

《数据结构》

实

验

报

告

|  |  |
| --- | --- |
| 项目名称： | 图的实验--图的主要遍历算法 |
| 班 级： |  |
| 学生姓名**:** |  |
| 学生学号： |  |
| 指导教师： |  |
| 成 绩： |  |

20 年 月 日

成都信息工程大学计算机学院

1. 实现功能描述

1.分别采用邻接表实现图的深度优先与广度优先遍历算法。

2.采用邻接矩阵实现图的广度优先遍历和深度优先遍历算法。

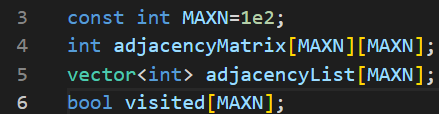
2. 方案比较与选择

1. 从数据结构的逻辑结构与存储结构角度提供多种解决方案；
2. 从时空效率角度分析决定最终采用方案的原因。

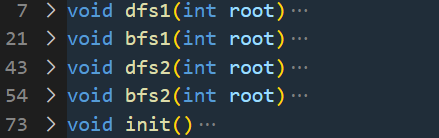
顺序结构实现邻接矩阵，链式结构实现邻接表

3. 设计算法描述

1. 用简单示例结合所设计算法采用的数据逻辑结构图、存储结构图说明算法思想。



1. 进行模块划分，给出主要功能组成框图。形式如下：



# 4. 算法实现（即完整源程序，带注解）

#include<bits/stdc++.h>

using namespace std;

const int MAXN=1e2;

int adjacencyMatrix[MAXN][MAXN];

vector<int> adjacencyList[MAXN];

bool visited[MAXN];

void dfs1(int root)

{

cout<<root<<" ";

for(int i=0;i<MAXN;++i)

{

if(adjacencyMatrix[root][i])

{

if(visited[i])

continue;

visited[i]=1;

dfs1(i);

}

}

}

void bfs1(int root)

{

int p;

queue<int> q;

q.push(root);

while(!q.empty())

{

p=q.front();

q.pop();

cout<<p<<" ";

for(int i=0;i<MAXN;++i)

{

if(adjacencyMatrix[p][i])

{

if(visited[i])

continue;

visited[i]=1;

q.push(i);

}

}

}

}

void dfs2(int root)

{

cout<<root<<" ";

for(int i=0;i<adjacencyList[root].size();++i)

{

if(visited[adjacencyList[root][i]])

continue;

visited[adjacencyList[root][i]]=1;

dfs2(adjacencyList[root][i]);

}

}

void bfs2(int root)

{

int p;

queue<int> q;

q.push(root);

while(!q.empty())

{

p=q.front();

q.pop();

cout<<p<<" ";

for(int i=0;i<adjacencyList[p].size();++i)

{

if(visited[adjacencyList[p][i]])

continue;

visited[adjacencyList[p][i]]=1;

q.push(adjacencyList[p][i]);

}

}

}

void init()

{

memset(visited,0,sizeof(visited));

}

int main()

{

std::ios::sync\_with\_stdio(0);

std::cin.tie(0);

std::cout.tie(0);

adjacencyMatrix[1][2]=adjacencyMatrix[1][3]=1;

adjacencyMatrix[2][4]=adjacencyMatrix[2][5]=1;

adjacencyList[1].push\_back(2);

adjacencyList[1].push\_back(3);

adjacencyList[2].push\_back(4);

adjacencyList[2].push\_back(5);

init();

dfs1(1);

cout<<endl;

init();

bfs1(1);

cout<<endl;

init();

dfs2(1);

cout<<endl;

init();

bfs2(1);

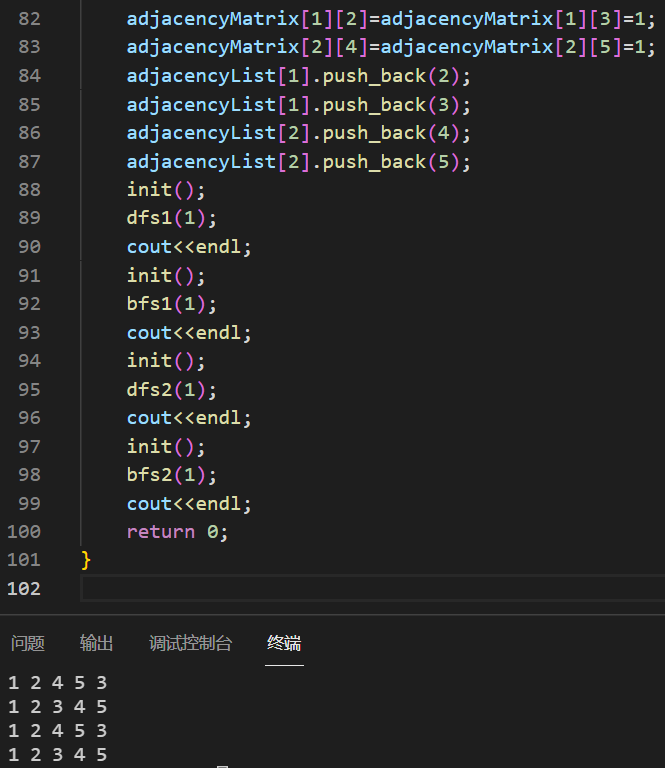
cout<<endl;

return 0;

}

5. 实验结果测试与分析

1. 用各种可能数据测试程序，取截图；
2. 对结果进行分析，说明算法的有效性。



# 6. 思考及学习心得（得分与文字数量和描述真实性相关）

（1）描述实验过程中对此部分知识的认识；

早就会的，花了点时间写完，更熟练吧

（2）特别描述在学习方法上的收获及体会；

…………

（3）针对前面的思考题内容选择部分在此回答。

1.对于顺序存储结构和链式存储的遍历算法，在时空效率上与进行分析对比，并得出结论。

设边数E，点数V，

E接近V^2,邻接矩阵时空更优，没有指针要存

E接近V，邻接链表时空更优，没有空间冗余