****

USER GUIDE

# **PURPOSE:-**

# This program focus on carrying out a thorough analysis of how the network is spread in the virtual social site or a media. Consider if a system has vertex as people accounts and links as they are following each other and liking each new post. By then this program is capable of plotting a stats graph which shows the overall changes done to the network which any variable modifications are done from the command line arguments.

# This program has features like:-

# Creating accounts, creating a link between accounts for showing their relationship as they are following each other over the network. It has the probability of liking and following separate consideration based on the possibility given from the command line argument.

# A user of this module has functionalities of creating, adding, deleting and searching for any profile over the network to check all of its properties.

# In the same code, F represents people following other people.

# “:” shows the connection between two profiles.

# P shows what someone has or posting.

# U shows if someone has unfollowed someone’s profile or account.

# A shows to add a new account.

# Finally, R shows to remove/deleting an account.

# Unzipping the folder:-

# Unzip the folder, into a /dir after downloading the zipped folder.

# Set the command prompt directory location to a folder. By using:-

## cd (change directory to:)

# How to use or run the program (step by step):-

# The code or application can be run by using any platform capable of supporting python3 or later.

# For Direct executing of the program type the above command in command prompt.

## python3 SocialSim.py

# [The above command will run the code with initials sets for some accounts created and taken from the provided network and event files for two famous series of “The dark crystal” and “Toy Story-3 ” option and will list it out as graph and a linked list”]

# These values are default set as:-

# Network files = network1.txt

# Events file = events1.txt

# For providing value manually, run the code by giving command-line arguments as shown below accordingly as specified above:-

## python3 SocialSim.py

Then choose (D) for default functionalities.

\OR/ choose (-I)nteractive Mode

\OR/ (S)imulation Mode

For Example:

## Python3 SocialSim.py

Set run mode (D)efault, (-I)nteractive, (-S)imulation and (E)xit.

D (For default mode with the default parameter:- network1.txt and events1.txt)

-I (For interactive way, will give a drop-down menu like list options to choose from which will provide all the needed opportunity to do any wished change to the network )

-S (for running the code in simulation mode which will the timesteps based on files inputted along with the probability of liking a profile/post and likelihood of following a profile

# Running program using Interactive Mode(-I):

* It will list all the necessary option that must be needed to change any operation In a network.

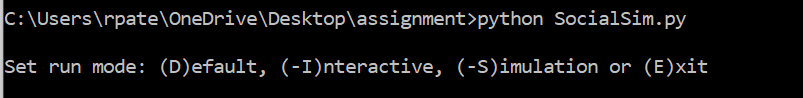


Figure: snipped from the command line of the original code showing the three function

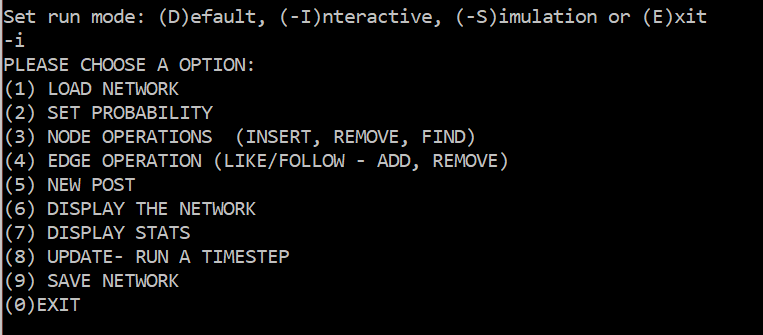


Figure: clipped from the command line of unique code showing the interactive features

* Class NodeFunc() is used to generate and create all the functionality as that of DSAVertex from practical 5 of “Data Structure and Algorithms”, which has sub-function such AddAccounts(), CheckAccExits(), AddLinks() and other as shown in the above figure.



