Bansilal Ramnath Agarwal Charitable Trust's

VISHWKARMA INSTITUTE OF INFORMATION TECHNOLOGY,

PUNE DEPARTMENT OF COMPUTER ENGINEERING

**PROJECT SYNOPSIS**

1 Group Id

STUDENTS: -  
1. Eudes Linheiro – 222066 (B3)  
2. Pournima Shinde – 222067(B3)

2 Project Title

Graph run time complexity

3 Project Option

4 Internal Guide  
5 Sponsorship and External Guide

6 Technical Keywords

1. Graph
2. Nodes
3. Vertex
4. Adjacency List

5. Adjacency Matrix

7 Problem Statement

Finding The running time complexity of Graph implemented by using Adjacency Matrix and Adjacency List

8 Abstract

The program is designed to find the running time complexity of graph when implemented by Adjacency Matrix/Adjacency List.  
The adjacency matrix is one way of representing a graph using a two-dimensional array (NxN matrix). In the intersection of nodes, we add 1 (or other weight) if they are connected and 0 or - if they are not connected. the matrix list all nodes horizontally and vertically. If there a few connections we called **sparse graph** if there are many connections (close to the max number of links) we called it **dense graph**. If all possible connections are reached, then we have a **complete graph**.

Adjacency List is one of the most common ways to represent graphs. Each node has a list of all the nodes connected to it.

Graphs can be represented as an adjacency list using an Array (or HashMap) containing the nodes. Each of these node entries includes a list (array, linked list, set, etc.) that list its adjacent nodes.

9 Goals and Objectives

The main objective of the program is to the user how the run time complexity of Graph data structure is calculated and we implemented this by using Adjacency Matrix as well as Adjacency List.

10 Relevant mathematics associated with the Project

11 Names of Conferences / Journals where papers can be published

12 Review of Conference/Journal Papers supporting Project idea

13. Plan of Project Execution