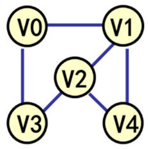
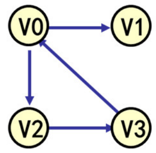
第 5 章 图

5.1 图的基本概念

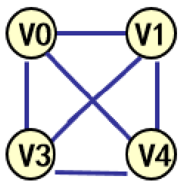
(1)无向图：<u>每条边都是无方向的</u>；

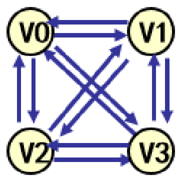


(2)有向图：<u>每条边都是有方向的</u>；



(3)完全图：<u>任意两个顶点上都存在一个边</u>；





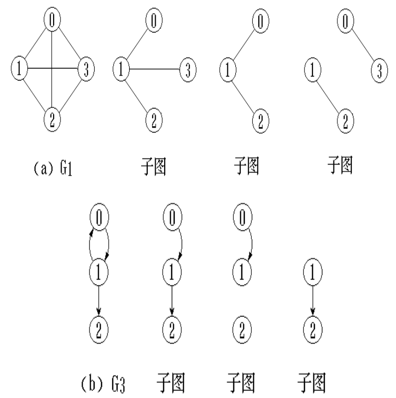
完全无向图　　　完全有向图

a、n 个顶点的完全无向图有<u> n(n-1)/2 </u>条边；

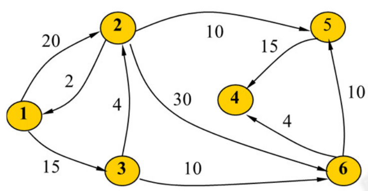
b、n 个顶点的完全有向图有<u>　 n(n-1)　</u>条边；

c、无向图 G 中顶点数为 n，则图 G 至少有<u>　0　</u>条边，至多有<u>　n(n-1)/2　</u>条边；若 G 为有向图，则至少有<u>　0　</u>条边，至多有<u>　n(n-1)/2　</u>条边。

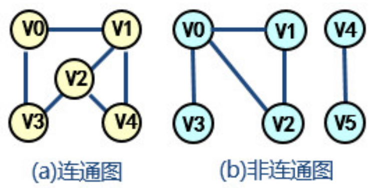
(4)子图：<u>从图中取出的部分集合</u>；



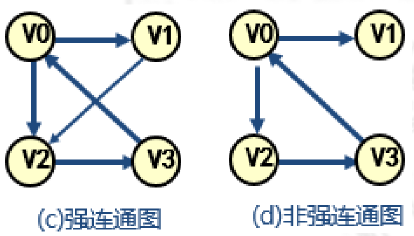
(5)带权图：<u>边上带权的图</u>；(权：具有某种含义的数值，如表示两点之间的距离)



(6)连通图：无向图中任意两个点都存在路径可达；



(7)强连通图：有向图中，若任意两个顶点都存在路径可达；



(8)顶点的度：与顶点有关的边的数目；

有向图有分为出度和入度；

顶点 V 的出度=以 V 为起点有向边数；

顶点 V 的入度=以 V 为终点有向边数；

顶点 V 的度=V 的出度+V 的入度；

图的度=图中所有顶点度的和；

问题：一个图的顶点数为 n，边数为 e，则该图的度 =<u>　　2*e　　</u>。

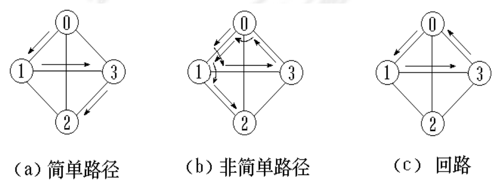
(9)路径与回路

路径：顶点 A 到顶点 B 的经过的所有的边；

简单路径：在一条路径中，除起点和终点外，若其余顶点各不相同；

回路：起点和终点相同的路径；

简单回路：由简单路径组成的回路称为简单回路；



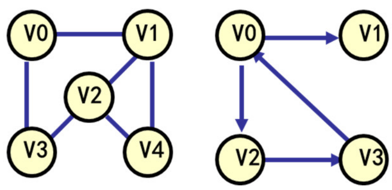
例题：在右图所示的的无向图中；

V0,V1,V2,V3 是不是简单路径?<u>　　　是　　　</u>

V0,V1,V2,V4,V1 是不是简单路径?<u>　　不是　　　</u>

写出无向图的一条简单回路<u>　　V0,V1,V2,V3,V0　</u>

写出有向图的一条简单回路<u>　　V0,V2,V3,V0　　</u>



(10)连通分量：在无向图中，如果从顶点 vi 到顶点 vj 有路径，则称 vi 和 vj 连通。如果图中任意两个顶点之间都连通，则称该图为连通图，否则，将其中的较大连通子图称为连通分量。

