

## 一、单选题 ( )

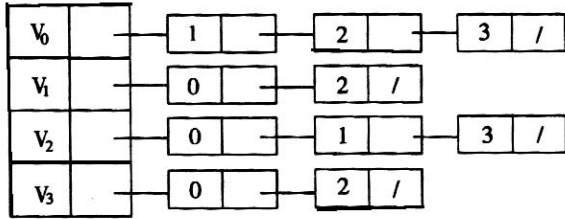
- ( ) 1. 在一个图中, 所有顶点的度数之和等于图的边数的\_\_\_\_\_倍。  
A. 1/2                      B. 1                      C. 2                      D. 4
- ( ) 2. 在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的\_\_\_\_\_倍。  
A. 1/2                      B. 1                      C. 2                      D. 4
- ( ) 3. 有 8 个结点的无向图最多有 \_\_\_\_\_ 条边。  
A. 14                      B. 28                      C. 56                      D. 112
- ( ) 4. 有 8 个结点的无向连通图最少有 \_\_\_\_\_ 条边。  
A. 5                      B. 6                      C. 7                      D. 8
- ( ) 5. 有 8 个结点的有向完全图有 \_\_\_\_\_ 条边。  
A. 14                      B. 28                      C. 56                      D. 112
- ( ) 6. 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
- ( ) 7. 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
- ( ) 8. 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序列是

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

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

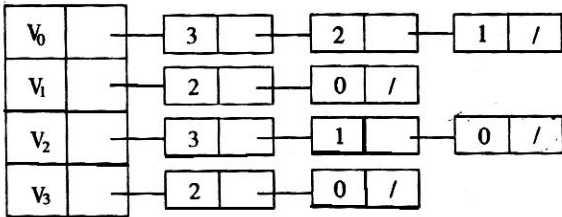
- ( ) 9. 已知图的邻接矩阵同上题 8, 根据算法, 则从顶点 0 出发, 按深度优先遍历的结点序列是  
A. 0 2 4 3 1 5 6      B. 0 1 3 5 6 4 2      C. 0 4 2 3 1 6 5      D. 0 1 3 4 2 5 6
- ( ) 10. 已知图的邻接矩阵同上题 8, 根据算法, 则从顶点 0 出发, 按广度优先遍历的结点序列是  
A. 0 2 4 3 6 5 1      B. 0 1 3 6 4 2 5      C. 0 4 2 3 1 5 6      D. 0 1 3 4 2 5 6
- ( ) 11. 已知图的邻接矩阵同上题 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. 0 2 4 3 1 6 5      B. 0 1 3 5 6 4 2      C. 0 1 2 3 4 6 5      D. 0 1 2 3 4 5 6

( ) 12. 已知图的邻接表如下所示, 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 1 3 2      B. 0 2 3 1  
C. 0 3 2 1      D. 0 1 2 3

( ) 13. 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. 0 3 2 1      B. 0 1 2 3  
C. 0 1 3 2      D. 0 3 1 2

( ) 14. 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

( ) 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

( ) 16. 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

## 二、填空题 ( )

1. 图有\_\_\_\_\_、\_\_\_\_\_等存储结构, 遍历图有\_\_\_\_\_、\_\_\_\_\_等方法。

2. 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\_\_\_\_\_。

3. 如果 n 个顶点的图是一个环, 则它有\_\_\_\_\_棵生成树。

4. 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\_\_\_\_\_。

5. n 个顶点 e 条边的图, 若采用邻接表存储, 则空间复杂度为\_\_\_\_\_。

6. 设有一稀疏图 G, 则 G 采用\_\_\_\_\_存储较省空间。

7. 设有一稠密图 G, 则 G 采用\_\_\_\_\_存储较省空间。

8. 图的逆邻接表存储结构只适用于\_\_\_\_\_图。

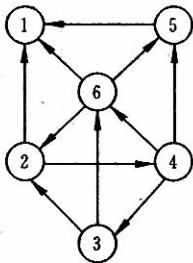
9. 已知一个图的邻接矩阵表示, 删除所有从第 i 个顶点出发的方法是\_\_\_\_\_。

