优先级队列是一种带有优先级的队列，它是一种比栈和队列更为专用的数据结构。具体结构跟普通的队列一样，但是每个数据元素都有自己的优先级，优先级高的数据元素每次插入都排在队首，而优先级低的每次插入都在优先级高的后边。

哈夫曼树算法就用到了优先级队列、抢先式任务操作系统中的程序被排在优先级队列中。

优先级队列也可以采用顺序和链式两种存储结构，但是考虑到优先级队列要保证安全快速的访问到优先级高的数据元素，又要保证可以实现较快的插入操作，所以通常使用链式存储结构实现优先级队列。

优先级队列的实现，最大的区别在与数据元素的添加上，要区分优先级找到要插入的位置然后再插入，别的方法实现没有很大变化。

public void offer(Object x) {

PriorityDate priorityDate =(PriorityDate)x;

Node q = new Node(priorityDate);

if(isEmpty())

front=rear=q;

else {

Node p = front;

Node pp = front;

while(p!=null&&priorityDate.priority>=((PriorityDate)p.val).priority){//找插入位置

pp=p; //记录当前位置

p=p.next;//p走到下一个位置

}

if(p==null){ //当插入的位置在末尾的时候直接查到尾

rear.next=q;

rear=q;

}

else if(p==front){//当插入的位置在头的时候

q.next=front;

front=q;

}

else {//当插入的位置在中间的时候 pp和p之间。

pp.next=q;

q.next=p;

}

}

}