算分回课05-16

黄道吉 1600017857

冒泡排序-算法

- 加一个flag标记排序是否完成
- 也可以看循环中是否有交换

- 每一次交换一对相邻的元素

算法 bubbleSort

```
FLAG ← n
while FLAG > 1 do
    k ← FLAG −1
    FLAG ← 1
    for j=1 to k do
      if L(j) > L(j+1) then do
          L(j) ↔ L(j+1)
          FLAG ← j
```

冒泡排序-逆序

- 每交换一次逆序数减一
- 逆序序列: 在i右边小于i的元素个数
- 置换和逆序序列一一对应
  - 3 1 6 5 8 7 2 4
  - (0, 0, 2, 0, 2, 3, 2, 3)
  - 12

冒泡排序-分析

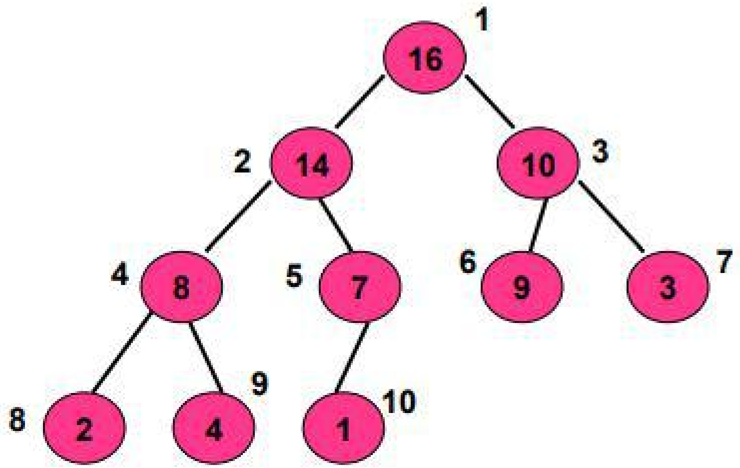
- 最坏情况: O(n^2) = O(n) * O(n), 并且在(n, n - 1, ..., 2, 1)取到

- 平均情况: 对所有可能的输入情况(n!)做平均, 每一个置换都能找到另一个和它和为O(n^2)的置换

$$E(\sum_{1}^{n} b_i) = \sum_{1}^{n} E(b_i)$$

堆-定义, 运算

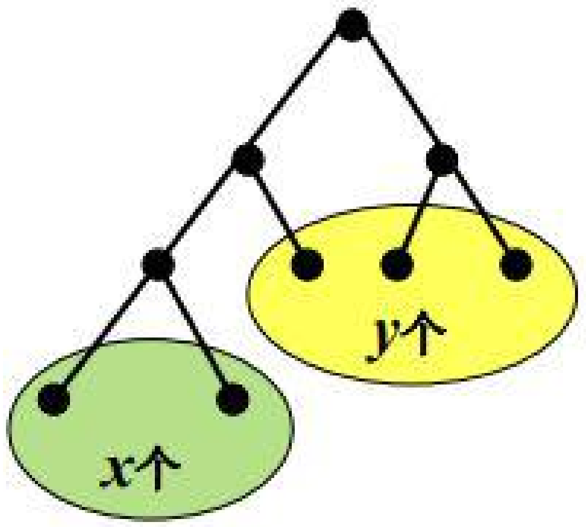
- 完全二叉树, 每一个节点元素不小于子节点(最大堆)
- 建堆: 对每一个有孩子的节点, 把它和孩子中最大的数调整上去, 递归向下处理
- 复杂度: 每一个节点处理O(1)时间,
  最多递归到堆底, O(h)



堆-建堆时间

- 建堆总时间 $$T(n) = \sum_{h=0}^{\log n} n_h \times O(h)$$
- 引理: 高度为n的层最多[n / 2^{h + 1}]个节点(归纳)
- h = 0 时,

$$\begin{aligned} & x + y \\ =& x + 2^{d-1} - \frac{x}{2} \\ =& \frac{2^d + x}{2} \\ =& [\frac{n}{2}] \end{aligned}$$



