

9.1.1概述

void X_Sort(ElementType A[],int N)

1. 默认讨论整数从小到大的排序
2. N 是正整数
3. 只讨论基于比较的排序
4. 只讨论内部排序
5. 稳定性：任意两个相等的数据排序前后的相对位置不发生变化
6. 没有一种排序是任何情况下都表现最好的

9.1.2冒泡排序

```
typedef int ElementType;

void Bubble_Sort(ElementType A[],int N)
{
    int P;
    int i;
    int flag;
    for(P=N-1;P>=0;P--)
    {
        flag=0;
        for(i=0;i<P;i++) /* 一趟冒泡排序 */
        {
            if(A[i]>A[i+1])
            {
                Swap(A[i],A[i+1]);
                flag=1; /* 标识发生了交换 */
            }
        }
        if(flag==0)
            break ; /* 全程无交换 */
    }
}
```

最好情况： $T = O(N)$

最坏情况： $T = O(N^2)$

只有严格大于的时候才做交换，否则不做交换

这是极其稳定的

9.1.3插入排序

```
void Insertion_Sort(ElementType A[],int N)
{
    for(P=1;P<N;P++)
    {
        Tmp=A[P]; /* 摸下一张牌 */
        for(i=P;i>0&&A[i-1]>Tmp;i--)
        {
            A[i]=A[i-1]; /* 移出空位 */
        }
        A[i]=Tmp; /* 新牌落位 */
    }
}
```

最好情况： $T = O(N)$

最坏情况： $T = O(N^2)$

插入排序也有稳定性

9.1.4时间复杂度下界

交换两个相邻元素正好消去一个逆序对

$$T(N, I) = O(N + I)$$

如果序列基本有序，则插入排序简单而高效

定理：任意 N 个不同元素组成的序列平均具有 $N(N - 1)/4$ 个逆序对

定理：任何仅以交换相邻两元素来排序的算法，其平均时间复杂度为 $\Omega(N^2)$

因此为了提高算法效率：

1. 每次消去不止一个逆序对
2. 每次交换相隔较远的两个元素