

得分

一、单项选择题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每题回答正确得 1.5 分, 错误不得分!

- 1、图的 Depth-First Search(DFS)遍历思想实际上是二叉树 ( A ) 遍历方法的推广。  
A. 先序                      B. 中序                      C. 后序                      D. 层序
- 2、AOV 网是一种 ( D )。  
A. 有向图                      B. 无向图                      C. 无向无环图                      D. 有向无环图
- 3、设有 6 个结点的无向图, 该图最少要有 ( A ) 条边才能使图是一个连通图。  
A. 5                      B. 6                      C. 7                      D. 8
- 4、设有 6 个结点的无向图, 该图至少应有 ( D ) 条边才能确保是一个连通图。  
A. 5                      B. 6                      C. 10                      D. 11
- 5、在含  $n$  个顶点和  $e$  条边的无向图的邻接矩阵中, 零元素的个数为 ( D )。  
A.  $e$                       B.  $2e$                       C.  $n^2 - e$                       D.  $n^2 - 2e$
- 6、假设一个有  $n$  个顶点和  $e$  条弧的有向图用邻接表表示, 则删除与某个顶点  $v_i$  相关的所有弧的时间复杂度是 ( C )。  
A.  $O(n)$                       B.  $O(e)$                       C.  $O(n+e)$                       D.  $O(n \cdot e)$
- 7、设某完全无向图中有  $n$  个顶点, 则该完全无向图中有 ( A ) 条边。  
A.  $n(n-1)/2$                       B.  $n(n-1)$                       C.  $n^2$                       D.  $n^2 - 1$
- 8、设某有向图中有  $n$  个顶点, 则该有向图对应的邻接表中有 ( B ) 个表头结点。  
A.  $n-1$                       B.  $n$                       C.  $n+1$                       D.  $2n-1$
- 9、设无向图  $G$  中有  $n$  个顶点  $e$  条边, 则其对应的邻接表中的表头结点和表结点的个数分别为 ( D )。  
A.  $n, e$                       B.  $e, n$                       C.  $2n, e$                       D.  $n, 2e$
- 10、设某强连通图中有  $n$  个顶点, 则该强连通图中至少有 ( C ) 条边。  
A.  $n(n-1)$                       B.  $n+1$                       C.  $n$                       D.  $n(n+1)$
- 11、设某无向图中有  $n$  个顶点  $e$  条边, 则该无向图中所有顶点的入度之和为 ( D )。  
A.  $n$                       B.  $e$                       C.  $2n$                       D.  $2e$
- 12、设某有向图的邻接表中有  $n$  个表头结点和  $m$  个表结点, 则该图中有 ( C ) 条有向边。  
A.  $n$                       B.  $n-1$                       C.  $m$                       D.  $m-1$

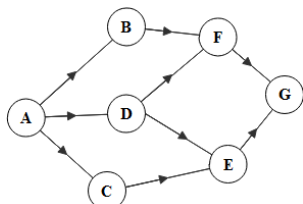
- 13、设连通图  $G$  中的边集  $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点  $a$  出发进行深度优先遍历无法得到的顶点序列为 ( **B** )。
- A. abedfc                      B. acfebd                      C. aebdfc                      D. aedfcb
- 14、设无向图  $G$  中的边的集合  $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点  $a$  出发进行深度优先遍历可以得到的一种顶点序列为 ( **A** )。
- A. aedfcb                      B. acfebd                      C. aebcfcd                      D. aedfbc
- 15、设某无向图中有  $n$  个顶点  $e$  条边，则建立该图邻接表的时间复杂度为 ( **A** )。
- A.  $O(n+e)$                       B.  $O(n^2)$                       C.  $O(ne)$                       D.  $O(n^3)$
- 16、设用邻接矩阵  $A$  表示有向图  $G$  的存储结构，则有向图  $G$  中顶点  $i$  的入度为 ( **B** )。
- A. 第  $i$  行非 0 元素的个数之和                      B. 第  $i$  列非 0 元素的个数之和
- C. 第  $i$  行 0 元素的个数之和                      D. 第  $i$  列 0 元素的个数之和
- 17、设某无向图有  $n$  个顶点，则该无向图的邻接表中有 ( **B** ) 个表头结点。
- A.  $2n$                       B.  $n$                       C.  $n/2$                       D.  $n(n-1)$
- 18、设无向图  $G$  中有  $n$  个顶点，则该无向图的最小生成树上有 ( **B** ) 条边。
- A.  $n$                       B.  $n-1$                       C.  $2n$                       D.  $2n-1$
- 19、设完全无向图中有  $n$  个顶点，则该完全无向图中有 ( **A** ) 条边。
- A.  $n(n-1)/2$                       B.  $n(n-1)$                       C.  $n(n+1)/2$                       D.  $(n-1)/2$
- 20、设有向无环图  $G$  中的有向边集合  $E=\{<1, 2>, <2, 3>, <3, 4>, <1, 4>\}$ ，则下列属于该有向图  $G$  的一种拓扑排序序列的是 ( **A** )。
- A. 1, 2, 3, 4                      B. 2, 3, 4, 1
- C. 1, 4, 2, 3                      D. 1, 2, 4, 3

