

**IF 100 - Fall 2017**  
**Homework 4**  
**Due December 20<sup>th</sup> 2017 Wednesday 23:55 (Sharp Deadline)**

In this homework, you will pretend like a software engineer in Facebook. We have a real facebook dataset (***friendship.txt***) and this dataset contains information regarding the friends of the users. Based on this dataset, you will suggest a number of friends to a specific user depending on a basic algorithm.

### **Prepared Dataset**

The dataset consists of 'friend pairs' from Facebook. Facebook data was collected from survey participants using the Facebook app. For privacy issues, Facebook data has been anonymized by replacing the Facebook-internal IDs with new dummy values.

In each line of the file containing this dataset, there are two integer values and these integer values are separated by the horizontal tab character (`\t`). Here, each integer value represents the IDs of the users. Hence, each line has the format given below, where *user1* represents the ID of the first user, *user2* represents the ID of the second user and `\t` is the tab character:

*user1\tuser2*

Facebook has an undirected relationship format which means that when you become a friend of a user, then this user also becomes your friend as well. For instance, if you see the line given below in the file (integers are separated by a tab), it means that the user with ID 10 is a friend of the user with ID 158, and it also means that the user with ID 158 is a friend of the user with ID 10. In other words, such a line means that the users with IDs 158 and 10 are friends with each other.

*10      158*

You may assume that the file is in correct format for each line and there are no empty lines. However, you cannot make any assumptions on the number of lines and the number of users in the file.

### **Inputs and Outputs**

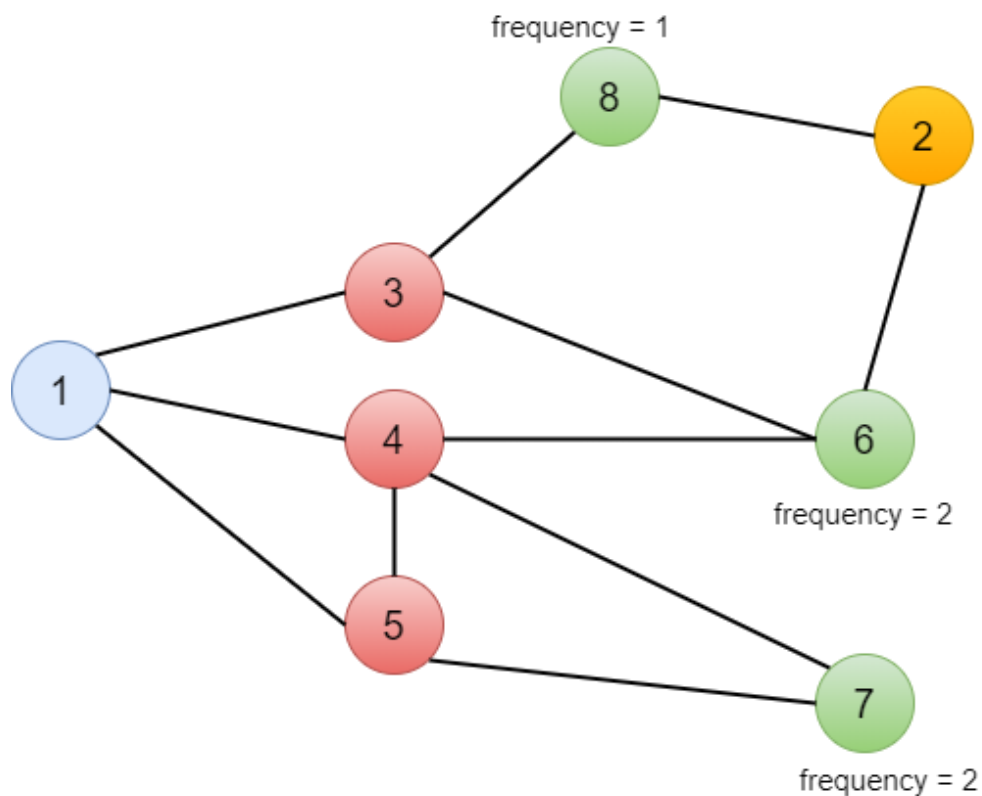
Your program will prompt for a single input, which is an integer that corresponds to the ID of the user, to whom the friend suggestion will be made. You may assume that this input is really an integer, i.e. it consists only of digits. If this user does not

exist in our dataset, i.e. the input user ID does not appear in the prepared dataset file, then your program will prompt "*There is no such user*". Otherwise, your program will suggest (i.e. print) the most frequent user(s) among the friends of the friends of the input user. In other words, it will suggest the most frequent user among the 2<sup>nd</sup> degree connections of the input user. In Facebook, a friend of you is called as ***your 1<sup>st</sup> degree connection***. A user is called ***your 2<sup>nd</sup> degree connection*** if:

- It is not you, and
- It is not your friend, and
- It is a friend of one of your 1<sup>st</sup> degree connections.