在有向图中，如果对于每一对顶点 vi 和 vj，从 vi 到 vj 和从 vj 到 vi 都有路径，则称该图为强连通图；否则，将其中的极大连通子图称为强连通分量。

5.2 图的存储结构

1、邻接矩阵（数组）表示法

邻接数组表示法是以一个 n*n 的数组来表示一个具有 n 个顶点的图形。我们以数组的索引（下标）值来表示顶点，以数组的内容之来表示顶点间边是否存在（1 表示存在， 0 表示不存在）。

例子：无向图的邻接矩阵

根据上述无向图得到邻接矩阵，可以得出如下结论：

结论 1：<u>　　　无向图的邻接矩阵是对称的　　　　　　　　</u>；

结论 2：<u>　顶点 i 的度＝第 i 行（列）中 1 的个数　</u>；

注意：完全图的邻接矩阵中，对角元素为 0，其余全 1。

例子：有向图的邻接矩阵

根据上述有向图得到邻接矩阵，可以得出如下结论：

结论 1：<u>　　有向图的邻接矩阵可能是不对称的　　　　</u>；

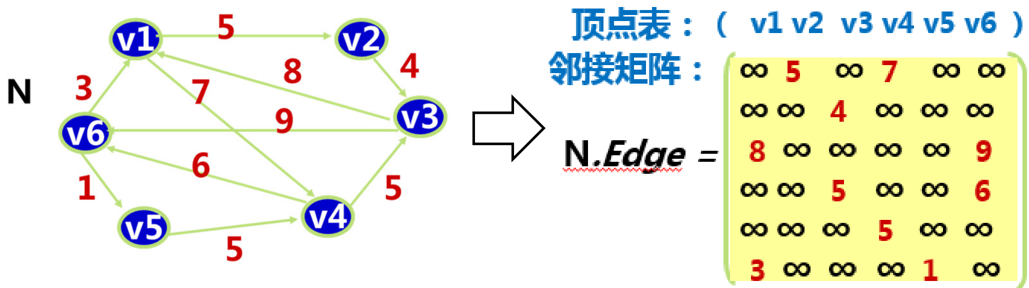
结论 2：<u>　顶点的出度 = 第 i 行元素之和　　　　　　</u>；

<u>　　顶点的入度 = 第 i 列元素之和　　　　　　</u>；

<u>　　图的度 = 矩阵中 1 的个数 * 2　　　　　　</u>；

结论 3：<u>　图的存储空间占用量只与它的顶点数有关，与边数无关；适用于边稠密的图；　　</u>；

例子：带权图的邻接矩阵



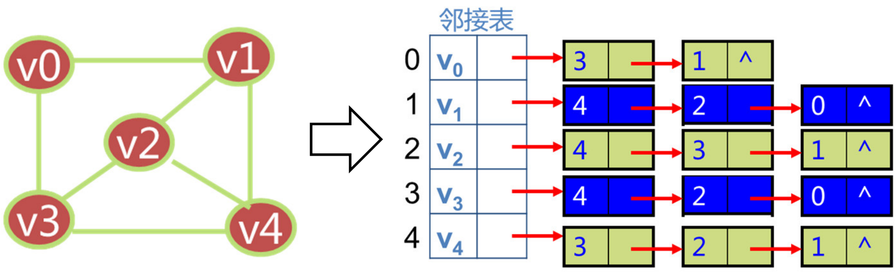
邻接矩阵法优点：容易实现图的操作，如：求某顶点的度、判断顶点之间是否有边（弧）、找顶点的邻接点等等。

