# 查找

查找表

a) 概念：是由同一类型的数据元素(或记录)构成的集合，由于“集合”中的数据元素之间存在着松散的关系，因此查找表是一种应用灵便的数据结构。

b) 对查找表的操作:

查询：某个“特定的”数据元素是否在查找表中；

检索：某个“特定的”数据元素的各种属性；

插入：在查找表中插入一个数据元素；

删除：从查找表中删去某个数据元素

查找表的分类：

1）静态查找表：仅作查询和检索操作的查找表

2）动态查找表：在查找过程中同时插入查找表中

不存在的数据元素，或者从查找表中删除已存在的某个

数据元素,此类表为动态查找表

关键字（key）

是数据元素（或记录）中某个数据项的值，用以标识（识别）一个数据元素（或记录）。若此关键字可以识别唯一的一个记录，则称之谓“主关键字”。若此关键字能识别若干记录，则称之谓“次关键字”。

查找（searching）

根据给定的某个值，在查找表中确定一个其关键字等于给定值的数据元素或（记录）。

查找成功：若查找表中存在这样一个记录，则称“查找成

功”，给出整个记录的信息，或指示该记录在

查找表中的位置；

查找不成功：否则称“查找不成功”，查找结果：给出

“空记录”或“空指针”。

### 顺序查找:最简单的查找方法

数据一项一项按按顺序逐个查找，无论数据是否有序，都是从头到尾找到底

优点：在查找前不需要做任何处理

缺点：查找速度慢

实例

74 53 61 28 99 47 88

设计一个python程序，随机数生成1-150之间的80个整数，然后实现顺序查找。

写一个python算法，实现折半查找

已知如下11个元素的有序表：  
（05 13 19 21 37 56 64 75 80 88 92）, 请查找关键字为21和85的数据元素。

解：① 先设定3个辅助标志: low,high,mid，

② 运算步骤：

（1） low =1,high =11 ,mid =6 ，待查范围是 [1,11]；

（2）若 ST.elem[mid].key < key，说明 key[ mid+1,high] ，

则令：low =mid+1;重算 mid＝ (low+high)/2；.

（3）若 ST.elem[mid].key > key，说明key[low ,mid-1]，

则令：high =mid–1;重算 mid ；

（4）若 ST.elem[ mid ].key = key，说明查找成功，元素序号=mid;

③ 结束条件： （1）查找成功 ： ST.elem[mid].key = key

（2）查找不成功 ： high≤low （意即区间长度小于0）