得分

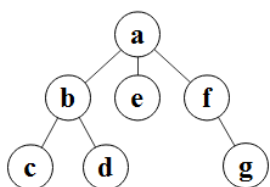
二、填空题 (X 题，每题 X 分，共 X 分)

评分标准：每空回答正确得 1 分，错误不得分，不完全正确酌情给分！

- 1、 $n$  个结点无向完全图的的边数为 【1】  $n(n-1)/2$  ， $n$  个结点的生成树的边数为 【2】  $n-1$  。
- 2、已知一有向无环图如下，任意写出二种拓扑排序序列： 【3】  $ADCBFEG$  、 【4】  $ABCDEF$  。



- 3、在图型结构中，每个结点的前趋结点数和后续结点数可以 【5】 有多个 。
- 4、对于一个具有  $n$  个顶点和  $e$  条边的有向图和无向图，在其对应的邻接表中，所含边结点分别有 【6】  $e$   个和 【7】  $2e$   个。
- 5、AOV 网是一种 【8】 有向无回路  的图。
- 6、在一个具有  $n$  个顶点的无向完全图中，包含有 【9】  $n(n-1)/2$   条边，在一个具有  $n$  个顶点的有向完全图中，包含有 【10】  $n(n-1)$   条边。
- 7、在图的邻接表中，每个结点被称为 【11】 边结点 ，通常它包含三个域：一是 【12】 邻接点域 ；二是 【13】 权域 ；三是 【14】 链域 。
- 8、已知一个图的广度优先生成树如图所示，则与此相应的广度优先遍历序列为 【15】  $abefcdg$  。



- 9、设无向图  $G$  中有  $n$  个顶点和  $e$  条边，则其对应的邻接表中有 【16】  $n$   个表头结点和 【17】  $2e$   个表结点。
- 10、设无向图  $G$  中有  $n$  个顶点  $e$  条边，所有顶点的度数之和为  $m$ ，则  $e$  和  $m$  有 【18】  $m=2e$   关系。
- 11、设一个连通图  $G$  中有  $n$  个顶点  $e$  条边，则其最小生成树上有 【19】  $n-1$   条边。
- 12、设某无向图中顶点数和边数分别为  $n$  和  $e$ ，所有顶点的度数之和为  $d$ ，则  $e=$  【20】  $d/2$  。

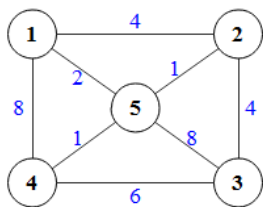
$v_1 \rightarrow 3 \rightarrow 2 \rightarrow 4$

