**数据结构第四次上机报告：图的深度优先遍历**

学号：19030100141 姓名：杨祥瑞

**题目原文**

问题描述

已知无向图的邻接矩阵，以该矩阵为基础，给出深度优先搜索遍历序列，并且给出该无向图的连通分量的个数。在遍历时，当有多个点可选时，优先选择编号小的顶点。（即从顶点1开始进行遍历）

**标题**

图的深度优先遍历

**时间限制**

1 S

**内存限制**

10000

**问题描述**

已知无向图的邻接矩阵，以该矩阵为基础，给出深度优先搜索遍历序列，并且给出该无向图的连通分量的个数。在遍历时，当有多个点可选时，优先选择编号小的顶点。（即从顶点1开始进行遍历）

**问题输入**

第一行是1个正整数，为顶点个数n（n<100），顶点编号依次为1，2，…，n。后面是邻接矩阵，n行n列。

**问题输出**

共2行。第一行输出为无向图的深度优先搜索遍历序列，输出为顶点编号，顶点编号之间用空格隔开；第二行为无向图的连通分量的个数。

**输入样例**

6

0 1 0 0 0 0

1 0 0 0 1 0

0 0 0 1 0 0

0 0 1 0 0 0

0 1 0 0 0 1

0 0 0 0 1 0

### 输出样例

1 2 5 6 3 4

2

**算法实现**

**源代码:**

#include<iostream>  
#include<cstring>  
  
using namespace std;  
  
int graph[1005][1005];  
bool visited[1005];  
int vexnum;  
  
int firstneighbor(int begin){  
    for(int i=1;i<=vexnum;i++){  
        if(graph[begin][i]==1){  
            return i;  
        }  
    }  
    return -1;  
}  
  
int nextneighbor(int begin,int nownode){  
    for(int i=begin+1;i<=vexnum;i++){  
        if(graph[nownode][i]==1){  
            return i;  
        }  
    }  
    return -1;  
}  
  
void Dfs(int v){  
    cout<<v<<" ";  
    visited[v]=true;  
    for(int i=firstneighbor(v);i>0;i=nextneighbor(i,v)){  
       if(!visited[i]){  
           Dfs(i);  
       }  
    }  
}  
  
int DfsTraverse(){  
    for(int i=1;i<=vexnum;i++){  
       visited[i]=false;  
    }  
    int count=0;  
    for(int i=1;i<=vexnum;i++){  
        if(!visited[i]){  
           Dfs(i);  
           count++;  
        }  
    }  
    return count;  
}  
  
int main(){  
    cin>>vexnum;  
    for(int i=1;i<=vexnum;i++){  
        for(int j=1;j<=vexnum;j++){  
            cin>>graph[i][j];  
        }  
    }  
    int ans;  
    ans=DfsTraverse();  
    cout<<endl;  
    cout<<ans<<endl;  
    return 0;  
}

**提交结果**

貌似提交了两次,前两次居然是编译错误QAQ.经过修改发现是忘记带;了



**总结**

这道题让我更加了解了C语言数组的使用,最开始觉得题干有些难懂,后来逐渐慢慢在纸上画画过程才读懂全部过程.

所以说以后做题一定要更加的细心,不能让自己浮躁的心来做题.这样反而事倍功半.

通过做这道题,也了解了一些C语言的stl操作,感觉自己写题越来越顺手了!