**优先队列、图等总结及习题**

**一、优先队列**

**1、定义**

* + **优先队列（priority queue）是0个或多个元素的集合， 每个元素都有一个优先权或值。**
  + **与FIFO结构的队列不同，优先队列中元素出队列的顺序由元素的优先级决定。**
  + **从优先队列中删除元素是根据优先权高或低的次序，而不是元素进入队列的次序.**
  + **对优先队列执行的操作有:**
    - **1)查找一个元素**
    - **2)插入一个新元素**
    - **3)删除一个元素**

**2、描述**

**线性表、堆、左高树**

**（1）线性表**

* **采用无序线性表描述最大优先队列**
  + **公式化描述（利用公式Location(i)=i-1)**
    - **插入：表的右端末尾执行，时间: Θ(1) ;**
    - **删除:查找优先权最大的元素，时间: Θ(n) ;**
  + **使用链表，**
    - **插入：在链头执行，时间: Θ(1) ;**
    - **删除: Θ(n) ;**
* **采用有序线性表描述最大优先队列**
  + **公式化描述（利用公式Location(i)=i-1，元素按递增次序排列)**
    - **插入: O(n) ;删除: O(1) ;**
  + **使用链表（按递减次序排列）**

**插入: O(n) ;删除: O(1)**

**（2）堆**

* **最大/最小树**
* **最大/最小堆**

**初始化、插入、删除。**

**（3）左高树**

**定义（重量优先、高度优先）；**

**操作：创建、插入、删除。**

**3、应用**

**（1）堆排序**

**（2）哈夫曼编码**

**（3）归并排序**

**二、图**

**1、定义**

* **图(graph)是一个用线或边连接在一起的顶点或节点的集合。G = (V,E)**
  + **V 是顶点集. E是边集.**
  + **顶点也叫作节点( nodes)和点(points).**
  + **E中的每一条边连接V中两个不同的顶点。边也叫作弧(arcs)或连线(lines) 。可以用(*i,j)*来表示一条边，其中*i*和*j*是*E*所连接的两个顶点。**
  + **不带方向的边叫无向边(undirected edge)。对无向边来说，(*i*,*j*)和(*j*,*i*)是一样的。**
  + **带方向的边叫有向边(directed edge)，而对有向边来说，(*i*,*j*)和(*j*,*i*)是不同的。**

**有向图、无向图、带权有向图、带权无向图**

**2、描述**

**邻接矩阵、邻接压缩表、邻接链表**

**（1）邻接矩阵**

**（2）邻接压缩表**

**（3）邻接链表**

**3、基本操作及应用**

**（1）基本操作**

**求是否存在边、顶点度数…,DFS、BFS**

**（2）应用**

**求路径、求连通分支、判别连通性…**

**三、贪心算法**

**1、单源最短路径**

**2、拓扑排序**

**3、最小耗费生成树**

**四、分而治之**

**1、快速排序**

**2、归并排序**

**习题：**

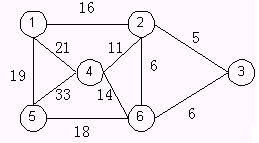
**1、归并排序-优先队列练习23**

**2、已知图G的邻接矩阵如下所示：**



**由邻接矩阵画出相应的图G；图中所有顶点是否都在它的拓扑有序序列中？**

**3、分别用深度优先搜索和宽度优先搜索遍历下图所示的无向图，给出以1为起点的顶点访问序列（同一个顶点的多个邻接点，按数字顺序访问），给出一棵深度优先生成树和宽度优先生成树。**

****

**4、求最小生成树**