VE 444 Networks
Homework3, Dec.5. 2020

Instructions: 1. Compress your code file, readme file, and pdf report file into a zip file . 2. Name your submitted file as studentID_Name_Hwk3, for example, for submissions from Andy Bi with student id 123, his file name should be 123_Andy_Bi_Hwk3. 3. Write necessary explanations rather than simply your answer to demonstrate your reasoning process. 4. The programming language is restricted to either Matlab, C, or C++. 5. The Readme file has to be clearly demonstrate how to run your code successfully.

1. Consider the SIR-model from class, with permanent immunity and recovery rate δ= 0.1. Plot a simulation of the infection process (S,I, R denoted as the fraction of people in each group) in a population of 1000 people, and the transmission rate of 0.2. The initial recovery group size is 0, and the initial infected group size is 1. (Hint: You might want to use first-order differential equation solver in your language. Or just calculate the size in an iterative way)

2. Suppose we have eight employees at a small company, where their relationship score can be measured as values between -100 to +100. A value of zero means the two employees haven't interacted or are indifferent (Recorded in Employee_Relationships.txt). Each employee is further asked to choose 3 movies that they would most enjoy watching for the upcoming company movie night (Recorded in Employee_Movie_Choices.txt).

2.1 Build a friend (with friendship value greater than zero) network, and write a program to plot the degree distribution graph of this friend graph.

2.2 Implement the greedy algorithm we introduced in the class to generate a list of movies to display in the movie night, so that each employee could at least have one movie they like, and the total number of movies that needs to be displayed is also minimized.
Suggested output format:
MovieName1 \t MovieName2 … MovieNameN \n

3. Suppose we have 6 nodes in the network which forms a directed graph with 9 edges. The connection relationship is given as source node set s = [1 1 2 2 3 3 3 4 5], and destination node set t = [2 5 3 4 4 5 6 1 1], where edge i is from s[i] to t[i].

3.1 Display the adjacency matrix M of this directed graph, where element $Mij =$ 1 if there is an edge from node i to node j.

3.2 Implement the pagerank algorithm we introduced in page49 of slide: 12-web_pagerank.pdf and display the pagerank score for each node. The random teleport probability β is set to be 0.85.

Suggested output format:

NodeName \t PageRankScore \n