散列冲突处理：链地址法

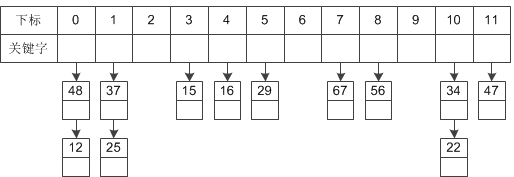
**链地址法（拉链法）**

前面我们谈到了散列冲突处理的[开放定址法](http://www.nowamagic.net/academy/detail/3008050)，它的思路就是一旦发生了冲突，就去寻找下一个空的散列地址。那么，有冲突就非要换地方呢，我们直接就在原地处理行不行呢？

可以的，于是我们就有了[链地址法](http://www.nowamagic.net/academy/tag/%E9%AD%94%E6%B3%95)。

将所有关键字为同义词的记录存储在一个单链表中，我们称这种表为同义词子表，在散列表中只存储所有同义词子表的头指针。

对于关键字集合{12,67,56,16,25,37, 22,29,15,47,48,34}，我们用前面同样的12为除数，进行除留余数法：



* 此时，已经不存在什么冲突换址的问题，无论有多少个冲突，都只是在当前位置给单链表增加结点的问题。很不错的解决思路吧？

拉链法解决冲突的做法是：将所有关键字为同义词的结点链接在同一个单链表中。若选定的散列表长度为m，则可将散列表定义为一个由m个头指针组成的指针数组T[0..m-1]。凡是散列地址为i的结点，均插入到以T[i]为头指针的单链表中。T中各分量的初值均应为空指针。在拉链法中，装填因子α可以大于 1，但一般均取α≤1。

**拉链法的优势与缺点**

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。而对开放地址法构造的散列表，删除结点不能简单地将被删结 点的空间置为空，否则将截断在它之后填人散列表的同义词结点的查找路径。这是因为各种开放地址法中，空地址单元（即开放地址）都是查找失败的条件。因此在用开放地址法处理冲突的散列表上执行删除操作，只能在被删结点上做删除标记，而不能真正删除结点。

拉链法的缺点：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。

* 链地址法的优势是对于可能会造成很多[冲突](http://www.nowamagic.net/academy/tag/%E5%86%B2%E7%AA%81)的散列函数来说，提供了绝不会出现找不到地址的保障。当然，这也就带来了査找时需要遍历单链表的性能损耗，不过性能损耗在很多场合下也不是什么大问题。