**American International University Bangladesh (AIUB)**

****

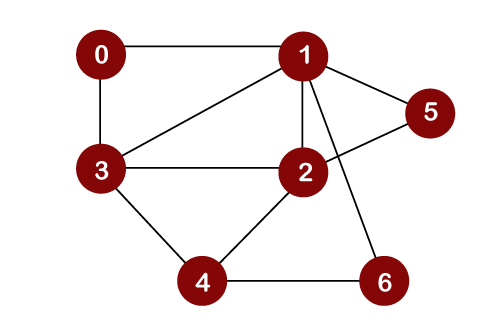
**Faculty of science & Technology**

**Department of Computer Science**

**LAB MANUAL BFS**  
CSC 2211 Algorithms

|  |
| --- |
| **TITLE** |

**BFS Algorithm**



Input Graph as a adjacency List

/\*

7 11

0 1

0 3

1 3

2 3

4 3

1 2

4 2

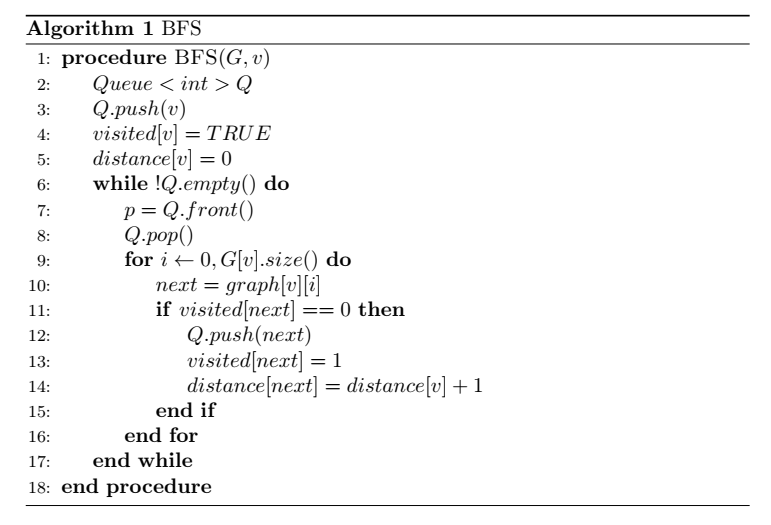
4 6

1 6

2 5

1 5

\*/



BFS implementation using adjacency List

/\*

#include <bits/stdc++.h>

using namespace std;

# define m 100

vector<int> g[m];

int visited[m];

int dis[m];

void bfs(int v){

queue<int> q;

q.push(v);

visited[v]=1;

dis[v]=0;

while(!q.empty()){

int p=q.front();

q.pop();

cout<<p<<" ";

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

int next = g[p][i];

if(visited[next]==0){

q.push(next);

visited[next]=1;

dis[next]=dis[p]+1;

}

}

}

}

int main(){

int node, edge;

cin>>node>>edge;

int u,v;

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

cin>>u>>v;

g[u].push\_back(v);

g[v].push\_back(u);

}

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

cout<<i<<"-->";

for(int j=0;j<g[i].size();j++){

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

}

cout<<endl;

}

int source;cout<<"Enter source for BFS: ";

cin>>source;

bfs(source);

cout<<"\nDistance from "<<source<<" : "<<endl;

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

cout<<i<<" "<<dis[i]<<endl;

}

return 0;

}

\*/

BFS implementation using adjacency Matrix

\*/