$v_2 \rightarrow 1 \rightarrow 3$

$v_3 \rightarrow 1 \rightarrow 4 \rightarrow 2$

- 13、设某无向图  $G$  的邻接表为  $v_4 \rightarrow 1 \rightarrow 3$ ，则从顶点  $V_1$  开始的深度优先遍历序列为 【21】  $(1, 3, 4, 2)$  ；广度优先遍历序列为 【22】  $(1, 3, 2, 4)$  。
- 14、设有向图  $G$  用邻接矩阵  $A[n][n]$  作为存储结构，则该邻接矩阵中第  $i$  行上所有非无穷大( $\infty$ )元素个数等于顶点  $i$  的 【23】 出度 ，第  $i$  列上所有非无穷大( $\infty$ )元素个数等于顶点  $i$  的 【24】 入度 （填出度或入度）。

- 15、设有向图  $G$  中有  $n$  个顶点  $e$  条有向边，所有的顶点入度数之和为  $d$ ，则  $e$  和  $d$  的关系为 **【25】  $e=d$** 。
- 16、设有向图  $G$  中有向边的集合  $E=\{<1, 2>, <2, 3>, <1, 4>, <4, 2>, <4, 3>\}$ ，则该图的一种拓扑序列为 **【26】  $(1, 4, 2, 3)$** 。
- 17、设某无向图  $G$  中有  $n$  个顶点，用邻接矩阵  $A$  作为该图的存储结构，则顶点  $i$  和顶点  $j$  互为邻接点的条件是 **【27】  $A[i][j]=1$** 。
- 18、设无向图对应的邻接矩阵为  $A$ ，则  $A$  中第  $i$  上非 0 元素的个数 **【28】 等于** 第  $i$  列上非 0 元素的个数（填等于，大于或小于）。
- 19、在图的邻接表中用顺序存储结构存储表头结点的优点是 **【29】 可以随机访问到任何一个顶点的简单链表**。
- 20、设有向图  $G$  的存储结构用邻接矩阵  $A$  来表示，则  $A$  中第  $i$  行中所有非零元素个数之和等于顶点  $i$  的 **【30】 出度**，第  $i$  列中所有非零元素个数之和等于顶点  $i$  的 **【31】 入度**。
- 21、设有向图  $G$  的元组形式表示为  $G = (D, R)$ ， $D=\{1, 2, 3, 4, 5\}$ ， $R=\{r\}$ ， $r=\{<1,2>, <2,4>, <4,5>, <1,3>, <3,2>, <3,5>\}$ ，则给出该图的一种拓扑排序序列 **【32】  $(1, 3, 2, 4, 5)$** 。
- 22、设无向图  $G$  中有  $n$  个顶点，则该无向图中每个顶点的度数最多是 **【33】  $n-1$** 。
- 23、设完全有向图中有  $n$  个顶点，则该完全有向图中共有 **【34】  $n(n-1)$**  条有向边；设完全无向图中有  $n$  个顶点，则该完全无向图中共有 **【35】  $n(n-1)/2$**  条无向边。
- 24、设有向图中不存在有向边  $<V_i, V_j>$ ，则其对应的邻接矩阵  $A$  中的数组元素  $A[i][j]$  的值等于 **【36】 0**。
- 25、设连通图  $G$  中有  $n$  个顶点  $e$  条边，则对应的最小生成树上有 **【37】  $n-1$**  条边。
- 26、设无向图  $G$ （如右图所示），则其最小生成树上所有边的权值之和为 **【38】 8**。



- 27、设无向图  $G$  中有  $n$  个顶点  $e$  条边，则用邻接矩阵作为图的存储结构进行深度优先或广度优先遍历时的时间复杂度为 **【39】  $O(n^2)$** ；用邻接表作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 **【40】  $O(n+e)$** 。
- 28、设有向图  $G$  中的有向边的集合  $E=\{<1, 2>, <2, 3>, <1, 4>, <4, 5>, <5, 3>, <4, 6>, <6, 5>\}$ ，则该图的一个拓扑序列为 **【41】  $124653$  或者  $14[2]653, 146[2]53, 1465[2]3$** 。