邻接矩阵法缺点：n 个顶点需要 n*n 个单元存储边(弧)。 对稀疏图而言尤其浪费空间。

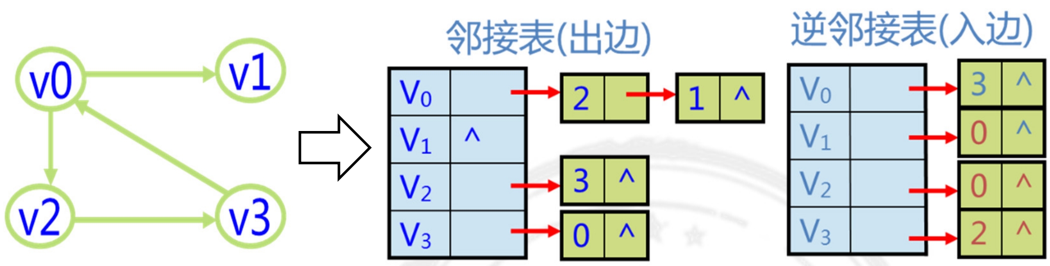
2、邻接（链式）表表示法

邻接链表法：以链表来记录个顶点的邻接顶点。

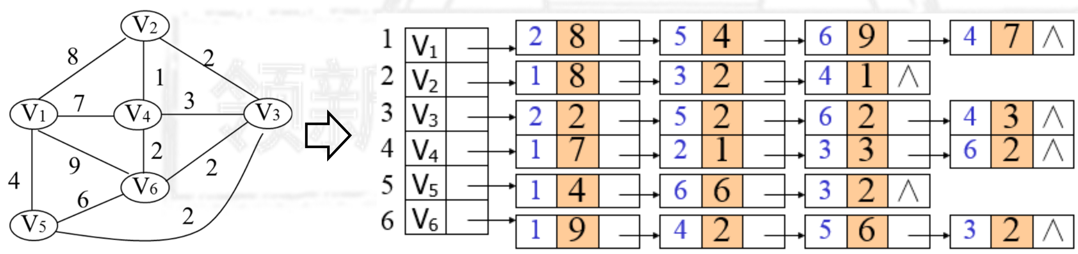
例 1：无向图的邻接表



例 2：有向图的邻接表



例 3：网（带权图）的邻接链表表示



5.3 图的遍历

从图的某个顶点出发，访问图中的所有顶点，且使每个顶点仅被访问一次。这一过程叫做图的遍历。遍历方法：深度优先遍历和广度优先遍历。

1、深度优先搜索（ DFS ）

基本思想：——仿树的先序遍历过程。

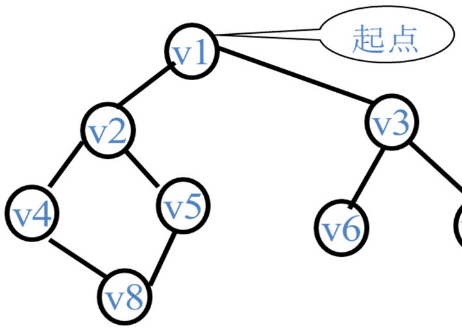
步骤：

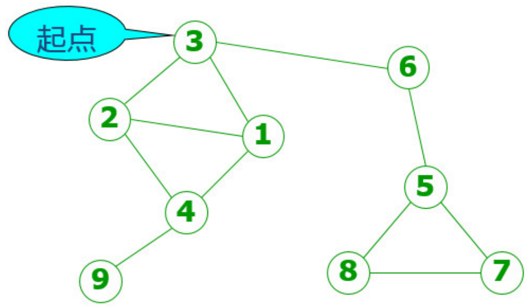
1)访问起始点 v;

2)若 v 的第 1 个邻接点没访问过，深度遍历此邻接点；

3)若当前邻接点已访问过，再找 v 的第 2 个邻接点重新遍历；

例子：





DFS 搜索结果：v1, v2, v4, v8, v5, v3, v6, v7　DFS 搜索结果：v3, v2, v4, v9, v1, v6, v5, v8, v7

