**Use – Depth First Search Algorithm**

**Problem Statement**

**In this Problem, you have to write an application in Python 3.7 that maps Companies and Directors as per the below guidelines.**

Assume that you are working for corporate fraud investigation agency and you want to analyse frauds conducted through money transfer between companies. Agency has identified an association pattern between directors and companies and based on that they want to develop a software application.

**Model the following problem as a graph-based problem. Clearly state how the vertices and edges can be modelled such that this graph can be used to answer the following queries efficiently.**

1. List of unique companies and directors the agency has stored in the system.

2. List of companies associated with a director

3. List of directors associated with a company

4. Validate if two companies have a common director

5. Find if two companies can be connected through a network of directors and companies (you can assume that a director can be associated with max two companies)

6. If you start with a company and reach another company traversing through your graph, then they are connected.

7. Perform an analysis for the features above and give the running time in terms of input size: n.

**The basic structure of the graph will be:**

class CorporateNetwork:

DirectorCompany=[] # list containing companies and directors

Edges=[[],[]] # matrix of edges/ associations

**Operations:**

1. **def readCompanyDirfile (self, inputfile)**: This function reads the input file **inputPS17.txt** containing the name of the companies and associated directors in one line. The name of the Director and Companies should be separated by a slash.

ABCD Corp / James Smith / Maria Garcia

The function should create relevant vertices for the Directors and Companies and relevant edges to indicate the connection of a Company and its Directors. Ensure that the vertices are unique and there are no duplicates.

2. **def displayAll(self):** This function displays the count of unique Directors and Companies entered through the input file. It should also list out the unique Directors and Companies. The output of this function should be stored in **outputPS17.txt** file. The output format should be as mentioned below.

--------Function displayAll--------

Total no. of Companies: 8

Total no. of Directors: 20

List of Companies:

ABCD Corp

HaHaHa Laughing Corp

List of Directors:

…….

3. **def displayCompanies(self, Director):** This function displays all the Companies a particular Director is associated with. The function reads the input Company name from the file **promptsPS17.txt** where the search id is mentioned with the tag as shown below.

findCompany: James Smith

findCompany: Maria Garcia

The output of this function should be appended into **outputPS17.txt** file. If a Company is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

--------Function displayCompanies --------

Director name: James Smith

List of Companies:

ABCD Corp

(if Company is not found display appropriate message)

-----------------------------------------

Note: This is only an indicative output and not the actual output of the program.

4. **def displayDirectors(self, Company):** This function displays all the Directors associated with a Company. The function reads the input Company name from the file **promptsPS17.txt** where the search id is mentioned with the tag as shown below.

listDirectors: ABCD Corp

listDirectors: HaHaHa Laughing Corp

The output of this function should be appended into **output PS17.txt** file. If a Director is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

--------Function displayDirectors --------

Company name: ABCD Corp

List of Directors:

James Smith

Maria Garcia

(if Director is not found, display appropriate message)

-----------------------------------------

5. **def findCommonDirector(self, CompanyA, CompanyB):**

Check if two Companies A and B are related to each other through a common Director. The function reads the input Company names from the file **promptsPS17.txt** where the search id is mentioned with the tag as shown below.

CommonDirector: Company A: Company B

Identify the Director that links Company A and Company B. The output of this function should be appended into **output PS17.txt** file. If a relation is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

--------Function findDirectorConnect --------

Company A: ABCD Corp

Company B: HaHaHa Laughing Corp

Related: Yes, James Smith

(if no, display appropriate message)

-----------------------------------------

6. **def findRelatedCompany(self, CompanyA, CompanyB):**

Check if two Companies A and B are connected to each other through a path on the graph. The function reads the input

Company names from the file **promptsPS17.txt** where the search id is mentioned with the tag as shown below.

RelatedCompany: Company A: Company B

The output of this function should be appended into **outputPS17.txt** file. If a relation is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

--------Function findRelatedCompany --------

Company A: ABCD Corp

Company B: HaHaHa Laughing Corp

Related: Yes

7. Add other functions that are required to perform the above minimum requirement

**Sample Input file**

The input file **inputPS17.txt** contains names of the Directors and its associated Companies in one line. The name of the Director and Companies should be separated by a slash (/).

**Sample inputPS17.txt**

ABCD Corp / James Smith / Maria Garcia

HAHAHA Laughing Corp / Maria Garcia / Maria Hernandez / Juan Carlos / Marcus Ceaser / Sean Maxwell

**Sample promptsPS17.txt**

findCompany: ABCD COrp

findCompany: HaHaHa Laughing Corp

listDirectors: James Smith

listDirectors: Maria Garcia

CommonDirector: James Smith: Maria Hernandez

RelatedCompany: ABCD Corp: Glass Corp

*Note that the input/output data shown here is only for understanding and testing, the actual file used for evaluation will be different.*

**Sample output file**

**Sample outputPS17.txt**

--------Function displayAll--------

Total no. of Companies: 8

Total no. of Directors: 20

List of Companies:

ABCD Corp

HaHa Laughing Corp

Movies Corp

Glass Corp

Almond Corp

List of Directors:

James Smith

Maria Garcia

Maria Hernandez

Juan Carlos

Kayleigh

Ethan

…….

-----------------------------------------

--------Function displayCompanies --------

Director name: James Smith

List of Companies:

ABCD Corp

HaHa Laughing Corp

(if Company is not found display appropriate message)

-----------------------------------------

……

Rest of the function outputs.

*Note that the input/output data shown here is only for understanding and testing, the actual file used for evaluation will be different.*

**2. Deliverables**

1. Word document **designPS17\_<group id>.docx** detailing your design and time complexity of the algorithm.

2. **[Group id]\_Contribution.xlsx** mentioning the contribution of each student in terms of percentage of work done. Download the Contribution.xlsx template from the link shared in the Assignment Announcement.

3. **inputPS17.txt** file used for testing

4. **promptsPS17.txt** file used for testing

5. **outputPS17.txt** file generated while testing

6. **.py file** containing the python code. Create a single \*.py file for code. Do not fragment your code into multiple files

**Zip all of the above files including the design document and contribution file in a folder with the name:**

1. **Instructions**

1. It is compulsory to make use of the data structure(s) / algorithms mentioned in the problem statement.

2. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full. Also ensure basic error handling is implemented.

3. For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.

4. Make sure that your read, understand, and follow all the instructions

5. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.

6. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to evaluate the submissions will be different. Hence, do not hard code any values into the code.

**Instructions for use of Python:**

1. Implement the above problem statement using Python 3.7.

2. Use only native data types like lists and tuples in Python, do not use dictionaries provided in Python. Use of external libraries like graph, numpy, pandas library etc. is not allowed.

3. Create a single \*.py file for code. Do not fragment your code into multiple files.

4. Do not submit a Jupyter Notebook (no \*.ipynb). These submissions will not be evaluated.

5. Read the input file and create the output file in the root folder itself along with your .py file. Do not create separate folders for input and output files.

**Readings**

**Text book:** Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). **Chapters:** 6.1