Report for Algorithms & Analysis Assignment 1

Faizan Elahi(s3647111) & Babloo Ratan(s3620927)

INTRODUCTION:

This assignment covers the implementation of Adjacency Matrix and Incidence Matrix. Both Adjacency matrix and Incidence matrix were developed in 2D Array.

The vertices of the matrices are referred to as people and the edges is represented as the connection between the people. The motive of this assignment is to show that the vertices (people) are connected to its edges (connection) by 6 connections if existing between them.

In this assignment, we used *System.nanoTime()* method for each operation to get the time taken to complete each operation.

In this assignment, we are performing some operations such as adding vertices or edges, removing the vertices or edges, finding the neighbours and calculating the shortest path between them.

Both the graph implementations were evaluated to get the time complexity and efficiency of generating the graph.

We created a data subset that has around 2000 vertices and 30000 undirected edges. We tested our graph with 11 different graph density.

The formula to get the density is to divide the number of edges by number of vertices.

Evaluation:

Scenario 1: Growing Friendship Graph

The figure shows that the adjacency matrix is not stable. But for incidence matrix, it is growing higher. When adding vertices (Figure 1), the Adjacency matrix starts fluctuating and incidence matrix shows a steady growth. The incidence matrix is much faster than the adjacency matrix as it takes less time.

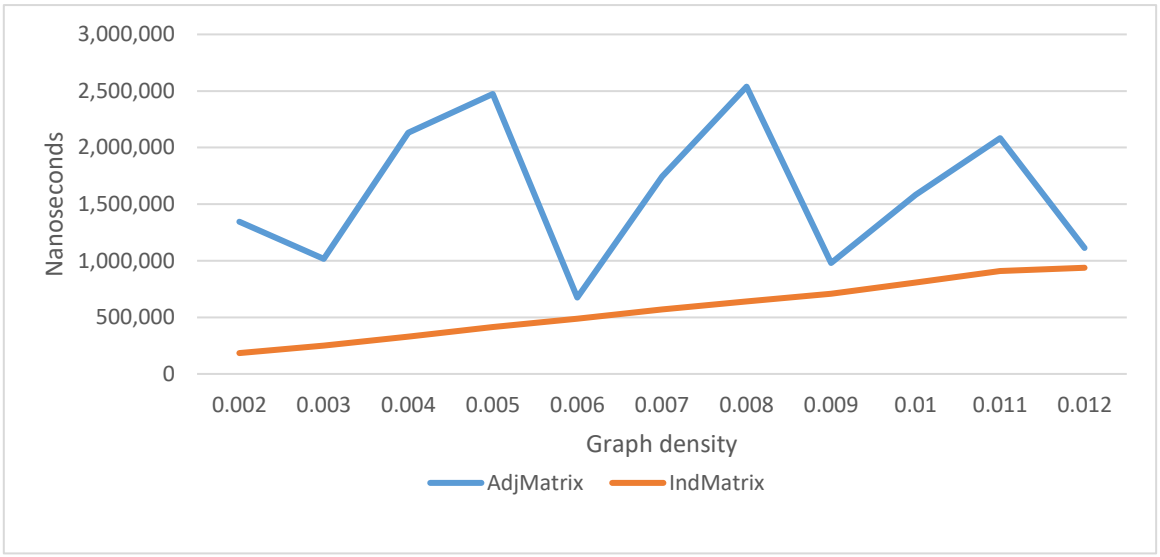

Figure 1: Adding vertices

When adding edges (Figure 2), both incidence and adjacency matrix fluctuates. Here, adjacency matrix takes less time than incidence matrix.

