第七章图 自测卷答案　　姓名_____　班级_____

题号	一	二	三	四	五	总分
题分	16	20	24	10	30	100
得分						

一、单选题（每题 1 分，共 16 分）

（ C ）1. 有 8 个结点的有向完全图有_____条边。

A. 14　　B. 28　　C. 56　　D. 112

（ A ）2. 图中有关路径的定义是_____。

A. 由顶点和相邻顶点序偶构成的边所形成的序列

B. 由不同顶点所形成的序列

C. 由不同边所形成的序列

D. 上述定义都不是

（ B ）3. 一个有 n 个结点的图，最少有_____个连通分量

A. 0　　B. 1　　C. n-1　　D. n

（ A ）4. 深度优先遍历类似于二叉树的

A. 先序遍历　　B. 中序遍历　　C. 后序遍历　　D. 层次遍历

（ B ）5. 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

（ A ）6. 用邻接表表示图进行深度优先遍历时，通常是采用_____来实现算法的。

A. 栈　　B. 队列　　C. 树　　D. 图

（ D ）7. 已知图的邻接矩阵同上题 8，根据算法，则从顶点 0 出发，按深度优先遍历的结点序列是

A. 0 2 4 3 1 5 6　　B. 0 1 3 5 6 4 2　　C. 0 4 2 3 1 6 5　　D. 0 1 3 4 2 5 6

（ C ）8. 有 8 个结点的无向连通图最少有_____条边。

A. 5　　B. 6　　C. 7　　D. 8

（ B ）9. 用邻接表表示图进行广度优先遍历时，通常是采用_____来实现算法的。

A. 栈　　B. 队列　　C. 树　　D. 图

（ C ）10. 已知图的邻接矩阵，根据算法思想，则从顶点 0 出发按深度优先遍历的结点

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

A. 0243156
B. 0136542
C. 0423165
D. 0361542

序列是

（ A ）11 任何一个无向连通图的最小生成树

A. 只有一棵　　B. 一棵或多棵　　C. 一定有多棵　　D. 可能不存在

（注，生成树不唯一，但最小生成树唯一，即边权之和或树权最小的情况唯一）

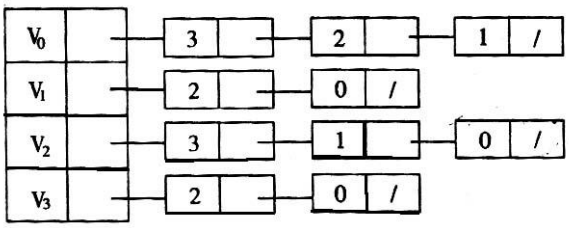
（ B ）12. 已知图的邻接矩阵同上题 8，根据算法，则从顶点 0 出发，按广度优先遍历的结点序列是

