Work Integrated Learning Programmes Division
M.Tech (Data Science and Engineering)

(S2-19_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2019-2020

Assignment 1 – PS10 - [IPL Bench] - [Weightage 12%]

1. Problem Statement

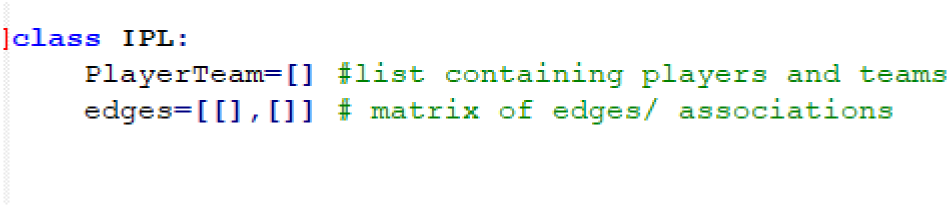
In this Problem, you have to write an application in Python 3.7 that maps IPL franchises and players as per the below guidelines.

Assume that you are a news reporter and you want to map which players have been associated with a franchise (either in the past or present). For this you need to have some system of storing these players and the franchise they have been with. Assume that you have a list of N franchises and M players. For the sake of this assignment, let us assume that a particular player could be associated with only two franchises at max.

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 the unique franchises and players the reporter has collected in the system.
2. For a particular player, help the reporter recollect the franchises he has represented.
3. For a particular franchise, list the players that have been associated with it (past or present).
4. Identify if two players are franchise buddies. Player A and Player B are considered to be franchise buddies if they have been associated with the same franchise (not necessarily at the same time or in the same year)
5. Can two players A and B be connected such that there exists another player C where A and C are franchise buddies and C and B are franchise buddies.
6. Perform an analysis for the questions above and give the running time in terms of n.

The basic structure of the graph will be:



```
class IPL:
    PlayerTeam=[] #list containing players and teams
    edges=[[],[]] # matrix of edges/ associations
```

Functions:

1. **def readInputfile(self, inputfile)**: This function reads the input file **inputPS10.txt** containing the name of the franchises and associated players in one line. The name of the player and franchises should be separated by a slash.

   RR / Ben Stokes / Jaydev Unadkat / Sanju Samson / Jofra Archer / G Krishnappa

   The function should create relevant vertices for the players and franchises and relevant edges to indicate the connection of a franchise and its players. Ensure that none of the players or franchises get repeated while creating the vertices of the graph.

2. **def displayAll(self):** This function displays the total number (count) of unique players and franchises entered through the input file. It should also list out the unique players and franchises. The output of this function should be pushed into **outputPS10.txt** file. The output format should be as mentioned below.

```
--------Function displayAll--------

Total no. of franchises: 10

Total no. of players: 28

List of franchises:

RR

KXIP

KKR

SRH

MI

CSK

RCB

DD

GL

RPS


List of players:

Ben Stokes

Jaydev Unadkat
```

```
Sanju Samson
Jofra Archer
…….
```

3. **def displayFranchises(self, player):** This function displays all the franchises a particular player is associated with. The function reads the input franchise name from the file **promptsPS10.txt** where the search id is mentioned with the tag as shown below.

   ```
   findFranchise: Ben Stokes
   findFranchise: Robin Uthappa
   ```

   The output of this function should be appended into **outputPS10.txt** file. If a franchise is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

   ```
   --------Function displayFranchises --------
   Player name: Ben Stokes
   List of Franchises:
   RR
   RPS          (if franchise is not found display appropriate message)
   -----------------------------------------
   ```

4. **def displayPlayers(self, franchise):** This function displays all the players associated with a franchise. The function reads the input franchise name from the file **promptsPS10.txt** where the search id is mentioned with the tag as shown below.

   ```
   listPlayers: RR
   listPlayers: SRH
   ```

   The output of this function should be appended into **outputPS10.txt** file. If a player is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

   ```
   --------Function displayPlayers --------
   Franchise name: RR
   List of Players:
   Ben Stokes
   Jaydev Unadkat
   Sanju Samson
   ```

Jofra Archer

G Krishnappa (if player is not found, display appropriate message)

----

5. **def franchiseBuddies(self, playerA, playerB):** Use one of the traversal techniques to find out if two players are related to each other through one common franchise. The function reads the input player names from the file **promptsPS10.txt** where the search id is mentioned with the tag as shown below.

   franchiseBuddies: Krunal Pandya : Kieron Pollard

   The output of this function should be appended into **outputPS10.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 franchiseBuddies --------

