1、以邻接表作存储结构，编写程序对给定的无向图G（包含n个顶点，编号为0至n-1）进行广度优先遍历，并在遍历的过程中计算图G的连通分量个数及边的数目。

本题限定在遍历过程中，如果同时出现多个待访问的顶点，则优先选择编号最小的一个进行访问，以顶点0为遍历起点。

邻接表的类型描述

#define MaxVexNum 30 //最大顶点数

typedef struct ArcNode //表结点定义

{

int adjvex;

struct ArcNode \*nextarc;

}ArcNode;

typedef struct //头结点定义

{

ArcNode \*firstarc;

}VerNode;

typedef struct

{

VerNode vertices[MaxVexNum];

int vernum, arcnum;

}ALGraph;

输入格式:

第一行输入图的顶点数和边数e。

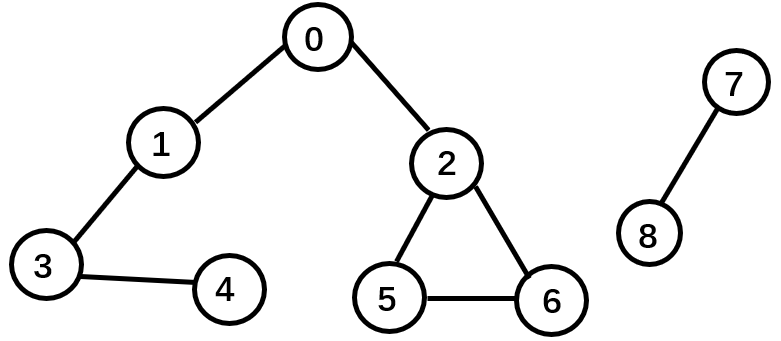
接下来共e行。每行代表一条边，输入边依附的两个顶点的编号。用头插法建邻接表，各边按第一个顶点编号升序输入，第一个顶点相同时按第二个顶点降序输入。注意：边不能重复输入。

输出格式:

输出分三行

* 第一行 广度优先遍历序列。序列中每个顶点编号后跟一个空格。
* 第二行 连通分量个数
* 第三行 边数

对于下面给出的无向图G



输入样例:

9 8

0 2

0 1

1 3

2 6

2 5

3 4

5 6

7 8

输出样例:

0 1 2 3 5 6 4 7 8

2

8

2、【练习题】假设上面这些点都是通信主机，最初，任意两台机器的通信额度均为0。为了能发送数据，主机需要向你申请额度。每个申请形如u v x的格式，表示机器u和v的可用额度变为x MB。定义每台机器的“通信主要对象”为当前时刻与该机器的可用额度最大的机器（如果有并列，则取其中编号最小的机器）。

**[输入格式]**

第一行：2个正整数m, n，表示主机总数、申请通信额度的机器数。

接下来n行，每行一个非负整数k，表示额度申请请求，格式如题面中所述。

**[输出格式]**

输出m行，每行一个非负整数表示该主机的“通信主要对象”。

输入样例：

9 7

0 1 3

0 4 2

1 3 5

3 4 6

5 6 2

2 5 7

7 8 9

输出样例：

1

3

5

4

3

2

5

8

7

**0**

**3**

6

**4**

9

**8**

**7**

2

7

**6**

**2**

**5**

5

3

**1**

3、【练习题】以邻接表作存储结构，编写程序对给定的无向图G（包含n个顶点，编号为0至n-1）进行**深度优先遍历**，找出一条包含所有顶点的简单路径。

邻接表的类型描述

#define MaxVexNum 30 //最大顶点数

typedef struct ArcNode //表结点定义

{

int adjvex;

struct ArcNode \*nextarc;

}ArcNode;

typedef struct //头结点定义

{

ArcNode \*firstarc;

}VerNode;

typedef struct

{

VerNode vertices[MaxVexNum];

int vernum, arcnum;

}ALGraph;

输入格式:

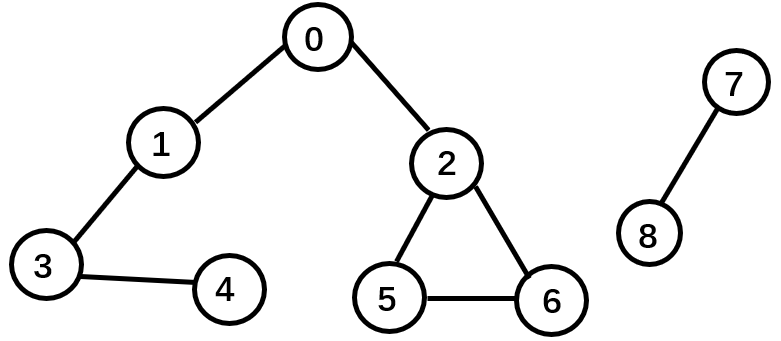
第一行输入图的顶点数和边数e。

接下来共e行。每行代表一条边，输入边依附的两个顶点的编号。用头插法建邻接表，各边按第一个顶点编号升序输入，第一个顶点相同时按第二个顶点降序输入。注意：边不能重复输入。

输出格式:·

一行：深度优先遍历序列。序列中每个顶点编号后跟一个空格。

对于下面给出的无向图G



输入样例:

9 8

0 2

0 1

1 3

2 6

2 5

3 4

5 6

7 8

输出样例:

0 1 3 4 2 5 6 7 8