得分

三、判断题（X 题，每题 X 分，共 X 分。正确填 ‘T’，错误 ‘F’。）

评分标准：每题回答正确得 2 分，错误不得分！

- 1、调用一次深度优先遍历可以访问到图中的所有顶点。 ( F )
- 2、对连通图进行深度优先遍历可以访问到该图中的所有顶点。 ( T )
- 3、带权无向图的最小生成树是唯一的。 ( F )
- 4、如果某个有向图的邻接表中第 i 条单链表为空，则第 i 个顶点的出度为零。 ( T )
- 5、图的深度优先遍历算法中需要设置一个标志数组，以便区分图中的每个顶点是否被访问过。 ( T )
- 6、有向图的邻接表和逆邻接表中表结点的个数不一定相等。 ( F )
- 7、用邻接矩阵作为图的存储结构时，则其所占用的存储空间与图中顶点数无关而与图中边数有关。 ( F )

得分

四、阅读程序题（X 题，每题 X 分，共 X 分）

评分标准：每空回答正确得 1 分，错误不得分，不完全正确酌情给分！

- 1、阅读程序回答问题。（评分标准：第 1 点 4 分，第 2 点 1 分！）

```
void AJ(adjlist GL, int i, int n) {
    Queue Q;
    InitQueue(Q);
    cout<<i<<' ';
    visited[i]=true;
    QInsert(Q,i);
    while(!QueueEmpty(Q)) {
        int k=QDelete(Q);
        edgenode* p=GL[k];
        while(p!=NULL) {
            int j=p->adjvex;
            if(!visited[j]) {
                cout<<j<<' ';
                visited[j]=true;
                QInsert(Q,j);
            }
            p=p->next;
        }
    }
}
```

```

    }
}

```

(1) 写出算法实现的功能是什么。

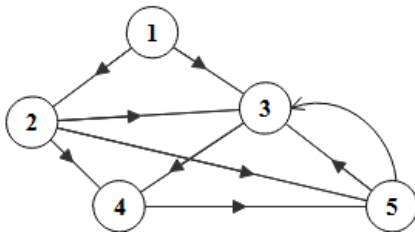
答：功能为：从初始点  $v_i$  出发广度优先搜索由邻接表 GL 所表示的图。

得分

五、分析题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每题回答完全正确得 5 分, 其余按得分点给分!

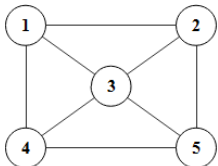
- 1、对于图所示的有向图, 若其存储采用邻接表, 并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的, 试写出: (1) 从顶点①出发进行深度优先搜索所得到的深度优先生成树; (2) 从顶点②出发进行广度优先搜索所得到的广度优先生成树。(评分标准: 第1点4分, 第2点1分!)



答: (1) DFS: ①②③④⑤

(2) BFS: ②③④⑤: ①

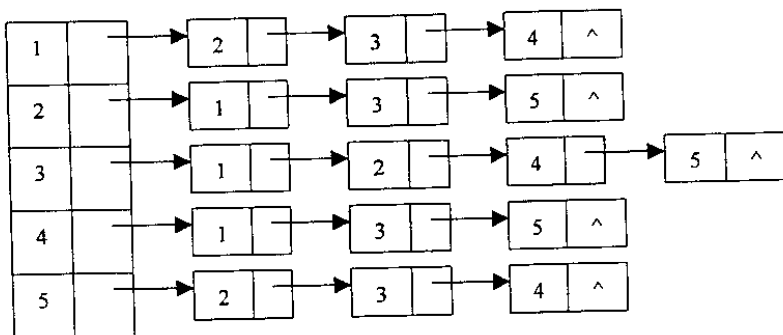
- 2、请画出图对应的邻接矩阵和邻接表。(评分标准: 第1点4分, 第2点1分!)



