Find out if a vertex v is reachable from vertex u for all vertex(v,u) in a graph.

IV semester - Bachelor's of Technology in Information technology with specialization in Business Informatics,

Indian Institute of Information Technology Allahabad, India

1st Aditya Raj
IIB2019007
iib2019007@iiita.ac.in
Adityahulk

2nd Shyam Tayal
IIB2019008
iib2019008@iiita.ac.in
shyamTayal

3rd Abhijeet Sonkar
IIB2019009
iiB2019009@iiita.ac.in
Abhijeet-sonkar

***Abstract*—In this paper, we are devising an algorithm to a given directed graph, find out if a vertex v is reachable from another vertex u for all vertex pairs (u, v) in the given graph. Here reachable mean that there is a path from vertex u to v. The reach-ability matrix is called transitive closure of a graph.. This paper also analyzes the time and space complexity of the algorithms used and provides the most efficient approach to solve the given problem.**

***Index Terms*—depth first search,floyd warshall algo,reach-ability matrix**

I. Introduction

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking. The Floyd Warshall Algorithm is for solving the All Pairs Shortest Path problem. The problem is to find shortest distances between every pair of vertices in a given edge weighted directed Graph.

Steps of doing Dfs:- Start by putting any one of the graph's vertices on top of a stack. Take the top item of the stack and add it to the visited list. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the top of the stack. Keep repeating steps 2 and 3 until the stack is empty.

Steps of Floyd-Warshall:- We initialize the solution matrix same as the input graph matrix as a first step. Then we update the solution matrix by considering all vertices as an intermediate vertex. The idea is to one by one pick all vertices and updates all shortest paths which include the picked vertex as an intermediate vertex in the shortest path.

II. Algorithm Design

We have devised two algorithms to Find the possible pairs among n friends.

Implementing steps of DFS Algorithm-¿ We take all nodes and edges input and generate graph with an adjency list.Then we make an empty reachability matrix for representation of all paths.We run DFS algo function and mark nodes visted and then move to unvisited child of those nodes from adjency matrix correspondingly.

Algorithm 1: Depth first Search

```
function dfs(curr, src):
    closure[src][curr]=1

    for(child in graph[curr]) :
        if closure[src][child] = 0 :
            dfs(child,src)

for v and u in n :
    dfs(v,v)
answer = closure[v][u]
```

The same problem can be solved out using floyd Warshall's shortest path obtaining algorithm for all nodes.
We initially maintain an nxn matrix(that stores shortest path between any two nodes) storing distance between adjacent nodes as given and other connected nodes as infinity.
Then, We use Dynamic Programming approach to fill and update that matrix.We calculate distance between two nodes via other mid-nodes or direct distance and update with minimum distance as shortest path.

Algorithm 2: Floyd Warshall Algorithm

```
for k from 0 to n :
    for i from 0 to n :
        for j from 0 to n :
            x = closure[i][k]
            y = closure[k][j]
            z = closure[i][j]
            closure[i][j]= z || (x AND y)

given v and u
answer = closure[v][u]
```

III. Algorithm Analysis

Time and Space Complexity Comparison		
Algorithms	**Time**	**Space**
DFS	$O(N^2)$	$O(N^2)$
Floyd Warshall	$O(N^3)$	$O(N^2)$

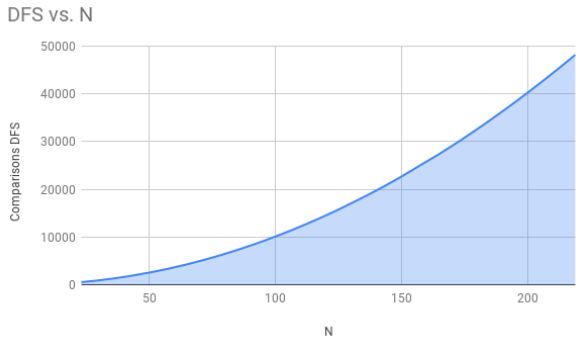

Fig. 1: Algorithm 1 VS N

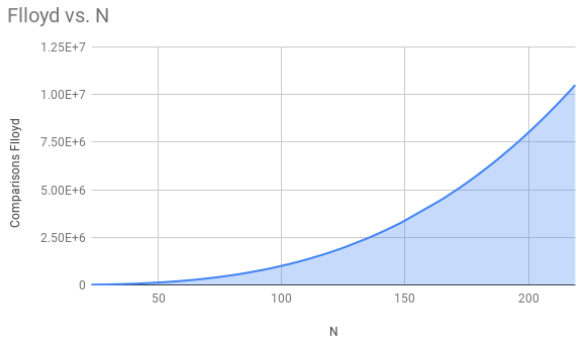

Fig. 2: Algorithm 2 VS N

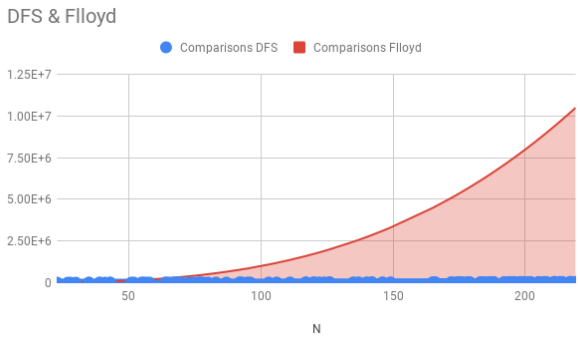

Fig. 3: Algorithm 1 vs Algorithm 2

IV. Conclusion

References

[1] https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm
[2] https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/