# ECE 275 - Project 5

**D2L:** Project 5 (alpha)

**Due**: 11/20/2018, 11:59 PM

**D2L:** Project 5

**Due**:11/27/2018, 11:59 PM

# Usage

Usage: sna inputFile userName outputFile

# Requirements Summary

Create a program that will read in the text file containing graph specification of Twitter account followers and write to the output a list of all users within a depth of 3 of the specified root users (the user specified from the userName command line argument), that are not already being followed by that user. The program should be able to create the directed graph from the input file and output the results according to the In-Degree of Centrality.

# Assignment Name

The assignment name for this assignment is: sna

# Social Network Analysis and Common Metrics

The basic structure of SNA is a graph.

To evaluate the characteristics of a graph and the nodes inside, people gradually develop a set of metrics, some of them are:

## Degree of Centrality

In an undirected graph, the degree centrality of a node i (denoted by CD(i)) is the node degree (number of edges), denoted by deg (i).

CD(*i*) = deg (*i*)

Centrality captures “how connected" a node is in general. In a directed graph, the degree of centrality comes in two types: the *in-degree centrality* and the *out-degree centrality*.

The in-degree centrality is number of edges that end at the node aka. Followers (i.e., the number of edges going to the node).

The out-degree centrality is the number edges starting at the node (i.e., the number of edges leaving the node).

# Dataset Description: Twitter accounts and their friends

The inputFile will consist of real Twitter accounts with their friends and for each friend his/her friends etc.

By using this data build up a graph and calculate the In-Degree of Centrality for each user.

The input file will have the following format:

*user follower(user listed below follows user to the left)*

github john\_stewart

github microsoft

microsoft oracle

Each line contains two usernames separated by three spaces.

Each username is no longer than 15 characters and contain only alphanumeric characters (letters A-Z,a-z, numbers 0-9) with the exception of underscores.

The two columns define the relationship between two accounts. The user in the second column is following the user in the first column. Thus, this is a directed graph.

For an empty file, a file that does not have the correct format, or if the specified userName is not found, the program should create an empty output file.

You can also assume that the input graph will be connected. Starting at any node, your program will be able to visit all other nodes using a breadth-first or depth-first search, or a variant thereof.

# Task Decomposition

## Define User class and read input file

Define a User class representing each node in the graph. For this class there are some essential concepts that must be captured:

* name: store a username
* followers: other users following this account
* following: other users that this account is following

The created User class objects need to be stored properly. Decide a way to store this list of all users from input data sets; class definitions are up to you, but the fundamental structure will reflect a directed graph with nodes and edges.

## Calculate the in-Degree centrality

Calculate the in-degree centrality for all users. Using the following relationship, the program should find all users within a depth of 3 from the specified root user who are not already followed by the specified root user.

## Write results to output file

The program should create the outputFile consisting of

a list of all the users within a depth of 3 of the specified root users, that are not already being followed by that user.

The list should be printed in descending order of in-degree of centrality.

In case of ties, the users should be output alphabetically by username (case-insensitive).

The outputFile should have the following format. inDegreeofCentrality should be output using the default precision.

Looking for new accounts for userName (inDegreeofCentrality) to follow

userName1 (inDegreeofCentrality)

userName2 (inDegreeofCentrality)

# Alpha Submission

For the alpha submission, submit a **complete project** that reads in the data, creates the directed graph, and calculates the in-degree of centrality for all users. Then print to the output file the following single line with the specified root users in-degree centrality. inDegreeofCentrality should be output using the default precision.

Looking for new accounts for userName (inDegreeofCentrality) to follow