A PROJECT REPORT ON

“**FRIEND SUGGESTION – USING GRAPH**”

SUBMITTED TO

Department of computer Science & Engineering, in fulfillment of Project Based Learning for the semester-III of academic year 2018-2019

SUBMITTED BY

1. Vaishnavi Patil (42)

2. Prathmesh Patil(45)

3. Ruturaj Sankpal (54)

4. Shailesh Mohite (55)

UNDER THE GUIDANCE OF

1. Ms. Shital Dinde
2. Ms. Manali Bhandari



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KIT’S COLLEGE OF ENGINEERING,

KOLHARUR.

YEAR-2018-2019

**CERTIFICATE**



KIT’s COLLEGE OF ENGINEERING

This is to certify that, the project entitled “**FRIEND SUGGESTION – USING GRAPH**”,

has been satisfactorily completed by ,

1. Vaishnavi Patil

2. Prathamesh Patil

3. Ruturaj Sankpal

4. Shailesh Mohite

, the students of S Y B Tech, Department of computer Science & Engineering, in fulfillment of Project Based Learning for the semester-III of academic year 2018-2019.

This project report is a record of student’s own work carried by him/her under my supervision and guidance in satisfactory manner.

Date: 09 November 2019

**GUIDE HoD**

**Ms. Shital Dinde Mr. Ajit patil**

**CSE Dept. CSE Dept.**

**Director**

**Kolhapur Institute of Technology’s**

**College of Engineering, Kolhapur.**

**Year 2018-2019**

**ACKNOWLEDGMENT**

We are highly grateful to the **Mr. A. S. Patil**, HOD CSE Dept., KIT’s College of Engineering, Kolhapur, for providing this opportunity to carry out the Project at CSE department. We would like to expresses our gratitude to other faculty members of CSE department for providing academic inputs, guidance & encouragement throughout this period. We would like to express a deep sense of gratitude and thank to **Ms. Dinde Shital** without whose permission, wise counsel and able guidance, it would have not been possible to carry out our project in this manner.

Finally, we express my indebtedness to all who have directly or indirectly contributed to the successful completion of our project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Title** | **Page no.** |
| 1. | Problem statement | 5 |
| 2. | Introduction | 6 |
| 3. | Algorithm | 7 |
| 4. | Implementation | 10 |
| 5. | Inputs | 11 |
| 6. | Result | 12 |
| 7. | Conclusion | 13 |
| 8. | References | 14 |

**PROBLEM STATEMENT**

**Friend Suggestion/recommendation - using graph** :

1. You will be given a graph - (i/p to program - Adjacency Matrix)
2. Each node in the graph will represent a Person. Links between nodes will represent the association between persons.
3. The program shall suggest/recommend the friends by considering the association between nodes. - If friend is added the corresponding link shall be added to the adjacency matrix

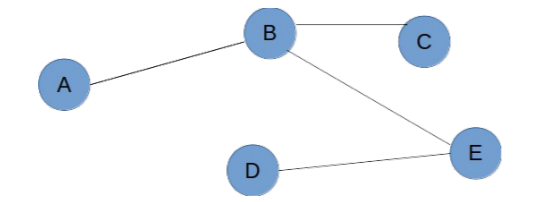


Figure A:- Example Graph

In above fig A. the Person A shall get recommendation to connect with Person C as B is common Friend. Similarly Person D will get recommendation to connect to B and so on.

**INTRODUCTION**

Graph structure is used to the friend suggestion / recommendation.

A Graph is a non-linear data structure consisting of nodes and edges.

Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like LinkedIn, Facebook. For example, in Facebook, each person is represented with a vertex (or node).

**ALGORITHM**

**Adjacency\_count :**

1. Set i, j, count = 0;
2. Repeat j = 0 to j < number ;
3. If matrix[no][j] == 1, then increment count;

[end loop]

1. Return count;

**Adjacency\_cell :**

1. Set i, j, k, count = 0;
2. count = adjacency\_count(matrix, no);
3. Repeat j = 0 to j < number;
   1. If matrix [no][j] == 1, then result [k] = j;
   2. Increment k;

[end loop]

1. Return result;

**isPresent :**

1. Set i, flag = 0;
2. Repeat for i = 0 to I <= top
3. If stack[i] == no

Set flag = 1;

[end loop]

1. Return flag;

**Push :**

1. If isPresent(stack, no) = 1

[end if]

1. Else
2. Increment top;
3. Stack[top] = no;
4. Return stack;

**Display\_stack :**

1. For i = 0 to i <= top
2. Print stack[i]

**Algorithm**

1. Get adjacency matrix.
2. If operation is finding friends
3. count = adjacency\_count(matrix, no);
4. friends = adjacency\_result(matrix, no, friends);
5. Repeat for i = 0 to i < count

Print friends[i];

1. If operation is suggesting friends,
2. count = adjacency\_count(matrix, no)
3. count\_adj[i] = adjacency\_count(matrix, i);
4. final\_result\_size = final\_result\_size + count\_adj[i];
5. Repeat for i = 0 to i < count

