



第13周小结

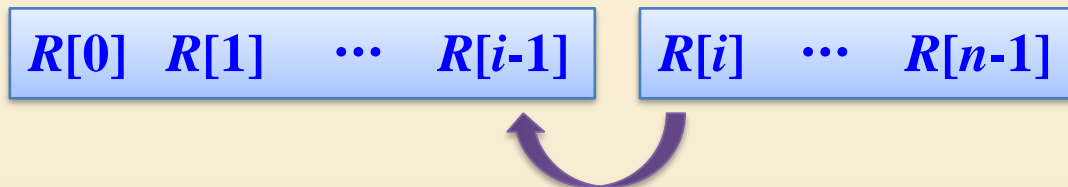
1

插入排序

- 直接插入排序
- 折半插入排序
- 希尔排序

① 直接插入排序

- 思路：

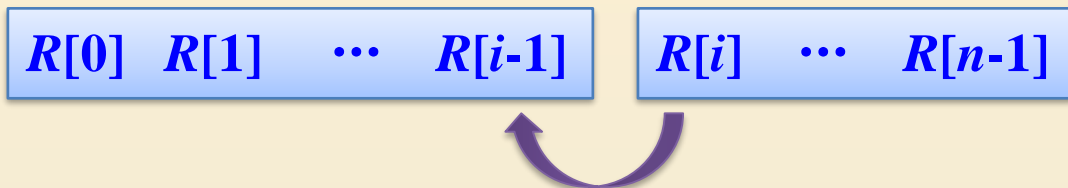


将 $R[i]$ 有序插入到有序区 $R[0..i-1]$ 中

- 趟数： $i=1 \sim n-1$ ，共 $n-1$ 趟
- 有序区：局部有序区（最后一趟前，所有数据不一定有序）
- 性能：最好（正序）： $O(n)$ ；最坏（反序）： $O(n^2)$ ；平均： $O(n^2)$
- 稳定性：稳定

② 折半插入排序

- 思路：



在有序区 $R[0..i-1]$ 折半查找插入位置

- 趟数： $i=1 \sim n-1$ ，共 $n-1$ 趟
- 有序区：局部有序区（最后一趟前，所有数据不一定有序）
- 性能：最好（正序）： $O(n)$ ；最坏（反序）： $O(n^2)$ ；平均： $O(n^2)$
- 稳定性：稳定

③ 希尔排序

- 思路:

```
d=n/2
while (d>0)
{   将R[0..n-1]分为到d个组（相距d个位置的元素为一组）
    每组进行直接插入排序
    d=d/2;
}
```

