2016 Fall EE205

**Project #2**

**Twitter characteristics**

**Due date: Dec. 9 (Wed), 11:59 PM (project #2.1~3, 50 pts.)**

**Due date: Dec. 22 (Tue), 11:59 PM (project #2.4~5, 50 pts.)**

**TA: Minseok Kim (**[**kms4105@kaist.ac.kr**](mailto:kms4105@kaist.ac.kr)**)**

Make sure understand clearly, what you do in project#2 by carefully reading this assignment. If you have anything unclear, ask your questions on the klms Lecture Q&A Board. Other private questions can be forwarded to TAs with CC to me ([rhee.jk@kaist.ac.kr](mailto:rhee.jk@kaist.ac.kr)).

**Ⅰ. Introduction**

In this project, you are expected to learn about the framework of twitter, a famous social network. Twitter is an online social networking service, which enables users to send and read short 140-character messages called “twits”.

There are several important terms to inform.

1. Twit : To generate a message and send it to followers of yours.
2. Follow : To subscribe someone’s twit.
3. Follower : People who subscribe your twits are your followers.
4. Followee : People who are followed by followers
5. Retwit : To repost or forward another user's twit

A user’s every follower can see the twit and if a follower retwit it, that follower’s followers can also

see it.

**Ⅱ. Goal**

You are given with an input file of several commands that should carry out corresponding functions. For example,

Add Noah *(Start of twit social network with creating a new user node of Noah.)*

Add Emma

Add Olivia

Add Liam

Add Mason

Follow Noah Emma *(Emma subscribes to Liam’s twit.)*

Follow Emma Liam

Follow Emma Olivia

Follow Mason Liam

Twit Noah “thankyou” retwit Emma Olivia

*(Noah initiates twit that are retwitted by Emma and Olivia.)*

Hop Noah Mason

Del Olivia *(Delete Olivia’s node from twit network. Make sure to fix related follows)*

Unfollow Emma Liam

(…)

* Functions

1. Add *username*

This function add new user.

1. Del *username*

This function delete existing user.

1. Follow *username1 username2*

*Username2* (follower) follows *username1* (followee)

1. Unfollow *username1 username 2*

*Username2* (follower) no more follows *username*1 (followee)

1. Twit *username1* *message* retwit *username2 username3 …*

Print out people who see the twit *message* from *usernam*1 when *username2*, *username3* … retwit the twit

Ex)

Q. Twit Noah “thankyou” retwit Emma Olivia

A. *Emma Liam Olivia Mason Sophia* see the twit *thankyou*

1. Hop *username1 username2*

Print out minimum hop from *usernam1* to *username2*

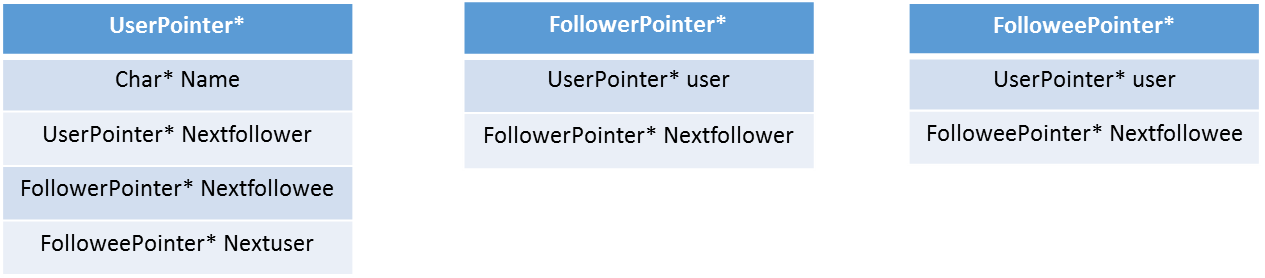
Ex)