堆-建堆时间

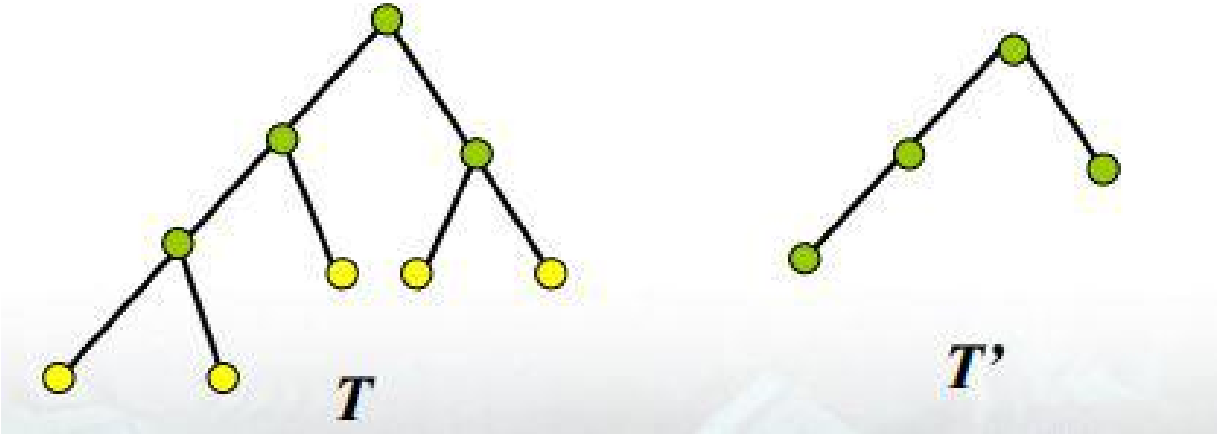
- 对于高度h的层, 拿掉所有树叶让它变成h - 1的层

$$n_0 = \lceil \frac{n}{2} \rceil$$

$$n^{'} = n - n_0 = \lfloor \frac{n}{2} \rfloor$$

$$n_h = n^{'}_{h-1}$$

$$n_h \le \lceil \frac{n^{'}}{2^h} \rceil \le \lceil \frac{n}{2^{h+1}} \rceil$$



堆-建堆时间

- 建堆时间为O(n)
- 右面的无穷级数收敛

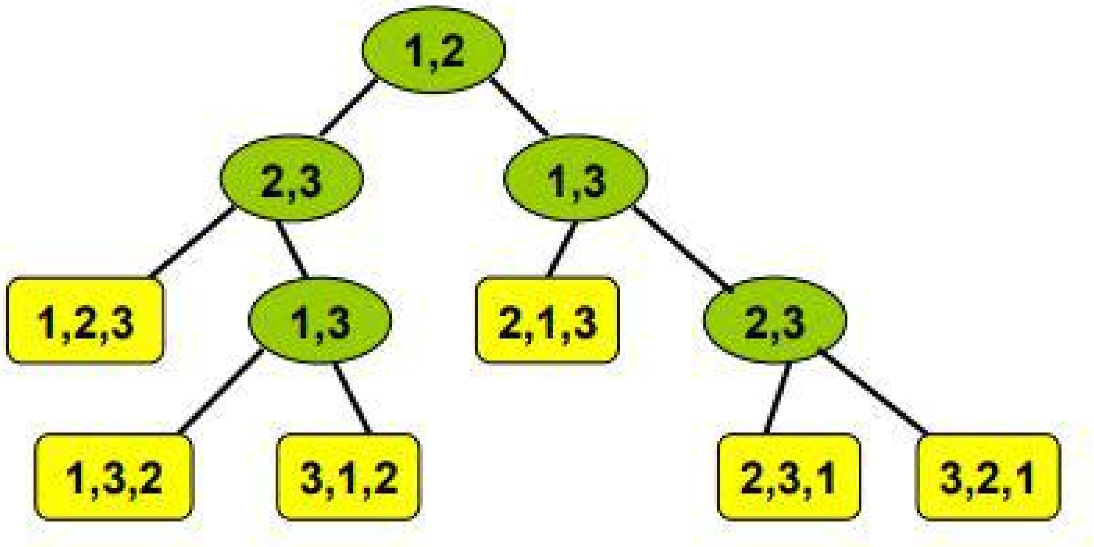
$$
\begin{aligned}
T(n) &= \sum_{h=0}^{\lfloor \log n \rfloor} \lceil \frac{n}{2^{h+1}} \rceil O(h) \\
&= O(n \sum_{h=0}^{\infty} \frac{h}{2^{h+1}}) \\
&= O(n)
\end{aligned}
$$

