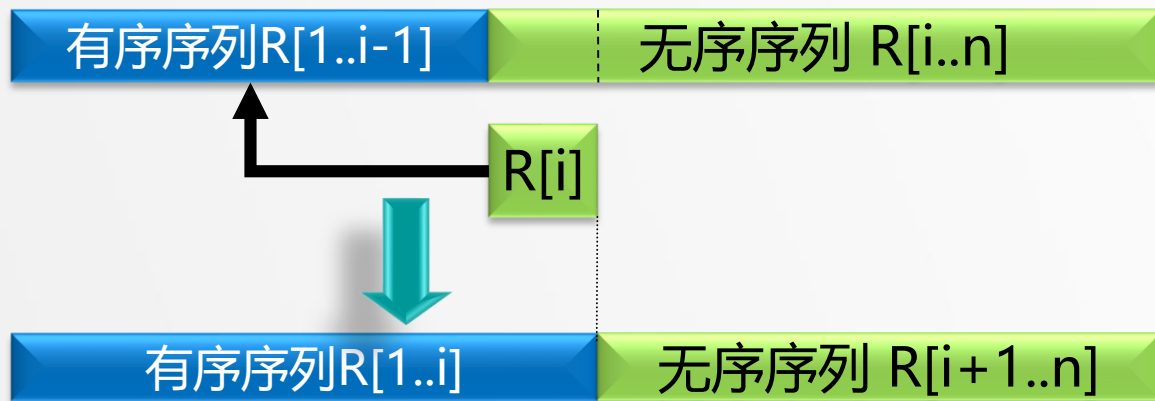




简单插入排序

插入排序

基本思想：将无序子序列中的一个或几个记录“插入”到有序子序列中，从而增加有序子序列的长度。

9, 5, 6, 4, 2, 1

(a)

5, 9, 6, 4, 2, 1

(b)

5, 6, 9, 4, 2, 1

(c)

4, 5, 6, 9, 2, 1

(d)

2, 4, 5, 6, 9, 1

(e)

1, 2, 4, 5, 6, 9

(f)

实现“一趟插入排序”可分三步进行：

- 插入排序三步曲

1. 在 $R[1..i-1]$ 中查找 $R[i]$ 的插入位置,
 $R[1..j].key \leq R[i].key < R[j+1..i-1].key$;

定位

2. 将 $R[j+1..i-1]$ 中的所有记录均后移
一个位置;

挤空

3. 将 $R[i]$ 插入(复制)到 $R[j+1]$ 的位置上。

插入

不同的定位方法导致不同的插入算法

插入排序

- 直接插入排序（基于顺序查找定位）
- 折半插入排序（基于折半查找定位）
- 希尔排序（基于逐趟缩小增量）

直接插入排序

排序过程：整个排序过程为 $n-1$ 趟插入，即先将序列中第1个记录看成是一个有序子序列，然后从第2个记录开始，逐个进行插入，直至整个序列有序

例

i=1 (49) 38 65 97 76 13 27
 ↓
 i=2 **38** (38 49) 65 97 76 13 27
 ↓
 i=3 **65** (38 49 65) 97 76 13 27
 ↓
 i=4 **97** (38 49 65 97) 76 13 27
 ↓
 i=5 **76** (38 49 65 76 97) 13 27
 ↓
 i=6 **13** (13 38 49 65 76 97) 27
 ↓
 i=7 **27** (13 27 38 49 65 76 97)
 ↓ ↓ ↓ ↓ ↓ ↓
 排序结果: (13 27 38 49 65 76 97)

算法描述

```
typedef struct  
{ int key;  
  float info;  
}JD;
```

```
void straisort(JD r[],int n)//对长度为n的序列排序  
{ int i,j;  
  for(i=2;i<=n;i++)  
  { r[0]=r[i];  
    j=i-1;  
    while(r[0].key<r[j].key)  
    { r[j+1]=r[j];  
      j--;  
    }  
    r[j+1]=r[0];  
  }  
}
```

算法评价

时间复杂度

若待排序记录按关键字从小到大排列(正序)

关键字比较次数:

$$\sum_{i=2}^n 1 = n - 1$$

记录移动次数:

$$\sum_{i=2}^n 2 = 2(n - 1)$$

若待排序记录按关键字从大到小排列(逆序)

关键字比较次数:

$$\sum_{i=2}^n i = \frac{(n + 2)(n - 1)}{2}$$

记录移动次数:

$$\sum_{i=2}^n (i + 1) = \frac{(n + 4)(n - 1)}{2}$$

若待排序记录是随机的, 取平均值

关键字比较次数:

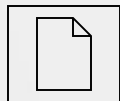
$$\frac{n^2}{4}$$

$$T(n) = O(n^3)$$

记录移动次数:

$$\frac{n^2}{4}$$

空间复杂度: $S(n) = O(1)$



Ch8_1.c

表 直接插入排序算法的性能

时间复杂度			空间复杂度	稳定性	复杂性
平均情况	最坏情况	最好情况			
$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定	简单

[习题 1] 用直接插入排序对下面 4 个序列进行递增排序, 元素比较次数最少的是()。

A. 94, 32, 40, 90, 80, 46, 21, 69

B. 32, 40, 21, 46, 69, 94, 90, 80

C. 21, 32, 46, 40, 80, 69, 90, 94

D. 90, 69, 80, 46, 21, 32, 94, 40

C

分析: 用直接插入排序, 数据序列越接近有序, 比较次数越少。

思考

简单插入排序的本质？

比较和交换

序列中 **逆序的个数** 决定交换次数

平均逆序数量为 $C(n,2)/2$,所以 $T(n)= O(n^2)$

简单插入排序复杂度由什么决定？ **逆序个数**

如何改进简单插入排序复杂度？

- **分组**，比如 $C(n,2)/2 > 2C((n/2),2)/2$
- 3,2,1有3组逆序对(3,1) (3,2) (2,1)需要交换3次。但**相隔较远**的3,1交换一次后1,2,3就没有逆序对了。
- 基本有序的插入排序算法复杂度接近 $O(n)$