散列表（哈希表）的定义

**一般的查找**

给你一个顺序表，你会如何查找某个给定的元素？

* 一般思路就是从表头开始，一个挨一个的比较记录a[i]与key的值是“=”还是“≠”，直到有相等才算是査找成功，返回i的值。

到了有序表（已排序的表）査找时，我们可以利用a[i]与key的“<” 或 “>”，来折半査找，直到相等时査找成功返回i。

反正我们的目标就是要找到那个 i 值，是不是还有其它好点的方法呢？

**一个小故事**

试想这样的场景，你很想学太极拳，听说学校有个叫张三丰的人打得特别好，于是你到学校学生处找人。学生处的工作人员可能会拿出学生名单，一个一个的査找， 最终告诉你，学校没这个人，并说张三丰几百年前就已经在武当山作古了。可如果你找对了人，比如在操场上找那些爱运动的同学，人家会告诉你，“哦，你找张三丰呀， 有有有，我带你去。”于是他把你带到了体育馆内，并告诉你，那个教大家打太极的小伙子就是“张三丰”，原来“张三丰” 是因为他太极拳打得好而得到的外号。

* 数据结构的知识可以和生活上的事情类比，查找其实跟找人就很像。学生处的老师找张三丰，那就是顺序表査找，依赖的是姓名关键字的比较。而通过爱好运动的同学询问时，没有遍历，没有比较，就凭他们“欲找太极‘张三丰’，必在体育馆当中”的经验，直接告诉你位置。

也就是说，我们只需要通过某个函数f，使得

存储位置 = f(关键字)

那样我们可以通过査找关键字不需要比较就可获得需要的记录的存储位置。

这就是一种新的存储技术——散列技术。

**散列技术与散列表**

散列技术是在记录的存储位置和它的关键字之间建立一个确定的对应关系f，使得每个关键字key对应一个存储位置f(key)。

査找时，根据这个确定的对应关系找到给定值key的映射f(key),若査找集合中存在这个记录，则必定在f(key)的位置上。

这里我们把这种对应关系f称为散列函数，又称为哈希(Hash)函数。按这个思想，采用散列技术将记录存储在一块连续的存储空间中，这块连续存储空间称为散列表或哈希表（Hashtable)。那么关键字对应的记录存储位置我们称为散列地址。