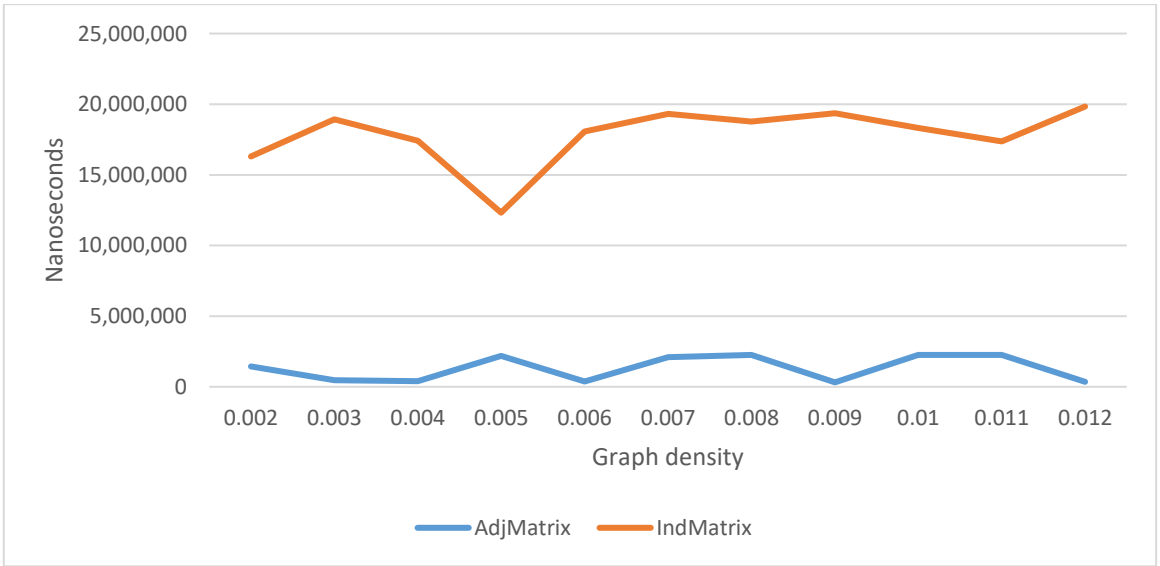

Figure 2: Adding edges

Scenario 2: Neighbours and Shortest path

To calculate the shortest path for adjacency matrix, we used Dijkstra's algorithm and for incidence matrix, we used BFS.

To find the neighbours, incidence matrix took more time than adjacency matrix.

