Programming Assignment #2

Management & sorting of EdgeRank scores in Single-Person Social Network Application

Due date: March 8th, 5:00PM
URL: http://www.cs.colostate.edu/~cs200

A. Objective

In this assignment you will build: (1) a method to calculate the EdgeRank score for each of the edges to be listed on the news feed page. (2) a method to sort the edges using the quicksort algorithm, and (3) your own method to pick the pivot.

B. Description of Task

a. Calculate Edge Rank Score

You will be provided sample code (in `NewsFeed.java`) for this part. You should integrate the sample code into your software.

i. Add `edgeRankScore` property and getter and setter methods in the `Edge` class

ii. Create a new class `NewsFeed` with the following new methods:

```
LinkedList<Edge> getSingleMemberUnsortedNewsFeed(Member viewer)
LinkedList<Edge> getSingleMemberSortedNewsFeed(LinkedList<Edge>
unsorted)
int getWeight(int type)
int getTimeDecay(int age)
```

To calculate the Edge Rank Score, you will need three values for each of the `Edges`: (1) affinity score, (2) weight, and (3) time decay. You stored these values in the `Item` class

Table A. Members' Activities and Weight score

Activity	Weight
Visit (just browsing) (type: 6)	1
Click like (type: 5)	2
Add a Comment (type: 2)	3
Posting on the wall (type: 1)	4

Status Change(type: 4)	5
There will be no type 3 in the activity	0

Table B. Time Decay

Age of the NewsFeed Object in minutes	Time Decay value
0 < age ≤ 60 minutes old	6
60 < age ≤ 180 minutes old	5
180 < age ≤ 360 minutes old	4
360 < age ≤ 720 minutes old	3
720 < age ≤ 1440 minutes old	2
1440 < age ≤ 2880 minutes old	1
2880 < age minutes old (older than 2 days)	0

Since this assignment is for the viewer himself, **the affinity score is 1**.

For each of the `NewsItems(Wall posting/Status Change)`, we will calculate the EdgeRank score (Σ *affinity* x *weight* x *time decay*). If there are other edges related to this edge, (comment on this wall posting, like this status change, etc), you need to add their EdgeRank scores to this EdgeRank score. You can determine if an Edge is related to other Edge, based on the item ID of the reference item. For example, if your wall-posting (ID10) has the EdgeRank score of 20, and there is another item (comment on ID10) having the EdgeRank score of 10, the total EdgeRank score of your wall-posting will be 20+10=30.

Special Requirements

(1) Your NewsFeed page will list **wall postings** and **status changes** only.

(2) All of the items will have a reference to the homepage, wall posting, or status change. We **DO NOT** consider references to anything else.

b. Sorting the scores

Now you have a list of scores for the `Edges`. You should build a method that returns a list of Edges **sorted in descending order**. Modify the sample quicksort code included in the skeleton files (`Qsort.java`). Sample code handles only a very simple integer array.

1. You should modify this code to sort the `Edge` objects.
2. You need to create your own `getPivot()` method.

NOTE: This code is based on the quicksort implementation written by James Gosling. Please note that the `partition()` method works this sample. Based on your getPivot() method, you might have to modify the `partition()` method.

c. Counting the operations
In your quicksort code, add lines to count comparisons and data movements. (An example is included in the `Qsort.java`) Add a method to retrieve that value.

d. Build your own getPivot() method
You will be provided a default `getPivot()` method that always returns the item in the middle. Create your own `getPivot()` method and compare the number of operations.

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 the unsorted list of edges that have type wall-posting or status change and their `EdgeRank` scores.

Command:
```
java PA2 PA2TestSmall.txt getSingleMemberUnsortedNewsFeed
```

Output:
```
edge 33
item 10 4 eh0721 message:10 495
reference 5 3 eh0721 http://classbook.myclass.org/eh0721
35000
edge 18
item 11 1 eh0721 message:11 485
reference 5 3 eh0721 http://classbook.myclass.org/eh0721
35000
```

(2) Test case 2

Objective:

Print the unsorted list of edges that have types, **wall posting or status change**. Print their EdgeRank scores next to the tag "`edge`".

Command:
```
java PA2 PA2TestMedium.txt getSingleMemberUnsortedNewsFeed
```

Output:
```
edge 27
item 10 4 eh0721 message:10 493
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 24
item 11 4 eh0721 message:11 484
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 33
item 12 1 eh0721 message:12 481
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 12
item 13 1 eh0721 message:13 474
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 15
item 16 4 eh0721 message:16 458
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 27
item 17 1 eh0721 message:17 454
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 21
item 18 4 eh0721 message:18 447
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 15
item 22 4 eh0721 message:22 423
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 12
item 25 1 eh0721 message:25 400
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 24
item 26 4 eh0721 message:26 392
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
```

(3) Test case 3
Objective:
Print the sorted list of edges that have type wall posting or status change and their `EdgeRank` scores. For the edges with same scores, they can be placed in any order.

Command:
```
java PA2 PA2TestSmall.txt getSingleMemberSortedNewsFeed
```

Output:
```
edge 33
item 10 4 eh0721 message:10 495
```

```
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
edge 18
item 11 1 eh0721 message:11 485
reference 5 3 eh0721 http://classbook.myclass.org/eh0721 35000
```

(4) Test case 4
Objective:
Print the number of operations during the quicksort algorithm with the default `getPivot()` method:PA2TestSmall.txt

Command:
```
java PA2 PA2TestSmall.txt getTotalNumberOfOperations_Default
```

Output: `3` **(This can vary based on your input order)**

(5) Test case 5
Objective:
Print the number of operations during the quicksort algorithm with the default getPivot() method: PA2TestMedium.txt.

Command:
```
java PA2 PA2TestMedium.txt getTotalNumberOfOperations_Default
```

Output: `57 (with this sample program)` **(This can vary based on your input order)**

(6) Test case 6
Objective:
Print the number of operations during the quicksort algorithm with **your** getPivot() method.

Command:
```
java PA2 PA2TestMedium.txt getTotalNumberOfOperations_Yours
```

Output:
```
Your count
```

PA2 file, input file, and submission instructions will be posted on the class web site along with this document. **DO NOT MODIFY the `main()` of PA2.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.