National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Description of the Term Project for Data Structures and Algorithms

Course: Data Structures and Algorithms **Instructor:** Dr. Yasir Faheem

You are given a data set of research publications in the standard Bibtex format. Your task is to first parse the data of each of the articles from the given data set. In the second phase, you should declare the appropriate classes and data structures to be able to store the data of all the articles. In the last phase, you have to implement various functions to analyse the data the details of which are given in this document.

PHASE 1:

Write a parser that reads data from the standard Bibtex format file one article at a time. You should extract only data of those fields which are mentioned below in the **class** Article. For instance, an article might have been written by multiple authors. In such a case, the author field contains the data of all the authors. First and last name of an author are separated by a comma whereas names of two authors are separated by "and". Your parser should be able to extract the data accordingly. The parser should read the data for the fields mentioned in the below given classes.

PHASE 2:

Declare appropriate classes to store the data of each of the research articles. Each element of the class should have an appropriate data type. For instance, the title of an article is a string. It may be co-authored by multiple authors the count of which is not fixed. Hence, you need to maintain a list of authors in an appropriate data structure such as array list, a singly/circular/doubly linked list, a BST, an AVL tree or a Hash table. You should choose the most efficient data structure.

Class Article

Declare a class Article which stored the data of all the fields mentioned in the dataset for each article. For instance,

- List of Authors
  An article might have been authored by one or more authors. Use an appropriate data structure to store the list of authors. As explained in the class, you should maintain a list of the addresses of the objects of class author.
- Title
- Journal name
- Year
- Month
- Publisher
- Type (article)
- DOI
- ISSN
- Cited references (store only a list of the name of the journals (3rd field) which have cited this article and the year (2nd field)
- Number of citations

- Journal ISO

Hint: You may store the data of all the articles in a linked list, an AVL-tree or any other suitable data structure that offers optimal space and time complexity for various operations.

Class Author: Declare this class to store information related to an author such as his name, affiliation, the number of articles published by him. Moreover, you need to maintain a list of all the research papers published by an author. To do so, store a list of pointers to those objects of the class article which contain the data of those articles this author has published.

- Name (you may separate first, middle last names)
- Author Affiliation details.
- A list of pointers to the papers published by an author x (store address of those nodes of the class article in which this person x is an author.
- Total number of articles published by an author x (count of the above list).
- Total number of articles published by an author x per year (sorted year-wise).
- A list of the number of publications per year produced by this author.
- A list of the count of the number of co-authors of this author for each publication year.
-

Hint: You may store the data of all the authors in the most suitable data structure which may be an array-list, a linked list (singly, circular or doubly), a BST, an AVL-tree or a Hash table.

Once, you have stored the data of all the articles provided to you in the given data set, it is time to move to the third phase of this project which is explained next.

PHASE 3:

Now, we want to analyse the research publications and the trends of authorship from multiple aspects. For instance, we want to calculate the average number of authors in an article. We may want to check how many papers an author has published in different years. We may want to check the position of authorship of a particular researcher in his/her publications – how many papers he/she has published as first, second or third author etc. To do such analysis, you have to implement various functions which are explained next.

In this phase, your task is to implement functions that do the following:

1. Calculate the total number of articles published by an author x.
2. Calculates the number of publications of an author x per year

2013	2014	2015	2016	2017
2	3	5	1	10

3. Calculates the number of co-authors of an author x per publication sorted timewise.

PUBLICATIONS	1	2	3	4	5	6	7	8	9

NO OF CO-authors	2	3	4	3	7	2	2	2	2
Publication year	2007	2008	2010	2012	2012	2014	2018	2019	2019

4. Counts the numbers of papers published by an author x for each author position

Position	1	2	3	4	5	6
Number of papers	3	6	7	3	1	0

5. Sorted timewise, prints the name of the journal in which an author x has published his research work

Position	1	2	3	4	5	6
Journal	EEE SYSTEMS JOURNAL	IEEE Access	Elsevier JNCA	IEEE JSAC	IEEE Access	Springer Networks
Publication year	2012	2012	2014	2018	2019	2019

6. Store a weighted graph of all authors based on their authorship relationship. The weight of an edge between author vertices vi and vj will be the number of articles they have co-authored together.
    a. Implement a function that takes as input the name of an author **x** and the distance **d**. If the author is found, it should print all authors from the database who are at distance **d.** For instance, d=1 should print all co-authors, d=2 should print all co-authors of the co-authors of **x and so on.**
7. More functions to be added later.