**Practical 11**

**Name: Shantanu Sethi**

**Roll no. 163**

**Aim:To create graph structure using adjacency matrix**

**Objectives:**

1. To represent graph using adjacency matrix

**Theory:**

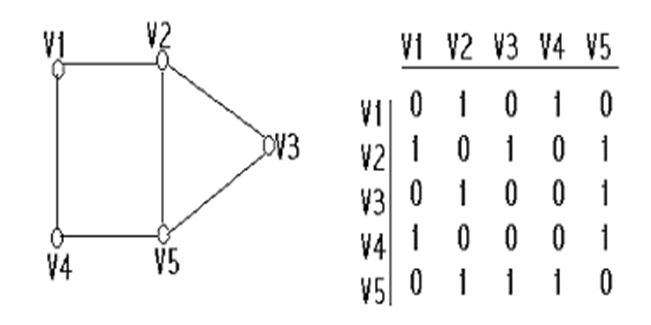
• n by n matrix, where n is number of vertices

– A(m,n) = 1 iff (m,n) Є E ,

– 0 otherwise

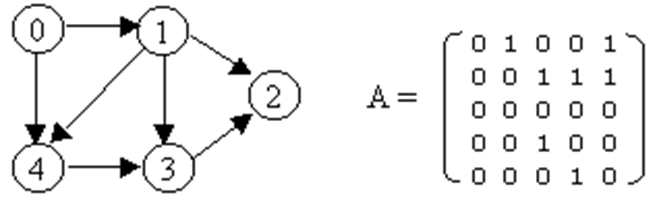
• If Vi is adjacent to Vj, then place 1 at the ith row and jth column and 1 at jth row and ith column; place 0 otherwise.

• The adjacency matrix is symmetric for a simple graph.



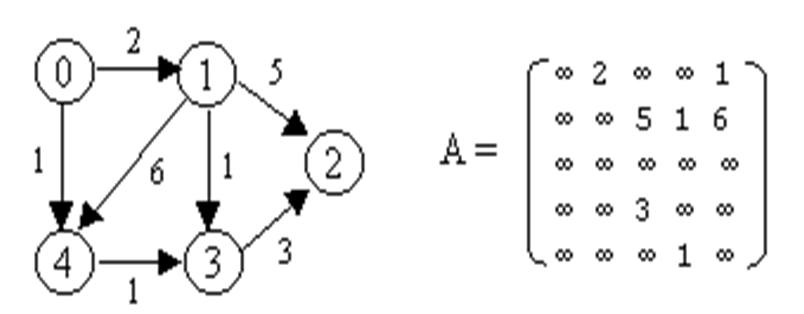
• **Digraph** - for each directed edge (Vi, Vj), we place 1 at ith row and jth column

ü Adjacency matrix, in general, is not symmetric for a digraph.

****

• **Weighted graph:**

A(m,n) = w (weight of edge), or positive infinity (or 0) otherwise



**Program:**

**package** matrix;

**import** java.util.Scanner;

**public** **class** adjMatrix {

**int** totalNodes;

String[] nodes;

**int**[][] adjMat;

Scanner sc = **new** Scanner(System.***in***);

// Constructor

**public** adjMatrix() {

System.***out***.print("How many nodes are there (less than 10): ");

totalNodes = sc.nextInt();

// Validate total nodes

**if**(totalNodes <= 0 || totalNodes > 10) {

System.***out***.println("Invalid number of nodes! Must be between 1 and 10.");

System.*exit*(1);

}

nodes = **new** String[totalNodes];

adjMat = **new** **int**[totalNodes][totalNodes];

// Accept node names

**for** (**int** i = 0; i < totalNodes; i++) {

System.***out***.print("Enter the name of node " + (i + 1) + ": ");

nodes[i] = sc.next();

}

}

// Method to accept edges

**public** **void** acceptEdges() {

**for** (**int** i = 0; i < totalNodes; i++) {

**for** (**int** j = 0; j < totalNodes; j++) {

System.***out***.print("Is there an edge between" + nodes[i] +"--> " + nodes[j] + " (Y/N)? ");

**char** ans = sc.next().charAt(0);

**if** (ans == 'Y' || ans == 'y') {

adjMat[i][j] = 1;

}

**else** {

adjMat[i][j] = 0;

}

}

}

}

// Method to print the adjacency matrix

**public** **void** print() {

System.***out***.print(" ");

**for** (**int** i = 0; i < totalNodes; i++) {

System.***out***.print(nodes[i] + " ");

}

System.***out***.println();

**for** (**int** i = 0; i < totalNodes; i++) {

System.***out***.print(nodes[i] + " ");

**for** (**int** j = 0; j < totalNodes; j++) {

System.***out***.print(adjMat[i][j] + " ");

}

System.***out***.println();

}

}

**public** **static** **void** main(String[] args) {

System.***out***.println("184-ABHINAV SINGH");

adjMatrix adj = **new** adjMatrix();

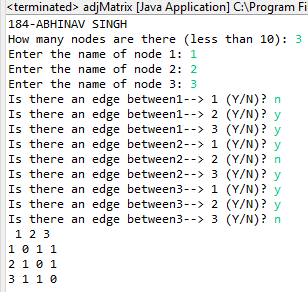
adj.acceptEdges();

adj.print();

}

}

**OutPut:**

****

**Conclusion**: Successfully represented graph using adjacency matrix .