Plot the graph form given adjacency matrix (For undirected, directed and mixed graphs)

Anshul Agarwal *(IWM2017008)*, Vardhan Malik *(IWM2017007)*, Aman Gupta *(IWM2017006)*

***Abstract*—This paper plots undirected, directed and mixed graphs for a given adjacency matrix.**

I. Keywords

Graph, Undirected graph, Directed graph, Mixed graph, Adjacency matrix, Networksx, Edge, Vertex.

II. Introduction

In mathematics, a **matrix** is a collection of numbers arranged into a fixed number of rows and columns. Here is an example of a matrix with three rows and three columns:

2 3 4
2 4 6
7 8 9

An adjacency matrix is a square matrix used to represent a finite graph. Its rows and columns are labeled by graph vertices, with a 1 or 0 in position (i,j) according to whether i and j are adjacent or not.

The adjacency matrix of a complete graph contains all ones except along the diagonal where there are only zeros. The adjacency matrix of an empty graph is a zero matrix.

For an undirected graph,the adjacency matrix is symmetric. All the edges of undirected graph are bidirectional in nature.

0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

For a directed graph,the adjacency matrix need not to be symmetric. If the value of adjacency matrix is 1 (adjacency[i][j]=1), then there is an edge between the vertex i and vertex j, otherwise there is no edge between the two vertices.

0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

For mixed graph, if the value of adjacency matrix (adjacency[i][j]==1) then ij is an edge, or if the value of adjacency matrix(adjacency[i][j]==1) then (i,j) is an arc, or if the value of adjacency matrix(adjacency[i][j]== 1 and adjacency[j][i]== 1) then (j,i) and (i,j) are an arc, otherwise there will be no connection between the two vertices.

0 0 1 0
1 0 0 0
1 1 0 0
1 0 0 0

Here we are plotting the graphs for each case. Adjacency matrix will be taken as an input by the user.

III. ALGORITHM DESIGN AND EXPLANATION

Our approach to given problem includes the following steps:
1. Taking edges of the graph as user input.

2. Using edges, create the adjacency matrix for the graph by marking the graph[i][j] =1 for an edge between ith and jth node ,0 otherwise.(Directed or bi-directed depends on the input edges).

3.By using the Networkx library in Python, we drew the Graph.

4.Traverse the matrix elements one by one and add the edge to the Graph object if the element value is 1.

5.If the graph type is Directed or Mixed, then add attribute arrow = True in the Networkx parameter else, just plot the graph.

The pseudo-code for this algorithm is:-

Algorithm 1 Undirected graph

```
$call\ Graph()\ function$                          ▷ t ← t+1
while $i < rows$ do                                ▷ t ← t+1
    while $j < columns$ do                         ▷ t ← t+1
        if $adjacency[i][j] == 1$ then             ▷ t ← t+1
            $addedge\ from\ i\ to\ j$              ▷ t ← t+1
        end if
    end while
end while
```

Algorithm 2 Directed graph and Mixed Graph

```
$call\ Graph()\ function$                                ▷ t ← t+1
while $i < rows$ do                                      ▷ t ← t+1
    while $j < columns$ do                               ▷ t ← t+1
        if $adjacency[i][j] == 1$ then                   ▷ t ← t+1
            $addedge\ from\ i\ to\ j$                    ▷ t ← t+1
        end if
    end while
end while
$call\ Networkx.Draw(graph, withlabelsarrow = True)$
$inbuilt\ library\ function$                             ▷ t ← t+edge
```

Algorithm 3 User input

```
$vertices$                                               ▷ t ← t+1
$edges$                                                  ▷ t ← t+1
    return 0                                             ▷ t ← t+1
```

IV. Algorithm Analysis

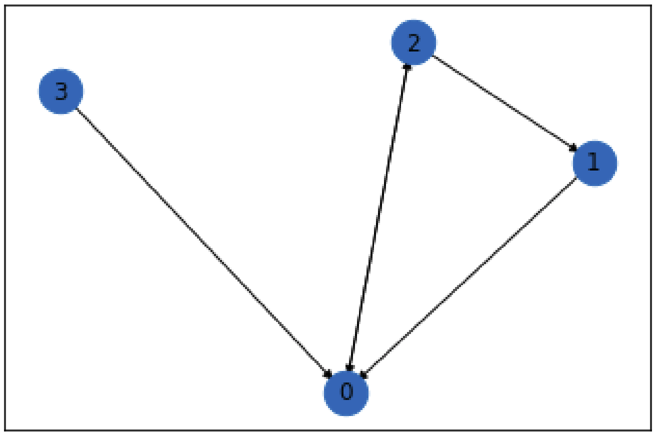
The above algorithm is designed to reduce the time complexity for this question as much as possible. The Networkx Library performs all the operations in liner or constant time thus making it best suitable for the graph to be drawn.

V. Experimental Study

Let the input matrix be :

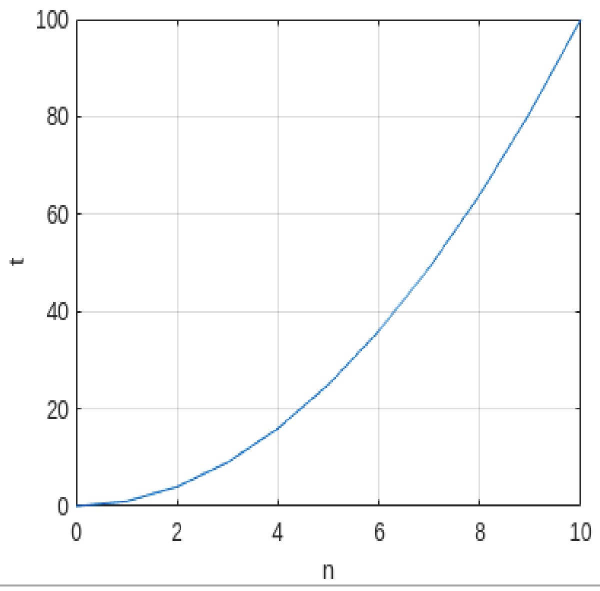
0 0 1 0
1 0 0 0
1 1 0 0
1 0 0 0

So the corresponding graph is



A. Time Complexity

The time complexity analysis of an Algorithm is quite necessary to learn about it's efficiency and optimization.



The time complexity for this algorithm is-
$T = O(n^2 + E)$
Where n is the no. of nodes and E is the no. edges between nodes. According to algorithm we check for each possible edges between the nodes which will takes n^2 time and edges Representation between nodes takes $O(1)$ and there are E edges so $O(E)$ time for all the edges Representation.

The Space Complexity for this algorithm is- $O(n^2)$. where n is no. of nodes. For Storing the graph in adjacency matrix take $O(n^2)$ spaces.

VI. Conclusion

In this paper, we were given an adjacency matrix representing a graph, we were required to plot the graph back from the matrix in most optimised fashion.

References

[1] The Wachowski Brothers, Online ”https://en.wikipedia.org/wiki/Adjacencymatrix”,availal on March 31, 1999

[2] Introduction to Algorithm 3rd Edition By Thomas H.Cormen

[3] Online ”https://www.overleaf.com”