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

答: (1) 邻接矩阵:

(2) 邻接表:



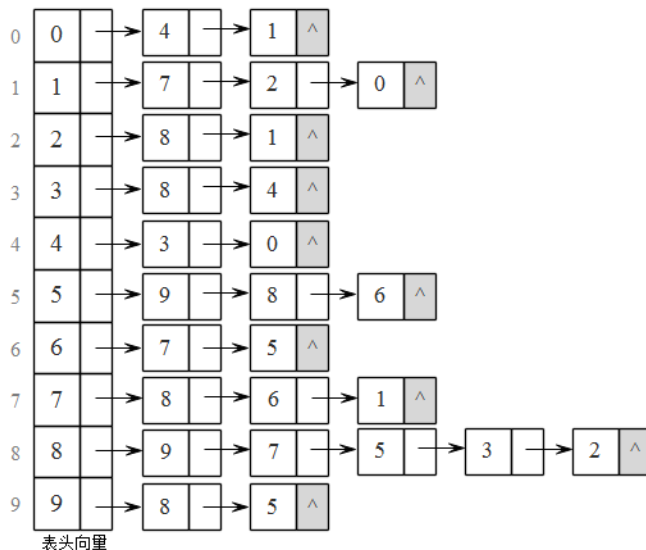
3、已知一个图的顶点集  $V$  和边集  $E$  分别为:  $V=\{1, 2, 3, 4, 5, 6, 7\}$ ;  $E=\{(1,2)3, (1,3)5, (1,4)8, (2,5)10, (2,3)6, (3,4)15, (3,5)12, (3,6)9, (4,6)4, (4,7)20, (5,6)18, (6,7)25\}$ 。用克鲁斯卡尔算法得到最小生成树, 试写出在最小生成树中依次得到的各条边。(评分标准: 第1点4分, 第2点1分!)

答: 用克鲁斯卡尔算法得到的最小生成树为:  $(1,2)3$ ,  $(4,6)4$ ,  $(1,3)5$ ,  $(1,4)8$ ,  $(2,5)10$ ,  $(4,7)20$ ;

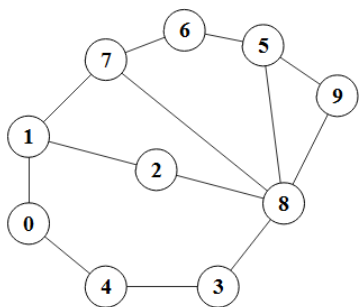
4、已知一个图的顶点集  $V$  各边集  $G$  如下:  $V=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ;  $E=\{(0, 1), (0, 4), (1, 2), (1, 7), (2, 8), (3, 4), (3, 8), (5, 6), (5, 8), (5, 9), (6, 7), (7, 8), (8, 9)\}$  当它用邻接矩阵表示和邻接表表示时, 分别写出从顶点  $V=0$  出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历等到的顶点序列。假定每个顶点邻接表中的结点是按顶点序号从大到小的次序链接的。(评分标准: 第1点4分, 第2点1分!)

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

(a) 邻接矩阵



(b) 邻接链表



答:

图	深度优先序列	广度优先序列
1	0, 1, 2, 8, 3, 4, 5, 6, 7, 9	0, 1, 4, 2, 7, 3, 8, 6, 5, 9
邻接表表示时	0, 4, 3, 8, 9, 5, 6, 7, 1, 2	0, 4, 1, 3, 7, 2, 8, 6, 9, 5

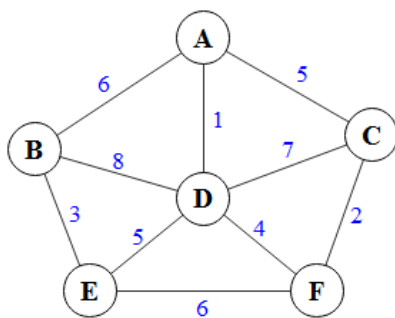
得分

六、应用题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每题回答完全正确得 5 分, 其余按得分点给分!

- 1、已知一网络的邻接矩阵如下, 求从顶点 A 开始的最小生成树 (要求有过程)。(评分标准: 共 8 分。第 1 点 4 分, 第 2 点 1 分!)

