开放地址法：

1.线性探测法

线性再散列法是形式最简单的处理冲突的方法。插入元素时，如果发生冲突，算法会简单的从该槽位置向后循环遍历hash表，直到找到表中的下一个空槽，并将该元素放入该槽中（会导致相同hash值的元素挨在一起和其他hash值对应的槽被占用）。查找元素时，首先散列值所指向的槽，如果没有找到匹配，则继续从该槽遍历hash表，直到：（1）找到相应的元素；（2）找到一个空槽，指示查找的元素不存在，（所以不能随便删除元素）；（3）整个hash表遍历完毕（指示该元素不存在并且hash表是满的）

用线性探测法处理冲突，思路清晰，算法简单，但存在下列缺点：
① 处理溢出需另编程序。一般可另外设立一个溢出表，专门用来存放上述哈希表中放不下的记录。此溢出表最简单的结构是顺序表，查找方法可用顺序查找。
② 按上述算法建立起来的哈希表，删除工作非常困难。如果将此元素删除，查找的时会发现空槽，则会认为要找的元素不存在。只能标上已被删除的标记，否则，将会影响以后的查找。
③ 线性探测法很容易产生堆聚现象。所谓堆聚现象，就是存入哈希表的记录在表中连成一片。按照线性探测法处理冲突，如果生成哈希地址的连续序列愈长（即不同关键字值的哈希地址相邻在一起愈长），则当新的记录加入该表时，与这个序列发生冲突的可能性愈大。因此，哈希地址的较长连续序列比较短连续序列生长得快，这就意味着，一旦出现堆聚（伴随着冲突），就将引起进一步的堆聚。

2.线性补偿探测法

线性补偿探测法的基本思想是：将线性探测的步长从 1 改为 Q ，即将上述算法中的
hash ＝ (hash ＋ 1) % m 改为：hash ＝ (hash ＋ Q) % m = hash % m + Q % m，而且要求 Q 与 m 是互质的，以便能探测到哈希表中的所有单元。
【例】 PDP-11 小型计算机中的汇编程序所用的符合表，就采用此方法来解决冲突，所用表长 m ＝ 1321 ，选用 Q ＝ 25 。

3.伪随机探测

随机探测的基本思想是：将线性探测的步长从常数改为随机数，即令： hash ＝ (hash ＋ RN) % m ，其中 RN 是一个随机数。在实际程序中应预先用随机数发生器产生一个随机序列，将此序列作为依次探测的步长。这样就能使不同的关键字具有不同的探测次序，从而可以避 免或减少堆聚。基于与线性探测法相同的理由，在线性补偿探测法和随机探测法中，删除一个记录后也要打上删除标记。

拉链法

拉链法 : hashmap

拉链法的优点
与开放定址法相比，拉链法有如下几个优点：
①拉链法处理冲突简单，且无堆积现象，即非同义词决不会发生冲突，因此平均查找长度较短；
②由于拉链法中各链表上的结点空间是动态申请的，故它更适合于造表前无法确定表长的情况；
③开放定址法为减少冲突，要求装填因子α较小，故当结点规模较大时会浪费很多空间。而拉链法中可取α≥1，且结点较大时，拉链法中增加的指针域可忽略不计，因此节省空间；
④在用拉链法构造的散列表中，删除结点的操作易于实现。只要简单地删去链表上相应的结点即可。

拉链法的缺点
拉链法的缺点是：指针需要额外的空间，故当结点规模较小时，开放定址法较为节省空间，而若将节省的指针空间用来扩大散列表的规模，可使装填因子变小，这又减少了开放定址法中的冲突，从而提高平均查找速度。

再散列（双重散列，多重散列）

当发生冲突时，使用第二个、第三个、哈希函数计算地址，直到无冲突时。缺点：计算时间增加。

建立一个公共溢出区

假设哈希函数的值域为[0,m-1],则设向量HashTable[0..m-1]为基本表，另外设立存储空间向量OverTable[0..v]用以存储发生冲突的记录。