Assignment 6 Write-Up

Classes Relations
There are five major classes in our program: main class RecommendationSystem.java and four auxiliary classes InputParser.java, ReadWriteCSV.java, User.java, and Recommendation.java. RecommendationSystem.java calls InputParser.java first to check whether input arguments are valid. After the validation, RecommendationSystem.java will read the node csv file and edge csv file(ReadWriteCSV class will be called) to generate a map of users (User.java will be called iteratively for each Twitter user.) Then, Recommendation.java will be called in main class to generate the recommended users for some users as required. After all of this, main class will write the output into a csv file(ReadWriteCSV.java) and provide meta-information on the console.

Handle Errors/Exceptions:
In InputParser, if the given input arguments are invalid such as they don't have enough arguments or they are not csv files. The class will generate appropriate mesages.

Some edges in edges.csv contain an edge to a node ID that doesn't exist, my program will handle this case when adding the connections (following & followers).

When reading input and output files, FileNotFoundException will be catched if the given file is not found and IOException may be throws if some cases happened such as no permission given to read/write.

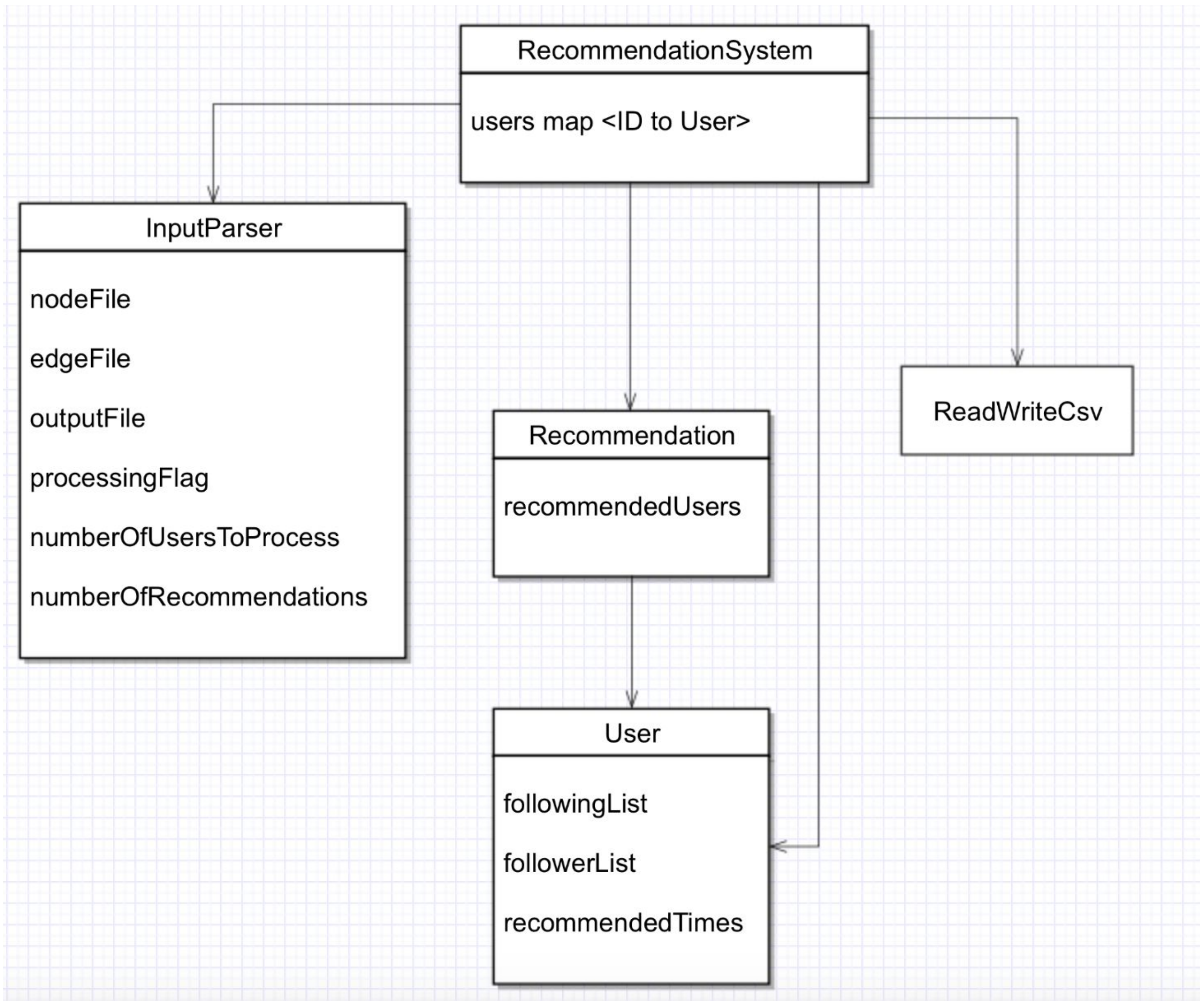
Encoding and Accessing the Information
The program encoded the user information into a map with user id as Key and User itself as Value. We use the idea of directed graph. Assume the whole Twitter social network as a graph. Every single user is a node and every connection (following or follower) is an edge in the graph. The edge directed outwards means this user is following that user. The edge directed inwards means that user is a follower of this user.

Pros: O(1) time of get a single node
Cons: costs O(m+n) space complexity

Time Complexity
The encoding file takes runtime of O(n), where n is the line number of the file.
There are four criteria for the recommendation. Criterion One(Newbies mimic a friendliest friend) takes runtime of O(n). Criterion Two(Friend of a Friend is a Friend) takes runtime of O(m+n), where m is total number of edge and n is total number of nodes. Criterion Three(Always Follow the Influencer) takes runtime of O(n). Criterion Four(When in Doubt, Branch Out) takes runtime

of O(n). Therefore, the runtime of generation recommendation for one user if O(m+n), and total runtime of the program is **O(n(m+n))**.

UML:


RecommendationSystem
users map <ID to User>
InputParser
nodeFile
edgeFile
outputFile
processingFlag
numberOfUsersToProcess
numberOfRecommendations
Recommendation
recommendedUsers
ReadWriteCsv
User
followingList
followerList
recommendedTimes