- 趟数: $\lfloor \log_2 n \rfloor$ 趟

- 有序区: 不产生有序区（最后一趟前，所有数据不一定有

- 性能: 平均: $O(n^{1.3})$

- 稳定性: 不稳定



对同一待排序序列分别进行折半插入排序和直接插入排序，两者之间可能的不同之处是（ ）。

A.排序的总趟数

B.元素的移动次数

C.使用辅助空间的数量

D.元素之间的比较次数 ✓

在有序区查找插入 $R[i]$ 的位置：

● 直接插入排序采用逐个比较

● 折半插入排序采用折半查找方法

} 元素之间的比较次数不同

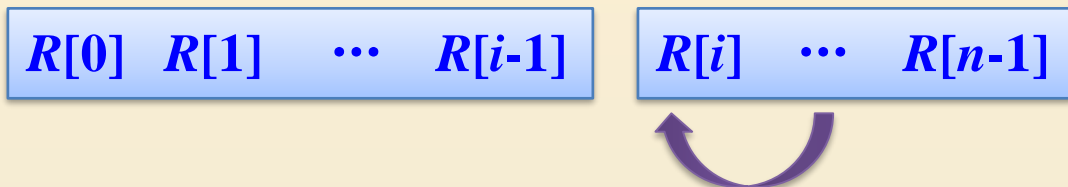
2

交换排序

- 冒泡排序
- 快速排序

① 冒泡排序

- 思路：

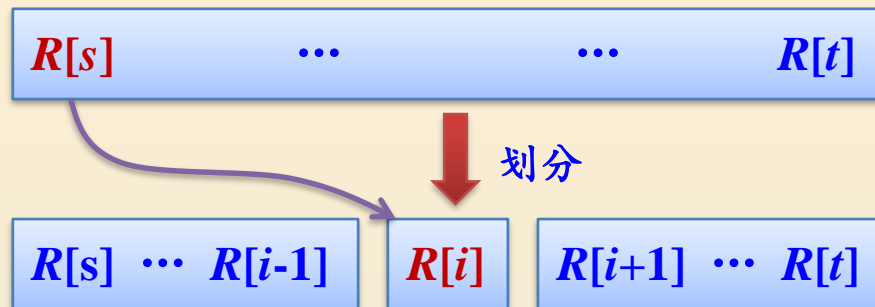


通过交换将 $R[i]$ 放在无序区开头

- 趟数： $i=0 \sim n-2$ ，共 $n-1$ 趟
- 有序区：全局有序区
- 性能：最好（正序）： $O(n)$ ；最坏（反序）： $O(n^2)$ ；平均： $O(n^2)$
- 稳定性：稳定

② 快速排序

● 思路：



- 递归树高度： $\log_2 n \sim n$
- 有序区：每次划分归位一个元素
- 性能：最好（随机）： $O(n \log_2 n)$ ；最坏（正反序）： $O(n^2)$ ；
平均： $O(n \log_2 n)$
- 稳定性：不稳定
- 空间： $O(\log_2 n)$



对数据序列 (8, 9, 10, 4, 5, 6, 20, 1, 2) 进行递增排序, 采用每趟冒出一个最小元素的冒泡排序算法, 需要进行的趟数至少是 ()。

- A.3 B.4 C.5 √ D.8

$i=0$: 1 8 9 10 4 5 6 20 2

$i=1$: 1 2 8 9 10 4 5 6 20

$i=2$: 1 2 4 8 9 10 5 6 20

$i=3$: 1 2 4 5 8 9 10 6 20

$i=4$: 1 2 4 5 6 8 9 10 20



以下关于快速排序的叙述中正确的是（ ）。

- A.快速排序在所有排序方法中 fastest, 而且所需辅助空间也最少
- B.在快速排序中, 不可以用队列替代栈
- C.快速排序的空间复杂度为 $O(n)$
- D.快速排序在待排序的数据随机分布时效率最高 ✓

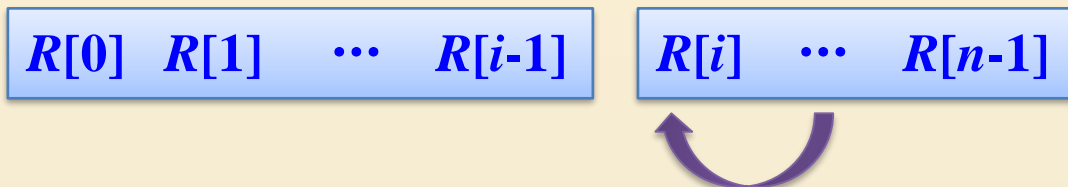
3

选择排序

- 简单选择排序
- 堆排序

① 简单选择排序

- 思路：



通过简单比较将 $R[i]$ 放在无序区开头

- 趟数： $i=0 \sim n-2$ ，共 $n-1$ 趟
- 有序区：全局有序区
- 性能：最好、最坏、平均： $O(n^2)$
- 稳定性：不稳定

② 堆排序

- 思路： 借助堆将 $R[i]$ 放在无序区末尾



- 趟数： $i=n \sim 2$ ，共 $n-1$ 趟
- 有序区： 全局有序区
- 性能： 最好、最坏、平均： $O(n \log_2 n)$
- 稳定性： 不稳定



有一个关键字序列(22, 86, 19, 49, 12, 30, 65, 35, 18), 在进行一趟排序后得到的结果为(18, 12, 19, 22, 49, 30, 65, 35, 86), 则采用的排序方法可能是 ()。

