

8.2.1 直接插入排序

直接插入排序 (Straight Insertion Sort) 是一种最简单的排序方法, 其基本操作是将一条记录插入已排好序的表, 从而得到一个新的、记录数量增1的有序表。

算法8.1 直接插入排序

【算法步骤】

- ① 设待排序的记录存放在数组 $r[1..n]$ 中, $r[1]$ 是一个有序序列。
- ② 循环 $n-1$ 次, 每次使用顺序查找法, 查找 $r[i]$ ($i = 2, \dots, n$) 在已排好序的序列 $r[1 \dots i-1]$ 中的插入位置, 然后将 $r[i]$ 插入表长为 $i-1$ 的有序序列 $r[1 \dots i-1]$, 直到将 $r[n]$ 插入表长为 $n-1$ 的有序序列 $r[1 \dots n-1]$, 最后得到一个表长为 n 的有序序列。

【例8.1】 已知待排序记录的关键字序列为 $\{49, 38, 65, 97, 76, 13, 27, 49\}$, 请给出用直接插入排序法进行排序的过程。

直接插入排序过程如图8.1所示, 其中()中为已排好序的记录的关键字。

初始关键字	(49)	38	65	97	76	13	27	49
$i=2$	(38	49)	65	97	76	13	27	49
$i=3$	(38	49	65)	97	76	13	27	49
$i=4$	(38	49	65	97)	76	13	27	49
$i=5$	(38	49	65	76	97)	13	27	49
$i=6$	(13	38	49	65	76	97)	27	49
$i=7$	(13	27	38	49	65	76	97)	49
$i=8$	(13	27	38	49	49)	65	76	97)

图8.1 直接插入排序过程

在具体实现 $r[i]$ 向前面的有序序列插入时, 有两种方法: 一种是将 $r[i]$ 与 $r[1], r[2], \dots, r[i-1]$ 从前向后顺序比较; 另一种是将 $r[i]$ 与 $r[i-1], r[i-2], \dots, r[1]$ 从后向前顺序比较。这里采用后一种方法, 和顺序查找类似, 为了在查找插入位置的过程中避免数组下标越界, 在 $r[0]$ 处设置监视哨。在自 $i-1$ 起往前查找插入位置的过程中, 可以同时后移记录。

【算法描述】

```
void InsertSort (SqList &L)
{ // 对顺序表L进行直接插入排序
  for (i=2; i<=L.length; ++i)
    if (L.r[i].key < L.r[i-1].key) // "<", 需将 r[i] 插入有序子表
    {
      L.r[0]=L.r[i]; // 将待插入的记录暂存到监视哨中
      L.r[i]=L.r[i-1]; // r[i-1] 后移
      for (j=i-2; L.r[0].key < L.r[j].key; --j) // 从后向前寻找插入位置
        L.r[j+1]=L.r[j]; // 记录逐个后移, 直到找到插入位置
      L.r[j+1]=L.r[0]; // 将 r[0] 即原 r[i], 插入正确位置
    } // if
}
```



直接插入排序