

## Lab Assignment 3

Instructor: Yim Pan, CHUI

Due: 23:59 on Web. Nov. 2nd, 2022

## 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 Anaconda<sup>1</sup> and Pycharm<sup>2</sup> 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.
4. For those of you using the Windows PC in SHB 924A (NOT recommended) with your CSDOMAIN account<sup>3</sup>, 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.
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 `unique(list)` that takes a list `list` as argument. It returns a list. This list contains two sublists. One includes all duplicated elements of the original list, and the other one is a list of numbers which show how many times the corresponding element 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', 'b'], [3, 2]]`.

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

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

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

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

---

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

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

<sup>3</sup>A non-CSE student should ask the TA for a CSDOMAIN account.

## Exercise 2 (20 marks)

The quicksort algorithm is a recursive algorithm that works by selecting some element, termed the pivot; dividing the original list into three parts, namely those that are smaller than the pivot, those equal to the pivot, and those larger 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 descending order.

You **HAVE TO** use **List Comprehension** to do the sorting, otherwise you will get significant mark deductions for Exercise 2.

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

```
def quicksort(a):
```

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

## Exercise 3 (60 marks)

The given file `dept-prof.pydata` contains data about school departments and their corresponding professors.

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., `dept-prof.pydata`) as argument, through which we can load the given data file `*.pydata` to a `dictionary`, where the keys are names of the departments and the value to each key is a list of professors who work for this department. And the function should return the `dictionary` load from `*.pydata`. **(15 marks)**

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

```
def load_data(file):
```

- (b) Implement a function that takes a professor's name as argument (case sensitive), through which we can query which department(s) he/she works for. And the function should return a `list` of department(s). **(15 marks)**

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

```
def query(prof_name):
```

- (c) Imagine that it is year 2028, and the school want to establish some new departments and remove the outdated ones. Implement a function including following operations to the `dictionary` we get from `dept-prof.pydata`.

- remove the old department **Artificial Intelligence**, and move all professors in this department to the department **Computer Science**
- add a new department **Metaverse** with professors **Mark**, **Neo** and **Trinity**
- use **pickle** to save the updated dictionary to the same directory hierarchy as your script for this exercise as file **dept-prof-updated.pydata**

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

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

```
def update():
```

- (d) Implement a function to analyze the updated data in **dept-prof-updated.pydata** which we get from (c), and return a new **dictionary** where the keys are names of the departments and the value to each key is the number of its professors. **(15 marks)**

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

```
def get_depts_size():
```

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

## Submission rules

1. Please name the functions, script files and data 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 well as the data files **dept-prof.pydata** and **dept-prof-updated.pydata** in Exercise 3 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 Web. Nov. 2nd, 2022
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!