A.简单选择排序

B.冒泡排序

C.快速排序 ✓

D.堆排序

选项A、B、D每一趟产生的有序区是全局有序区



一个有 n 个整数的数组 $R[1..n]$ ，其中所有元素是有序的，将其看成是一棵完全二叉树，该树构成一个堆吗？若不是，请给一个反例，若是，请说明理由。

该数组一定构成一个堆，**递增有序数组构成一个小根堆，递减有序数组构成一个大根堆。**

以递增有序数组为例，假设数组元素为 k_1 、 k_2 、...、 k_n 是递增有序的，从中看出下标越大的元素值也越大，对于任一元素 k_i ，有：

$$k_i < k_{2i}, \quad k_i < k_{2i+1} \quad (i < n/2)$$

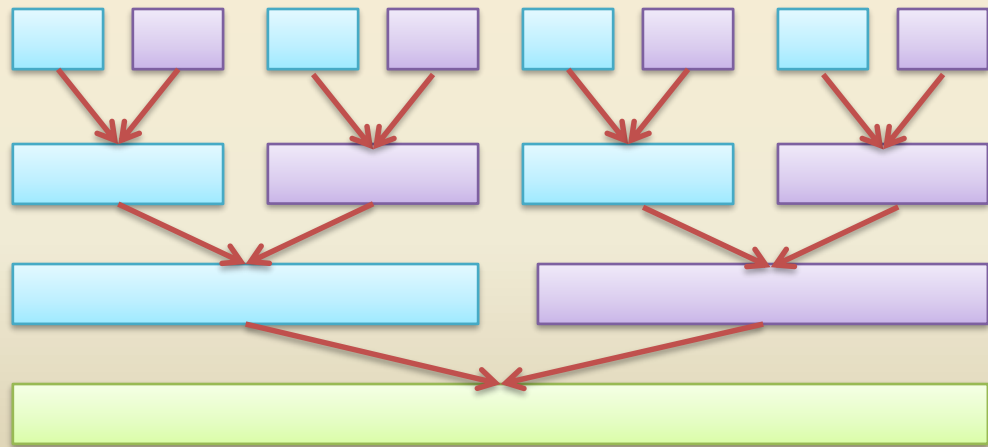
这正好满足小根堆的特性，所以构成一个**小根堆**。

4

归并排序

二路归并排序

● 思路：



二路归并排序

- 趟数：共 $\lceil \log_2 n \rceil$ 趟
- 有序区：局部有序区
- 性能：最好、最坏、平均： $O(n \log_2 n)$
- 稳定性：稳定
- 空间： $O(n)$



两个各含有 n 个元素的有序序列归并成一个有序序列时，关键字比较次数为 $n-1 \sim 2n-1$ ，也就是说关键字比较次数与初始序列有关。为什么通常说二路归并排序与初始序列无关呢？

二路归并排序中使用了辅助空间R1:

$R \Rightarrow R1 \quad R1 \Rightarrow R$ 每趟移动元素的次数 $2n$ ，总的移动次数总是 $O(n \log_2 n)$



尽管待排序的初始序列对关键字的比较有一定的影响，但不改变算法的总体时间性能，所以通常说二路归并排序与初始序列无关。

5

基数排序

思路：

将关键字分离出位，对每一位进行排序（共 d 位）



多关键字排序

- 所有位的取值 \Rightarrow 基数 r
- 每一位进行排序：分配、收集（不需要关键字比较）
- 数据特性 \Rightarrow 按最高位优先，按最低位优先

基数排序

- 趟数：共 d 趟
- 性能：最好、最坏、平均： $O(d(n+r))$
- 稳定性：稳定
- 空间： $O(r)$



有 n 个不同的英文单词（均为小写字母），它们的长度相等，均为 m 。若 $n=500$ ， $m<5$ ，试问采用什么排序方法时其时间复杂度最小？为什么？

- 采用基数排序方法时， $r=26$ ，时间复杂度为 $O(m(n+26m))$ 。
- 其他排序方法的时间复杂度最小为 $O(n\log_2 n)$ 。
- 当 $n=500$ ， $m<5$ 时， $m(n+26m)<n\log_2 n$ 。
- 采用基数排序方法最好。