图

姓名：付炎平 学号： 2019217819

**1 实验目的：**

掌握图的存储结构，能够进行图的遍历，理解最小生成树算法。

**2 实验学时：4**

**3 实验内容：**

1. 编写图类，实现基于邻接矩阵的存储；
2. 实现图的DFS、BFS算法
3. 实现图的最小生成树算法（选做）

**源码：**

#include <iostream>

#include <queue>

using namespace std;

class Graph{

public:

int visit[8]={0};

int size;

int data[8];

int graph[8][8]={

{0,1,2,0,0,0,0,8},

{1,0,0,3,6,0,0,0},

{2,0,0,0,0,9,0,0},

{0,3,0,0,4,0,0,0},

{0,6,0,4,0,5,0,0},

{0,0,9,0,5,0,7,0},

{0,0,0,0,0,7,0,12},

{1,0,0,0,0,0,12,0},

};

Graph(){

for(int i=0;i<8;i++){

data[i]=i+1;

}

size=8;

}

void dfs(int index){

index--;

visit[index]=1;

cout<<data[index]<<" ";

for(int j=0;j<size;j++){

if(graph[index][j]!=0&&visit[j]==0&&j!=index){

dfs(j+1);

}

}

}

void bfs(int index){

queue<int> q;

q.push(index);

visit[index-1]=1;

while(!q.empty()){

int idx=q.front();

q.pop();

idx--;

cout<<data[idx]<<" ";

for(int i=0;i<size;i++){

if(graph[idx][i]!=0&&visit[i]==0&&i!=idx){

visit[i]=1;

q.push(i+1);

}

}

}

}

};

int main(){

Graph g1,g2;

cout<<"深度遍历："<<endl;

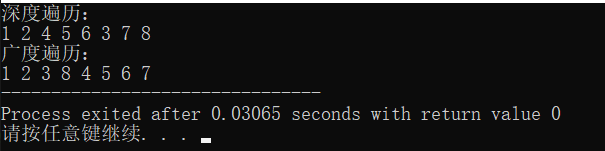
g1.dfs(1);

cout<<endl<<"广度遍历："<<endl;

g2.bfs(1);

}

**截图：**



**4 心得体会：**