

## Programming Assignment Overview

### *Search, Sort, and EdgeRank for the Social Network Application*

Students in CS200 will build a social network API similar to Facebook. Your system will manage (1) your personal information and (2) information about your friends and their activities. You will also provide APIs to generate News Feeds for each of the users. Your News Feeds should reflect the *relation* between friends and *importance* of the news: news from a loosely connected friend who's faded off into obscurity or aged items is less important. To build News Feed, we will use the EdgeRank algorithm and this will involve searching, sorting, and data traversal techniques that will be covered in the class.

### **How does Facebook build the News Feed page?**

EdgeRank is an algorithm developed by Facebook to organize their News Feed page to list items that are newer and more relevant. In the EdgeRank Algorithm, every item that shows up in your News Feeds is considered an *object*. Whenever another user interacts with that object, an *edge* will be created between the user and the object. Of course, if someone creates a new object, the "creation" edge between new object and creator will be automatically created. For existing objects such as wall-postings or a status change, if your friend adds comment or tag on it, it will create an edge between the object and your friend.

Based on the concepts described above, EdgeRank provides a formula to optimize the News Feed:

$$\sum_{\text{edge } e} u_e w_e d_e$$

$u_e$ - affinity score between the viewing user and the edge creator.

$w_e$ - weight for this edge type (create, comment, like, tag, etc.)

$d_e$ - time decay factor based on how long ago the edge was created

Friends with whom you regularly interact receive a higher affinity score. Please note that the affinity score is not mutual. Friends A and B do not need to have the same affinity score for each other; you also *cannot change* your friends' affinity score towards you. Weight relates to how important this object is. The "freshness" factor for all edged objects is reflected in the time decay factor.

News Feeds is a list of objects sorted based on their EdgeRank scores. The objects are listed in descending order of their scores, with the highest EdgeRank score listed first. In this assignment, you will build simpler version of actual Facebook's News Feed. Detailed instruction and requirements will be posted in each of the assignments.

You can find more information about this algorithm at:

*Everything you need to know about facebook's edgerank.*

<http://thenextweb.com/socialmedia/2011/05/09/everything-you-need-to-know-about-facebook's-edgerank/>

### *Let's Break Down the Task.*

This software will be separated into 5 steps and each of them will be a programming assignment. You will start with building a single-person social network software. In assignment 3, you will extend the software to support multiple users.

Assignment #	Objectives	Features
PA1 (Individual submission)	Building a single-person social network APIs	*Managing personal information (F1-1) *Managing a Stack of edges (F1-2)
PA2 (F1-1, F1-2 will be reused.) (Team submission)	Calculating the EdgeRank score for personal edges and sorting them	*Calculating EdgeRank score for personal edges(F2-1) *Sorting edges with the Quicksort algorithm (F2-2)
PA3 (F1-1 will be reused.) (Team submission)	Extending your software to support multiple members	*Managing a Queue of "Friends"(F3-1) *Building a binary-tree of members (F3-2)
PA4 (F1-1, F3-2 will be reused) (Team submission)	Managing multiple members	*Searching for a member in the binary-tree(F4-1)
PA5 (F1-1, F1-2, F2-1, F2-2, F3-1, F4-1 will be reused.) (Team submission)	Implementing the News Feed page	*Calculating EdgeRank score of the specified person's friends' edges (including viewer's edges)(F5-1) *Providing the first N items for the first page of the News Feed page(F5-2)

Each of the assignments will include a description, skeleton file, sample input, test cases and example output.

### *Required Programming Platform*

All assignments must be implemented in the Java programming language. For all assignments please ensure that your code compiles without problems.

## ***Submission Requirement***

All programming assignments are to be submitted using the checkin program. (<http://www.cs.colostate.edu/~cs200/Spring12/checkin.html>) When checking in assignment number  $i$  user PA $i$  as the name of the assignment. (e.g. PA0 for the first in-lab assignment). Make sure you submit your assignment as a tarball, as PA $i$ .tar, and only include your .java files unless specified otherwise. You will get familiar with the checkin system in assignment 0 in your recitation class. All programming assignments are due at 5pm on the due date.

Assignments can be submitted up to a maximum of 24 hours past the deadline. There will be a deduction of 10%. No submissions will be accepted after 2:00 pm on next day of the due date and you will be given a 0 for that assignment.

## ***Grading***

The scores for individual assignments will be posted within 5 working days from the late submission deadline. For further information, please check the grading policy page: <http://www.cs.colostate.edu/~cs200/Spring12/GradingPolicy.html>