# 堆

是一个二叉树，满足条件：小顶堆，每个节点小于它的两个子节点，大顶堆，每个节点小于,它的两个子节点。

子节点之间顺序可以任意，因此一个堆很多种建法。

## 插入：

为了尽量保持平衡，每次按顺序插入，按照一层一层的，从左到右的顺序。对于数组实现的二叉树的堆，相当于直接插入到数组末尾。

## 调整：

调整全部：从堆最后一层开始，从下往上，每个点与它父节点比，不满足就条件就交换两者的值，时间复杂度是n \*logn

调整单独一个点：循环的向上与它父节点比较，不满足就条件就交换两者的值，时间复杂度是logn

对于数组实现的二叉树的堆，有两种建堆和插入策略

* 一个是用o1时间插入，插入很多次，统一调整。
* 还可以插入一个点调整一个点

上面两种方式效率相同(nlogn)，如果使用指针实现二叉堆，就只能用第二个方法了，而且代码比较长，因此二叉堆数组实现比较简单

## 删除：

二叉堆一般都是删除堆顶，但这个方法删除非顶端的点也适用。

对于小顶堆，从上往下看，

从顶端开始，令当前点元素 等于它两个子节点最小的，向下走，如果某个子节点不存在就直接走另一个。循环直到走到叶子结点为止。

## 排序：

这是属于堆的用途之一，每次取堆顶元素，删除它，再取，这样可以生成一个有序序列。

# 败者树

https://blog.csdn.net/youwuwei2012/article/details/32326897