棵子树
                                所有叶子结点都出现在同一层次上, 不带任何信息
                                1.在B树上查找结点
                           杳找 ⊙
                               2.在结点上找关键字
                                定位最底层的某个非叶子结点
                                插入,如果没有破坏B树定义,则不修改
                           插入。
                                                          插入后的结点的中间位置[m/2]关键字并入父结点
                      B树 🤄
                                若插入后,关键字数量大于m-1,则进行分裂操作。 中间结点左侧结点留在原先的结点中,右侧结点放入新的结点中,
                                                          若并入父结点后,父结点超出范围,则继续向上分裂
                                                  若被删除关键字个数依然满足>[m/2]-1,则表明删除后仍满足定
                                                  若删除后=[m/2]-1,而此结点临近的结点关键字>=[m/2],则从
                                                  兄弟结点借一个关键字,并调整该结点、双亲结点和兄弟结点的
                                删除终端结点(底层非叶子结点) 3
                                                  关键字
                                                  若删除后兄弟不够借,则删除关键字,并与一个不够借的兄弟结
            查找 ○ 查找 ○
                                                  点和双亲结点中两兄弟子树中间的关键字合并。合并后如果双亲
数据结构
                                                  结点因减少一个结点不符合定义,则继续合并。
                           删除(
                                          若小于k的子树中关键字字数>[m/2]-1,则找出k的前驱值,和k交
                                          换位置,然后递归的删除k
                                          若小于k的子树中关键字字数>[m/2]-1,则找出k的后继值,和k交
                                删除非终端结点 🌣
                                          换位置,然后递归的删除k
                                          若前后两子树关键字均为[m/2],则直接合并两个子结点,删除k
                                B+树中所有结点的孩子结点数的最大值称为B+树的阶
                                一棵m阶B+树,树中每个结点至多有m棵子树
                                若根结点不是终端结点,则至少有2棵子树
                                除根结点外的所有非叶子结点至少有[m/2]上界棵子树,子树和关
                      B+树 ⊙ 定义 ⊙
                                键字个数相等
                                所有叶子结点包含全部关键字和指向相应记录的指针。叶子结点
                                按关键字按大小顺序排列,并将相邻节点按大小顺序连接起来
                                所有分支结点中仅包含他的各个子结点中关键字的最大值和指向
                                其子结点的指针
                            散列函数: 把查找表中关键字映射为改关键字对应的地址的函数
                            散列表:根据关键字而直接进行访问的数据结构。建立了关键字
                            和存储关系之间的直接映射关系
                            冲突: 散列函数可能把多个不同的关键字映射到同一地址下
                                   直接定址法 ② 没有冲突,当关键字不连续时比较浪费空间
                                   除留取余法 ○ 选择除数是关键,可以减少冲突。比如选质数
                                           选取关键字中等概率均匀的几位作为散列地址计算。但如果关键
                            散列函数 ⊙ 数字分析法 ⊙ 字改变,需要重新构造
                                   平方取中法 ○ 取关键字平方值的中间几位作为散列地址
                                         将关键字分割成位数相同的几部分,然后取这几部分的叠加和作
                                   折叠法 ○
                                         为散列地址
                      散列表
                                            可存放新表项的空袭那地址既向同义词开放,也向非同义词开放
                                                  线性探查法 ○ d=0,1,2,3,...m-1
                                                  平方探测法 ○ d=0^2,1^2,-1^2...k^2,-k^2
                                   开放定址法 ○ 增量序列 ○
                                                  再散列法 ⊙ di=i*Hash2(key)
                                                  伪随机法 ⊙ di=伪随机序列
                            外理冲突 €
                                           弊端:不能随便删除某个元素
                                         把所有同义词放在一个线性链表中,散列表中每个单元存放该链
                                         表头指针
                                   拉链法 ♀
                                         适用于经常进行插入和删除的情况
                            查找效率 ○ 填装因子 ○ 等于表中记录数/散列表长度。平均查找长度依赖于填装因子
            排序 •
```