

# Tutorial

November 2, 2021

- 堆的建立与删除节点的过程.
  - 线性建堆, 复杂度分析.
  - 删除和加入节点的互逆过程.

- k sorted lists.
  - 使用双关键字堆记录value所在的位置, 每次删除后插入下一个.
  - Time Complexity:  $O(n \log k)$ .

- 正确性由归纳法证明.
- 错误性可以构造反例证明.
- 复杂度由主定理或归纳法均可.

- 考察快速排序和递归.
  - 为什么快速排序可以使用调用结构控制, 归并排序不行?
  - 它们都是分治算法, 但区别在于对于子问题的信息有没有合并的过程.
  - 考察 *partition* 函数的作用.
  - 考察函数调用的本质: 堆栈实现.

- 冒泡排序和复杂度分析.
  - 冒泡排序是每次把一个元素“冒泡”，规约为一个规模减少1的子问题.
  - 如何用  $SqrtSort(k)$ , 实现冒泡一个元素?
  - 如何用  $SqrtSort(k)$ , 实现冒泡一组元素?
  - 利用哪些性质可以给出复杂度的下界?
  - 复杂度可以用归纳法分析得出.

- 一道乱入的概率题.
  - 随机变量的概率收敛性.
  - 概率方程.

- lowerbound and upperbound.
  - upperbound 比 lowerbound 简单很多.
  - upperbound 只需要构造一个算法, 证明其正确性, 就给出了一个上界.
  - 如何证明lowerbound?
  - 信息论角度: 决策树.
  - 其他特征度量.



- 堆排序复杂度.
  - 这是对于一个算法的复杂度分析, 与上一题不同.
  - 对于复杂度级别没有影响的放缩.
  - 要求理解`heapify`.

- 近似排序: 分为k块.
- 分治的思想, 快速排序怎么修改.
- *RANDOMIZED – SELECT* 函数的实现与使用.
- 复杂度分析: 决策树的深度变化.
- lowerbound 证明.

- 两枚假币.
  - 显然是古典概型, 简单的组合数学.
  - 两侧分别是轻重假币后二分查找.
  - 有没有更好的办法?
  - 三除法.

- 排序后预处理前缀后缀和, 挨个check显然满足a.
- 考虑如何不排序求出?
- 关键性质: answer取值只能有一个.
- 类似 *RANDOMIZED – SELECT* 函数.

- 二路归并.
- 题目有点小问题, 考点是基数排序.

- 反证法求重心.
- dfs维护信息.

- morris 遍历.
- 二分, 分治各有做法.