

一、选择题

DDDDA BA

二、填空题

1. 0; $n(n-1)/2$; $n-1$

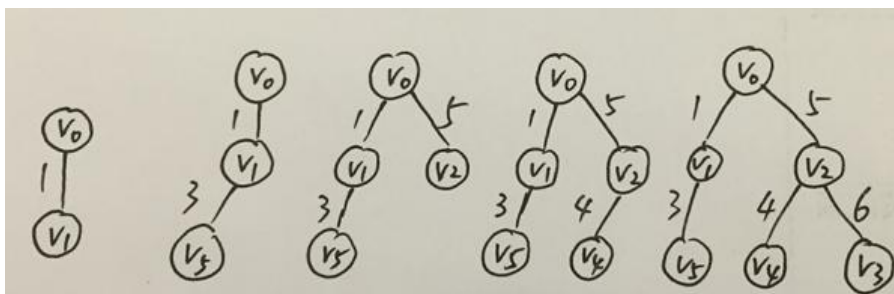
2. 先序遍历; 层序遍历

3. AEDCB; BCADE

4. n ; $n(n-1)$

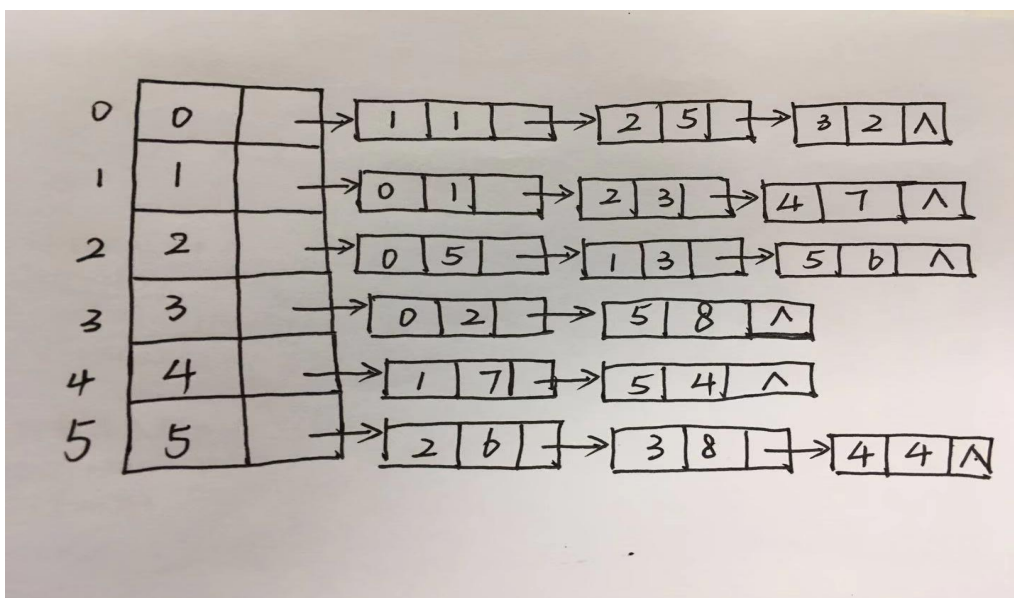
三、应用题

1.



2.

对应的邻接表如图所示，深度优先遍历该图的结果为 012534。

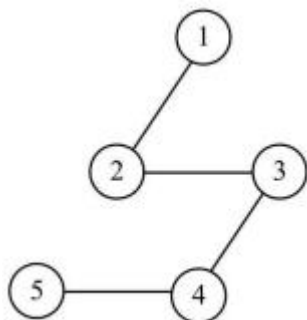


3.

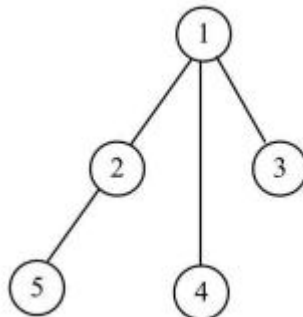
过程 终点		=> 从顶点a到各终点的D值和最短路径父顶点P的变化过程 =>											
		初始 (第0步)		选c (第1步)		选b (第2步)		选d (第3步)		选e (第4步)		选d (第5步)	
		D	P	D	P	D	P	D	P	D	P	D	P
b		6	a	5	c	5	c	5	c	5	c	5	c
c		3	a	3	a	3	a	3	a	3	a	3	a
d		∞		6	c	6	c	6	c	6	c	6	c
e		∞		7	c	7	c	7	c	7	c	7	c
f		∞		∞		∞		9	d	9	d	9	d

4.

(1)



(2)



四、算法设计题

```

1. int countout(frontnodetype adjlist[],int n) //n 为有向图中的顶点个数
{
    int i,j,num=0; //num 为出度为 2 的顶点个数
    nodetype *p;
    for(i=0;i<n;i++)
    {
        j=0;
        p=adjlist[i].next;
        while(p != NULL)
        {
            j++;
            p=p->next;
        }
        if(j == 2)
            num++;
    }
}
  
```

```

        return num;
    }

```

2. int SumOfWeight(int G[][],int Nv) //二维数组 G 为邻接矩阵, Nv 为顶点数

```

{   int sum,i,j;
    sum=0;
    for(i = 0;i < Nv;i++)
        for(j = 0;j < Nv ;j++)
            if(G[i][j] != 0 && G[i][j] <  INF) //INF 为符号常量, 表示无穷大
                sum = sum+G[i][j];
    return  sum;
}

```

3. void DegreeOut(int G[][],int Nv) //统计各顶点的出度

```

{   int i,j;
    int degree[maxvertexnum] = {0};
    for(i = 0;i < Nv; i++)
        for(j = 0;j < Nv; j++)
            if(G[i][j] >0 && G[i][j] < INFINITY) //INFINITY 表示无穷大
                degree[i]++;
}

```

void DegreeIn(int G[][],int Nv) //统计各顶点的入度

```

{   int i,j;
    int degree[maxvertexnum] = {0};
    for(i = 0;i < Nv; i++)
        for(j = 0;j < Nv; j++)
            if(G[j][i] >0 && G[j][i] < INFINITY) //INFINITY 表示无穷大
                degree[i]++;
}

```