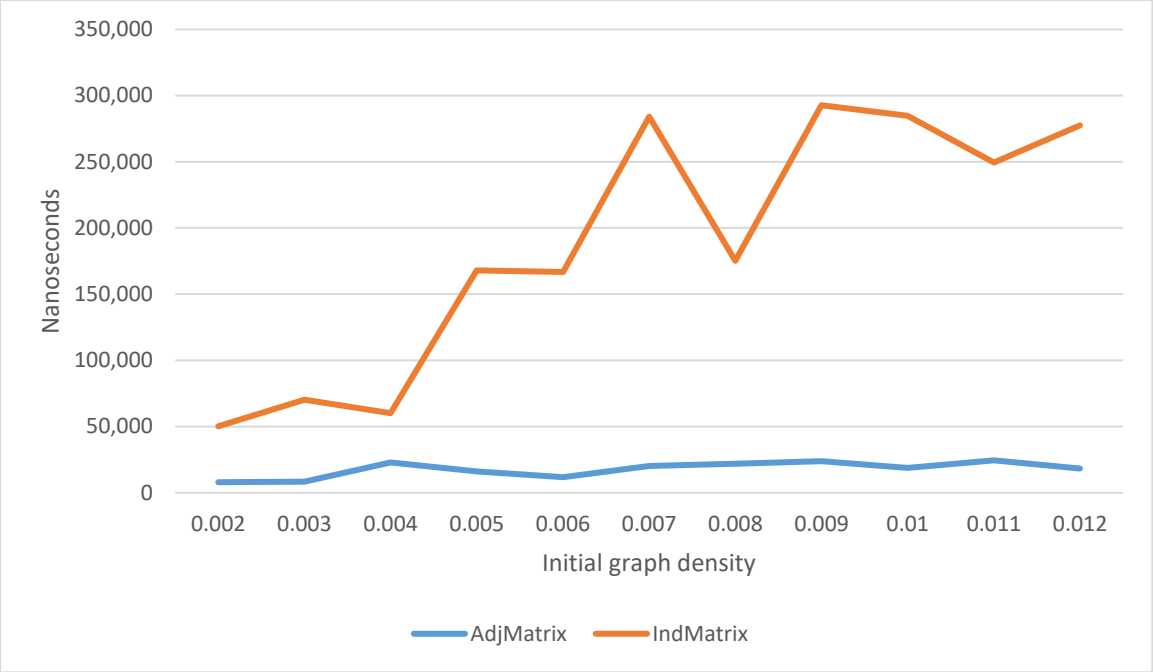

Figure 3: Finding neighbours

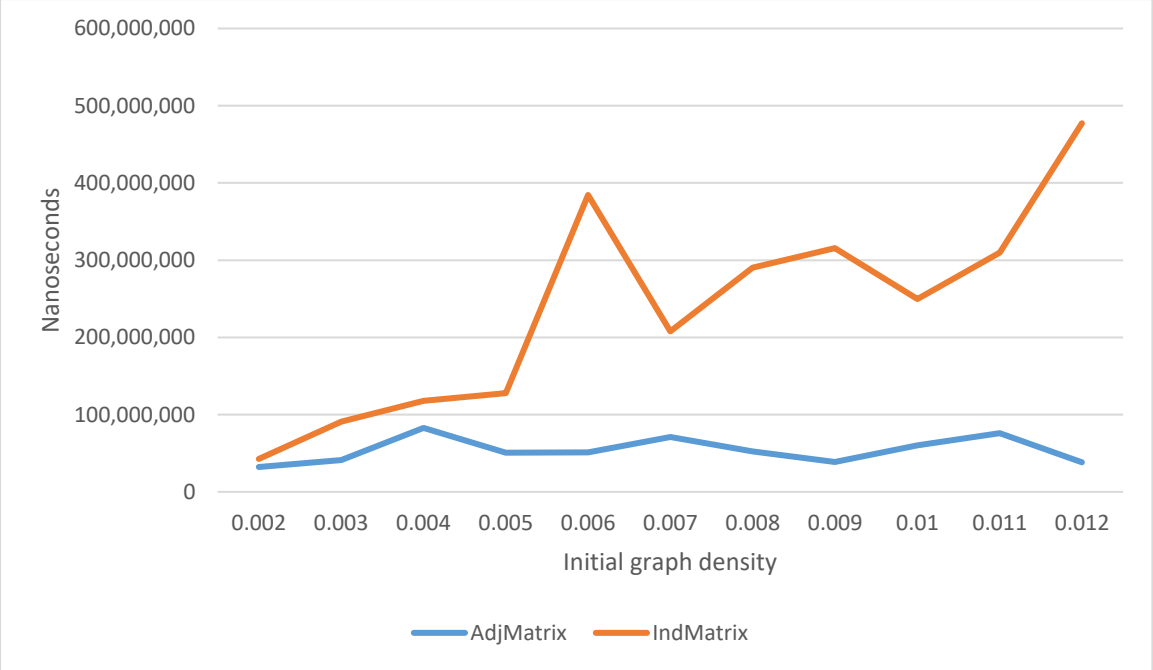

Figure 4: Finding Shortest path

Scenario 3: Shrinking Friendship graph

As seen from the graph (Figure 5), adjacency matrix takes same time in removing the vertices and incidence matrix takes more time when the number of vertices increases.

While removing the edges, the incidence matrix takes more time than adjacency matrix.