A. 0243651　　B. 0136425　　C. 0423156　　D. 0134256

（ C ）13. 已知图的邻接矩阵同上题 8，根据算法，则从顶点 0 出发，按广度优先遍历的结点序列是

A. 0243165　　B. 0135642　　C. 0123465　　D. 0123456

（ A ）14. 已知图的邻接表如下所示，根据算法，则从顶点 0 出发按广度优先遍历的结点序列是



A. 0321　　B. 0123
C. 0132　　D. 0312

（ D ）15. 广度优先遍历类似于二叉树的

A. 先序遍历　　B. 中序遍历　　C. 后序遍历　　D. 层次遍历

（ A ）16. 用有向无环图描述表达式(A+B)*（(A+B）/A)，至少需要顶点的数目为_____

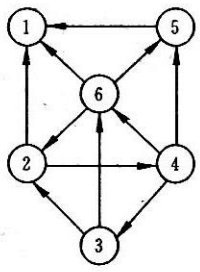
A. 5　　B. 6　　C. 8　　D. 9

二、填空题（每空 1 分，共 20 分）

1. 设有一稀疏图 G，则 G 采用__邻接表___存储较省空间。
2. 图的 BFS 生成树的树高比 DFS 生成树的树高_小或相等____。
3. 有向图 G 用邻接表矩阵存储，其第 i 行的所有元素之和等于顶点 i 的___出度___。
4. 有向图 G 的强连通分量是指_极大强连通子图_。
5. 设 G 为具有 N 个顶点的无向连通图，则 G 中至少有__N-1__条边。
6. 如果含 n 个顶点的图形形成一个环，则它有__n__棵生成树。
7. 对于一个具有 n 个顶点 e 条边的无向图的邻接表的表示，则表头向量大小为__n__，邻接表的边结点个数为__2*e___。
8. 若要求一个稀疏图 G 的最小生成树，最好用_克鲁斯卡尔(Kruskal)_算法来求解。
9 图的逆邻接表存储结构只适用于___有向__图。
10. 用 Dijkstra 算法求某一顶点到其余各顶点间的最短路径是按路径长度___递增___的次序来得到最短路径的。
11. n 个顶点 e 条边的图，若采用邻接矩阵存储，则空间复杂度为___$O(n^2)$____。
12. n 个顶点 e 条边的图采用邻接矩阵存储，广度优先遍历算法的时间复杂度为__$O(n^2)$__；若采用邻接表存储，该算法的时间复杂度为___O(n+e)___。
13. n 个顶点 e 条边的图，若采用邻接表存储，则空间复杂度为____O(n+e)____。
14. 图的深度优先遍历序列__不是__惟一的。
15. 设有一稠密图 G，则 G 采用__邻接矩阵___存储较省空间。
16. 若要求一个稀疏图 G 的最小生成树，最好用_克鲁斯卡尔(Kruskal)_算法来求解。
17. n 个顶点 e 条边的图采用邻接矩阵存储，深度优先遍历算法的时间复杂度为 $O(n^2)$___；若采用邻接表存储时，该算法的时间复杂度为__O(n+e)___。
18. 若要求一个稠密图 G 的最小生成树，最好用__普里姆(Prim)___算法来求解。
19. 用普里姆(Prim)算法求具有 n 个顶点 e 条边的图的最小生成树的时间复杂度为 $O(n^2)$_____；用克鲁斯卡尔(Kruskal)算法的时间复杂度是___$O(elog_2e)$_____。
20 已知一个图的邻接矩阵表示，删除所有从第 i 个顶点出发的方法是_将邻接矩阵的第 i 行全部置 0__。

三、简答题（每题 8 分，共 24 分）

1. 【严题集 7.1①】已知如图所示的有向图，请给出该图的:



（1） 每个顶点的入/出度；
（2） 邻接矩阵；
（3） 邻接表；
（4） 逆邻接表。

答案：

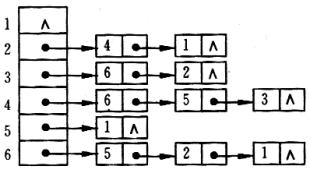
7.1 (1)

顶点	1	2	3	4	5	6
入度	3	2	1	1	2	2
出度	0	2	2	3	1	3

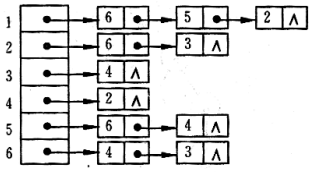
(2) 邻接矩阵

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(3) 邻接表



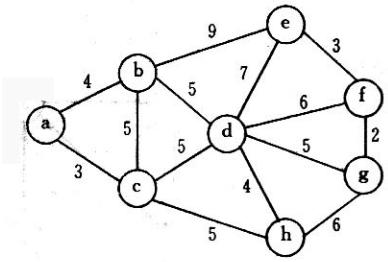
(4) 逆邻接表



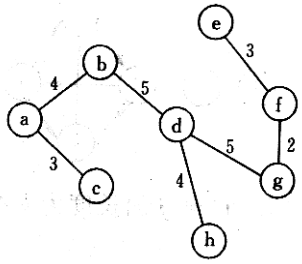
2. 【严题集 7.7②】请对下图的无向带权图：

（1） 写出它的邻接矩阵，并按普里姆算法求其最小生成树；

（2） 写出它的邻接表，并按克鲁斯卡尔算法求其最小生成树。



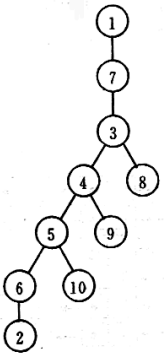
最小生成树：



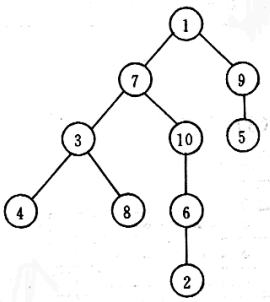
3. 【严题集 7.5②】已知二维数组表示的图的邻接矩阵如下图所示。试分别画出自顶点 1 出发进行遍历所得的深度优先生成树和广度优先生成树。

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	1	0	1	0
2	0	0	1	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0	1	0	0
4	0	0	0	0	1	0	0	0	1	0
5	0	0	0	0	0	1	0	0	0	1
6	1	1	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	1
8	1	0	0	1	0	0	0	0	1	0
9	0	0	0	0	1	0	1	0	0	1
10	1	0	0	0	0	1	0	0	0	0

