**2.4 优先队列**

**2.4.1 API**

支持删除最大元素和插入元素的数据类型叫做优先队列。

通过插入一列元素然后一个个地删除掉其中最小的元素，我们可以用优先队列实现排序算法，这种算法称为堆排序。

优先队列MaxPQ的主要接口是delMax和insert。

**2.4.2 初级实现**

数组实现（无序）

insert方法的代码和栈的push方法完全一样，要实现删除最大元素，使用类似选择排序的代码，将最大元素和边界元素交换然后删除它。

数组实现（有序）

将所有较大的元素向右移动一格以使数组保持有序。

链表表示法

可以选择修改pop来找到并返回最大元素，或是修改push来保证所有元素为逆序并用pop来删除并返回链表的首元素。

使用无序序列是解决这个问题的惰性方法，使用有序序列则是解决问题的积极方法。

**2.4.3 堆的定义**

当一棵二叉树的每个节点都大于等于它的两个子节点时，它被称为堆有序。

根节点是堆有序的二叉树中最大的节点。

完全二叉树只用数组而不需要用指针就可以表示。将二叉树的节点按照层级顺序放入数组中，根节点在位置1，它的子节点在位置2，3，以此类推。

二叉堆是一组能够用堆有序的完全二叉树排序的元素，并在数组中按照层级存储（不使用数组的第一个位置）。

在一个堆中，位置k的节点的父结点的位置为k/2，而它的两个子节点的位置则分别为2k和2k+1。

一棵大小为N的完全二叉树的高度为lgN。

**2.4.4 堆的算法**

我们用长度为N+1的私有数组来表示一个大小为N的堆，我们不会使用pg[0]。

堆的操作会首先进行一些简单的改动，打破堆的状态，然后再遍历堆并按照要求将堆的状态恢复。这个过程叫做堆的有序化。

由下至上的堆有序化（上浮）：

如果堆的有序状态因为某个结点变得比它的父结点更大而被打破，那么我们就需要通过交换它和它的父结点来修复堆。

由上而下的堆有序化（下沉）：

通过将它和两个子节点中的较大者交换来恢复堆。

插入元素：将新元素加到数组末尾，增加堆的大小并让这个新元素上浮到合适的位置。

删除最大元素：我们从数组顶端删去最大的元素并将数组的最后一个元素放到顶端，减小堆的大小并让这个元素下沉到合适的位置。

插入元素和删除最大元素这两个操作的用时和队列的大小成对数关系。

编程：基于堆的优先队列

对于一个含有N个元素的基于堆的优先队列，插入元素操作只需不超过lgN+1次比较，删除最大元素的操作需要不超过2lgN次比较。

多叉堆：基于数组表示的完全三叉树构造堆并修改相应的代码并不困难。甚至对于给定的d，将其修改为任意的d叉树也并不困难。

**2.4.5 堆有序**

我们可以把任意优先队列变成一种排序方法。

堆排序可以分为两个阶段：在堆的构造阶段，我们将原始数组重新组织安排进一个堆中；然后在下沉阶段，我们从堆中按递减顺序取出所有元素并得到排序结果。

堆的构造：只需从左向右遍历数组，用swim保证扫描指针左侧的所有元素已经是一棵堆有序的完全树即可。

更聪明高效的办法是从右至左用sink函数构造子堆。

编程：堆排序

将N个元素排序，堆排序只需少于(2NlgN+2N)次比较（以及一半次数的交换）。

堆排序需要恒定的额外空间，当空间十分紧张的时候（例如在嵌入式系统）它很流行。

但现代系统的许多应用中很少使用它，因为它无法利用缓存。数组元素很少和相邻的其他元素进行比较，因此缓存未命中次数要远远高于大多数算法。