七、排序（上）

7.1. 格式：void X_Sort(ElementType A[], intN)

- X：排序名称
- 讨论从小到大的整数排序
- N是正整数
- 只讨论继续比较的排序（>=<有定义）
- 只讨论内部排序（所有数据的排序在内存空间中一次完成）
- 稳定性：保证任意两个相等的数据，排序前后相对位置不变
- 没有一种排序是任何情况下都表现最好的

7.2. 冒泡排序：Bubble_Sort()

- 比较相邻两数据，前大后小则交换，指针一次+1
- 一次排序之后，最大的冒到最后一个
- 最后一部分的数据总是有序的
- 代码：

```
for(P=N-1;P\>=0;P--){

 flag=0;

 for(i=0;i\<P;i++){

 if(A[i]\>A[i+1]){ //判断写\>不写\>=，保证稳定性

 Swap();

 flag++}

}

if(flag==0) break;  //全程无交换
```

- 时间复杂度T=：
    - 最好：O(N)
    - 最坏：O(N^2) N+(N-1)+(N-2)+....+1

7.3.插入排序：Insertion_Sort()

- 从前往后摸牌
- 在摸过的牌中，从后往前比较，大则放在最后，小则往前找到合适的位置插入
- 前面摸过的牌总是有序的
- 代码：

for(P=1;P<N;P++){ //默认第一张牌已经摸过

Tmp=A[P]; //摸下一张牌

for(i=P;i>0 && A[i-1]>Tmp);i--){ //判断写>不写>=，保证稳定性

A[i]=A[i-1]； //移出空位

}

A[i]=Tmp; //新牌落位

}

- 时间复杂度T=
  - 最好：O(N)
  - 最坏：O(N^2)

7.4.时间复杂度下界

- 逆序对inversion：对于i<j,若A[I]>A[J]，则称(i,j)是一对逆序对
- 插入排序：T(N,I)=O(N+I)
- 定理：N个不同元素的序列平均具有N(N-1)/4个逆序对(I)
- 定理：任何仅以交换两相邻元素来排序的算法，平均时间复杂度为Ω(N^2)
  - →想提高算法效率，每次要消去不止一个逆序对
  - 每次交换相隔较远的两个元素

7.5.希尔排序---by Donald Shell

- 定义希尔增量序列：Dm>Dm-1>...>D1=1
- 对每个Dk进行Dk间隔排序
- 每次间隔排序不会打乱前一次的间隔排序
- 原始希尔排序：
  - Dm=[N/2]，之后每次都除二取下整数
  - 代码：

```
void Shell_sort(ElementType A[],int N){
  for(D=N/2;D>0;D/=2){
    for(P=D;P<N;P++){ //P+D???
      Tmp=A[P];
      for(i=P;i\>=D&&A[i-D]>Tmp;i-=D){
        A[i]=A[i-D]
      }
    A[i]=Tmp
    }
  }
}
```

  - 最坏情况：$T = \theta(N^2)$

- 希尔增量不互质导致
- D=1之前都没有交换

7.6.堆排序