**VARUN KUMAR**

**2K19 / IT / 140**

Problem 23: Write a program to implement Graph Data Structure and Its traversal BFS and DFS

#include <iostream>

using namespace std;

struct node

{

int data;

node\* next;

};

node\* head=0;

node\* last=0;

void enqueue(int x)

{

if(head==0)

{

node\* newnode= new node;

newnode->data=x;

newnode->next=0;

head=last=newnode;

}

else

{

node\* newnode=new node;

newnode->data=x;

newnode->next=0;

last->next=newnode;

last=newnode;

}

}

void dequeue()

{

node\* temp=head;

head=temp->next;

delete temp;

}

int front()

{

return head->data;

}

int empty()

{

int empty;

if(head==0)

empty=0;

else

empty=1;

return empty;

}

void add\_edge(int\* a,int n)

{

int u,v;

cout<<"\t\t\t\t\*\*\*\*\*Enter the Edge\*\*\*\*\*\n";

cout<<"From: ";

cin>>u;

cout<<"To: ";

cin>>v;

\*((a+u\*n)+v)=1;

\*((a+v\*n)+u)=1;

}

void adj\_matrix (int \* a,int n)

{

cout<<"Adjacency Matrix\n\n";

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

{

for(int j=0;j<n;j++)

{

cout<<\*((a+i\*n)+j)<<" ";

}

cout<<endl;

}

}

void dfs(int\* a,int n,int start,int visited[])

{

if(visited[start]==0)

{

cout<<start<<" ";

visited[start]=1;

for(int j=0;j<n;j++)

{

if(\*((a+start\*n)+j)==1&&visited[j]==0)

dfs((int\*) a,n,j,visited);

}

}

}

void bfs(int \*a,int n,int start,int visited[] )

{

enqueue(start);

visited[start]=1;

while(empty()!=0)

{

cout<<front()<<" ";

int j=front();

dequeue();

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

{

if(\*((a+j\*n)+i)==1&&visited[i]==0)

{

enqueue(i);

visited[i]=1;

}

}

}

}

int main() {

int n,x;

cout<<"Enter the no of vertices: ";

cin>>n;

int a[n][n];

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

{

for(int j=0;j<n;j++)

a[i][j]=0;

}

int visited [n];

do

{

cout<<"1.ADD EDGE\t2.ADJ MATRIX\t3.DFS\t4.BFS\t5.EXIT\n";

cout<<"Enter your choice: ";

cin>>x;

switch(x)

{

case 1:

{

add\_edge(&a[0][0],n);

break;

}

case 2:

{

adj\_matrix((int \*)a,n);

break;

}

case 3:

{

int start;

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

visited[i]=0;

cout<<"DFS BEGINS!!!\n";

cout<<"Enter the starting vertex: ";

cin>>start;

dfs((int\*)a,n,start,visited);

cout<<endl;

break;

}

case 4:

{

int start;

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

visited[i]=0;

cout<<"BFS BEGINS!!!\n";

cout<<"Enter the starting vertex: ";

cin>>start;

bfs((int \*)a,n,start,visited);

cout<<endl;

break;

}

case 5:

{

cout<<"Good Bye\n";

break;

}

default:

{

cout<<"Invalid Choice!!\n";

}

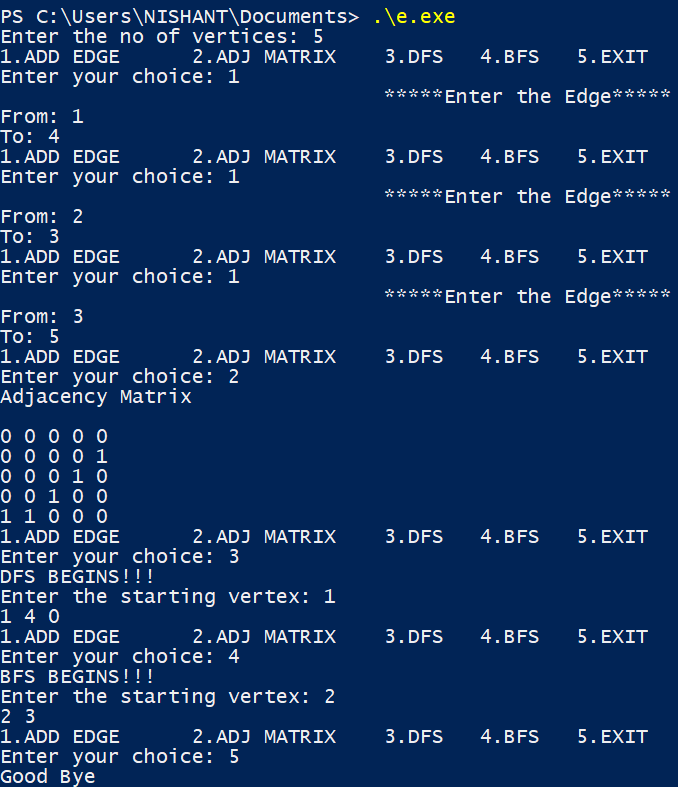
}

}

while(x!=5);

return 0;

}

****