   Player A: Krunal Pandya

   Player B: Kieron Pollard

   Franchise Buddies: Yes, MI (if no, display appropriate message)

   ----

6. **def findPlayerConnect(self, playerA, playerB):** Use one of the traversal techniques to find out if two players A and B are related to each other through a common franchise member C as defined in the question above. The function reads the input player names from the file **promptsPS10.txt** where the search id is mentioned with the tag as shown below.

   playerConnect: Kedar Jadhav : Ishan Kishan

   Display the entire relation that links player A and player B. The output of this function should be appended into **outputPS10.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 findPlayerConnect --------

   Player A: Kedar Jadhav

   Player B: Ishan Kishan

   Related: Yes, Kedar Jadhav > MI > Dwayne Bravo > GL > Ishan Kishan

   (if no, display appropriate message)

```
-----------------------------------------
```

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

2. Sample file formats

Sample Input file

The input file **inputPS10.txt** contains names of the players and its associated franchises in one line. The name of the player and franchises should be separated by a slash (/).

Sample inputPS10.txt

```
RR / Ben Stokes / Jaydev Unadkat / Sanju Samson / Jofra Archer / G Krishnappa
KXIP / KL Rahul / R Ashwin / Andrew Tye / Marcus Stoinis / Glenn Maxwell
KKR / Chris Lynn / Mitchell Starc / Dinesh Karthik / Robin Uthappa / Manish Pandey / Chris Woakes
SRH / Manish Pandey / Rashid Khan / Shikhar Dhawan
MI / Krunal Pandya / Ishan Kishan  / G Krishnappa / Kieron Pollard
CSK / Kedar Jadhav / Dwayne Bravo / Karn Sharma
RCB / Chris Woakes / Yuzvendra Chahal / Umesh Yadav / KL Rahul / Mitchell Starc
DD / Glenn Maxwell / Sanju Samson / Kedar Jadhav / Karun Nair
GL / Dinesh Karthik / Andrew Tye / Dwayne Bravo / Ishan Kishan
RPS / Ben Stokes / Jaydev Unadkat
```

Sample promptsPS10.txt

```
findFranchise: Ben Stokes
findFranchise: Robin Uthappa
listPlayers: RR
listPlayers: SRH
franchiseBuddies: Krunal Pandya : Kieron Pollard
playerConnect: Kedar Jadhav : Ishan Kishan
```

Sample outputPS10.txt

```
--------Function displayAll--------
Total no. of franchises: 10
Total no. of players: 28
```

```
List of franchises:
RR
KXIP
KKR
SRH
MI
CSK
RCB
DD
GL
RPS

List of players:
Ben Stokes
Jaydev Unadkat
Sanju Samson
Jofra Archer
…….
----------------------------------------
--------Function displayFranchises --------
Player name: Ben Stokes
List of Franchises:
RR
RPS            (if franchise is not found display appropriate message)
----------------------------------------


……
Rest of the function outputs.
```

3. **Deliverables**

   a. **A1_PS10_IPL_[Group id] package folder** containing modules and package files for the entire program code and associated functions

   b. **inputPS10.txt** file used for testing

   c. **promptsPS10.txt** file used for testing

   d. **outputPS10.txt** file generated while testing

   e. **analysisPS10.txt** file containing the running time analysis for the program.

4. Instructions

a. It is compulsory to make use of the data structure/s mentioned in the problem statement.

b. Do not use inbuilt data structures available in Python. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.

c. It is compulsory to use Python 3.7 for implementation.

d. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full.

e. 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.

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

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

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

i. Run time analysis is provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

5. Deadline

a. The strict deadline for submission of the assignment is 18th June, 2020.

b. The deadline already accounts for extra days for the students to work on the assignment. No further extension of the deadline will be entertained.

c. Late submissions will not be evaluated.

6. How to submit

a. This is a group assignment.

b. Each group has to **make one submission** (only one, no resubmission) of solutions.

c. Each group should zip **all the deliverables** into one file and name the zipped file as below

“ASSIGNMENT1_[G1/G2/…].zip” and upload in CANVAS in respective location under ASSIGNMENT Tab.

d. Assignments submitted via means other than CANVAS **will not be graded**.

7. Evaluation

a. The assignment carries 12 Marks.

b. Grading will depend on

   a. Fully executable code with all functionality

   b. Well-structured and commented code

   c. Accuracy of the run time analysis

c. Every bug in the functionality will have negative marking.

d. Source code files which contain compilation errors will get at most 25% of the value of that question.

8. Readings

Section 6: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition)