# 7.4散列表（下）

- **处理冲突的方法**
  - 开放定址法
    - 增量的取值方法
      - 数学递推公式 $H_i = (H(\text{key}) + d_i)\ \%\ m$
        - H（key）为散列函数
        - m表示散列表表长;di为增量序列
      - 线性探测法 $d_i = 0, 1, 2, \cdots, m-1$
        - 当出现冲突时，就会顺序的向下一个单元探测，直到单元没有发生冲突
        - 优点：简单 容易实现
        - 缺点：会出现聚集现象，降低查找效率
      - 平方探测法 $d_i = 0^2, 1^2, -1^2, 2^2, -2^2, \cdots, k^2, -k^2$
        - 优点：可以避免出现"堆积"问题,
        - 缺点：不能探测到散列表上的所有单元,但至少能探测到一半单元
      - 再散列法 $d_i = \text{Hash}_2(\text{key})$
        - 当通过第一个散列函数H(key）得到的地址发生冲突时,则利用第二个散列函数Hash(key)计算该关键字的地址增量
        - 计算公式公式 $H_i = (H(\text{key}) + i \times \text{Hash}_2(\text{key}))\ \%\ m$
      - 伪随机序列法 $d_i =$ 伪随机数序列时,称为伪随机序列法
  - 拉链法
    - 把所有的同义词存储在一个线性链表中,这个线性链表由其散列地址唯一标识
    - 所有同义词就像被拉链串在一起一样
- **散列查找及性能分析**
  - 实现过程
    - ①检测查找表中地址为Addr的位置上是否有记录,若无记录，返回查找失败；若有记录，比较它与key的值，若相等，则返回查找成功标志，否则执行步骤②
    - ②用给定的处理冲突方法计算"下一个散列地址",并把Addr置为此地址,转入步骤①
  - 特点
    - 平均查找长度作为衡量散列表的查找效率的度量
    - 散列表的查找效率取决于三个因素:散列函数、处理冲突的方法和装填因子
    - 装填因子 $\alpha = \frac{\text{表中记录数}n}{\text{散列表长度}m}$
      - 装填因子越大越容易发生冲突