DS2030 Data Structures and Algorithms for Data Science
Lab 0 Due on August 6, 11.59pm

Instructions

- You are to use Python as the programming language. Use may use Visual Studio Code (or any other editor you are comfortable with) as the IDE.
- You have to work individually for this lab.
- You are not allowed to share code with your classmates nor allowed to use code from the internet. You are encouraged engage in high level discussions with your classmates; however ensure to include their names in the report/code documentation. If you refer to any source on the Internet, include the corresponding citation in the report/code documentation. If we find that you have copied code from your classmate or from the Internet, you will get a straight fail grade in the course.
- The submission must be a zip file with the following naming convention - rollnumber.zip. The Python files should be contained in a folder named after the question number.
- Include appropriate comments to document the code. Include a `read me` file containing the instructions on for executing the code. The code should run on institute linux machines.
- Upload your submission to moodle by the due date and time. Do not email the submission to the instructor or the TA.

The goal of this lab is to get you familiar with Python programming language and environment. We will be using Python for all the labs. Hence it is important that you spend a couple of weeks getting familiar with Python. This lab consists of 3 programming questions.

1 Smart Imposition (5 points)

When I was young, my typing instructor teacher used to give imposition as punishment whenever I made a typing mistake. I always wondered if there was a smart way to complete the imposition to make it look natural by randomly inserting a fixed number of spelling mistakes. I could not use copy paste as it gives the same result every time and the outcome does not appear natural. Write a simple Python function that takes as input the sentence to be repeated, the number of repetitions and the number of errors and outputs the sentence the specified number of times with specified number of random errors. The program must number each sentence and it should make a certain number of random errors so that it looks like typed by a human. However there can be only one error in each repetition.

Input

I will always use OOP programming
5
3

Output

1 I will always use OOP programming
2 I will always use OaP programming
3 I will klways use OOP programming
4 I will always use OOP programmisg
5 I will always use OOP programming

2 Toppers in DS2030(5 points)

Write a class that maintains information about students securing the top ten ranks in DS2030. Student information consists of student id and rank. Use a doubly linked list to maintain this information that is sorted on basis of rank. You can refer to Code Fragments 3.16 and 3.17 in the textbook for the starter code. The student information and their percentile rankings can be obtained from the standard input.

You have to first create a class `StudentRecord` that provides the interface for a student record with the attributes student id and rank. Define and implement the constructors, `set` update and `get` access methods for this class. You will also implement comparators `isEqual`, `isGreater` and `isLesser` that will be helpful while sorting the list of student records based on rank.

The input to the program will be in the following format `1 id_val rank`. 1 indicates that the record has to be inserted (if possible) into the list. After every insert, the program should output the current list of top ten rankers. The program is exited with `0` as the input. Below is sample input and output of the program.

Input	Output
1 Ram 10 1 Rahul 20 0	Ram 10 Ram 10 Rahul 20

3 The Matrix (5 points)

In this problem, we will conduct an empirical (experimental) analysis of a widely used function - matrix multiplication. Write a Python program for multiplying two $n \times n$ randomly generated matrices. Given n, the size of a matrix, your program must perform 20 matrix multiplications, each time on two random matrices of size n. Run experiments to find the average time required for multiplication of the matrices for varying values of $n = 5, 10, 20, 30, 40, 50$. Use GNU plot to see the trend of growth rate in time with n. Analyze the result and make a small report.

4 References

1. M Goodrich, R Tamassia, and M. Goldwasser, "Data Structures and Algorithms in Python", 1^{st} edition, Wiley, 2013.