**§5.4 图的遍历**

**习题5.4**

1. 给出图5.2、图5.7的邻接矩阵、连通矩阵、关联矩阵。

解：（1）图5.2的

邻接矩阵A=

连通矩阵B=

关联矩阵C=

（2）图5.7的

邻接矩阵A=

连通矩阵B=

关联矩阵C=

1. 根据图5.7中图的邻接矩阵，求出：（1）顶点3和顶点5之间长度小于或等于3的基本通路的条数；（2）通过顶点3且长度等于4的基本回路的条数。

解：A2= A3=

A4=

（1）顶点3和顶点5之间长度小于或等于3的基本通路有0+3+6=9条；

（2）通过顶点3且长度等于4的基本回路有83条。

1. 给出图5.9中两个图的邻接矩阵，并根据邻接矩阵求（1）图（a）中顶点2与顶点12之间的距离；（2）图（b）中顶点1和顶点5之间的距离。

解：图（*a*）的邻接矩阵为：

A=

由于第2行第12列的值是0，可以一直计算A2,A3,A4,A5，直到对应的这个值不是0，得到A5中对应的值不是0，所以矩离为5。

图（*b*）的邻接矩阵为

B=。

由于第1行第5列的值是0，可以一直计算B2,B3,B4，直到对应的这个值不是0，得到B4中对应的值不是0，所以矩离为4。

4．用广度优先搜索算法求出下面图5.22中每个连通图的一棵生成树（从A开始，在选择顶点时，使用字母顺序）。







图5.22 习题4的图

解：1图的生成树是



2图的生成树是



3图的生成树是

4图的生成树是

5图的生成树是

6图的生成树是

5．用深度优先搜索算法求出下面图5.23中每个连通图的一棵生成树（从A开始，在选择顶点时，使用字母顺序）。





图5.23 习题5的图

解：

1图的生成树为

2图的生成树是

3图的生成树是

4图的生成树是

5图的生成树是

6图的生成树是

6．写出Prim\_Alternate算法的算法步骤，并证明Prim\_Alternate算法是正确的，即算法执行的结果会产生一个最小生成树。

解：

1． **procedure** *Prim* (*G，s，ET* )

2． ￡//将起始顶点加入到集合￡中

3． 〒 //初始边集合为空

//第4～17行在边集〒放入条边

4． **for**  to  do

//第5～16行增加一条最小权值的边，它的一个顶点在￡中，另一个不在￡中

5． 

6． **for** ￡中最近的顶点 **do**

7． **for** 不在￡中的每个点 **do**

8． **if**  **then**

9． 

10． 

11． 

12． **end** *if*

13． **end** *for*

14． **end** *for*

15． ￡=￡//将选中的顶点放入￡中

16． 〒=〒//将选中的边放入〒中

17． **end** *for*

18． **return**（〒）

19．**end** *Prim*

7．**Kruskal**算法用来求解有个顶点的连通加权图的最小生成树。它假设图开始只包含的顶点，不包含边，每次循环都增加权值最小的边到中，且不产生回路，当有条边时，停止。请写出Kruskal算法的算法步骤，并证明Kruskal算法是正确的，即算法执行的结果会产生最小生成树。

Kruskal算法基本描述：

1. 先构造一个只含 n 个顶点，而边集为空的子图，若将该子图中各个顶点看成是各

棵树上的根结点，则它是一个含有n棵树的一个森林。

1. 从带权图的边集 E 中选取一条权值最小的边，若该条边的两个顶点分属不同的树，

则将其加入子图，也就是说，将这两个顶点分别所在的两棵树合成一棵树；反之，若该条边的两个顶点已落在同一棵树上，则不可取，而应该取下一条权值最小的边再试之。

（3）依次类推，直至森林中只有一棵树，也即子图中含有 n-1条边为止。

8．用Prim算法、Prim\_Alternate算法和Kruskal算法给图5.24的每个图找出最小生成树。

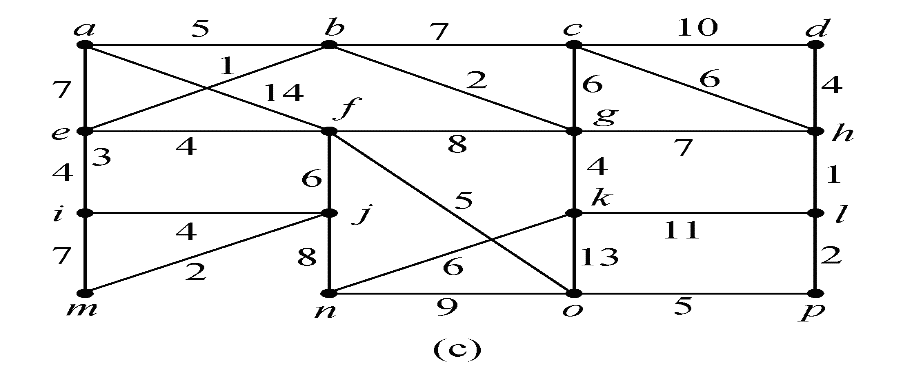


图5.24 习题8的图

解：用Prim算法得到的最小生成树为：



用Prim\_Alternate算法得到的最小生成树为：



用Kruskal算法得到的最小生成树为上面的二种都可以得到。

9．判断下面说法是否正确，如果是对的，加以证明，否则给出反例。其中是连通加权图。

（1）如果中所有边的权值都不一样，则不同的生成树的权值都不一样。

（2）如果是的一条边，权值最低，则被中任意一个最小生成树所包含。

（3）如果一直删除中权值最大的边且不导致非连通，则最后得到的图为的最小生成树。

解：

（1）错误。

例如二个生成树中只有二条枝不同，其他的枝都相同。其中一个生成树中的一条枝的权值是3，另一条枝的权值是7。而另一个生成树中一条枝的权值是4，另一条枝的权值是6。但是二个生成树的权值相同。

（2）正确。

用上述的最小生成树的算法都会包含这条权值最小的边。若是不包含就会得到的生成树不是权值最小的。

（3）错误。

应改成：如果一直删除中权值最大的边且不导致非连通，并且没有基本回路了，则最后得到的图为的最小生成树。