# 算法基础 HW2

PB18111697 王章瀚

2020 年 10 月 18 日

## 1 堆排序

对于一个按升序排列的包含 $n$ 个元素的有序数组 $A$ 来说, Heapsort 的时间复杂度是多少? 如果 $A$ 是降序的呢? 请简要分析给出结果.

答:

它们应当都是$O(n \lg n)$的.

- 对于升序排列的数组 $A$ 来说, 首先要经历一个 $O(n)$ 的建堆过程
- 对于降序排列的数组 $A$, 如果程序能够对" 降序排列" 这个信息作处理, 那么这个数组本身就已经是一个最大堆了, 这时将有 $O(1)$ 的建堆过程;
  但如果程序不能处理" 降序" 这个信息, 那么依然要遍历非叶子结点来做 Max-Heapify, 只不过每次判断结果都是不必交换, 这个过程依然是 $O(n)$ 的.

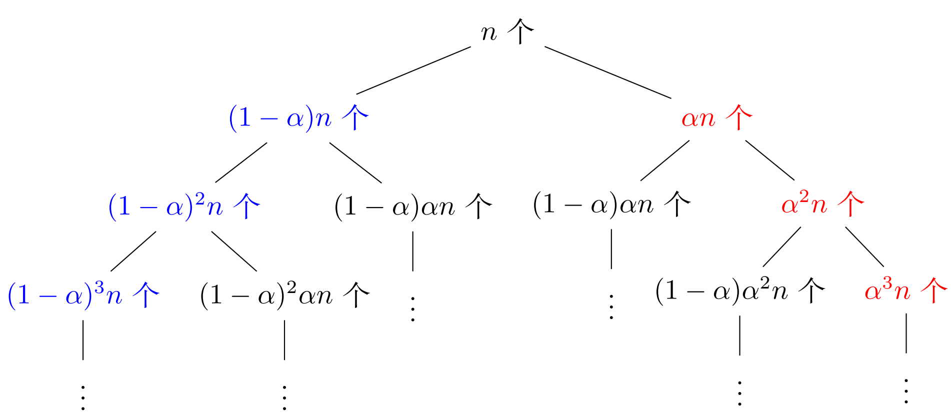
而更为耗时的排序过程都需要 $O(n \lg n)$, 因此总的来看, 两种情况的 Heapsort 时间复杂度都是$O(n \lg n)$ $\square$

## 2 快速排序

(a). 假设快速排序的每一层所做的划分比例都是 $1-\alpha;\alpha$, 其中 $0<\alpha\le 1/2$ 是一个常数. 试证明: 在相应的递归树中, 叶结点的最小深度大约是 $-\lg n/\lg\alpha$, 最大深度大约是 $-\lg n/\lg(1-\alpha)$(无需考虑舍入问题)

答:

不考虑舍入, 那么可以认为叶子节点只有一个元素 从根节点开始, 应按如下分支:




- 最小深度: 显然, 由于 $0<\alpha\le 1/2$, 最快到达叶子节点的分支应当是每次都选右节点的那条路径 (图中红色).

如果设深度为 $h$, 那么这条路径上的对应叶子节点就有 $\alpha^h n$ 个元素构成, 因此我们有

$$\alpha^h n = 1 \quad i.e. \quad h = \log_\alpha \frac{1}{n} = -\lg n / \lg \alpha$$

- 最大深度: 显然, 由于 $0 < \alpha \leq 1/2$, 最慢到达叶子节点的分支应当是每次都选左子结点的那条路径 (图中蓝色).
  如果设深度为 $h$, 那么这条路径上的对应叶子节点就有 $(1-\alpha)^h n$ 个元素构成, 因此我们有

$$(1-\alpha)^h n = 1 \quad i.e. \quad h = \log_{(1-\alpha)} \frac{1}{n} = -\lg n / \lg(1-\alpha)$$

□

(b). 试证明: 在一个随机输入数组上, 对于任何常数 $0 < \alpha \leq 1/2$, PARTITION 产生比 $1-\alpha : \alpha$ 更平衡的划分的概率约为 $1-2\alpha$

答:

在一次 PARTITION 中, 要产生比 $1-\alpha : \alpha$ 更不平衡的划分, 则必然是取到的pivot element为第 $\alpha n$ 大或第 $\alpha n$ 小的数, 这个的概率是

$$P(worse) = 2 \times \frac{\alpha n}{n} = 2\alpha$$

因此, 产生更平衡的划分的概率为

$$P(better) = 1 - 2\alpha$$

□