BT-3051 Data Structures and Algorithms for Biology
Assignment - 3

Submission : Since this is a coding based assignment, students need to submit their codes in a zipped folder. Your zip file should be named something like BTyyBxxx.zip, based on your roll number. This zip file must contain the codes used for each of the problems in separate **.ipynb** files with proper documentation.

Deadline : 16th September 2024, 23:59 hrs
Bonus marks and penalty will be applied as mentioned in the course plan.

Instructions:
 a. Total marks for this assignment - 30
 b. No restrictions on usage of functions, unless otherwise specified.
 c. Submissions will be evaluated only if they are provided in the required .ipynb format.

Problem 1: (7 marks)
A pharmaceutical company has many drugs in its production pipeline and will select a drug for large scale production based on two criteria: market demand (high demand = high priority) and production cost (high production cost = low priority). The criteria can change with time. The company's R&D team works on new drugs which can be added to the production pipeline. Choose a suitable data structure to represent the drugs in the production pipeline along with their market demand and production cost. Implement this as a class where you can add drugs to the production pipeline, change the priority of the company, and select a drug for production.

Simulate a company's one year production plan (Jan - Dec) based on the following conditions:
 1. At the start of the year, the company had 10 drugs on the production list. In January, the production criteria was market demand. A drug was selected based on its market demand score.
 2. Till April, the same drug was continued while 4 more drugs were added (1 in the beginning of each month) to the production list. In May, the company changed its criteria to production cost and an appropriate drug was selected from the list and was continued till the end of the year while new drugs were added to the list every month.
 3. Print the drug with the highest priority at the end of each month (not the one in production)

Note:
I. Generate random numbers between 0 and 100 for the drug properties (market demand and production cost).
II. You would have added 22 drugs in total at the end of the year.
III. You can use heapq.

Problem 2: (10 marks)
Two companies PiedPiper and Hooli are in their hiring process. They receive 10,000 resumes from the same set of applicants. Both the companies evaluate the resumes and score each of them between 0 to 1000. They sort the resumes from highest to lowest and add them to their database after which they call the top 100 applicants for the interview process. The person with the highest resume score will be the first one to be interviewed. Both companies use different strategies to sort the resumes: PiedPiper chooses a random resume and compares the others with that in a recursive fashion while Hooli divides the resumes into two halves to sort them recursively. Simulate this interview process and check which company goes through the interview process quickly and explain why.
Note: What is expected: Two sorting algorithms for resume sorting and a data structure to represent the database and how applicants are called for the interview. Generate random numbers between 0 to 1000 for resume scores.

Problem 3: (5 marks)
You are designing a digital note-taking system where you can add, update, retrieve and delete notes (heading and content). In this system, unique identifiers are generated from the note heading. Implement this using python.
Note: Python dictionaries are not allowed. You can use built-in hash functions.

Problem 4: (5 marks)
With a list of protein IDs, their names and function, construct a tree-like structure that you have studied which can efficiently 1) add proteins and 2) find proteins and their function through their IDs. Provide examples for adding and finding proteins.

Problem 5: (3 marks)
A large queue of people is waiting in a one-way passage to buy tickets for the newly released movie WOAT. Unfortunately, the show is canceled, and people need to go back in the reverse order they arrived. Use appropriate data structure to represent this; add 1000 people (identified by unique IDs 1 to 1000) to the queue and remove them when they are leaving.