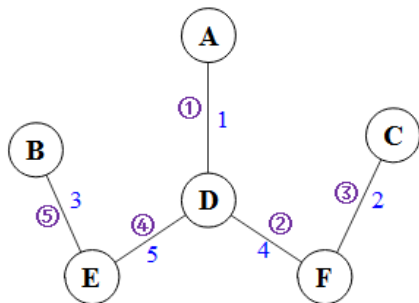
	A	B	C	D	E	F
A	$\infty$	6	5	1	$\infty$	$\infty$
B	6	$\infty$	$\infty$	8	3	$\infty$
C	5	$\infty$	$\infty$	7	$\infty$	2
D	1	8	7	$\infty$	5	4
E	$\infty$	3	$\infty$	5	$\infty$	6
F	$\infty$	$\infty$	2	4	6	$\infty$



(1) 求从顶点 A 开始的最小生成树。

(2) 分别画出以 A 为起点的 DFS 生成树和 BFS 生成树。

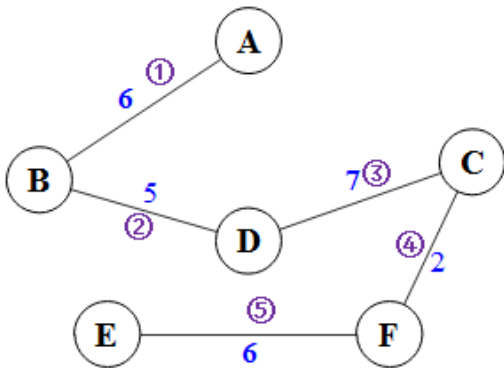
答: (1) 搜索顺序由①②③④⑤标识。(4 分)



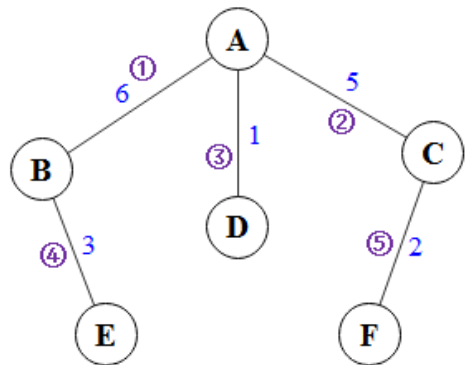
从 A 开始的最小生成树

(2) 搜索顺序由①②③④⑤标识。(4 分)





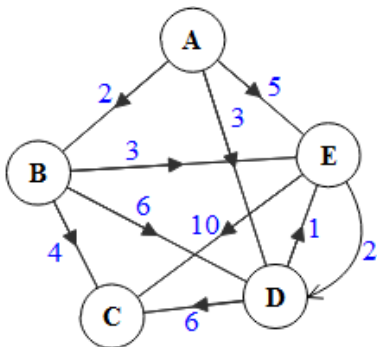
从 A 开始的 DFS 生成树  
DFS: ABDCFE



从 A 开始的 BFS 生成树  
BFS: ABCDEF

2、已知有如下的有向网，求顶点 A 到其它各顶点的最短路径（采用 Dijkstra 算法，要有过程）。

（评分标准：共 6 分。第 1 点 4 分，第 2 点 1 分！）



答：

第 1 步：初始时刻

下标 0 1 2 3 4

顶点	A	B	C	D	E	S
dist[]	-	2	$\infty$	3	5	{0, }
pre[]	0	0	0	0	0	

第 2 步：找到顶点 B 的最短路径（A->B[2]），更新顶点 C, D, E（实际只更新 C）

下标 0 1 2 3 4

顶点	A	B	C	D	E	S
dist[]	-	2	6	3	5	{0, 2}
pre[]	0	0	1	0	0	

第 3 步：找到顶点 D 的最短路径（A->D[3]），更新顶点 C, E（实际只更新 E）

下标 0 1 2 3 4

顶点	A	B	C	D	E	S
dist[]	-	2	6	3	4	{0, 2, 3}
pre[]	0	0	1	0	3	

第 4 步：找到顶点 E 的最短路径（A->D->E[4]），更新顶点 C（实际没发生更新）

下标 0 1 2 3 4

顶点	A	B	C	D	E	S
dist[]	-	2	6	3	4	{ 0, 2, 3, 4 }
pre[]	0	0	1	0	3	}

第 5 步：找到顶点 C 的最短路径 (A->B->C[6])，所有顶点已找到最短路径，结束！

【1】A-B: (A、B) (1 分)

