CS214: Social networking - 2021

Data Structure Course

Project (100 points)

Deadline & Submission

At least one team member should submit the compressed group solution as zip file containing the programs on blackboard –>Project Task(name your assignment file ("Project_ID1_ID2_ID3_ID4_ID5.zip").
Groups must be different student that you worked with in A1 and A2
The deadline for submitting the electronic solution of this assignment is 16/7/2021

About this Assignment

1. This project will be solved in teams from 4 to 5.
2. All team members should understand all project problems.
3. All code must be standard ANSI C++.
4. Assume any missing details and search for information yourself.
5. You may NOT use any built in data structures or algorithms in C++ in this project. YOU MUST BUILD all your data structures by yourself from scratch
6. Any cheating in any part of the assignment is the responsibility of the whole team and the whole team will be punished.

Introduction

You and your friends are building a smart social networking system (choose its name). You are going to compete with the facebook app. Competition is very hard, and you need to build strong features. One of the most critical parts of this system is user manager, and they decided you are the one to develop it. Are you up to the task?

Problem Statement

You are required to implement the user manager for your system. This is a program that keeps track of all user information (profile, friends, ..). You can also quickly reach a user through search.

User Manager (35 points)

- User linked list (10 points)

You are given a file with all users. This file is called "all-users.in". Each line of the file will be in the following format (without quotes):
"username","name","email"
A linked list will be created to save all user in the system and each user object contains (username -is a unique value-, name, email, friendsBST)

- Friend list Balanced BST (25 points)

You will get a file with all users relations. This file is called "all-users-relations.in". Each line of the file will be in the following format (without quotes):

"username1","username2"
"username2","username3"
"username1","username4"

You are required to build a friends list using a balanced BST with all the users sorted by username alphabetically. Each friend is an object, carrying the username and reference to the user object.

Using Linked List to keep all user friends is inefficient for this task. We need a better data structure to efficiently search for a username and get the name. We will use a “Balanced Binary Search Tree” because it allows us to search in O(log n) while keeping the friend list dynamic. Each node in the tree will carry a username and a pointer to the user.For this task you are required to implement a "Treap". It is a Randomized Balanced Binary Search Tree that performs very well in practice. Read the document attached parts 8.1 to 8.4 to understand more about them.

We won’t need all of the functionality of the tree for this problem. The required operations are:

Add(username, pointer to user): adds a username and pointer to user in its right position in the tree.

Find(username): Searches for a username in the tree. If found, returns the user, otherwise returns null.

Application (65 points):

1) Login: type “username” to be logged in (6 points)
2) Logout: type “logout” (2 points)
3) Exit: type “exit” (2 points)
4) Application actions
    - Login
    - Exit
5) List of all action logged in user can do (55 points)
    - List all friends (15 points)
    - Search by username (10 points)
    - Add friend (10 points)
    - Remove friend (10 points)
    - People you may know (10 points)
    - logout

Logged in user Actions Description:

- **List all friends**
    - Traverse “friendsBST” of the logged in user
    - Print these information for each friend in the tree

      Username1, name1
      Username2, name2
      Username3, name3

- **Search by username**
    - If the username belongs to the friend list, display the user “name”.
    - Otherwise display the not found message

      Enter “usernameX”
      If exist, print
      UsernameX, nameX, email
      If doesn’t exist, print
      “not found”

- **Add friend**
    - if userA become friend with userB this mean userB also become friend with userA so you need to update the two users friendsBST

      Enter “usernameX”
      Print
      You are now friends

If they are already friends then print
You are already friends

- **Remove friend**
  - if userA remove userB from his friends this mean userA will be removed from userB friends so you need to update the two users friendsBST

    Enter "usernameX"
    Print
    Done
    If they are already not friends then **do nothing**

- **people you may know**
  - List any users not in user friend list
  - Recommend 5 new friends - if the existed less than 5 view them -

    Username1, name1
    Username2, name2
    Username3, name3

Rules:

1. Cheating will be punished by giving -2 * assignment mark.
2. Cheating is submitting code or reports taken from any source that you did not fully write yourself (from the net, from a book, from a colleague, etc.)
3. Giving your code to others is also considered cheating both the giver and the taker.
4. People are encouraged to help others fix their code but cannot give them their own code.
5. Do not say we solved it together and we understand it. You can write the algorithm on paper together but each group should implement it alone.
6. If you do not follow the delivery style (time and files names), your assignment will be rejected