```cpp
//日期: 2018/ 时间:
#include <stdio.h>
#include <stdlib.h>
#include <vector>
using namespace std;

const int maxv = 1000;  //最大顶点数
const int INF = 1000000000; //INF为很大的数

#define p2

//**************************邻接矩阵****************************************
#ifdef p1
int n,G[maxv][maxv];        //n为顶点数
bool vis[maxv] = {false};   //如果顶点i已被访问，则vis[i]==true，初值为false

void DFS(int u,int depth){  //u为当前访问的顶点编号，depth为深度
    vis[u] = true;          //设置u已被访问
    //如果需要对u进行一些操作，可以在这里进行
    //下面对所有从u出发能到达的分支顶点进行枚举
    for(int v=0;v<n;v++){
        if(vis[v] == false && G[u][v] != INF){  //u出发能到达的节点v
            DFS(v,depth+1);
        }
    }
}

void DFSTrave(){                //遍历图
    for(int u=0;u<n;u++){       //对每个顶点
        if(vis[u] == false){    //如果u没有被访问
            DFS(u,1);
        }
    }
}
#endif

//**************************邻接表*****************************************
#ifdef p2
vector<int> adj[maxv];      //图G的邻接表
int n;
bool vis[maxv] = {false};

void DFS(int u,int depth){
    vis[u] = true;      //设置u以及被访问
    //在此处进行对u的一些其他操作

    for(int i=0;i<adj[u].size();i++){   //对从u出发可以到达的所有顶点v
        int v = adj[u][i];
        if(vis[v] == false){
            DFS(v,depth+1);
        }
    }
}

void DFSTrave(){        //遍历图
    for(int u=0;u<n;u++){
```

```
57          if(vis[u] == false){
58              DFS(u,1);
59          }
60      }
61  }
62
63  #endif
64
65  int main(){
66
67
68      return 0;
69  }
70
71
```