|  |
| --- |
|  |
| **CSE 2208 Algorithms Lab**  **Assignment No**:**02**    **Assignment Topic**:  **1.BFS**  **2.DFS** |

|  |  |
| --- | --- |
| **Date of Performance**: **19-02-2020**  **Date of Submission**: **26-02-2020** | **Name**: **Mubina Ashrafi**  **Student ID**: **180104030**  **Lab Group**: **A2**  **Department of CSE, AUST.** |

**1.BFS**

#include<bits/stdc++.h>

using namespace std;

vector<int>vec[100];

void bfs(int s)

{

int level[100],i;

bool vis[100];

queue<int>q;

q.push(s);

level[s]=0;

vis[s]=true;

while(!q.empty())

{

int p = q.front();

q.pop();

cout << p << " ";

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

{

if(vis[vec[p][i]]==false)

{

level[vec[p][i]]=level[p]+1;

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

vis[vec[p][i]]=true;

}

}

}

}

int main()

{

int node,edge;

int nodeA,nodeB,start;

cin >> node >> edge;

int i,j;

cin >> nodeA >> nodeB;

vec[nodeA].push\_back(nodeB);

start=nodeA;

for(i=2; i<=edge; i++)

{

cin >> nodeA >> nodeB;

vec[nodeA].push\_back(nodeB);

}

for(i=1; i<=node; i++)

{

cout << i << "->";

for(j=0; j<vec[i].size(); j++)

{

cout << vec[i][j] <<" ";

}

cout << endl;

}

bfs(start);

return 0;

}

**2.DFS**

#include<bits/stdc++.h>

using namespace std;

vector<int>vec[100];

void dfs(int s)

{

int level[100],i;

bool vis[100];

stack<int>st;

st.push(s);

level[s]=0;

vis[s]=true;

while(!st.empty())

{

int p = st.top();

st.pop();

cout << p << " ";

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

{

if(vis[vec[p][i]]==false)

{

level[vec[p][i]]=level[p]+1;

st.push(vec[p][i]);

vis[vec[p][i]]=true;

}

}

}

}

int main()

{

int node,edge;

int nodeA,nodeB,start;

cin >> node >> edge;

int i,j;

cin >> nodeA >> nodeB;

vec[nodeA].push\_back(nodeB);

start=nodeA;

for(i=2; i<=edge; i++)

{

cin >> nodeA >> nodeB;

vec[nodeA].push\_back(nodeB);

}

for(i=1; i<=node; i++)

{

cout << i << "->";

for(j=0; j<vec[i].size(); j++)

{

cout << vec[i][j] <<" ";

}

cout << endl;

}

dfs(start);

return 0;

}