Q. Hop Noah Mason

A. 3

* Structure

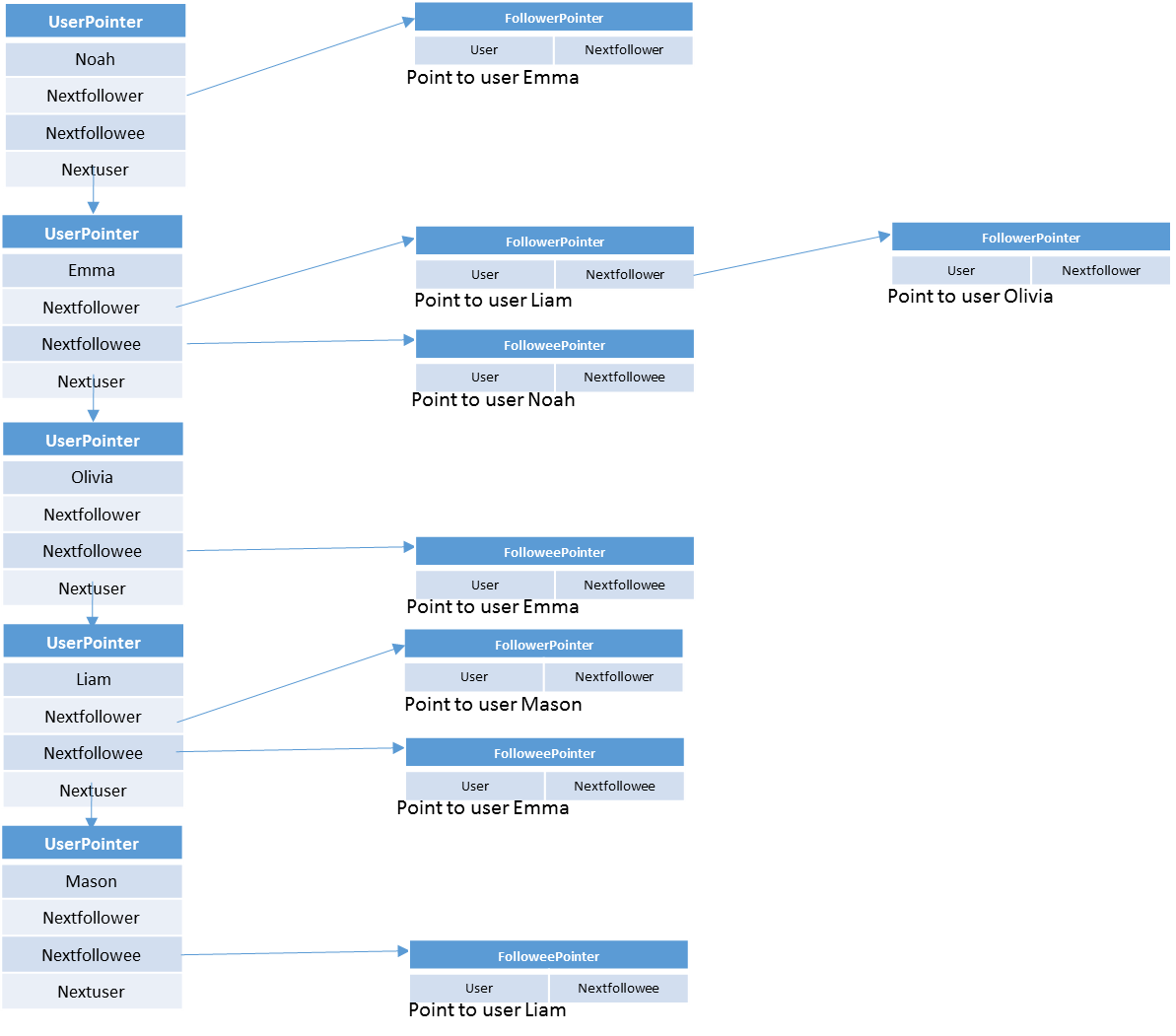
With an input file, you have to save all the users and their followers and followees information in forms of linked list. (You must use linked list. Array is not allowed.) The recommended data structures of a user node, follower list node and followee list node are as follows:



1. UserPointer
   1. Name
      * User name
   2. Nextuser
      * When the new user comes, links least added user and new user using ‘Nextuser’ pointer
   3. Nextfollower / Nextfollowee
      * When the function ‘follow’ comes, according to the follow and the followee relation, generate ‘FollowerPointer’ and ‘FolloweePointer’ and link it using ‘Nextfollower’ and ‘Nextfollowee’ pointers.
2. FollowerPointer / FolloweePointer
   1. User
      * indicates the user pointers with the follower and followee name.
   2. Nextfollower
      * When the new follower and followee arrives, link it using ‘Nextfollower’ and ‘Nextfollowee’ pointer

* Example

With the input above, initial ‘UserPointer’ has the name Noah and its ‘Nextuser’ with the name Emma. Then ‘Nextuser’ of Emma is Olivia, and so on. Also from the command ‘Follow Noah Emma’, Noah has the ‘Nextfollower’ list node whose pointer ‘user’ indicates user with name “Emma”. Then ‘Nextfollowee’ of Emma should be Noah. From the command ‘Follow Emma Liam’, ‘Nextfollower’ in least pointer of Emma’s follower list should indicate Liam using pointer ‘user’. Also ‘Nextfollowee’ of Liam should indicate Emma. See the figure below.



The goal of this process is to find who see a user’s twit. Along the twitter characteristic, a twit spread out through follower’s retwit. You have to make an algorithm of searching people using the linked list. You may use either recursive algorithm or queue.

* Error print
  1. If add the existing user
  2. If delete a user that is not in the user list
  3. If add a follower whose name is not in the user list
  4. If delete a follower that is not in the follower list
  5. If call the twit or hop function with the name that is not in the user or follower list
* Run

Your program should run in the format of

> twitter inputfilename

The result should appear as text on the command screen via “stdio”.

**Ⅲ. Ground rules**

Your program must be implemented by using C language and runnable on the provided Linux machines. You are not allowed to use any generics of C. In this project, you can use only the following header files**.**

*stdio.h, stdlib.h, string.h*

**IV. Queries on Database**

> **twitter** *inputfile*

With the input file above for instance, you should print out

From query ‘Twit *Noah thankyou retwit Emma Olivia’*

*Emma Liam Olivia Mason see the twit thankyou*

And From next query ‘*Hop Noah Mason’*

*3*

After carried out all of the functions like ‘Add’, ‘Delete’, ‘Follow’ and ‘Unfollow’.

#### V. input file

An *inputfile is a plain text ending with* <EOF>.

**VI.** **output\_file**

An *output* will be should be created by stdout redirection. Output file conditions will be provided by TAs shortly.

**VIII. Delay penalty**

- 90% within 4 hours

- 80% within 8 hours

- 70% within 12 hours

- 60% within 24 hours

- 50% within 48 hours

- 0% after 48 hours