

UNIVERSITY OF KARACHI

(UOK)

**ACADEMIC YEAR-2021**

DEPARTMENT OF COMPUTER SCIENCE

**MORNING PROGRAMME**

**MCS (FINAL)- 4th SEMESTER**

**COURSE TITLE*: “ARTIFICIAL INTELLIGENCE”***

**COURSE CODE: CS-616**

**SUBMITTED TO:**

**TEACHER NAME: MISS SAIMA ASHRAF**

**SUBMITTED BY:**

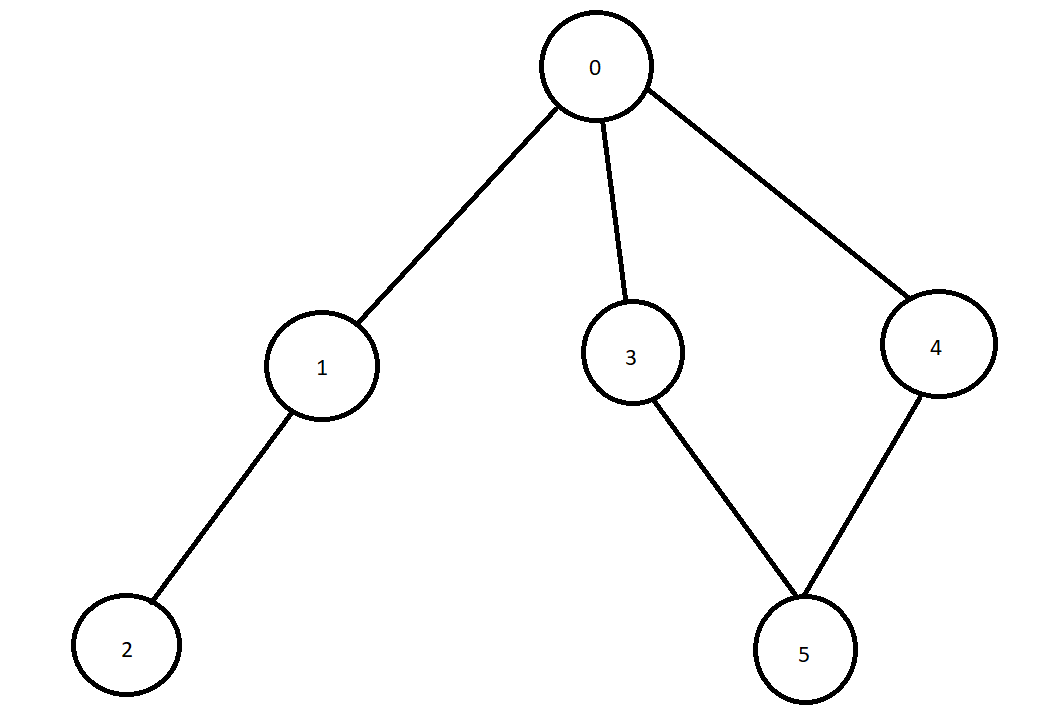
**STUDENT NAME: NEHA ADIL**

**SEAT NO: P19101049**

**ENROLLMENT NO:**  **SCI/DCS/KU-7276/2019**

**IMPLEMENTATION OF BREADTH FIRST SEARCH:**

**Input Tree:**

****

**Source Code:**

package graph;

import java.io.\*;

import java.util.\*;

public class Graph {

private int V; //number of nodes in the graph

private LinkedList<Integer> adj[]; //adjacency list

private Queue<Integer> queue; //maintaining a queue

Graph(int v)

{

V = v;

adj = new LinkedList[v];

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

{

adj[i] = new LinkedList<>();

}

queue = new LinkedList<Integer>();

}

void addEdge(int v,int w)

{

adj[v].add(w); //adding an edge to the adjacency list (edges are bidirectional in this example)

}

void BFS(int n)

{

boolean nodes[] = new boolean[V]; //initialize boolean array for holding the data

int a = 0;

nodes[n]=true;

queue.add(n); //root node is added to the top of the queue

while (queue.size() != 0)

{

n = queue.poll(); //remove the top element of the queue

System.out.print(n+" "); //print the top element of the queue

for (int i = 0; i < adj[n].size(); i++) //iterate through the linked list and push all neighbors into queue

{

a = adj[n].get(i);

if (!nodes[a]) //only insert nodes into queue if they have not been explored already

{

nodes[a] = true;

queue.add(a);

}

}

}

}

public static void main(String[] args) {

Graph graph = new Graph(6);

graph.addEdge(0, 1);

graph.addEdge(0, 3);

graph.addEdge(0, 4);

graph.addEdge(4, 5);

graph.addEdge(3, 5);

graph.addEdge(1, 2);

graph.addEdge(1, 0);

graph.addEdge(2, 1);

graph.addEdge(4, 1);

graph.addEdge(3, 1);

graph.addEdge(5, 4);

graph.addEdge(5, 3);

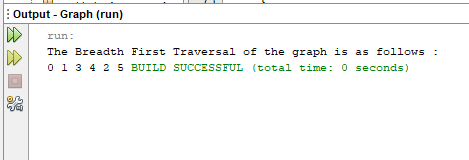
System.out.println("The Breadth First Traversal of the graph is as follows :");

graph.BFS(0);

}

}

**OUTPUT:**

****