

1. 绪论

算法分析

正确性

Beware of bugs in the above code;

I have only proved it correct, not tried it.

- D. Knuth

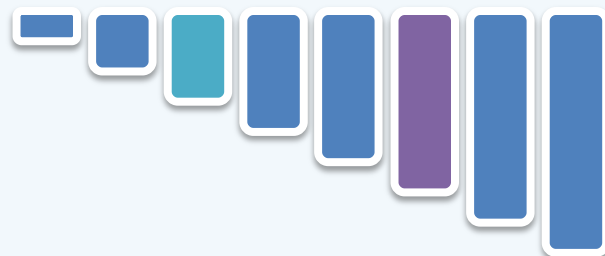
邓俊辉

deng@tsinghua.edu.cn

起泡排序

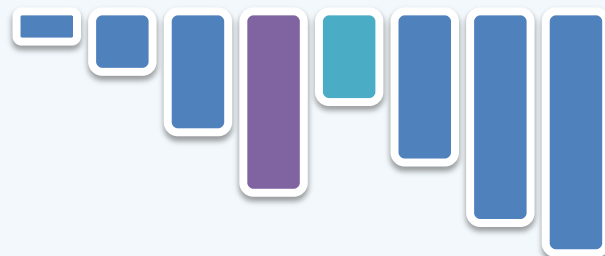
❖ 问题： 给定 n 个整数，

将它们按（非降）序排列



❖ 观察： 有序 / 无序 序列中，

任意 / 总有一对相邻元素 顺序 / 逆序



❖ 扫描交换：依次比较每一对相邻元素；如有必要，交换之

❖ 若整趟扫描都没有进行交换，则排序完成；否则，再做一趟扫描交换

实例



(a)



(b)



(c)



(d)



(e)



(f)



(g)

算法

❖ void bubblesort(int A[], int n) { //第二章将进一步改进

for (bool sorted = false; sorted = !sorted; n--) //反复扫描交换，每一趟...

for (int i = 1; i < n; i++) //自左向右，逐对检查A[0, n)内各相邻元素

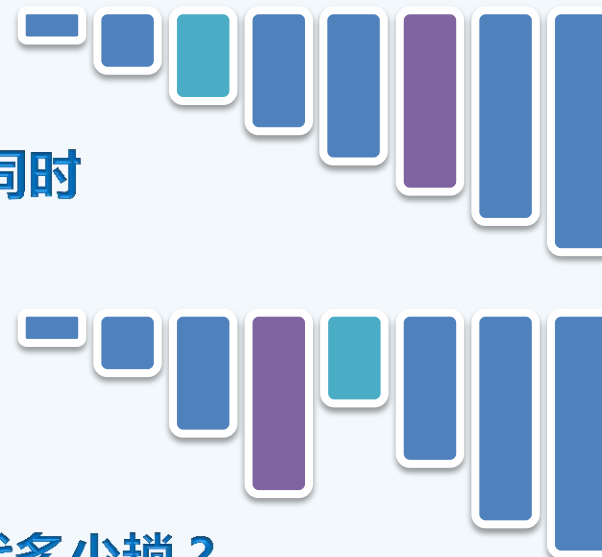
if (A[i - 1] > A[i]) { //若逆序，则

swap(A[i - 1], A[i]); //令其互换，同时

sorted = false; //清除（全局）有序标志

}

} //该算法...的确实现了所需的功能？必然会结束？至多需迭代多少趟？



证明

- ❖ **不变性**：经 k 趟扫描交换后，最大的 k 个元素必然就位
- ❖ **单调性**：经 k 趟扫描交换后，问题规模缩减至 $n - k$
- ❖ **正确性**：经至多 n 趟扫描后，算法必然终止，且能给出正确解答



(a)



(b)



(c)



(d)



(e)



(f)



(g)

性能

❖ 最坏情况：输入数据反序排列

共 $n - 1$ 趟扫描交换

每趟的效果，都等同于当前有效区间循环左移一位

第 k 趟中，需做 $n - k$ 次比较和 $3 \times (n - k)$ 次移动， $0 < k < n$

累计：#KMP = $(n-1) + (n-2) + \dots + 1 = n(n-1)/2$

#MOV = $3 \times n(n-1)/2$

$T(n) = 4 \times n(n-1)/2 = O(n^2)$

❖ 最好情况：所有输入元素已经完全（或接近）有序

外循环仅 1 次，做 $n - 1$ 次比较和 0 次元素交换

累计： $T(n) = n - 1 = \Omega(n)$