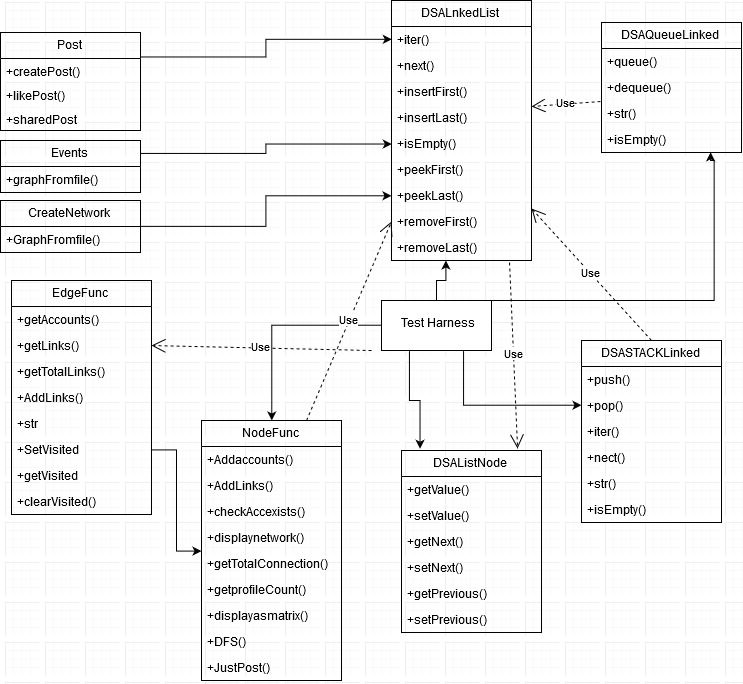
Figure: class NodeFunc with its function.

* The AddAccounts() function is used for creating profiles or adding the node in the DSALinkedList or adding a vertex.
* The CheckAccExist() checks for if any account already exists in the network if it does it would not create another one and if does not it will make a new account using AddAccount() function.
* Def AddLinks() will add a connection between two nodes or say profile, contact can be considered as adding an edge in the DSAGraph code.
* Def getProfileCount(), getTotalConnections and get links() all will do is that will get the total number profile counts, will get whole numbers of edges and connections between two accounts respectively.
* Defining a function for stats to plot a statistical graph for a versus graph.
* Defining displayNetwork() this function will be used whenever someone has to display his/her adjacency list as a network
* Def GetAllProfiles() function will simply extract all the profiles in the system.
* Def DFS() was going to be used for traversing the link but was unsuccessful in doing so was left as unused.
* Saving graphs for each timesteps output and the stats.
* Class EdgeFunc() is used to generate and create all the functionality as that of DSAEdge class from practical 5 of “Data Structure and Algorithms”, which has sub-function such as getAccount(), getLinks(), getTotalLinks(), AddLinks() and other as shown in the above figure.



FIGURE: CLASS EdgeFunc()(From original code of Author of this page.)

* Def getAccount() is used to get all the account in the profile, and this function is called in the NodeFunc() class.
* Def getLinks() will get all the connection and can be called in NodeFunc() class.
* Def AddLinks() will just create a link between two passed parameter if not already has once existed and can be in NodeFunc() class.
* While working in interactive mode, an option such as Load Network can be used by inputting the two files such network1.txt and events1.txt, which a user can change anytime.
* 2. Set Probability will be able to establish and run the timestep after taking probability from the user. Otherwise, it is default set as 0.5 and 0.6 respectively.
* The UML diagram was showcasing how classes were interconnected from other files and how types with InterCall by just merely calling their function.

FIGURE: UML DIAGRAM FOR OVERALL NETWORK OR SAY CODE MODULE SHOWCASING THE RELATION BETWEEN THE CLASSES, FILES AND TEST HARNESS.

To conclude, this user guide is sufficient to provide any user with adequate knowledge or running the SocialSim.py with all the functionalities.For any further queries, please refer to the report and readme files.

##################################################################

END OF THE USER GUIDE.

AUTHOR-

RAJ SURESH PATEL ##CURTIN UNIVERSITY##

STUDENT ID – 19701855.

POSTGRADUATES