**数据结构**

**——实验报告书**



姓名：樊雨晨

学号：201632110111

班级：软件工程(中外)16 班

学院：数理与信息工程学院

日期：2017年05月10日

# **实验 图的建立与遍历**

## **问题**

按邻接矩阵的方法创建图，分别用深度优先和广度优先方法遍历图。

## **实验思路**

建立邻接矩阵，主对角线为0，数组中的1代表连接，0代表没有连接。

深度遍历：当遍历到一个结点后继续向与这个结点有联系的下一个结点进行遍历。与当前顶点有联通的路径全部遍历，递归算法。

广度遍历：当遍历到一个结点后，与当前结点有链接的结点依次访问，当队列中的全部被访问后，从这些被访问过的点出发，访问他们的邻接点。递归。

## **实验步骤**

先建立邻接矩阵，然后建立一个bool数组，判断是否进行过遍历。

之后分别是深度和广度的遍历。

## **代码**

#**include**<iostream>

#**include**<vector>

#**include**<queue>

#**include**<cstdio>

#**define** N 1000

**using** **namespace** std;

**int** n,m;

**int** g[N][N];

**bool** visit[N];

**void** **dfs**(**int** v) {

visit[v] = true;

cout<<v<<" ";

**for**(**int** w=0;w<n;w++)

**if** (g[v][w] && !visit[w]) dfs(w);

}

**void** **bfs**(**int** v)

{

**int** w;

queue<**int**>Q;

**for** ( w=0; w<n; w++ )

**if** ( !visit[w])

{

visit[w] = true;

cout<<w<<" ";

Q.push(w);

**while** (!Q.empty())

{

**int** u=Q.front();

Q.pop();

**for**(w=0; w<n; w++)

**if** ( g[u][w] && ! visit[w])

{

visit[w]=true;

cout<<w<<" ";

Q.push(w);

}

}

}

}

**int** **main**()

{

**while**(~scanf("%d%d",&n,&m))

{

**for**(**int** i=0;i<m;i++)

{

**int** a,b;

scanf("%d%d",&a,&b);

g[a][b]=g[b][a]=1;

}

//DFS

**for**(**int** i=0;i<n;i++)

{

visit[i]=false;

}

**for**(**int** i=0;i<n;i++)

{

**if**(!visit[i])

dfs(i);

}

printf("\n");

//BFS

**for**(**int** i=0;i<n;i++)

{

visit[i]=false;

}

**for**(**int** i=0;i<n;i++)

{

**if**(!visit[i])

bfs(i);

}

printf("\n");

printf("\n");

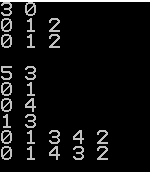
}

}

## **测试**

正常

## **结果**



## **总结**

在广度和深度遍历中，都要注意有一个总控制的循环，在从0到n都需要遍历，并判断是否进行过遍历，第二点是在内层循环中，需要递归调用。