

# 第九讲 排序（上）

浙江大学 陈 越

# 9.1 简单排序

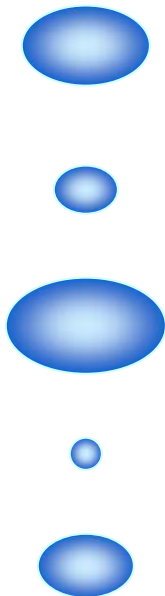
# 前提

```
void X_Sort ( ElementType A[], int N )
```

- 大多数情况下，为简单起见，讨论**从小到大的整数**排序
- **N**是**正整数**(表示我们要排的元素到底有多少个)
- 只讨论基于**比较**的排序 ( $> = <$  有定义)
- 只讨论**内部**排序(假设我们的内存空间充分大, 所有的数据都可以一次性的导入到内存空间里, 所有的排序过程是在内存里面一次性完成的)
- **稳定性**: 任意两个相等的数据, 排序前后的相对位置不发生改变
- 没有一种排序是任何情况下都是表现最好的

## ■ 冒泡排序

```
void Bubble_Sort( ElementType A[], int N )
{   for ( P=N-1; P>=0; P-- ) {
        flag = 0;
        for( i=0; i<P; i++ ) { /* 一趟冒泡 */
            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 */
    }
}
```



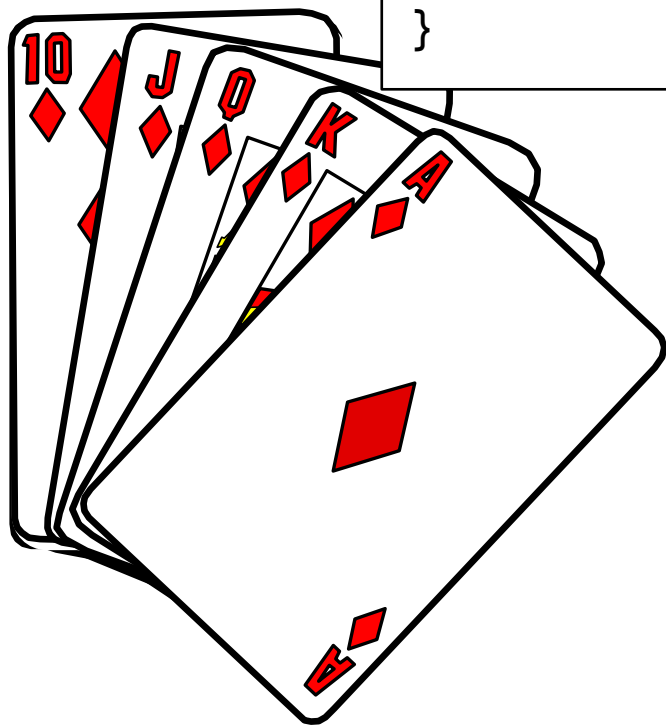
最好情况：顺序  $T = O(N)$   
最坏情况：逆序  $T = O(N^2)$

稳定

## ■ 插入排序

```
void Insertion_Sort( ElementType A[], int N )
{
    for ( P=1; P<N; P++ ) {
        Tmp = A[P]; /* 摸下一张牌放在一个临时的位置上 */
        for(i=P;i>0 && A[i-1]>Tmp;i--) /* 从最后一张牌向前比 */
            A[i] = A[i-1]; /* 向后移出空位 */
        A[i] = Tmp; /* 新牌落位 */
    }
}
```

稳定



最好情况：顺序  $T = O(N)$

最坏情况：逆序  $T = O(N^2)$

例：给定初始序列{34, 8, 64, 51, 32, 21}，冒泡排序和插入排序分别需要多少次数组元素交换才能完成？

# 时间复杂度下界

- 对于下标  $i < j$ ，如果  $A[i] > A[j]$ ，则称  $(i, j)$  是一对**逆序对 (inversion)**
- 问题：序列  $\{34, 8, 64, 51, 32, 21\}$  中有多少逆序对？  
(34, 8) (34, 32) (34, 21) (64, 51) (64, 32) (64, 21) (51, 32) (51, 21) (32, 21)
- 每次交换2个相邻元素正好消去1个逆序对！
- 插入排序：  $T(N, I) = O(N + I)$   
I: 原始的序列里面的逆序对的个数  
— 如果序列**基本有序**，则插入排序简单且高效

# 时间复杂度下界

更一般的情况：

- 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  $\Omega(N^2)$ 。
- 这意味着：要提高算法效率，我们必须
  - ⇒ 每次消去不止1个逆序对！
  - ⇒ 每次交换相隔较远的2个元素！