

## 3.6 查找算法



# 常见的查找算法

- 1、顺序查找
- 2、折半查找（二分查找）
- 3、索引顺序查找（分块查找）
- 4、树表查找
- 5、哈希查找



# 查找表及查找效率

- **查找表定义**：是指由同一类型的数据元素（或记录）构成的集合。
- **静态查找表**：只进行“查询”和“检索”操作。
- **动态查找表**：除了查询和检索外可能还会进行插入和删除操作。
- **关键字**：是数据元素(或记录)的某个数据项的值，用它来识别(标识)这个数据元素(或记录)。
- **查找**：根据给定的某个值，在查找表中确定是否存在一个其关键字等于给定值的记录或数据元素。
- **平均查找长度(ASL)**：对于查找算法来说，其基本操作是“将记录的关键字与给定值进行比较”。因此，通常以“其关键字和给定值进行比较的记录个数的平均值”作为衡量查找算法好坏的依据。



# 顺序查找

从表中的一端开始，逐个进行记录的关键字和给定值的比较，若找到一个记录的关键字与给定值相等，则查找成功；若整个表中的记录均比较过，仍未找到关键字等于给定值的记录，则查找失败。

7	14	3	10	6	12	1	15	9	13	2	5	8	4	11
---	----	---	----	---	----	---	----	---	----	---	---	---	---	----

- 查找效率低，但是算法简单且适应面广。



# 折半查找(二分查找)

- 基本思想：先令查找表中间位置记录的关键字和给定值比较，若相等，则查找成功；若不等，则缩小范围，直至新的查找区间中间位置记录的关键字等于给定值或者查找区间没有元素时（查找不成功）为止。
- 这种方法要求查找表进行顺序存储并且按关键字有序排列。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

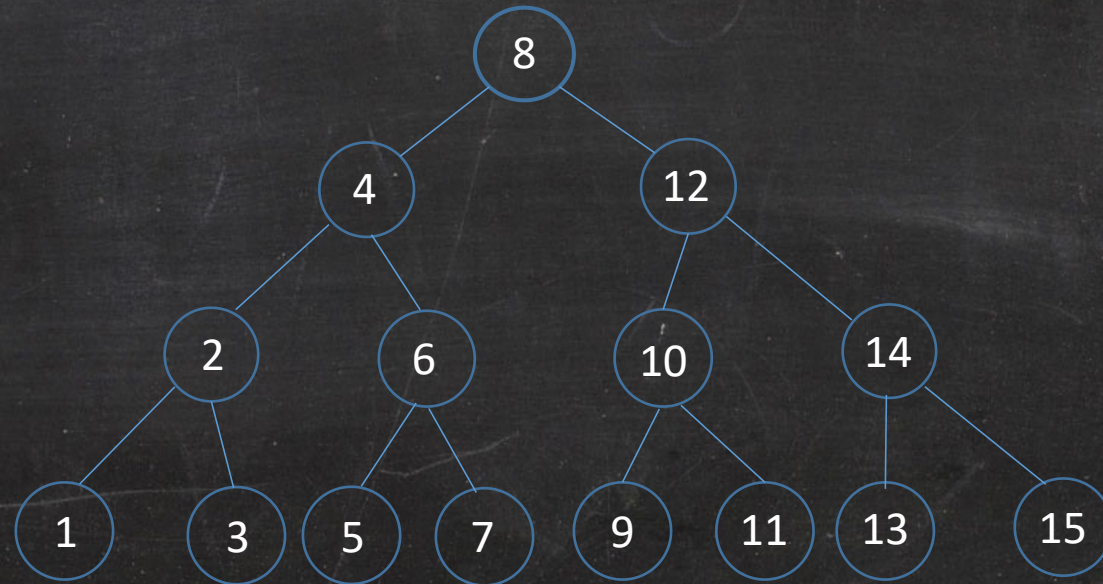
- 查找效率比顺序查找要高，但要求关键字有序排列。
- 适用于表不易变动，且又经常进行查找的情况。



# 折半查找判定树

- 从折半查找的过程来看，可以用一棵二叉树来描述。通常称这个描述折半查找二叉树的过程称为折半查找判定树。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----



不含外部节点的折半查找判定树



# 树表查找

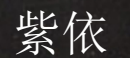
- 二叉查找树是一种动态查找表，表结构本身是在查找过程中动态生成的。
- 设关键字序列为 {46, 25, 54, 13, 29, 91}



构造的二叉查找树



- 是对顺序查找的一种改进
- 将表分成若干块，每一块中的关键字不一定有序，但块之间是有序的。





# 哈希查找

- 前面几种查找方法中，记录的存储位置和关键码之间不存在确定的关系，因此查找时都要建立在对关键字进行比较的基础之上。
- **哈希函数**：关键字作为自变量，关键字存放的地址作为因变量。
- $H_i = \text{Hash}(\text{Key})$
- 这一映射过程称为**哈希造表或散列**，所得的存储位置称为**哈希地址或散列地址**。
- **冲突**：对于某个哈希函数Hash和两个关键字 $K_1$ 和 $K_2$ ，如果 $K_1 \neq K_2$ 而 $\text{Hash}(K_1) = \text{Hash}(K_2)$ ，则称出现了冲突。
- 采用哈希法主要考虑的两个问题是**哈希函数的构造**和**冲突的解决**。



# 哈希查找

- 冲突的处理：

- (1) 开放定址法：

$$H_i = (\text{Hash}(\text{key}) + d_i) \% m \quad i=1, 2, \dots, k \quad (k \leq m-1)$$

- 其中， $\text{Hash}(\text{key})$ 为哈希函数， $m$ 为哈希表的表长， $d_i$ 为增量序列。

常见的增量序列有如下三种：

- $d_i=1, 2, 3, \dots, m-1$ ，称为线性探测再散列。
- $d_i=1^2, -1^2, 2^2, -2^2, 3^2, \dots, \pm k^2$  ( $k \leq m/2$ )，称为二次探测再散列。
- $d_i$ 为伪随机序列，称为随机探测再散列。



例：设关键码序列为47，34，19，12，52，38，33，57，63，21，哈希表表长为13，哈希函数为 $\text{Hash}(\text{key}) = \text{key} \bmod 11$ ，用线性探测法解决冲突构造哈希表。

哈希地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键码													



# 哈希查找

- (2) 链地址法：它将具有相同哈希函数值的记录组织成一个链表，当链域的值为NULL时，表示已没有后继记录。
- 地址里存放的是指针，而不是关键字，将哈希函数值相同的记录组成一个链表



例：哈希表表长为11，哈希函数为 $\text{hash}(\text{key}) = \text{key} \bmod 11$ ，对于关键码序列为47，34，12，52，38，33，27，3，使用链地址法构造哈希表。

