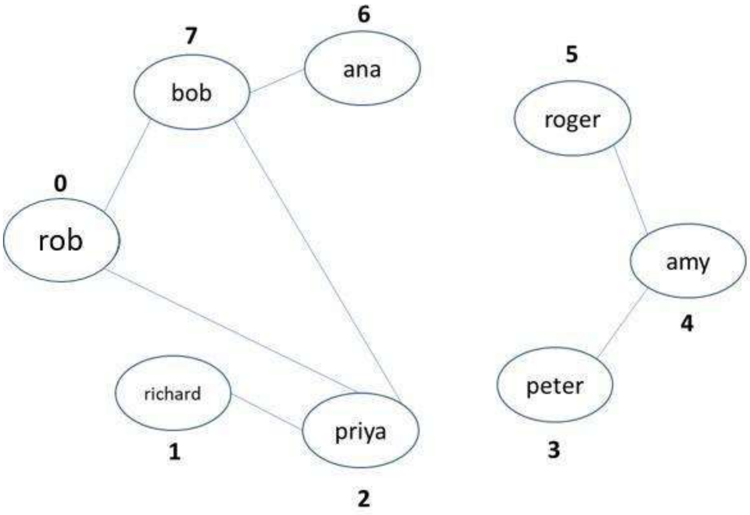
Objective

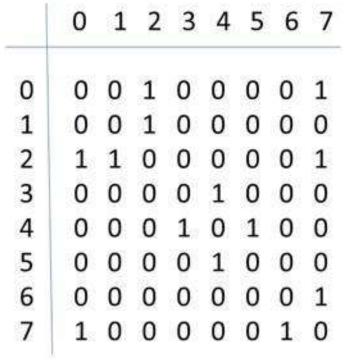
Let us consider a group of friends who are actively present in Social Networking platform which is created using an undirected graph, where people can find their friends, they can also follow up their friends, see their friend suggestion as well be a friend of them by sending request.

This inspires us to create this type of platform from available requisites that we have like algorithm and data structures.

Below is a pictorial representation to describe the social networking system using the graph and its adjacency matrix.



Graphical representation of a social network



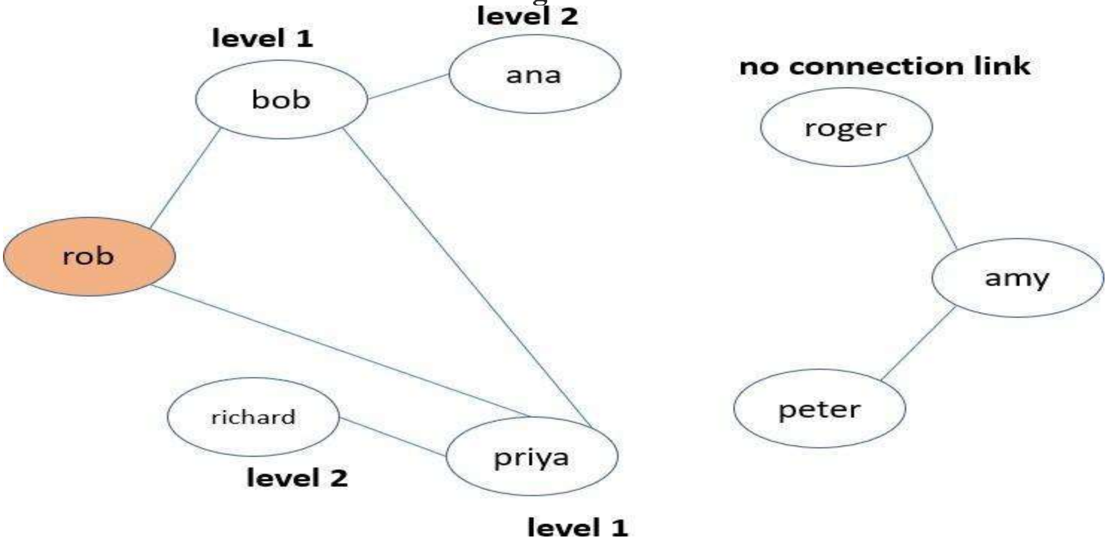
	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	0
2	1	1	0	0	0	0	0	1
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	1	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	1	0

Corresponding adjacency matrix

In the above figure the users are considered as the vertices of the graph. An edge from one user to another denotes that they are friends of each other. Here we can see that each user is provided with a user id (0,1,2,3….). And the corresponding adjacency matrix is also shown above.

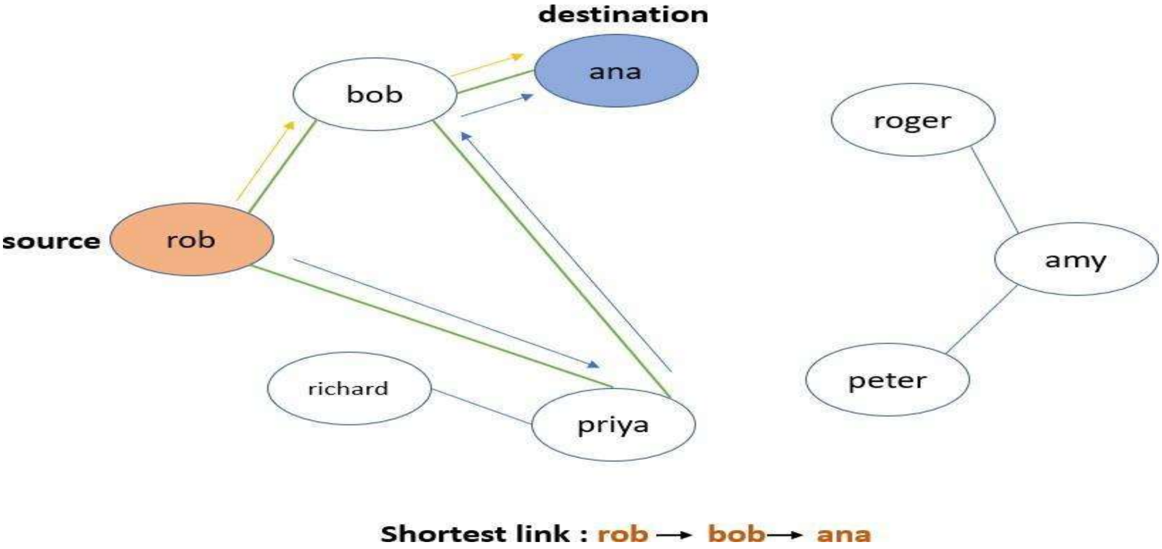
Consider Priya as an example, to find her connections we have to traverse the row represented by her user id in the adjacency matrix, if there is a 1 found then that person is her friend. Here rob, bob and Richard are connected to Priya.
In the similar way we can also find mutual friends of two given users.

To find the level of each user from the given user :



In the above figure for instance consider rob, the level of each user from him can be determined by using the BFS algorithm [5]. Here bob and Priya are at level 1, ana and Richard are at level 2 and there no connection from roger, amy and peter to rob. This is used while displaying suggestions where we will display the connection level along with the username.

To find the shortest connection link between two users :



In the above figure, we can see that there are two links between **rob** and **ana** i.e.,

1. Rob -> Priya ->bob ->ana

2. Rob -> bob -> ana

Here, we use BFS algorithm[4] to find the shortest link between them i.e., the second link.