

堆结构和堆排序

前置知识：无

堆结构

完全二叉树和数组前缀范围来对应，大小，单独的变量 $size$ 来控制

i 的父亲节点： $(i-1)/2$ ， i 的左孩子： $i*2 + 1$ ， i 的右孩子： $i*2 + 2$

堆的定义（大根堆、小根堆），本节课讲解按照大根堆来讲解，小根堆是同理的。

堆的调整： $heapInsert$ （向上调整）、 $heapify$ （向下调整）

$heapInsert$ 、 $heapify$ 方法的单次调用，时间复杂度 $O(\log n)$ ，完全二叉树的结构决定的

堆排序

A. 从顶到底建堆，时间复杂度 $O(n * \log n)$ ， $\log 1 + \log 2 + \log 3 + \dots + \log n \rightarrow O(n * \log n)$

或者用增倍分析法：建堆的复杂度分析+子矩阵数量的复杂度分析

B. 从底到顶建堆，时间复杂度 $O(n)$ ，总代价就是简单的等比数列关系，为啥会有差异？简单图解一下

C. 建好堆之后的调整阶段，从最大值到最小值依次归位，时间复杂度 $O(n * \log n)$

时间复杂度 $O(n * \log n)$ ，不管以什么方式建堆，调整阶段的时间复杂度都是这个，所以整体复杂度也是这个
额外空间复杂度是 $O(1)$ ，因为堆直接建立在了要排序的数组上，所以没有什么额外空间

注意：堆结构比堆排序有用的多，尤其是和比较器结合之后。后面几节课会重点讲述。