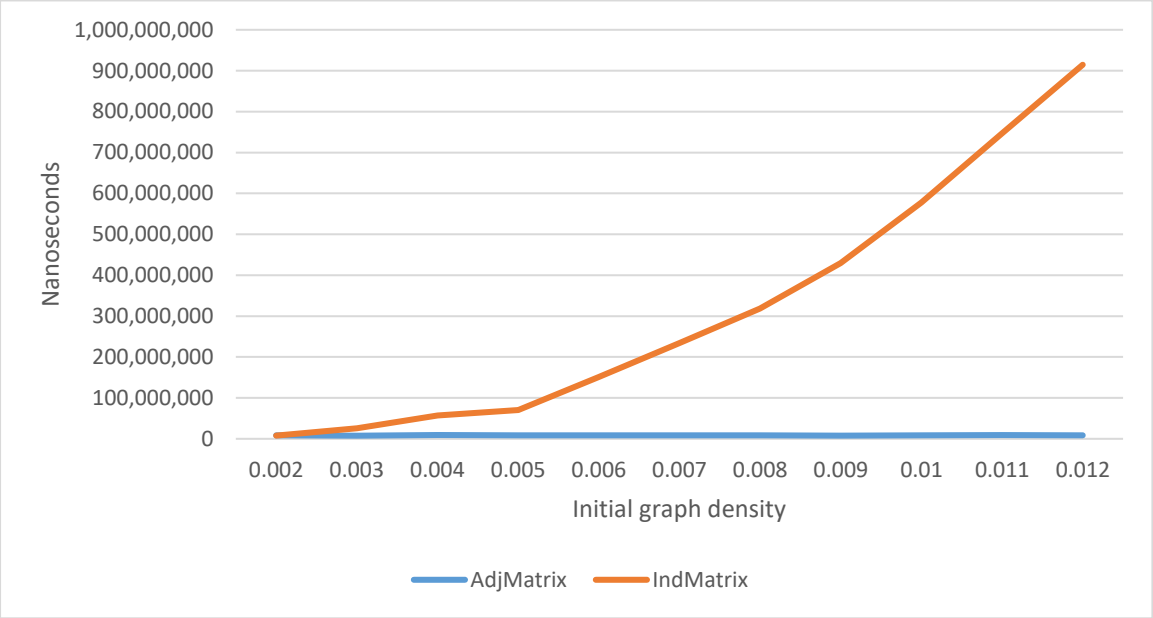

Figure 5: Removing vertices

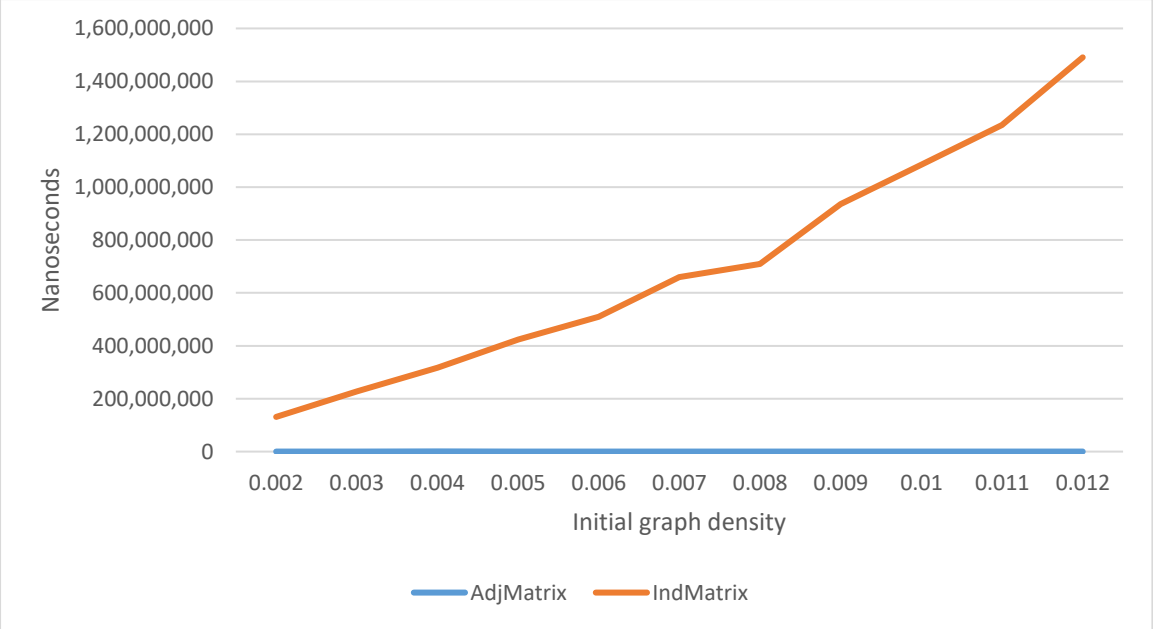

Figure 6: Removing edges

Recommendation:

From the above scenario, it can be said that adjacency matrix was much faster than incidence matrix in almost all operations.

To conclude, we can say that adjacency matrix is suitable for representing friendship graph.