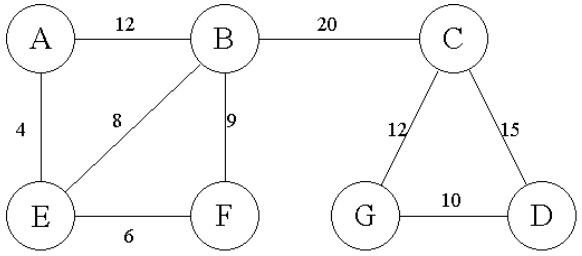
深度优先生成树



广度优先生成树



四、 【2001 年计考研题】给定下列网 G: （10 分）



1 试着找出网 G 的最小生成树，画出其逻辑结构图；

2 用两种不同的表示法画出网 G 的存储结构图；

3 用 C 语言（或其他算法语言）定义其中一种表示法（存储结构）的数据类型。

五、算法设计题（每题 10 分，共 30 分）

1 用伪码编写广度优先搜索。

解：bool visited[MAX_VERTEX_NUM];//访问数组，也就是顶点个数

```
void BFSTraverse(Graph G)
//外层的函数，为准备实现遍历做一些准备工作。
{
   for（int i=0;i<G.vexnum;++i）
        visited[i]=false;//先将所有的顶点都设置为没有被访问过
   InitQueue(Q);//初始化辅助队列方便遍历顶点
   for(int i=0;i<G.vexnum;++i)
        if(!visited[i])
              BFS(G,i);
//外层循环使用 if 语句来调用 BFS 的原因是为了防止有的顶点它不能从初始顶点出发而遍历到，所以这里需要一个完全的循环来避免这种极端情况。
}

void BFS(Graph G,int v)
//从顶点 v 出发，广度优先遍历图 G，算法借助了一个辅助队列 Q
 visit(v);//visit 函数访问这个顶点的信息
 visited[v]=true;//访问过了这个顶点之后就将这个顶点设置为已访问，即 true
 Enqueue(Q,v);//将顶点 v 入队列，这样就可以从队列中出队并访问它的相邻顶点
 while(!isEmpty(Q)){
      DeQueue(Q,v);//将队头的元素出队列存储在 v
      for(w=FirstNeighbor(G,v);w>=0;w=NextNeighbor(G,v,w))//这一步是检查 v 的所有邻接顶点
      if(!visited[w]){
      visit(w);
      visited[w]=true;
```

```
        EnQueue(Q,w);
        //如果 w 没有被访问过，那么访问这个顶点，并把它入队
        }
}
```

2. 【严题集 7.15③】试在邻接矩阵存储结构上实现图的基本操作：DeleteArc(G,v,w)。
（刘提示：删除所有从第 i 个顶点出发的边的方法是 将邻接矩阵的第 i 行全部置 0 ）
解：

```
//本题中的图 G 均为有向无权图,其余情况容易由此写出
Status Delete_Arc(MGraph &G,char v,char w)//在邻接矩阵表示的图 G 上删除边(v,w)
   {
   if((i=LocateVex(G,v))<0) return ERROR;
   if((j=LocateVex(G,w))<0) return ERROR;
   if(G.arcs[i][j].adj)
   {
      G.arcs[i][j].adj=0;
      G.arcnum--;
   }
   return OK;
}//Delete_Arc
```

3.【严题集 7.27④】采用邻接表存储结构，编写一个判别无向图中任意给定的两个顶点之间是否存在一条长度为 k 的简单路径的算法。
（注 1：一条路径为简单路径指的是其顶点序列中不含有重现的顶点。
注 2：此题可参见严题集 P207-208 中有关按“路径”遍历的算法基本框架。）

```
int visited[MAXSIZE];
int exist_path_len(ALGraph G,int i,int j,int k)//判断邻接表方式存储的有向图 G
的顶点 i 到 j 是否存在长度为 k 的简单路径
{
{
   if(i==j&&k==0) return 1; //找到了一条路径,且长度符合要求
   else if(k>0)
   {
      visited[i]=1;
      for(p=G.vertices[i].firstarc;p;p=p->nextarc)
      {
         l=p->adjvex;
         if(!visited[l])
            if(exist_path_len(G,l,j,k-1)) return 1; //剩余路径长度减一
      }//for
      visited[i]=0; //本题允许曾经被访问过的结点出现在另一条路径中
   }//else
   return 0; //没找到
}//exist_path_len
```