堆-堆排序

- 建好一个堆, 每次交换最大元素和堆尾元素, 再建堆
- 复杂度: $O(n \log n) = O(n) + O(n) * O(\log n)$

决策树

- 只考虑以比较为基本运算的排序算法
- 每一个节点标记比较的两个元素，
  根据比较结果走左或右儿子



最坏复杂度的下界

- 引理: d层深的B-树最多2^d片树叶(拿掉一层树叶, 得证)
- 引理: 决策树深度至少[log n!](因为有n!片树叶)

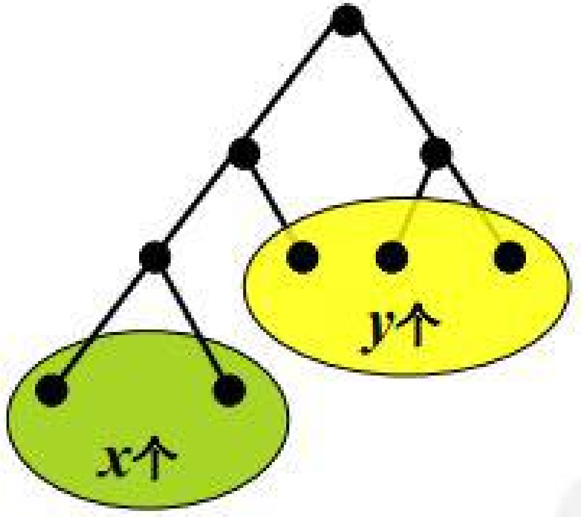
- 基于比较的排序最坏情况下的复杂度有[log n!] 近似nlogn - 1.5n

$$
\begin{aligned}
\log n! &= \sum_{j=1}^{n} \log j \geq \log e \int_{1}^{n} \ln x \; \mathrm{d}x \\
&= \log e(n \ln n - n + 1) \\
&\approx n \log n - 1.5n
\end{aligned}
$$

平均复杂度

- 定义epl(T), 表示所有树叶到根的路径长度之和, 那么要求的就是 epl(T) / n!的下界
- 引理: 所有树叶在相邻的两层的话, epl取到最小值(调整)
- 引理: epl最小的树有

$$\begin{aligned} epl(T) &= xd + y(d-1) \\ &= (2t - 2^d)d + (2^d - t)(d-1) \\ &= t(d-1) + 2(t - 2^{d-1}) \\ &= t\lfloor \log t \rfloor + 2(t - 2^{\lfloor \log t \rfloor}) \end{aligned}$$



平均复杂度的下界

- 基于比较的排序平均(输入等概分布)复杂度有[log n!]近似nlogn - 1.5n

$$
\begin{aligned}
A(n) &\geq \frac{1}{n!} epl(T) \\
&= \frac{1}{n!}(n!\lfloor \log n! \rfloor + 2(n! - 2^{\lfloor \log n! \rfloor})) \\
&= \lfloor \log n! \rfloor + \epsilon \\
&\approx n \log n - 1.5n
\end{aligned}
$$

排序算法的比较

算法	最坏情况	平均情况	占用空间	最优性
冒泡排序	$O(n^2)$	$O(n^2)$	原地	
快速排序	$O(n^2)$	$O(n\log n)$	$O(\log n)$	平均最优
归并排序	$O(n\log n)$	$O(n\log n)$	$O(n)$	最优
堆排序	$O(n\log n)$	$O(n\log n)$	原地	最优