Programming Assignment #4

Retrieving Information of multiple users' activities

Due date: April, 19 5:00 PM
URL: http://www.cs.colostate.edu/~cs200

A. Objective

In this assignment, you will build methods to (1) maintain a list of friends and (2) retrieve information about their activities from your system.

B. Description of Task

Please assume that every member has a policy that only their friends can comment, visit, and "like" their posting.

a. Maintaining a list of friends for the members

Each member has a list of friends. Each of these friends have an affinity score associated with that member. Add a data structure (e.g. java.util.HashMap) containing information about the member's friend to the **`Member`** class. For example, if you use `HashMap<String,Integer>` , for a list of friends this data structure will contain pairs of values (friend's userID, affinity score).

```
private HashMap<String,Integer> friends;

public void addFriend(String friend_memberID,int currentAffinityScore);

public void removeFriend(String friend_memberID);

public HashMap<String,Integer> getFriends();
```

b. Retrieving information

You should add methods to the existing class `NewsFeed`.

```
public HashMap<String,Integer> getFriends(String memberID)

public LinkedList <Item>
          getPostings(String memberID, int type, int max_age)

public LinkedList<Edge> getFriendsEdges(String memberID)

public LinkedList<Item>
            getRelatedItems(Item currentItem)
```

```
(1)   HashMap<String, Integer> getFriends(String my_memberID)
```

This method should return a HashMap containing information about friends of a user whose memberID is my_memberID. The information includes the tuple <memberID, affinity score> of each friend.

```
(2)  LinkedList <Item> getPostings (String memberID, int type, int
     max_age)
```

This method should return a LinkedList of Items posted by user specified by the memberID parameter. All of the items in this list should have the types specified in the type parameter. All of the items in this list should have an age value which is less than or equal to the max_age parameter.

```
(3)   LinkedList<Edge> getFriendsEdges(String memberID)
```

This method should return a LinkedList of Edges posted by all friends of the specified user.

```
(4)   LinkedList<Item> getRelatedItems(Item currentItem)
```

This method should return a LinkedList of Items related to the item specified in the currentItem parameter. There are two possible cases:

Case A: As depicted in the Figure 1, if the currentItem is either a wall posting or status change or homepage, your method simply returns all of the items that have the currentItem as their references.

Case B: If the specified item is either “visit”, “like” or “comment”, this item must have been added to a posting created by one of the creator’s friends. Figure 2 shows the relations. Currently item A is the currentItem parameter of getRelatedItems() method. To find all the related items, you should first find the original posting (here, item B). Now, your method should check all of the Edges those created by Member L’s (not K’s) friends. This process is identical to Case A. Please note that in this step, you should show all of the edges related to Item B even if its creator is not Member K’s friend.

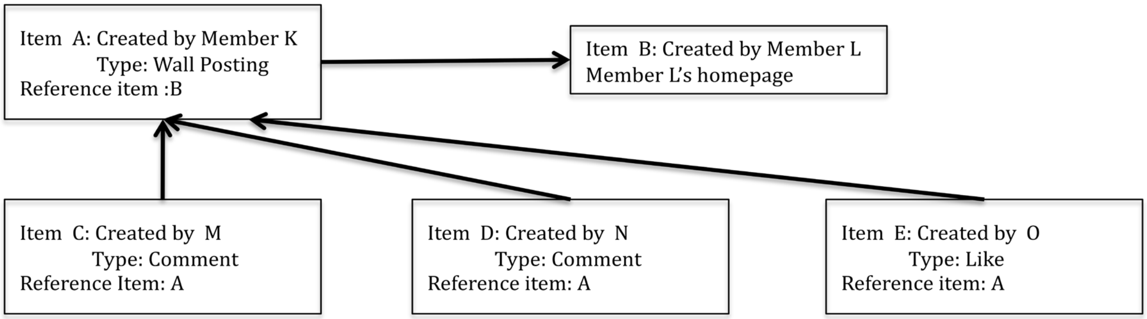

Figure 1. Retrieving related items of Item A - Case A: The current item is a wall posting or status change.
Result should be Item C,D, and E (Assume that the activity of building a homepage is older than the maximum age of these records.)

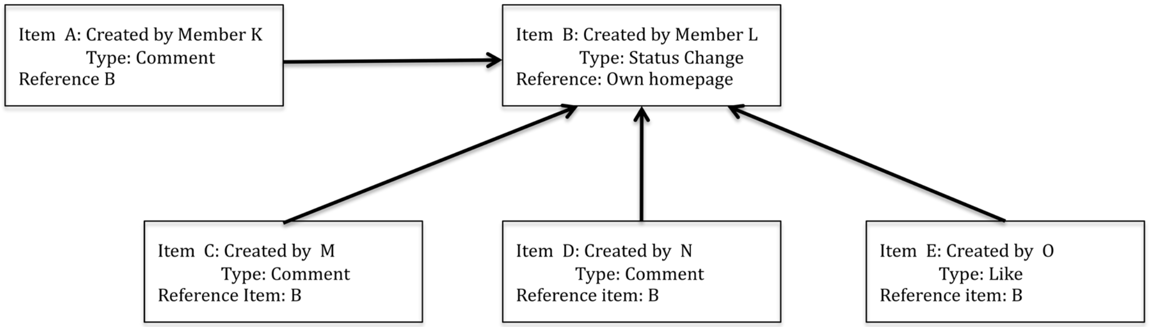

Figure 2. Retrieving related items of Item A -
Case B: if the current item is a comment, visit, or like. Result should be Item B, C, D, and E

c. Parsing Information of multiple users

Each of the users will have their friends and affinity score before their edges. Please modify your InformationParser class to accommodate new data.
Format:

```
      friend [friend's memberID] [affinity score]
```

