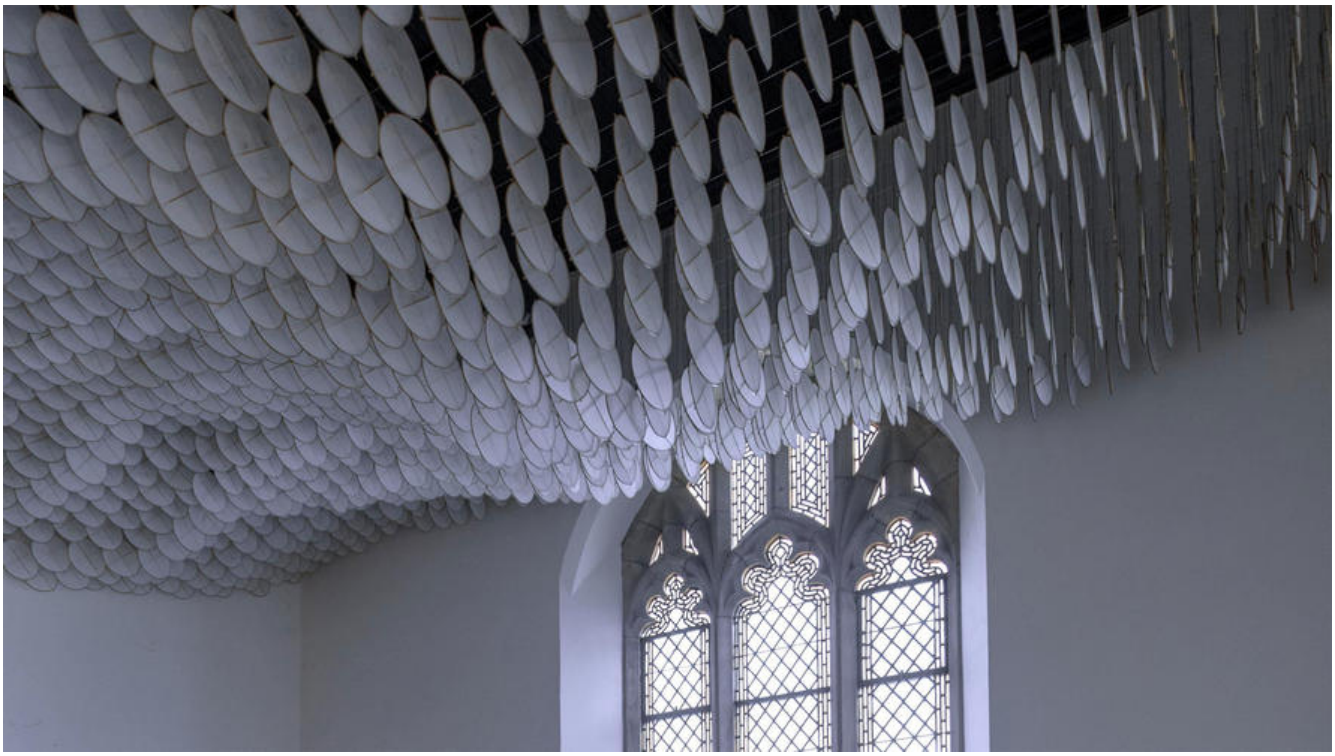
链滴

排序算法之比较排序

作者：someone33881
原文链接：https://ld246.com/article/1542723008617
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)



    本文主要是记录在学《算法笔记》等书籍中的排序算法，排序算法按照是否基于“比较”操作可以分为比较排序和非比较序两大类，这一篇文章主要是记录比较排序算法时的知识点和相关总结！

一、概述

    排序算法是非常基础重要的一类算法，本文主要是介绍比较排序算法的思想，主要涉及到冒泡排序、梳排序、堆排序、归排序（递归版与非递归版）、快速排序（递归版与非递归版）、内省排序、Timsort！

(1)基于比较的排序算法，也即根据待排序对象之间的大小关系进行排序的，比较规则必然是需要满足递性和全序性！

    排序算法下界O(nlog2 n))推导：

2^f(n) >= n!

=>

f(n) >= log2(n!) = nlog2(n)

(2)“合并多个有序列”问题

(3)“前k个小数”问题

二、冒泡排序与梳排序

1、冒泡排序

    冒泡排序是最最最基的一个排序算法，如果连这个都手写不出来伪代码，那也还是不要做程序员了早点滚蛋算了！

     思想：调整相邻两个象的位置，每进行一次内循环，就可以将最大值调整到最后！

因此，在经过**n-1次内循环**之后就可以得到整个完整的有序列！时间复杂度O(n^2)

伪代码如下：

```
for i = 1,2,.....,n-1 do
  for j = 1,2,.....,n-i do
    if a(j) > a(j+1) then
      交换a(j) 和 a(j+1)
    end if
  end for
end for
```

2、梳排序

    梳排序是冒泡排序的种改进，虽然没有很好的理论结果，但是实际效果非常好！

     思想：对固定距离处对象进行比较和交换，即在冒泡排序之前做了一些排序工作！

固定距离是待排序列长度n除以1.3向下取整（若小于1则取1）！时间复杂度O(nlog1.3(n))

伪代码如下：

```
j <- n, s <- 1.3, flag <- false
while j > 1 或者 flag = true do
  i <- 0, j <- max{j/s,1}, flag <- false
  while i + j <= n do
    if a(i) > a(i+j) then
      交换a(j) 和 a(j+1)
      flag <- true
    end if
    i <- i+1
  end while
end while
```

三、堆排序

    堆排序，即借助堆这数据结构来进行排序的！

     思想：对全部待排序象建堆，然后反复查找并删除最小值（最大值）！

     应用：

（1）多个序列合并问题：n个有序列，第i个序列长度为m(i)

    将每个有序列中的第1对象即最小值放入堆中，进行建堆 => 然后查找并删除最小值 => 若最小值来自第j个序列，则将第j

列中的下一个尚未处理的对象放入堆中 => 继续查找并删除最小值 => 直至全部比较完

    建堆时间复杂度O(n)
每次查找并删除最小值的复杂度是O(log(n))共需要次数sum[m(i)]「i=1.....n」，每次插入的复杂度O(log(n))共需要次数sum[m(i)-1]「i=1.....n」 => 总的时间复杂度是O(sum[m(i)log(n)]「i=1.....n)

(2)查找前k个数问题：
nbsp;将每个有序列中的第1个对象即最小值放入堆中，进行建堆 => 然后查找并删除最小值 => 若最值来自第j个序列，则将第j序列中的下一个尚未处理的对象放入堆中 => 继续查找并删除最小值 => 至第k个最小值即可

    建堆时间复杂度O(n)
，每次查找并删除最小值的复杂度是O(log(n))共需要次数k => 总的时间复杂度是O(n+klog(n))

四、归并排序

五、快速排序

六、内省排序

七、Timsort