2、广度优先搜索（BFS）

基本思想：——仿树的层次遍历过程。

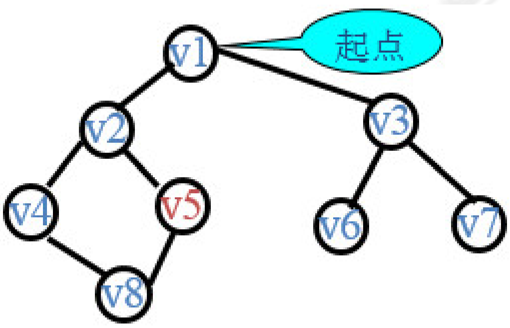
简单归纳：

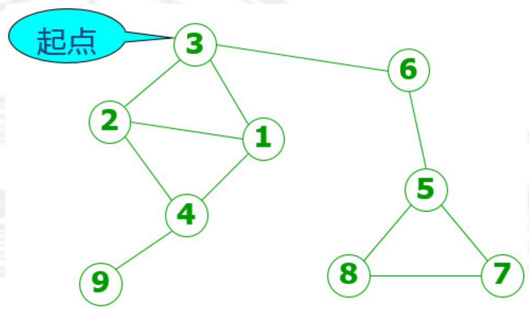
1）在访问了起始点 v 之后，依次访问 v 的邻接点；

2）然后再依次访问这些顶点中未被访问过的邻接点；

3）直到所有顶点都被访问过为止。

广度优先搜索是一种分层的搜索过程，每向前走一步可能访问一批顶点，不像深度优先搜索那样有回退的情况。因此，广度优先搜索不是一个递归的过程，其算法也不是递归的。





BFS 搜索结果：v1, v2, v3, v4, v5, v6, v7, v8

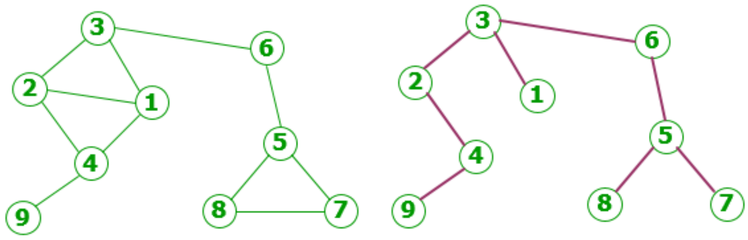
BFS 搜索结果：v3, v2, v1, v6, v4, v5, v9, v8, v7

5.4 图的最小生成树

1、生成树问题

在一个含有 n 个顶点的连通图中，必能从中选出 n-1 条 边构成一个极小连通子图，它含有图中全部 n 个顶点，但只有足以构成一棵树的 n-1 条边，称这棵树为连通图的生成树。例如，对下图(a)进行深度优先搜索遍历过程中经过的边和全部顶点构成的极小连通子图(b)即

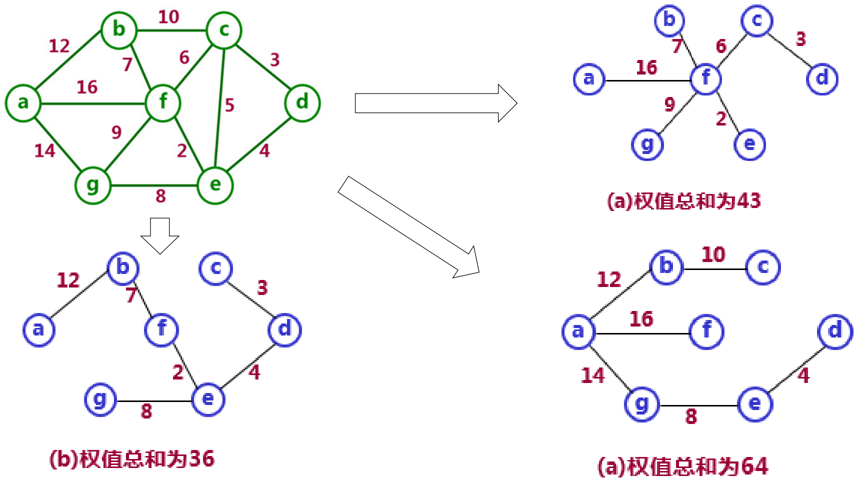
为(a)的一棵生成树。



(a) (b)

2、最小生成树

在含有 n 个顶点的连通网中选择 n-1 条边，构成一棵极小连通子图，并使该连通子图中 n-1 条边上权值之和达到最小，则称其为连通网的最小生成树。例如，对于如图所示的连通网可以有多棵权值总和不相同的生成树。



