EE 580 Lab 2

Python Programming

Summer 2017   Nazarian

Score:___/100

Student ID: ________________________   Name: ________________________

Assigned: Thursday, May 18th

Due: Wednesday, May 26th at 11:59pm.

Late submissions will be accepted only in the first two days after deadline with a maximum penalty of 15% per day: For each day, submissions between 12 and 1am: 2%, 1 and 2am: 4%, 2 and 3am: 8% and after 3am: 15%.

- **All assignments including this lab are based on individual work. No collaborations (including no discussions) are allowed.**
- **We may pick some students in random to demonstrate their design and simulations. Please watch the first lecture of this course regarding the academic integrity policies and also refer to the syllabus for a summary of AI policies (including the penalties for any violation).**
- **If you have any concerns or doubts about what is or is not allowed or prohibited in this course, please contact the instructor.**
- **ATTENTIONS: Start early otherwise you cannot finish this lab on time.**

Introduction

In this lab you will learn how to program in Python and learn some basic data structures.

Part A. Backtracking (50 points)

Create a program "Paths_in_matrix.py" to count all possible paths from of an MxN matrix**.** Given an input file "input_matrix.txt" in the format shown below:

```
1 1 1 1 1 1 1 0 1 1
1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1
1 1 0 1 1 1 1 1 1 1
```

The program should compute the number of paths from the Top-left to the Bottom-right of the MxN matrix. A path shouldn't have any zeros in it, it should only contain 1's. The value of M and N of the matrix is variable, make a generic program which computes the path for any size of MxN matrix given. The output file "output.txt" should be in the following format:

The number of paths found: #somenumber

Submission: Paths_in_matrix.py

Part B. Social Networks (50 points)

Create a program "social_network.py" which takes in the input of an adjacency matrix of a graph of a network of people's connections. The input will be of the format:

```
0 1 0 1 0 1 0
1 0 1 0 1 0 1
1 1 0 1 1 0 0
0 1 1 0 1 0 1
0 1 0 1 0 0 1
1 0 1 1 1 0 0
0 1 0 0 1 1 0
```

There will be an MxM matrix given as the input, so there will be M number of people in the social network. The names of the people will be "A for M[0], B for M[1], C for M[2], …., Z for M[25]". The size of the matrix can be to a maximum of 26x26. Given this adjacency matrix, the program should take two friend names as input arguments from the user. The two tasks to be performed on the inputs are:

a) Given 2 people, find all the mutual friends if there exists any and count how many friends in common or it should give a zero if they don't have any mutual friends.

Example:

$$\text{Input matrix} = \begin{array}{lcccc} & A & B & C & D \\ A: & 0 & 1 & 0 & 1 \\ B: & 1 & 0 & 1 & 0 \\ C: & 0 & 1 & 0 & 0 \\ D: & 1 & 0 & 0 & 0 \end{array}$$

If the input friends are A and C. Then both A and C have B as a mutual friend between them. The output should be "B".

b) Given 2 people, find a shortest chain of link between them if there exists any?

Example:

Input matrix =
$$\begin{array}{rcccc} & A & B & C & D \\ A: & 0 & 1 & 0 & 1 \\ B: & 1 & 0 & 1 & 0 \\ C: & 0 & 1 & 0 & 1 \\ D: & 1 & 0 & 1 & 0 \end{array}$$

If the input friends are B and D. Then B is friend with C who is in turn friends with D. The output can be “B-C-D” or “B-A-D”. The shortest among the links found are “B-C-D” or “B-A-D”. If there are more than one shortest link. Give all of them as the output. The output for this example: “B-C-D” or “B-A-D”.

The program should create an “output.txt” which gives the output of the two tasks above in the format:

Task 1 Output: #personName
Task 2 Output: #theChainLink

Submission: social_network.py

Submission

1. Zip all the files you need to submit into a zip file named: "firstname_lastname_lab2".zip.
2. Your zip file should include all the coding parts the assignment asks for, and also a Readme.pdf.
3. In your Readme.pdf, include any information that you think the course staff, especially the grader should know while grading your assignment: references, any non-working part, any concerns, etc.
   a. Any non-working part should be clearly stated
   b. The citations should be done carefully and clearly, e.g.: *“to write my code, lines 27 to 65, I used the Diijkstra's shortest path algorithm c++ code from the following website: www.SampleWebsite.com/...”*
4. Use the provided BB submission link to submit your zip file for this assignment