1. 查找

8.1查找表的基本概念

1. 用于查找的数据集合称为查找表，是由同一数据类型的对象所组成
2. 每个对象的某个属性的值可以唯一标识这个对象，称为关键码

//现实应用可能确实没有唯一关键码，也可以基于其他属性查找，但是结果不唯一

1. 查找的不同环境
   1. 静态查找表：查找表在插入和删除等操作的前后不发生改变
   2. 动态查找表：数据在操作前后会发生改变
2. 算法评价指标：**最多次数、平均次数**、最少次数

8.2静态表的查找

1. 顺序查找：遍历
2. 哨兵思想：在顺序表表头0位置存储需要寻找的值，然后从表尾开始向表头查找，这样无论如何都会找到而停止，就不需要判断指针是否走到头。（空间换时间）
3. 查找成功：最少次数1；最多次数n；平均次数（n+1）/2
4. 有序查找：二分
5. 环境：n个对象存放在一个有序顺序表中并已经按关键码从小到大排好了序
6. 二分思想：将待查区间分为两半（下标的中间值），查找时确定搜索值在左半区间还是右半区间
7. 二叉搜索树：搜索成功时指针停留在树中的某个节点，搜索失败时停留在树外的节点
8. 索引查找：嵌套

先查找索引表，再查找索引之间的顺序表

8.3动态查找表

1. 二叉排序树（BST - Binary Search Tree）
2. 二叉排序树定义：
   1. 若左子树不空，则左子树所有节点都小于其根节点
   2. 若右子树不空，则右子树所有节点都大于其根节点
   3. 它的左右子树也分别是二叉排序树
3. 二叉排序树的查找：

若小于根节点则在左子树中递归重复，若大于则在右子树中递归重复，指导进行到查找成功或者进行到一个空子树为止

1. 同样的数据输入顺序不同，会导致排序树不同，我们应该尽量减少排序树的层次高度，如单支树是我们不想要看到的情况
2. 平衡二叉树（AVL - Adelson Velskii Landis三位数学家）
3. 定义：AVL树是具有以下特征的二叉排序树
   1. 左子树与右子树的高度差绝对值不大于1
   2. 根的左子树与右子树都是AVL树

//这样的树更符合我们的要求，能够使得树的层次尽可能少

1. 平衡化旋转

在插入节点时就开始准备旋转

单左旋、单右旋、先左旋后右旋、先右旋后左旋

旋转之后要保持原来的大小关系，因为要保持它仍然是一个二叉排序树

1. B树
2. B树的性质：
   1. B树总是平衡的，所有的叶节点都在同一层
   2. B树把相近的值存储到同一磁盘页中，有利于符合访问局部性进行操作
   3. B树保证了树中内部的孩子个数最小，也即保证了关键字个数最少
3. B树的定义：B树是具有以下性质的m树
   1. 根是一个叶节点或者至少有两个孩子
   2. 除了根或叶的每个分支节点都有个孩子
   3. 所有叶节点在树的同一层并且不含任何信息（视作外部节点或查找失败）
4. B+树
   1. B+树的节点至少是半满
   2. 为了提高空间利用率，也存在B\*树，使得节点为2/3满

8.4 哈希表

一、哈希表的定义

1、根据哈希函数 H( key ) 和所选处理冲突的方法，将一组关键字映象到一个有限的、地址连续的地址集 (区间)上，并以关键字在地址集中的“映象”作为相应记录在表中的存储位置，如此构造所得的查找表称为“哈希表”（散列表）。

2、很难找到一个不产生冲突的哈希函数。一般情况下，只能选择恰当的哈希函数，使冲突尽可能少地产生。一般还需要处理非单值对应的冲突情况。

1. 构造哈希函数

1、原则：采用何种构造哈希函数的方法取决于建表的关键字集合的情况(包括关键字的范围和形态)，总的原则是使产生冲突的可能性降到尽可能地小。

2、造表方法：

直接定址法、数字分析法、平方取中法、折叠法、**除留余数法、随机数法**

//其中随机数法事实上是“伪随机数”，random(key)，即当key值相同时，总能产生相同的随机数

1. 处理冲突的方法

//“处理冲突”的实际含义就是“为冲突的地址寻找下一个哈希地址”

1. 开放定址法（闭域法）

为产生冲突的地址H(key)求得一个新地址序列Hi

H0 = H(key)

Hi = ( H(key) + di ) MOD m i=1, 2, …, s

Di的取法：

1)线性探测再散列  
 di = c×i 最简单的情况 c=1

2) 平方探测再散列  
 di = 12, -12, 22, -22, …,

3) 随机探测再散列  
 di 是一组伪随机数列

4)双散列探测再散列

di=i×H2(key)

1. 链地址法（开域法）

将所有哈希地址相同的记录都连接在同一链表中，即每个元素都储存为一个链表节点，将哈希地址相同的节点逐个链接在链表节点后面

四、哈希表的查找

1、查找过程：

对于给定值 K，计算哈希地址 i = H(K)或“求下一地址 Hi”，直至

r[Hi] = NULL (查找不成功) 或 r[Hi].key = K (查找成功) 为止

2、决定哈希表查找的ASL的因素

1) 选用的哈希函数；

2) 选用的处理冲突的方法；

3) 哈希表饱和的程度，即装载因子 α=n/m 值的大小（n—记录数，m—表的长度）