【2】A-D: (A、D) (1 分)

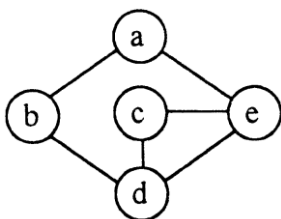
【3】A-E: (A、D、E) (2 分)

【4】A-C: (A、B、C) (2 分)

3、已知一个无向图的顶点集为{a, b, c, d, e} ,其邻接矩阵如下所示。请 (1) 画出该图的图形；  
(2)根据邻接矩阵从顶点 a 出发进行深度优先遍历和广度优先遍历，写出相应的遍历序列。

(评分标准：共 5 分。第 1 点 4 分，第 2 点 1 分!)

$$\begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

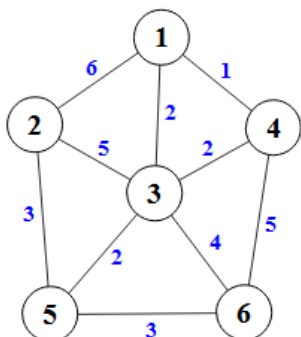
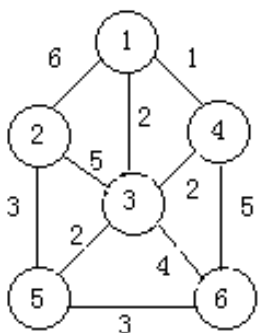


答：(1) 该图的图形为：

(2) 深度优先遍历序列为：abdce；

广度优先遍历序列为：abedc；

4、设无向图 G 及其存储结构如下所示，要求给出从顶点 1 开始的图的深度优先和广度优先遍历序列、并给出该图的最小生成树。(评分标准：共 5 分。第 1 点 4 分，第 2 点 1 分!)



$$\begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} \infty & 6 & 2 & 1 & \infty & \infty \\ 6 & \infty & 5 & \infty & 3 & \infty \\ 2 & 5 & \infty & 2 & 2 & 4 \\ 1 & \infty & 2 & \infty & \infty & 5 \\ \infty & 3 & 2 & \infty & \infty & 3 \\ \infty & \infty & 4 & 5 & 3 & \infty \end{bmatrix} \end{matrix}$$

