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 )
```

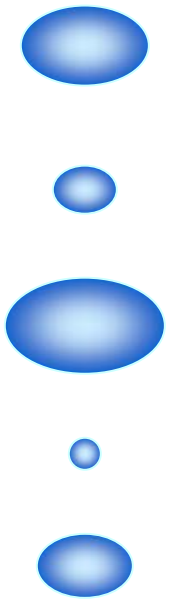
- 大多数情况下，为简单起见，讨论从小大的整数排序

- N是正整数
- 只讨论基于比较的排序（> = < 有定义）
- 只讨论内部排序 假设内存足够大，可以一次性把排序元素导入内存[对应有外部排序]
- 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
  都表现最好的 需要结合数据的特征选择

简单排序

- 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 */
   }
}
```

标识来缩减中间已经排好序的情况，无需继续循环

最好情况：顺序 $T = O(N)$

最坏情况：逆序 $T = O(N^2)$

稳定

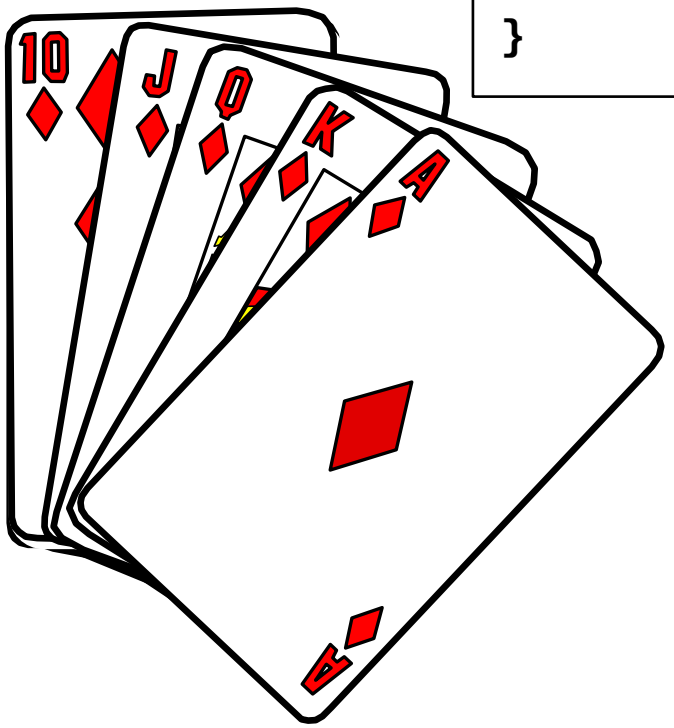
优点:对于链表，仍然可以顺序扫描，便于实现（对于其他算法，链表不容易实现）

简单排序

- 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 */
   }
}
```

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)

- 交换2个相邻元素正好消去1个逆序对！
- 插入排序：$T(N, I) = O(N+I)$

插入排序除了和数据规模有关，还和逆序对个数成正比

— 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

- 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！