Example:

```
member ar3090 Ayn Rand
friend eh0721 8
friend willr 7
friend lf8203 6
edge
item 1899 4 ar3090 Message_for_Object1898 598
reference 3 3 http://www.cs200classmatebook.org/~ar3090 97869
```

d. Classes

You can modify **your classes** from PA1, PA2, and PA3. Additional skeletons are provided. Please note that using these skeleton files is **OPTIONAL**. You are encouraged to build your software based on your own design. PA4.java will be provided but please do not change its **main**() method. Example data will also be provided.

D. Requirements/Test Cases

The test cases listed below are provided to assist you in verifying the correctness of your software. You are also required to test your software with different test cases that you will build yourself. Actual grading will be done by test cases using the same commands; however, the values that the specified input arguments take will be different.

(1) Test case 1

Objective: Print out memberIDs of a member's friends and the affinity score associated (with the specified member).

Command: `java PA4 PA4-exampleData.txt printListOfFriends remerson`

Output:

```
eh0721:9
nhawth:4
jambs:5
bettyfriedan:4
```

(2) Test case 2

Objective: Print out the affinity score of specified friend maintained by the member.
Command: The following command should return the affinity score of sinclair. This score is maintained by theodore789's account.

```
java PA4 PA4-exampleData.txt printAffinityScore theodore789 sinclair
```

Output:

```
7
```

(3) Test case 3

Objective: Print out postings performed by the specified member. Your results should contain only the specified type of postings and should not have age greater than the specified one.

Command: The following command should return all the postings published by "willr", with a type of 4 and a maximum age of 3000.

```
java PA4 PA4-exampleData.txt printPostings willr 4 3000
```

Output:

```
item 2890 4 willr Message_for_Object2890 356
item 4681 4 willr Message_for_Objec4681 98
```

(4) Test case 4

Objective: Print all of the edges posted by friends.
Command: The following command should return all the edges tracking sinclair's friends' activities.

```
java PA4 PA4-exampleData.txt printFriendsEdges sinclair
```

Output:

```
edge
item 1671 4 theodore789 Message_for_Object1671 776
reference 13 3 theodore789
http://www.cs200classmatebook.org/~theodore789 91211

edge
item 1675 4 theodore789 Message_for_Object1675 773
reference 13 3 theodore789
http://www.cs200classmatebook.org/~theodore789 91211
```

```
edge
item 1872 4 theodore789 Message_for_Object1872 509
reference 13 3 theodore789
http://www.cs200classmatebook.org/~theodore789 91211

edge
item 1874 4 theodore789 Message_for_Object1874 501
reference 13 3 theodore789
http://www.cs200classmatebook.org/~theodore789 91211

edge
item 2631 2 theodore789 Comment_for_Object2631 398
reference 2502 4 epyle Message_for_Object2502 411

edge
item 2600 2 willr Comment_for_Object2600 400
reference 2502 4 epyle Message_for_Object2502 411

edge
item 2890 4 willr Message_for_Object2890 356
reference 760 3 willr http://www.cs200classmatebook.org/~willr 12999

edge
item 4681 4 willr Message_for_Objec4681 98
reference 760 3 willr http://www.cs200classmatebook.org/~willr 12999
```

(5) Test case 5

Objective: The following command should return all the related items of an item 1186 posted by bettyfriedan containing “Message_for_Obnect1186” and an age of 685. This item has type 4, status change.

Command:

```
java PA4 PA4-exampleData.txt printRelatedPostings 1186 4 bettyfriedan
Message_for_Object1186 685
```

Output:

```
item 1189 0 eh0721  682
item 1194 5 mtwain  680
item 1192 2 jambs Comment_for_Object1192 678
```

(6) Test case 6

Objective: This is the same command as the one used in test case 5. However, in this test, the type of your item in your command is “comment”.

Command:

```
java PA4 PA4-exampleData.txt printRelatedPostings 2631 2
theodore789 Comment_for_Object2631 398
```

Output:

```
item 2502 4 epyle Message_for_Object2502 411
item 2630 2 theodore1 Comment_for_Object2630 399
item 2600 2 willr Comment_for_Object2600 400
```

PA4 file, input file, and submission instructions will be posted on the class web site along with this document. **DO NOT MODIFY the main() of the PA4.java that has been provided to you.**

E. Grading

This assignment will account for 5% of your final grade. The grading itself will be done on a 50 point scale.

G. Late Policy

Please check the late policy available from the course web page.