10. 图的深度优先遍历序列\_\_\_\_\_惟一的。

11.  $n$  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\_\_\_\_\_ ; 若采用邻接表存储时, 该算法的时间复杂度为\_\_\_\_\_。
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\_\_\_\_\_ ; 若采用邻接表存储, 该算法的时间复杂度为\_\_\_\_\_。
13. 图的 BFS 生成树的树高比 DFS 生成树的树高\_\_\_\_\_。
14. 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\_\_\_\_\_ ; 用克鲁斯卡尔(Kruskal)算法的时间复杂度是\_\_\_\_\_。
15. 若要求一个稀疏图  $G$  的最小生成树, 最好用\_\_\_\_\_算法来求解。
16. 若要求一个稠密图  $G$  的最小生成树, 最好用\_\_\_\_\_算法来求解。
17. 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\_\_\_\_\_的次序来得到最短路径的。
18. 拓扑排序算法是通过重复选择具有\_\_\_\_\_个前驱顶点的过程来完成的。

### 三、简答题 ( )

1. 已知如图所示的有向图, 请给出该图的:

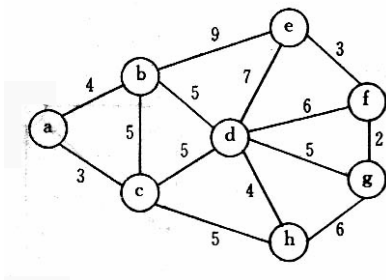


- (1) 每个顶点的入/出度;
- (2) 邻接矩阵;
- (3) 邻接表;
- (4) 逆邻接表。

顶点	1	2	3	4	5	6
入度						
出度						

2. 请对下图的无向带权图:

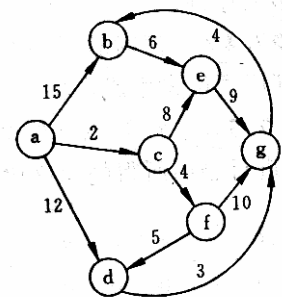
- (1) 写出它的邻接矩阵, 并按普里姆算法求其最小生成树;
- (2) 写出它的邻接表, 并按克鲁斯卡尔算法求其最小生成树。



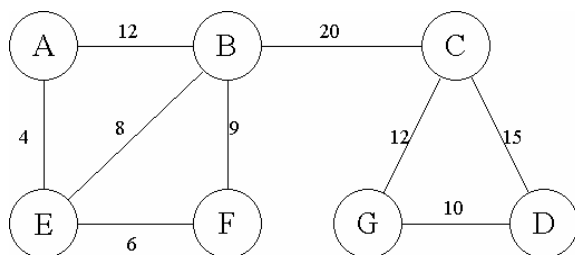
3. 已知二维数组表示的图的邻接矩阵如下图所示。试分别画出自顶点 1 出发进行遍历所得的深度优先生成树和广度优先生成树。

	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

4. 试利用 Dijkstra 算法求图中从顶点 a 到其他各顶点间的最短路径, 写出执行算法过程中各步的状态。



四、给定下列网 G:



- 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 ( )

1. 编写算法, 由依次输入的顶点数目、弧的数目、各顶点的信息和各条弧的信息建立有向图的邻接表。

解: `Status Build_AdjList(ALGraph &G)` //输入有向图的顶点数,边数,顶点信息和边的信息, 以建立邻接表

{

```
    return OK;
} //Build_AdjList
```

2. 试在邻接矩阵存储结构上实现图的基本操作: `DeleteArc(G,v,w)`, 即删除一条边的操作。  
(如果要删除所有从第  $i$  个顶点出发的边呢? 提示: 将邻接矩阵的第  $i$  行全部置 0 )

解: //设本题中的图  $G$  为有向无权图

```
Status DeleteArc(MGraph &G, char v, char w) //在邻接矩阵表示的图 G 上删除边(v,w)
{
```

```
}
```

```
    return OK;
} //Delete_Arc
```

3. 试基于图的深度优先搜索策略写一算法, 判别以邻接表方式存储的有向图中是否存在由顶点  $v_i$  到顶点  $v_j$  的路径 ( $i \neq j$ )。

附加题: 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。)