应用： n 个城市建网，如何选择 n－1 条线路，使总费用最少？

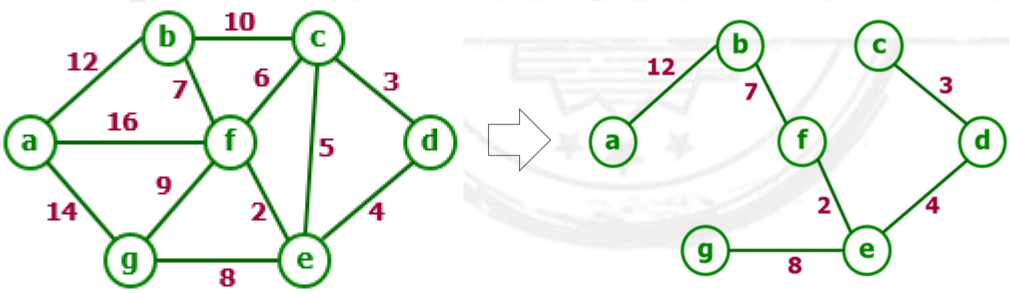
构造最小生成树的算法有许多，基本原则是：

- ◆ 尽可能选取权值最小的边，但不能构成回路；
- ◆ 选择 n-1 条边构成最小生成树。

（1）克鲁斯卡尔(Kruskal)算法

基本思想：按照权值从小到大的顺序选择 n-1 条边，并保证这 n-1 条边不构成回路。

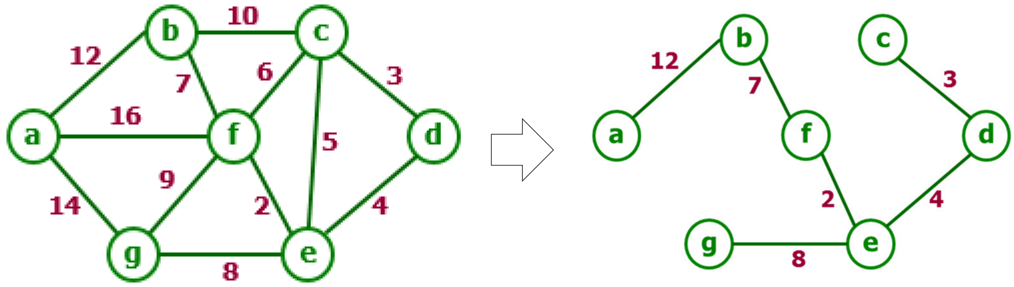
具体做法：首先构造一个只含 n 个顶点的森林，然后依权值从小到大从连通网中选择边加入到森林中，并使森林中不产生回路，直至森林变成一棵树为止。



