

Lab Assignment 3

Instructor: John C.S. Lui and S.H. Or

Due: 23:59 on Friday, Nov. 6, 2020

Notes

1. You are allowed to form a group of two to do this lab assignment.
2. You are strongly recommended to bring your own laptop to the lab with Anaconda¹ and Pycharm² installed. You don't even have to attend the lab session if you know what you are required to do by reading this assignment.
3. Only **Python 3.x** is acceptable. You need to specify your python version as the first line in your script. For example, if your scripts are required to run in **Python 3.6**, the following line should appear **in the first line of your scripts**:

```
#python_version == '3.6'
```

4. For those of you using the Windows PC in SHB 924A (NOT recommended) with your CSDOMAIN account³, please login and open “Computer” on the desktop to check if an “S:” drive is there. If not, then you need to click “Map network drive”, use “S:” for the drive letter, fill in the path \\ntsvr1\userapps and click “Finish”. Then open the “S:” drive, open the **Python3** folder, and click the “IDLE (Python 3.7 64-bit)” shortcut to start doing the lab exercises. You will also receive a paper document and if anything has changed, please be subject to the paper.
5. Your code should only contain specified functions. Please delete all the debug statements (e.g. print) before submission.

Exercise 1 (20 marks)

Using lists, write the function `non_unique(list)` that takes a list `list` as argument. It returns a list which duplicated elements remains and each duplicated element is followed by a number which shows how many times it appears. All elements in return list should be in the same order as their appearance in the original list.

For example, given the input `['a', 'b', 'c', 'a', 'b', 'd', 'a', 'e']`, the function would return `['a', 3, 'b', 2]`.

If no such `non_unique` list exist, just return an empty list.

Your program should contain the function with format shown as below:

¹An open data science platform powered by Python. <https://www.continuum.io/downloads>

²A powerful Python IDE. <https://www.jetbrains.com/pycharm/download/>

³A non-CSE student should ask the TA for a CSDOMAIN account.

```
def non_unique(list):  
    # Your codes here  
    return result # 'result' is a list.
```

Save your script for this exercise in `p1.py`

Exercise 2 (20 marks)

The quicksort algorithm is a recursive algorithm that works by selecting an element, termed the pivot; dividing the original list into three parts, namely those that are greater than the pivot, those equal to the pivot, and those less than the pivot; and recursively sorting the first and third, appending the results to obtain the final solution. Write a quicksort method that accept a `list` of numbers and use ***list comprehension*** to construct the method. Note that you need to sort the numbers in ascending order .

You **MUST** use **List Comprehension** to do the sorting, otherwise you will get zero marks for Exercise 2.

Your program should contain the function with format shown as below:

```
def quicksort(a):  
    # Your codes here. Use List Comprehension.  
    return result # 'result' is a list.
```

Save your script for this exercise in `p2.py`

Exercise 3 (60 marks)

The given file `followers.pydata` contains 10000 data entries of users and their followers in a social network.

Note that you need to place the data file to the same directory hierarchy as your script for this exercise. When reading a file, you should avoid the crash of your program in case that the file does not exist. (Hint: you may use `try...except..` statement). And you should include all the data files `*.pydata` for Exercise 3 when you submit lab 3 (see **Submission rules**).

- (a) We can use `pickle` to do data storage and retrieval. Implement a function that takes the name of data file (e.g., `followers.pydata`) as argument, through which we can load the given data file `*.pydata` to a dictionary, where the keys are names of the users and the value to each key is a list of users who are following him/her. And the function should return the dictionary loaded from `*.pydata`. **(15 marks)**

Your program should contain the function with format shown as below:

```
def load_data(file):  
    # Your codes here  
    return result # 'result' is the dictionary you load from the 'file'
```

- (b) Implement a function that takes a user's name as argument (case sensitive), through which we can query how many user(s) he/she is **following**. (Attention: It is **NOT** the number of the user's followers.)

The function should return an `int`. **(15 marks)**

Your program should contain the function with format shown as below:

```
def query_following(user_name):  
    # Your codes here  
    return result # 'result' is an int.
```

- (c) Imagine that someone join the social network and someone cancel their accounts.

Implement a function including following operations to the dictionary we get from `followers.pydata`.

- add a new follower Lorna Carrico to the user William Smith
- add a new user Anne Smelcer with followers Christine Phillips, Charles Mason and Tim Lathrop
- use `pickle` to save the updated dictionary to the same directory hierarchy as your script for this exercise as file `followers-updated.pydata`

There is no need to return anything. This function will be graded by testing the generated file `followers-updated.pydata`. **(15 marks)**

Your program should contain the function with format shown as below:

```
def update():  
    # Your codes here  
    # no need to return
```

- (d) Implement a function to analyze the updated data in `followers-updated.pydata` which we get from (c), and return a new `dictionary` where the keys are names of the users and the value to each key is the number of his/her followers. **(15 marks)**

Your program should contain the function with format shown as below:

```
def get_num_of_followers():  
    # Your codes here  
    return result # 'result' is a dictionary
```

Save your script for this exercise in `p3.py`

Submission rules

1. Please name the functions and script files with the **exact** names specified in this assignment and test all your scripts. Any script that has any wrong name or syntax error will not be marked.
2. For each group, please pack all your script files as a single archive named as
`<student-id1>_<student-id2>_lab3.zip`
For example, `1155012345_1155054321_lab3.zip`, i.e., just replace `<student-id1>` and `<student-id2>` with your own student IDs. If you are doing the assignment alone, just leave `<student-id2>` empty, e.g, `1155012345_lab3.zip`.
3. Upload the zip file to your blackboard (<https://blackboard.cuhk.edu.hk>),
 - Only one member of each group needs to upload the archive file.
 - Subject of your file should be `<student-id1>_<student-id2>_lab3` if you are in a two-person group or `<student-id1>_lab3` if not.
 - No later than 23:59 on Friday, Nov. 6, 2020
4. Students in the same group would get the same marks. Marks will be deducted if you do not follow the submission rules. Anyone/Anygroup who is caught plagiarizing would get 0 score!