1 堆的定义

堆是一棵近似完全二叉树，可以通过数组来表示；

2 对任意非叶子结点，其左孩子为，右孩子为；

对任意非根结点，其父结点为；

3 对于元序列，

如果满足：



则为最大堆；

如果满足：



则为最小堆；

4 堆操作

4.1 自上而下调整堆**percolate\_down**

1） 输入percolate\_down(a[], int n, int i)，n为堆元素个数，i为需要调整的结点val=a[i]，LeftChild(i)和RightChild(i)均为堆；

2） 计算j=LeftChild(i)，如果j >= n，循环结束；

3） 如果j+1 < n，且a[j] < a[j+1]，则j += 1；

4） 比较a[j]与val

如果val >= a[j]，循环结束；

如果val < a[j]，a[i] = a[j]，i=j；//结点下沉

5） 循环结束后，a[i] = val；

4.2 自下而上调整堆percolate\_up

1）输入(a[], i)，i为需要调整的结点，val=a[i]；

2）如果i<=0，循环终止；

3）计算j=Parent(i)，

如果val <= a[j]，循环结束；

如果val > a[j]，则a[j] = a[i]，i=j；

4）循环结束，a[i] = val；

5 堆化数组

将初始数组看成无序二叉树，叶子结点显然为堆，对非叶子结点依次自上而下调整，最终得到的二叉树即为堆；

n = n0 + n1 + n2；

n = n1 + 2n2 + 1

n1为0或者1，因此，

for(int i = n / 2 - 1; i >= 0; i--)

{

percolate\_down(a, n, i);

}

6 堆排序

1） 堆化数组；

2） 交换堆顶和堆尾元素(DeleteMax)，堆大小size-1；

3）自上而下调整堆percolate\_down(a, size-1, 0)；

4）循环执行size-1次，就得到升序排列数组；