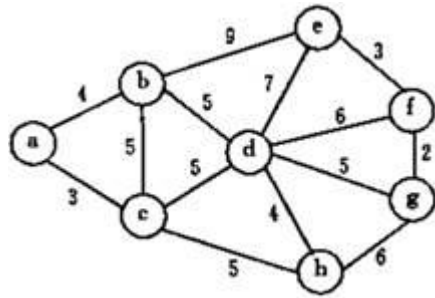


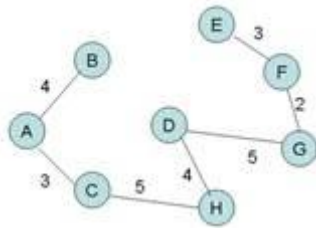
7.3 无向带权图如图

- (1) 画出它的邻接矩阵，并按 Prim 算法求其最小生成树
- (2) 画出它的邻接表，并按 Kruskal 算法求其最小生成树



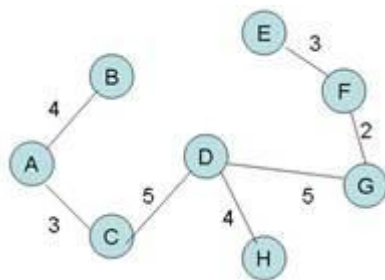
答：(1)

	a	b	c	d	e	f	g	h
a	∞	4	3	∞	∞	∞	∞	∞
b	4	∞	5	5	9	∞	∞	∞
c	3	5	∞	5	∞	∞	∞	5
d	∞	5	5	∞	7	6	5	4
e	∞	9	∞	7	3	∞	∞	∞
f	∞	∞	∞	6	3	∞	2	∞
g	∞	∞	∞	5	∞	2	∞	6
h	∞	∞	5	4	∞	∞	6	∞

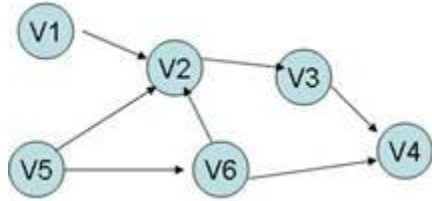


(2) 邻接表

0	A	→	1 4	→	2 3	Λ								
1	B	→	0 4	→	2 5	→	3 5	→	4 9	Λ				
2	C	→	0 3	→	1 5	→	3 5	→	7 5	Λ				
3	D	→	1 5	→	2 5	→	4 7	→	5 6	→	6 5	→	7 4	Λ
4	E	→	1 9	→	3 7	→	5 3	Λ						
5	F	→	3 6	→	4 3	→	6 2	Λ						
6	G	→	3 5	→	5 2	→	7 6	Λ						
7	H	→	2 5	→	3 4	→	6 6	Λ						



7.4 有向图如图所示，试写出所有可能的拓扑序列



答: 156234; 561234; 516234

7.5 试利用 Dijkstra 算法求图中顶点 A 到其他各顶点之间的最短路径。要求写出执行算法过程中，数组 D、P 和 S 各步的状态。

	1	2	3	4	5	6	D	P
B	15	15	15	15	15	15	15	AB
C	2						2	AC
D	12	12	11	11			11	ACFD
E		10	10				10	ACE
F		6					6	ACF
G			16	16	14		14	ACFDG
S	AC	ACF	ACFE	ACFED	ACFEDG	ACFEDGB		

7.9 假设有向图以邻接表作为存储结构。试基于图的深度优先搜索策略写一算法，判断有向图中是否存在由顶点 V_i 至顶点 V_j ($i \neq j$) 的搜索路径

```

int visited[MAXSIZE]; //指示顶点是否在当前路径上
int exist_path_DFS(ALGraph G, int i, int j) //深度优先判断有向图 G 中顶点 i 到顶点 j 是否有路径, 是则返回 1, 否则返回 0
{
    if(i==j) return 1; //i 就是 j
    else
    {
        visited[i]=1;
        for(p=G.vertices[i].firstarc;p;p=p->nextarc)
        {
            k=p->adjvex;
            if(!visited[k]&&exist_path_DFS(G, k, j)) return 1; //i 下游的
        }
    }
}

```

9.1 若对大小均为 n 的有序顺序表和无序顺序表分别进行顺序查找，试在下列三种情况下分别讨论两者在等概率时平均查找长度是否相同？

- (1) 查找不成功，即表中没有关键字等于给的值 K 的记录；
- (2) 查找成功，且表中只有一个关键字等于给定值 K 的记录；
- (3) 查找成功，且表中有若干关键字等于给定值 K 的记录，要求找出所有这些记录。

答：(1)相同，有序 $\frac{n+1}{2}$ ；无序 $\frac{n+1}{2}$ **(此题答案有争议)**

(2)相同，有序 $\frac{n+1}{2}$ ；无序 $\frac{n+1}{2}$

(3)不相同，对于有序表，找到了第一个与 K 相同的元素后，只要再找到与 K 不同的元素，即可停止查找；对于无序表，则要一直查找到最后最后一个元素。

9.3 画出对长度为 13 的有序表进行折半查找的判定树，并分别求其等概率时查找成功和查找不成功的 ASL。

答：查找成功：
$$ASL = \frac{1}{13} (1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 6) = \frac{41}{13}$$

查找失败：
$$ASL = \frac{2}{14} * 3 + \frac{6}{14} * 4 = \frac{15}{7}$$

(P220:查找不成功的过程就是走了一条从根节点到外部节点的路径，和给定值进行比较的关键字数等于该路径上内部结点数)

注：在折半查找判定树中，查找不成功时的比较次数即是查找相应外结点时与内结点的比较次数。整个判定树代表的有序表在查找失败时的平均查找长度即为查找每个外结点的比较次数之和除以外结点的个数。例如，长度为 10 的有序表在查找失败时的平均查找长度为：

$$ASL = (3 \times 5 + 4 \times 6) / 11 = 39 / 11$$