The frequency of a 2<sup>nd</sup> degree connection is the number of times that the user appears among the friends of the 1<sup>st</sup> degree connections. Therefore, your program will look for the friends of the 1<sup>st</sup> degree connections of the input user and find the most frequent user ID within those lists.

If we abstract this problem, we may end up with the following graph, where each user is represented by a node and the friendship information between two users is represented by an edge connecting the nodes of these users:



Here, the blue node (node#1) corresponds to the user to whom the suggestion will be made, the red nodes (node#3, node#4, node#5) correspond to this user's 1<sup>st</sup> degree connections and the green nodes (node#6, node#7, node#8) correspond to his/her 2<sup>nd</sup> degree connections. On the other hand, the yellow node (node#2) doesn't have either a 1<sup>st</sup> degree or a 2<sup>nd</sup> degree connection with the blue node. *frequency* underneath a green node denotes the frequency of the node in consideration with respect to its relation with the blue node. For instance, the frequency of node#6 is *frequency* = 2 because of the fact that this node shares 2 common connections (node#3 and node#4) with node#1. Similarly, *frequency* = 2 for node#7 because of the fact that this node also shares 2 common connections (node#4 and node#5) with node#1. In such a scenario, node#6 and node#7 will be the nodes to be suggested for node#1 since they have the highest frequency value.

When there are multiple candidates that shares the maximum frequency for friend suggestion, then your program should print each of these candidates in ascending order in terms of their IDs, by separating them with a comma (see sample runs). In case that the input user does not have any 2<sup>nd</sup> degree friends, then your program should prompt "*There is no friend to suggest*" as the output.

One important point is that the format (sentences, spaces, newlines, order, and everything) of **both** the input and the output **must** be **exactly the same** with the sample runs. The reason is that we automatize the process of grading your homeworks. Therefore, there must be an exact match in your output and the correct output that we specify by format, in order for you to get a full grade. Please see the sample runs for the input and output format.

## Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. Again, we want to emphasize that every single character (even spaces) both in the input and in the output should be exactly same with the sample runs. Whenever you see a space in the sample runs, it means a single space (not multiple consecutive spaces).

### Sample Run 1 (*non-existing user*)

```
Enter a user id to suggest some friends: 11
There is no such user
```

### Sample Run 2 (*no suggestions*)

```
Enter a user id to suggest some friends: 749
There is no friend to suggest
```

**Sample Run 3 (*single suggestion result*)**

Enter a user id to suggest some friends: **47**  
324

**Sample Run 4 (*single suggestion result*)**

Enter a user id to suggest some friends: **77**  
280

**Sample Run 5 (*multiple suggestions result*)**

Enter a user id to suggest some friends: **294**  
40, 332

**Sample Run 6 (*multiple suggestions result*)**

Enter a user id to suggest some friends: **153**  
1, 3, 9, 21, 25, 26, 31, 39, 40, 67, 98, 105, 117, 119, 121, 122,  
133, 141, 142, 169, 185, 188, 200, 231, 232, 236, 239, 252, 257, 271,  
272, 277, 290, 291, 297, 304, 315, 322, 323, 329, 332

**How to get help?**

You may ask questions to TAs, LAs or instructors. Information regarding the office hours of the TAs, LAs and the instructors are available at the SUCourse.

**What and where to submit?**

You should prepare (or at least test) your program using Python 3.6.x. We will use IDLE with Python 3.6.x while testing your homework.

It'd be a good idea to write your name and lastname in the program (as a comment line of course).

Submission guidelines are below. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 20 or more penalty points are to be deducted from your grade.

Name your py file that contains your program as follows:

`"username_hw4.py"`

For example: if your SuCourse username is "*inancarin*", then the name of the py file should be: *inancarin\_hw4.py*

Please make sure that this file is the latest version of your homework program.

However, this is NOT the only file that you are required to submit. In this homework, **you will submit two files** together in one submission:

*username\_hw4.py* and *friendship.txt*

If you forget to submit any of these files, then your program may not be executed and your grade may become 0.

You may visit the office hours if you have any questions regarding submissions.

## General Homework Rules

- Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your homework before the deadline. Please be careful that SUCourse time and your computer time may have a 1-2 minutes differences. You need to take this time difference into consideration.
- Do NOT submit your homework via email or in hardcopy! SUCourse is the only way that you can submit your homework.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Having a correct program is necessary, but not sufficient to get the full grade. Comments, meaningful and understandable identifier names will also affect your grade.
- Please do submit your **own** work only (even if it is not working correctly). It is really easy to find out "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in syllabus of the course.

Good luck!

Duygu Karaoğlu Altop, İnanç Arın, Hüsnü Yenigün