

## 11.1 散列的基本思路

---

编译中对变量的管理问题：

1. 查找
2. 插入
3. 删除

已知的查找方法

1. 顺序查找  $O(N)$
2. 二分查找（静态查找）  $O(\log_2 N)$
3. 二叉搜索树  $O(h)$  树高  $h$
4. 平衡二叉树  $O(\log_2 N)$

查找的本质：

1. 有序安排对象：全序、半序
2. 直接算出对象位置：散列

散列查找法的两项基本工作

1. 计算位置：构造散列函数确定关键词存储位置
2. 解决冲突：应用某种策略解决多个关键词位置相同的问题

时间复杂度几乎是常量： $O(1)$ ，即查找时间与问题规模无关

## 11.2 什么是散列表

---

类型名称：符号表（*SymbolTable*）

数据对象集：符号表是“名字 $name$  – 属性 $attribute$ ”对的集合

操作集： $Tbale \in SymbolTable, Name \in NameType, Attr \in AttributeType$

1. *SymbolTable* InitializeTable(int TableSize)：创建一个长度为 *TableSize* 的符号表
2. Boolean IsIn(*SymbolTable* Table, *NameType* Name)：查找特定的名字 *Name* 是否在符号表 *Table* 中
3. *AttributeType* Find(*SymbolTable* Table, *NameType* Name)：获取 *Table* 中指定名字 *Name* 对应的属性
4. *SymbolTable* Modelfy(*SymbolTable* Table, *NameType* Name, *AttributeType* Attr)：将 *Table* 中指定名字 *Name* 的属性修改为 *Attr*
5. *SymbolTable* Insert(*SymbolTable* Table, *NameType* Name, *AttributeType* Attr)：向 *Table* 中插入一个新名字 *Name* 及其属性 *Attr*
6. *SymbolTable* Delete(*SymbolTable* Table, *NameType* Name)：从 *Table* 中删除一个名字 *Name* 及其属性

装填因子：设散列表空间大小为  $m$ ，填入表中的元素个数是  $n$ ，则称  $\alpha = n/m$  为散列表的装填因子

散列（*Hashing*）的基本思想是

1. 以关键字  $key$  为自变量，通过一个确定的函数  $h$ （散列函数），计算出对应的函数值  $h(key)$ ，作为数据对象的存储地址
2. 可能不同的关键字会映射到同一个散列地址上，即  $h(key_1) = h(key_2)$ ，且  $(key_1 \neq key_2)$ ，称为冲突（*Collision*），需要某种冲突解决策略