1. 二叉排序树
2. 二叉排序树的定义

是一个递归定义：

·空树是二叉树

·满足以下条件的非空二叉树也是二叉排序树

若左子树不空，则左子树上所有结点的值均小于根结点的值；

若右子树不空，则右子树上所有结点的值均大于根结点的值；

它的左、右子树也都分别是二叉排序树

1. 二叉排序树的中序有序
2. 二叉排序树的插入(假设插入q)

先进行查找，没查找到(这时返回没查找到的最近一个结点p)→分情况讨论①p为空，说明为空树，则插入结点作为根节点;②q->data>p->data，插入为右结点；③q->data<p->data插入为左结点。

1. 二叉排序树的删除
   1. 删除结点为叶子结点，则直接删除；
   2. 删除结点只含左子树或只含右子树，孩子直接继承
   3. 删除结点既含左子树右含右子树，则在左子树找继承结点
2. 支持重复数据的二叉排序树，将重复数据放在重复数据结点的右子树上
3. 二叉排序树不唯一
4. 平均查找时间T(n)=O(height)
5. 平衡二叉树
6. 定义:在二叉树的条件下，满足:左右子树深度差不超过1
7. AVL树(严格平衡二叉树)

通过旋转保持删除前后的平衡(两种情况:单旋转和双旋转)

1. 红黑树
2. 哈希表(又叫散列表)
   1. 关键在于哈希函数，通过哈希函数缩小存储空间

由于哈希函数是一个压缩映像，因此会产生“冲突”，即:不同的地址对应同一个映像。需要一个解决冲突的办法。

* 1. 通过哈希函数及其解决冲突的方法，将一组关键字映射到一段连续的地址上的过程。被成为“哈希造表”或散列。
  2. 常见的哈希函数构造方法:

·直接定址法: 哈希函数为关键字的线性函数.

适合于：地址集合的大小 = = 关键字集合的大小

·数字分析法: 分析关键字集中的全体，并从中提取分布均匀的若干位或它们的组合作为地址。

适合于:能预先估计出全体关键字的每一位上各种数字出现的频度。

·平方取中法: 关键字的平方值的中间几位作为存储地址。

依据: 平方运算扩大差别,中间各位能受到整个关键字中各位的影响。

·折叠法:

·除留余数法

 其中，p≤m (表长),且p应为不大于m的素数或是不含 20 以下质因子的合数。

·随机数法

* 1. 处理冲突的方法

·开放地址法:将冲突的地址进行求得一个地址序列



增量有三种取法: