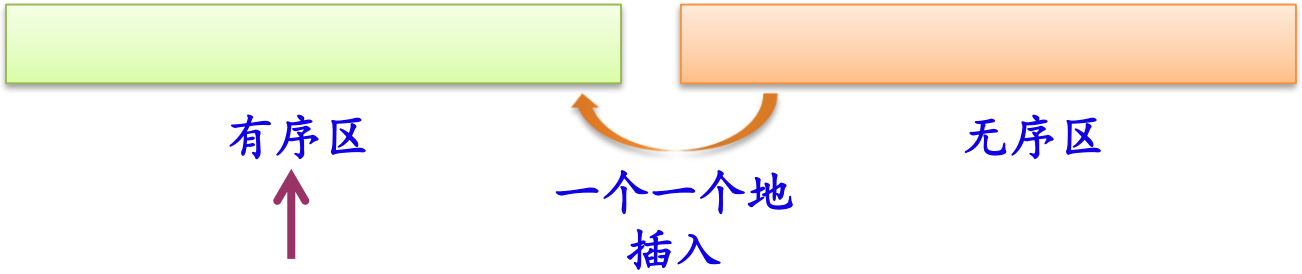
10.2 插入排序

基本思路



主要的插入排序方法：

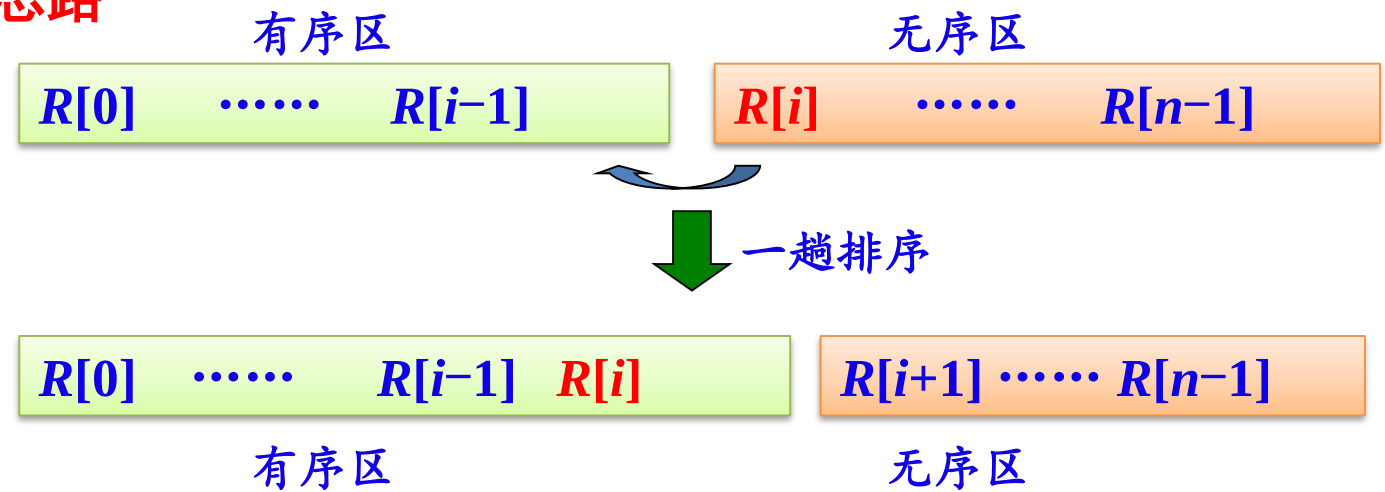
（1）直接插入排序

（2）折半插入排序

（3）希尔排序

10.2.1 直接插入排序

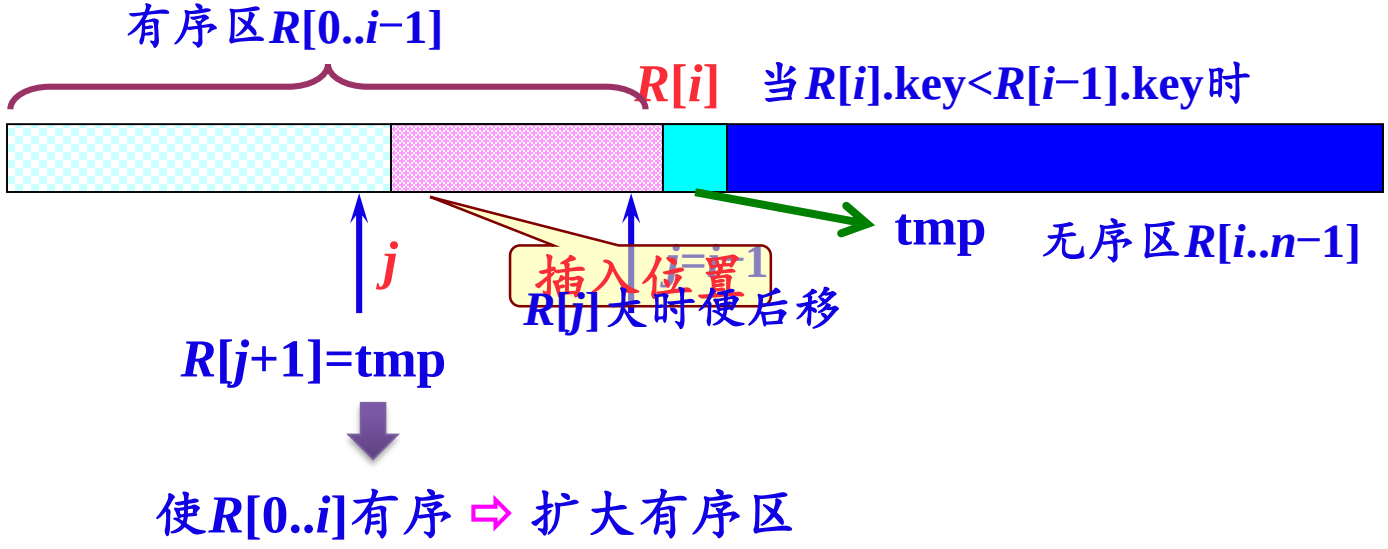
基本思路



初始时，有序区只有一个元素$R[0]$

$i=1\sim n-1$，共经过$n-1$趟排序

一趟直接插入排序：在有序区中插入$R[i]$的过程。



直接插入排序的算法：

```
void  InsertSort(RecType R[],int n)
{    int i, j;   RecType tmp;
     for (i=1;i<n;i++)
     {     if (R[i].key<R[i-1].key])          //反序时
           {      tmp=R[i];
                  j=i-1;
                 do                           //找R[i]的插入位置
                 {      R[j+1]=R[j];          //将关键字大于R[i].key的记录后移
                        j--;
                 }  while  (j>=0 && R[j].key>tmp.key)
                R[j+1]=tmp;                   //在j+1处插入R[i]
           }
     }
}
```

算法分析

最好的情况（关键字在记录序列中正序）：

“比较”的次数：

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

“移动”的次数：

$$0$$

最好：O(n)

最坏的情况（关键字在记录序列中反序）：

“比较”的次数：

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

“移动”的次数：

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

最坏：O(n^2)

总的平均比较和移动次数约为

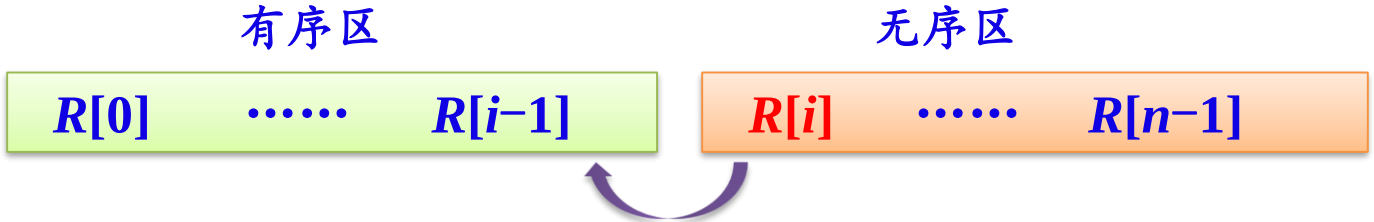
$$\sum_{i=1}^{n-1} \left(\frac{i}{2} + \frac{i}{2} + 2\right) = \sum_{i=1}^{n-1} (i+2) = \frac{(n-1)(n+4)}{2} = O(n^2)$$

平均：O(n^2)

10.2.2 折半插入排序

基本思路

查找采用折半查找方法，称为二分插入排序或折半插入排序。



折半插入排序算法：

```
void BinInsertSort(RecType R[],int n)
{    int i, j, low, high, mid;
     RecType tmp;
     for (i=1;i<n;i++)
     {    if (R[i].key<R[i-1].key])                    //反序时
          {    tmp=R[i];                               //将R[i]保存到tmp中
               low=0;  high=i-1;
               while (low<=high)                       //在R[low..high]中查找插入的位置
               {    mid=(low+high)/2;                  //取中间位置
                    if (tmp.key<R[mid].key)
                         high=mid-1;                   //插入点在左半区
                    else
                         low=mid+1;                    //插入点在右半区
               }                                       //找位置high
               for (j=i-1;j>=high+1;j--)               //记录后移
                    R[j+1]=R[j];
               R[high+1]=tmp;                          //插入tmp
          }
}
```

算法分析

折半插入排序：在$R[0..i-1]$中查找插入$R[i]$的位置，折半查找的平均关键字比较次数为$\log_2(i+1)-1$，平均移动元素的次数为$i/2+2$，所以平均时间复杂度为：

$$\sum_{i=1}^{n-1}\left(\log_2(i+1)-1+\frac{i}{2}+2\right)=O(n^2)$$

折半插入排序采用折半查找，查找效率提高。但元素移动次数不变，仅仅将分散移动改为集合移动。

【例】对同一待排序序列分别进行折半插入排序和直接插入排序，两者之间可能的不同之处是_____。

A.排序的总趟数

B.元素的移动次数

C.使用辅助空间的数量

D.元素之间的比较次数

说明：本题为2012年全国考研题

10.2.3 希尔排序

基本思路

① $d=n/2$

② 将排序序列分为d个组，在各组内进行直接插入排序

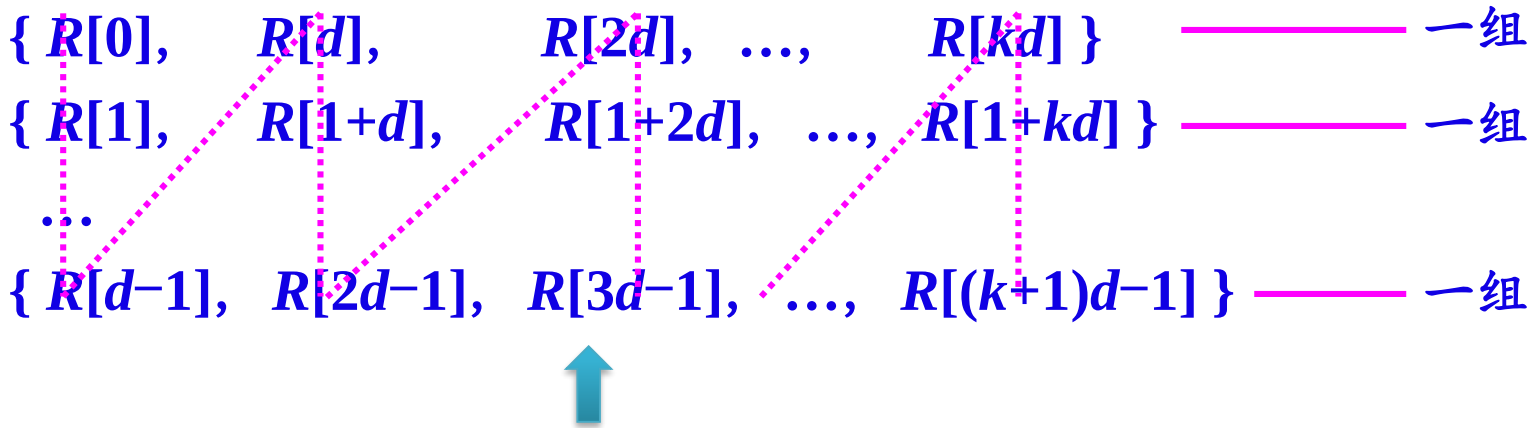
③ 递减$d=d/2$，重复②，直到$d=1$

算法最后一趟对所有数据进行了直接插入排序，所以结果一定是正确的。

一趟希尔排序过程

将记录序列分成若干子序列，分别对每个子序列进行直接插入排序。

例如：将 n 个记录分成 d 个子序列：



例如：$n=10$

注意：对于$d=1$的一趟，排序前的数据已将近正序！

希尔排序算法：

```
void ShellSort(RecType R[],int n)
{   int i, j, d;
    RecType tmp;
    d=n/2;                  //增量置初值
    while (d>0)
    {   for (i=d;i<n;i++)
        {                   //对相隔d位置的元素组直接插入排序
            tmp=R[i];
            j=i-d;
            while (j>=0&&tmp.key<R[j].key)
            {   R[j+d]=R[j];
                j=j-d;
            }
            R[j+d]=tmp;
        }
        d=d/2;              //减小增量
    }
}
```

