15-112, Spring 2015

Assignment 7

Submission

You must submit one (and only one) python file on Autolab. This python file must be named *userid*-hw7.py and must include all of the code that you have written for this assignment. Please write your name and userid as a comment at the beginning of each Python file. All of your test cases must be commented but not the functions.

Recommendations

- **Make sure to test your program**.
- Make sure that your program is executable. If you are unable to complete portions of the assignment, comment out the part of the code that does not work properly, and explain what you did, what worked, and what did not. It is your responsibility to explain as carefully as you can why you think you were unable to get the code working, what you think is wrong, and how you might go about fixing it. The quality of such an explanation will be important to us in deciding whether to give you partial credit.

1 Classbook [80 points]

Classbook is a CMU social network where CMU students can share information about their student life. Classbook is a collection of student profiles that contain the following information:

- `andrewid` (as a `string`) is the CMU Andrew ID of the student.
- `firstname` (as a `string`) is the first name.
- `lastname` (as a `string`) is the last name.
- `major` (as a `string`) is either `"BA"`, `"CS"` or `"IS"`.
- `class` (as a `number`) is the year of graduation, `2015` for instance.
- `TA` (as a `list of strings`) is the list of courses for which the student is one of the TA.
- `friends` (as a `list of strings`) is the list of Andrew IDs that are friends with the student.

To implement Classbook, we are going to use a dictionary to store the different student profiles. In this dictionary, a student profile is a key/value pair where the key is the CMU Andrew ID and the value is a sub dictionary that contains the information about the student. Here is a small example:

```
classbook = {
    "samakanbour":{
        "firstname": "Sama",
        "lastname": "Kanbour",
        "major": "IS",
        "year": 2015,
        "TA":["15-112","15-437"],
        "friends":["nadmaney","mpopatia"],
    },
    "nadmaney":{
        "firstname": "Noor-Ul-Huda",
        "lastname": "Admaney",
        "major": "BA",
        "year": 2015,
        "TA":[],
        "friends":["samakanbour","sabih"],
    },
    "sabih":{
        "firstname": "Sabih",
        "lastname": "Bin Wasi",
        "major": "CS",
        "year": 2015,
        "TA":["15-112","15-214"],
        "friends":["nadmaney","mpopatia"],
    },
    "mpopatia":{
        "firstname": "Musab",
        "lastname": "Popatia",
        "major": "CS",
        "year": 2016,
        "TA":["15-112"],
        "friends":["nadmaney","sabih","samakanbour"],
    },
}
```

Warning: the example given here is very small. You should define a bigger Classbook to be able to fully test your code.

[**10 points**] Write a function `register(classbook, andrewid, firstname, lastname, major, class)` that adds a new student profile to classbook. It returns `True` if the student has been added. It returns `False` if the Andrew ID already exists in classbook.

[**10 points**] Write a function `changeMajor(classbook, andrewid, major)` that changes the major of the given student. It returns `True` if the major has been changed. It returns `False` if the given Andrew ID is not in classbook.

[**10 points**] Write a function `becomeTA(classbook, andrewid, courseid)` that makes the given student becoming a TA for the given course. It returns `True` if the course has been added to the student's profile. It returns `False` if the given Andrew ID is not in classbook.

[**10 points**] Write a function `findTA(classbook, courseid)` that returns the list of Andrew IDs that are TA for the given course.

[**10 points**] Write a function `makeFriends(classbook, andrewid1,andrewid2)` that makes two given students friends on classbook. It returns `True` if **both** student's profiles have been modified. It returns `False` if any of them is not in classbook. In this case, the student's profiles should not be modified.

[**10 points**] Write a function `areFriends(classbook, andrewid1, andrewid2)` that returns `True` if two given students are friends on classbook. It returns `False` otherwise.

[**10 points**] Write a function `FriendsOfFriends(classbook, andrewid)` that returns the list of Andrew IDs that are friends of friends for the given student. If a student is already a friend of the given student, he or she should **not** appear in the result.

[**10 points**] Write a function `suggestFriends(classbook, andrewid)` that returns the list of Andrew IDs that are either friends of friends of the given student **or** that are from the class (same major, same year).

2 Degree of separation [20 points]

Read the Wikipedia article: `http://en.wikipedia.org/wiki/Six_degrees_of_separation`.

[**20 points**] Write a function `canReach(classbook, andrewid, degree)` that returns the list of Andrew IDs that can be reached within the given degree of separation on classbook.