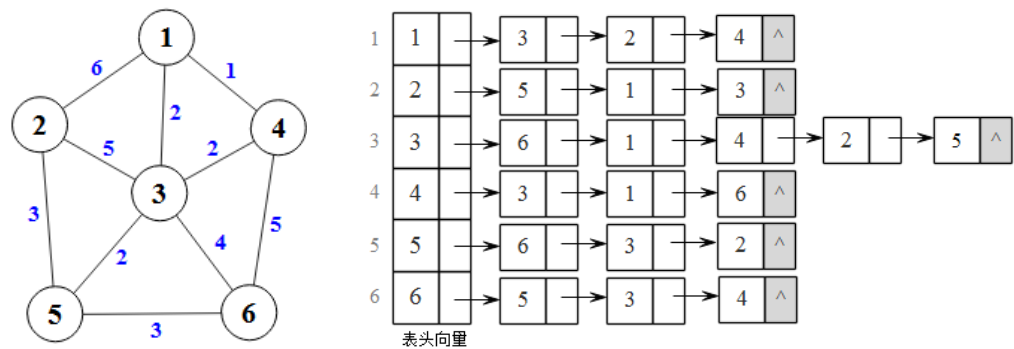
答：(1) 深度优先遍历序列：123465；

(2) 广度优先遍历序列：123456；

(3) 最小生成树 T 的边集为 E={(1, 4), (1, 3)或(4, 3), (3, 5), (5, 2), (5, 6)}，最后两条

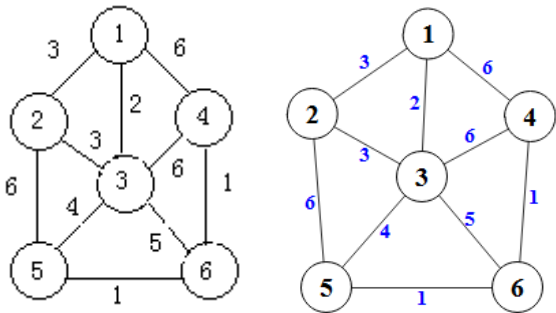
边(5, 2)和(5, 6)的顺序可交换。

5、设无向图 G 及其存储结构如下所示，要求给出从顶点 1 开始的图的深度优先和广度优先遍历序列、并给出该图的最小生成树。（评分标准：共 5 分。第 1 点 4 分，第 2 点 1 分!）

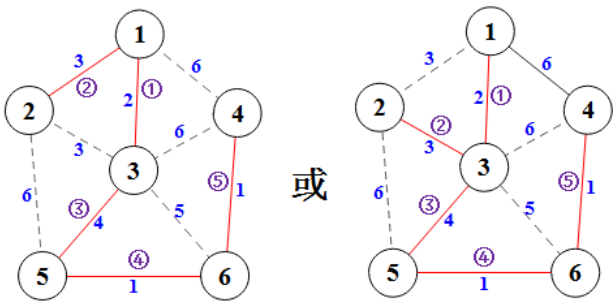


答：（1）深度优先遍历序列：136524；  
 （2）广度优先遍历序列：132465；  
 （3）最小生成树 T 的边集为  $E=\{(1, 4), (1, 3)\text{或}(4, 3), (3, 5), (5, 2), (5, 6)\}$ ，最后两条边(5, 2)和(5, 6)的顺序可交换。

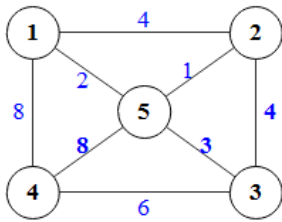
6、设有无向图 G（如右图所示），要求给出从顶点 1 出发、用普里姆算法构造最小生成树所走过的边的集合。（评分标准：共 5 分。第 1 点 4 分，第 2 点 1 分!）



答：  $E=\{(1, 3), (1, 2)\text{或}(3, 2), (3, 5), (5, 6), (6, 4)\}$



7、设无向图 G（如图所示），要求给出从顶点 1 出发的最小生成树上边的集合，并计算最小生成树各边上的权值之和。（评分标准：共 5 分。第 1 点 4 分，第 2 点 1 分!）



答：(1)  $E=\{(1, 5), (5, 2), (5, 3), (3, 4)\}$ ;

(2)  $W=12$ ;

得分

七、编程题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每小题回答正确得 10 分, 不完全正确则按得分点给分。

1、设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。实现算法的函数原型为: `void adjmatrixtoadjlist(gadjmatrix g1[], glinkheadnode g2[])`; (评分标准: 第 1 点 4 分, 第 2 点 1 分!)

答: `typedef struct {  
    int vertex[m];  
    int edge[m][m];  
} gadjmatrix;`

`typedef struct node1 {  
    int info;  
    int adjvertex;  
    struct node1 *nextarc;  
} glinklistnode;`

`typedef struct node2 {  
    int vertexinfo;  
    glinklistnode *firstarc;  
} glinkheadnode;`

`void adjmatrixtoadjlist( gadjmatrix g1[], glinkheadnode g2[] ) {  
    int i,j;  
    glinklistnode *p;  
    for(i=0; i<=n-1; i++)  
        g2[i].firstarc=0;  
  
    for(i=0; i<=n-1; i++)  
        for(j=0; j<=n-1; j++)  
            if (g1.edge[i][j]==1) {  
                p=(glinklistnode *)malloc(sizeof(glinklistnode));`

```
p->adjvertex=j;  
p->nextarc=g[i].firstarc;  
g[i].firstarc=p;
```

```
p=(glinklistnode *)malloc(sizeof(glinklistnode));  
p->adjvertex=i;  
p->nextarc=g[j].firstarc;  
g[j].firstarc=p;
```

```
}
```

```
}
```