*d*循环：使得每个记录都参加排序了

直接插入排序：

```
for (i=1;i<n;i++)
{   tmp=R[i];
    j=i-1;
    while (j>=0 &&
      tmp.key<R[j].key)
    {   R[j+1]=R[j];
        j=j-1;
    }
    R[j+1]=tmp;
}
```

希尔排序的时间复杂度约为$O(n^{1.3})$。

为什么希尔排序比直接插入排序好？

例如：有10个元素要排序。

直接插入排序

大约时间=10^2=100

希尔排序

d=5：分为5组，时间约为5×2^2＝20

+

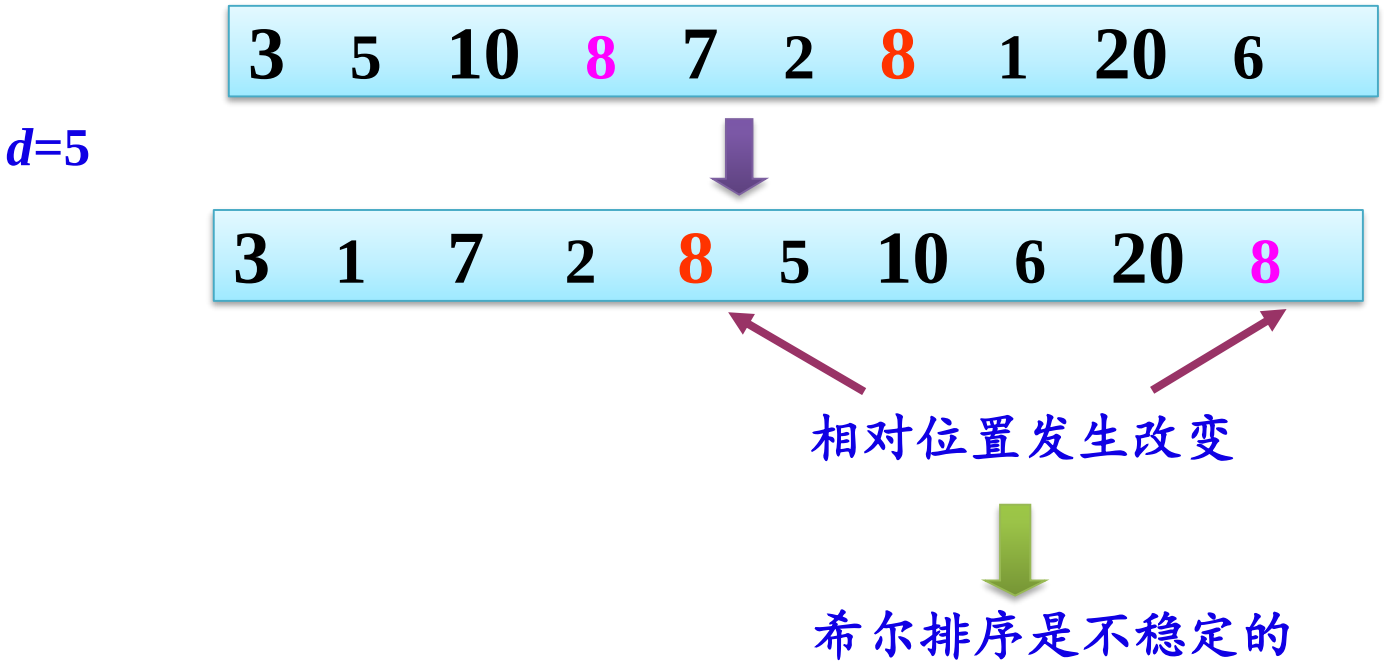
d=2：分为2组，时间约为2×5^2＝50

+

d=1：分为1组，几乎有序，时间约为10

= 80

希尔排序算法不稳定的反例：希尔排序法是一种不稳定的排序算法。



【例10-1】希尔排序的组内排序采用的是_____。

A.直接插入排序　　B.折半插入排序

C.快速排序　　D.归并排序

说明：本题为2015年全国考研题

思考题：

插入排序中每趟产生的有序区是全局有序吗？

该区域的元素位置不再改变

——本讲完——