Repeat for j = 0 to j <final\_result\_size

1. If final\_result[i] == no

Set flag = 0;

1. Else

Repeat for k = 0 to k < count

If final\_result[j] == result[k]

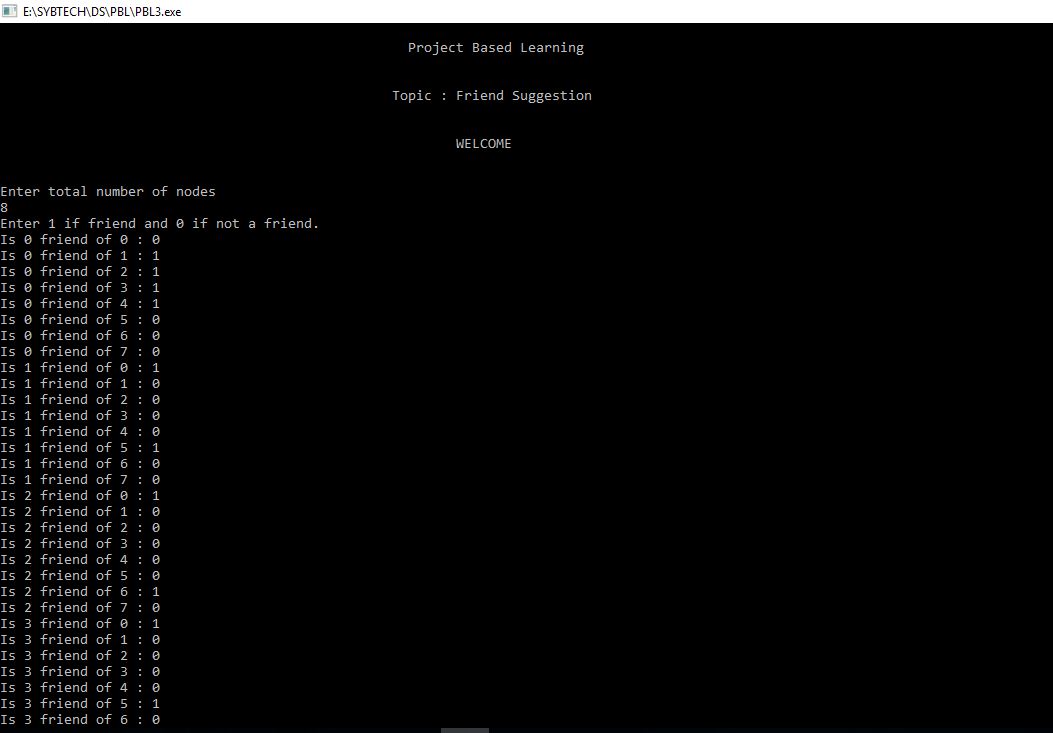
Set flag = 0;

1. If flag == 1, then push element on stack.
2. Display stack
3. If request is sent and is accepted then set the value at corresponding position in adjacency matrix to 1.

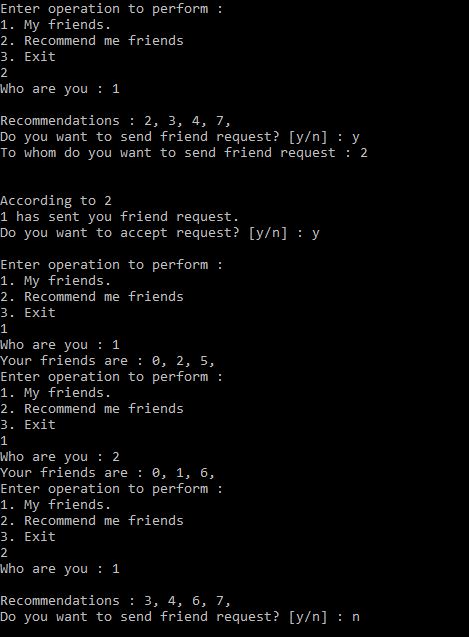
**IMPLIMENTAION**

1. Enter the data of total nodes present in graph.
2. Enter the adjacency matrix to represent whether the link is present between the nodes.
3. Find the count of adjacent nodes present for each node.
4. For each adjacent find count of adjacent nodes.
5. Add those nodes once in stack.
6. If the node is already present in stack then discard the node.
7. Then display the stack.
8. This stack represents the friend suggestion to be given to node.
9. If the request is accepted then change the corresponding value in adjacency matrix.

**INPUT**

****

**RESULTS**

****

**CONCLUSION**

* The different data structures are used to implement the various components in the design of social media.
* Hence we can conclude that friend request/recommendation problem can be solved using adjacency matrix, graph, etc.
* From this program we are able to suggest friends for people in friends circle.
* In this way Graph data structure can be used Friend Suggestion/recommendation

**REFERENCES**

* Data Structures - Lipschutz Seymour
* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)