散列表，是一种基于快速存取的角度设计，也是一种典型的“空间换时间”的做法，该结构可以理解为一个线性表，但是其中的元素不是紧密排列的，而是可能存在空隙。

散列函数：把关键码值映射到表中一个位置来访问记录，以加快查找的速度。存放记录的数组叫做散列表。

比如我们存储了70个元素，但可能为这70个元素申请了100个元素的空间，70/100=0.7，这个数字称为负载因子。

解决冲突是一个复杂问题。冲突主要取决于：
1.散列函数，一个好的散列函数的值应该尽可能平均分布。
2.处理冲突方法。

1.线性探查法：冲突后，线性向前试探，找到一个最近的空位置。缺点是会出现堆积现象。存取时，可能不是同义词的词也位于探查序列，影响效率。

2.双散列函数法：在位置d冲突后，再次使用另一个散列函数产生一个与散列表桶容量m互质的数c，依次试探（d + n*c）%m，使探查序列跳跃式分布。

常见的散列函数：
1.直接寻址法：取关键字或关键字的某个线性函数值为散列地址。H（key）=key或H（key）=a * key + b，其中a和b为常数。
2.数字分析法：分析一组数据，找出数字的规律，尽可能利用这些数据来构造冲突几率较低的散列地址。
3.平方取中法：取关键字平方后的中间几位作为散列地址。
4.折叠法：将关键字分割为位数相同的及部分，最后一部分位数可以不同，然后取这几部分的叠加和作为散列地址。
5.随机法：选择一随机数，取关键字的随机值作为散列地址，通常用于关键字长度不同的场合。
6.除留余数法：取关键字被某个不大于散列表表长m的数p除后所得的余数为散列地址。

查找过程中，关键码的比较次数，取决于产生冲突的多少，产生的冲突少， 查找的效率就高，产生的冲突多，查找效率就低。因此，影响产生冲突多少的因素，也就是影响查找效率的因素。影响产生冲突多少有以下三个因素。
1.散列函数是否均匀。
2.处理冲突的方法。
3.散列表的装填因子。 装填因子=填入表中的元素个数/散列表的长度。