（2）普里姆(Prim)算法

普里姆算法构造最小生成树的过程是从一个顶点 U={u0}作初态，不断寻找与 U 中顶点相邻

且代价最小的边的另一个顶点，扩充到 U 集合直至 U=V 为止。



比较以上两种算法可见：

A、Kurskal 算法主要对边进行操作，又称为“加边法”，其时间复杂度为 O(eloge)。这种方法比较适用于稀疏图。

B、Prim 算法主要对顶点进行操作，又称为“加点法”，其时间复杂度为 O(n^2)。比较适用于稠密图。

5.6 图的课堂练习

【NOIP2019 提高组】8. G 是一个非连通无向图（没有重边和自环），共有 28 条边，则该图至少有（ ）个顶点

A.10　　B.9　　C.11　　D.8

答案：B

解析：根据公式(8-1)*8/2 得到 28 条边，然后增加一个节点使其成为非连通图。

【NOIP2019 提高组】12.以下哪个结构可以用来存储图（　　）

A.栈　　　B.二叉树　　　C.队列　　　D.邻接矩阵

答案：D

解析：邻接矩阵和邻接表可以存储图，其他三项都是数据结构，不是存储结构。

【NOIP2017 普及组】10. 设 G 是有 n 个结点、m 条边（n ≤ m）的连通图，必须删去 G 的（　　）条边，才能使得 G 变成一棵树。

A. m – n + 1　　　B. m – n　　　C. m + n + 1　　　D. n – m + 1

答案：A

解析：m-(n-1)=m-n+1

【NOIP2015 普及组】15.设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（ ）个顶点。

A. 10　　B. 12　　C. 8　　D. 16

答案：D

解析：

抓住结点度数之和为边数的两倍来解题：

设顶点个数 x 个：

所以：2*x=16*2，x=16，所以答案选：D

5.6 图的练习题

1. 填空题

(1) 设无向图 G 中顶点数为 n，则图 G 至少有（　　）条边，至多有（　　）条边；若 G 为有向图，则至少有（　）条边，至多有（　）条边。

(2) 图的存储结构主要有两种，分别是（　　　）和（　　　）。

(3) 已知一个有向图的邻接矩阵表示，计算第 j 个顶点的入度的方法是（　　　）。

(4) 有向图 G 用邻接矩阵 A[n][n]存储，其第 i 行的所有元素之和等于顶点 i 的（　　）。

(5) n 个顶点的连通图用邻接矩阵表示时，该矩阵至少有（　　　）个非零元素。

(6) 表示一个有 100 个顶点，1000 条边的有向图的邻接矩阵有（　　　）个非零矩阵元素。

(7) 无向图中所有顶点的度数之和等于所有边数的（　　　）倍。

(8) 具有 n 个顶点的无向完全图中包含有（　　　）条边，具有 n 个顶点的有向完全图中包含有（　　　）条边。

(9) 一个具有 n 个顶点的无向图中，要连通所有顶点则至少需要（　　　）条边。

(10) 对于一个具有 n 个顶点和 e 条边的连通图，其生成树中的顶点数和边数分别为（　　　）和（　　　）。

(11) 对于一个图 G 的遍历，通常有两种方法，它们分别是（　　）和（　　）。

2. 选择题

(1) 在一个无向图中，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) 含 n 个顶点的连通图中的任意一条简单路径，其长度不可能超过（ ）。

A 1　　B n/2　　C n-1　　D n

(3) 对于一个具有 n 个顶点的无向图，若采用邻接矩阵存储，则该矩阵的大小是（　　）。

A n　　B $(n-1)^2$　　C n-1　　D n^2

(4) 图的生成树（　　　），n 个顶点的生成树有（　　　）条边。

A 唯一　　B 不唯一　　C 唯一性不能确定

D n　　E n +1　　F n-1

(5) G 是一个非连通无向图，共有 28 条边，则该图至少有（　　　）个顶点。

A 6　　　B 7　　　C 8　　　D 9

(6) 最小生成树指的是（　）。
A 由连通网所得到的边数最少的生成树
B 由连通网所得到的顶点数相对较少的生成树
C 连通网中所有生成树中权值之和为最小的生成树
D 连通网的极小连通子图

(7) 某无向图的邻接矩阵 A=$\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}$，可以看出，该图共有（ ）个顶点。
A 3　　B 6　　C 9　　D 以上答案均不正确

(8) 在一个具有 n 个顶点的有向完全图中包含有（　）条边
A n(n-1)/2　　B n(n-1)　　C n(n+1)/2　　D n^2

(9) 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

(10)　具有 4 个顶点的无向完全图有条边。（　）
A) 6　　B) 12　　C) 16　　D) 20

(11)　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

(12)　n 个结点的完全有向图含有边的数目（　）
A）n*n　　B）n *（n+1）　　C）n/2　　D）n*（n-1）

(13)　有向图中一个顶点的度是该顶点的（　）
A）入度　　B） 出度　　C） 入度与出度之和　　D）（入度+出度）/2

(14)　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（　）
A） e　　B）2e　　C） n^2-e　　D）n^2-2e

参考答案：

1.填空题

(1)【解答】0，n(n-1)/2，0，n(n-1)

【解析】图的顶点集合是有穷非空的，而边集可以是空集；边数达到最多的图称为完全图，在完全图中，任意两个顶点之间都存在边。

(2)【解答】邻接矩阵，邻接表

(3)【解答】求第 j 列的所有元素之和

(4)【解答】出度

(5)【解答】2(n-1)

【解析】所谓连通图一定是无向图,有向的叫做强连通图连通 n 个顶点,至少只需要 n-1 条边就可以了,或者说就是生成树由于无向图的每条边同时关联两个顶点,因此邻接矩阵中每条边被存储了两次（也就是说是对称矩阵）,因此至少有 2(n-1)个非零元素。

(6)【解答】1000

(7)2

(8)n(n-1)/2　　n(n-1)

(9)n-1

(10)n　　n-1

(11)深度优先法　　广度优先法

2.选择题

(1)C

(2)C

【分析】若超过 n-1，则路径中必存在重复的顶点。

(3)D　(4)C，F

(5)D

n 个顶点的无向图中，边数 e≤n(n-1)/2，将 e=28 代入，有 n≥8，现已知无向图非连通，则 n=9。

(6)C　(7)A　(8)B　(9)C　(10)A

(11)A　(12)D　(13)C　(14)D