第七章 图

一、判断题

1.一个无向图的邻接矩阵中各非零元素之和与图中边的条数相等。（ ）

2.一个有向图的邻接矩阵中各非零元素之和与图中边的条数相等。（ ）

3.一个对称矩阵一定对应着一个无向图。（ ）

4.一个有向图的邻接矩阵一定是一个非对称矩阵。（ ）

二、选择题

1.在一个图中,所有顶点的度数之和等于所有边数的（ ）倍。

（A）1/2（B）1（C）2（D）4

2.在一个有向图中,所有顶点的入度之和等于所有顶点的出度之和的（ ）倍。

（A）1/2（B）1（C）2（D）4

3.一个有 n 个顶点的无向图最多有（ ）条边。

（A）n （B）n(n-1)（C）n(n-1）/2（D）2n

4.具有 4 个顶点的无向完全图有（ ）条边。

（A）6（B）12（C）16（D）20

5.具有 6 个顶点的无向图至少应有（ ）条边才能确保是一个连通图。

（A）5（B）6（C）7（D）8

6.在一个具有 n 个顶点的无向图中,要连通全部顶点至少需要（ ）条边。

（A）n （B）n+1（C）n-1（D）n/2

7.对于一个具有 n 个顶点的无向图,若采用邻接矩阵表示,则该矩阵的大小（ ）

（A）n （B）(n-1) 2（C）n-1 （D）n2

8.对于一个具有 n 个顶点和 e 条边的无向图,若采用邻接表表示,则表头向量的大小为（ ），所有邻接表中的结点总数是（ ）。

①（A）n （B）n+1（C）n-1（D）n+e

②（A）e/2（B）e（C）2e （D）n+e

9.采用邻接表存储的图的深度优先遍历算法类似于二叉树的（ ）。

（A）先序遍历（B）中序遍历（C）后序遍历（D）按层遍历

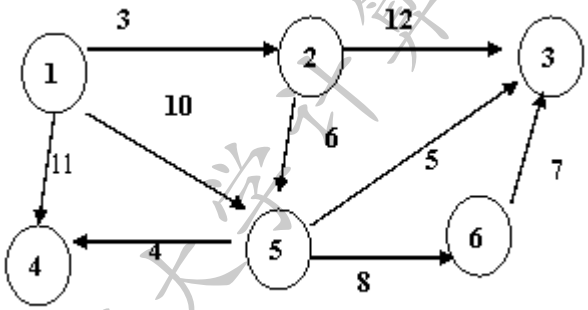
10.采用邻接表存储的图的宽度优先遍历算法类似于二叉树的（ ）。

（A）先序遍历（B）中序遍历（C）后序遍历（D）按层遍历

11.判定一个有向图是否存在回路除了可以利用拓扑排序方法外,还可以利用（ ）.

（A）求关键路径的方法（B）求最短路径的 Dijkstm 方法

（C）宽度优先遍历算法（D）深度优先遍历算法



12.用 Prim 算法求下列连通的带权图的最小代价生成树,在算法执行的某刻，已选取的顶点集合 U={1，2，5}，边的集合 TE={（1，2），（2，5）}，要选取下一条权值最小的边，应当从（ ）组中选取。

(A){（1，4），（3，4），（3，5），（2，5）}

(B){（5，4），（5，3），（5，6）}

(C){（1，2），（2，3），（3，5）}

(D){（3，4），（3，5），（4，5），（1，4）}

三、填空题

1.n 个顶点的连通图至少_____条边。

2.在一个无环有向图 G 中，若存在一条从顶点 i 到顶点 j 的弧，则在顶点的拓扑序列中，顶点 i 与顶点 j 的先后次序是_____。

3.在一个无向图的邻接表中，若表结点的个数是 m，则图中边的条数是_____条。

4. 如果从一个顶点出发又回到该顶点，则此路径叫做_____。

5.如果从一无向图的任意顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是_____。

6.若采用邻接表的存储结构，则图的广度优先搜索类似于二叉树的_____遍历。

7. 一个连通图的生成树是该图的_____连通子图。若这个连通图有 n 个顶点，则它的生成树有_____条边。

四、算法设计：

1.试在无向图的邻接矩阵和邻接链表上实现如下算法：

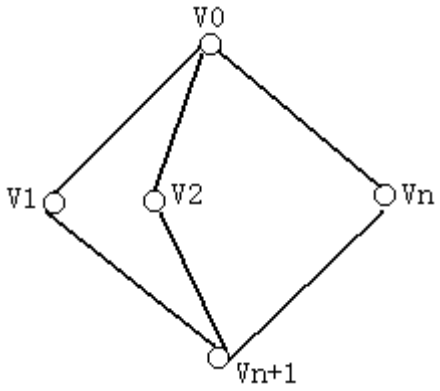
(1)往图中插入一个顶点

(2)往图中插入一条边

(3)删去图中某顶点

(4)删去图中某条边

2.下面的伪代码是一个广度优先搜索算法，试以下图中的 v0 为源点执行该



算法，请回答下述问题：

(1)对图中顶点 vn+1，它需入队多少次？它被重复访问多少次？

(2)若要避免重复访问同一个顶点的错误，应如何修改此算法？

```
void BFS(ALGraph *G,int k)
{//以下省略局部变量的说明，visited 各分量初值为假
InitQueue(&Q);//置空队列
EnQueue(&Q,k);//k 入队
while(!QueueEmpty(&Q)){
i=DeQueue(&Q);//vi 出队
visited[i]=TRUE;//置访问标记
printf("%c",G->adjlist[i].vertex;//访问 vi
for(p=G->adjlist[i].firstedge;p;p=p->next)
//依次搜索 vi 的邻接点 vj(不妨设 p->adjvex=j)
if(!visited[p->adjvex])//若 vj 没有访问过
EnQueue(&Q,p->adjvex);//vj 入队
}//endofwhile
}//BFS
```

3.试以邻接表和邻接矩阵为存储结构，分别写出基于 DFS 和 BFS 遍历的算

法来判别顶点 vi 和 vj(i<>j)之间是否有路径。

4.试分别写出求 DFS 和 BFS 生成树(或生成森林)的算法，要求打印出所有的树边。

5.写一算法求连通分量的个数并输出各连通分量的顶点集。

6.设图中各边的权值都相等，试以邻接矩阵和邻接表为存储结构，分别写出算法：

(1)求顶点 vi 到顶点 vj(i<>j)的最短路径

(2)求源点 vi 到其余各顶点的最短路径

要求输出路径上的所有顶点(提示：利用 BFS 遍历的思想)

7.以邻接表为存储结构，写一个基于 DFS 遍历策略的算法，求图中通过某顶点 vk 的简单回路(若存在)。

8.写一算法求有向图的所有根(若存在)，分析算法的时间复杂度。

参考答案：

一、判断题

1、×2、√3、×4、×

二、选择题

1. C 2.B 3. C 4. A 5. A 6.C 7、B 8、A、C 9、A 10、D 11、D 12、B

三、填空题

1、n-1 2、i 在前，j 在后 3、m/2 4、回路 5、强连通图 6、按层 7、极大；n-1