

MTech in Data Science
IIT Palakkad
DS5003A : Data Engineering

0800-0850

Test 1 (12 Sep 2025)

Marks : 10

Instructions

1. Use Blue/Black ink (red ink and pencils are not allowed). If your answer is not legible, you will not get any marks for that.
2. Do not use any language specific syntax or library routines.

1. Consider generating an analytics for a search engine, where queries come in sequence over a long time. The analytics intend to show the trend of queries on various topics. Ideally, we would like to save all queries and produce a histogram at the end. But due to memory constraint, we can store only K queries in a reservoir. The task is to capture a reasonable representative of all queries in this smaller set of K queries. Let I is the total number of queries, where I is very large compared to K . One more constraint is that, as the queries coming in sequence over time, we have seen only J number of queries at any point of time, where $K \leq J \leq I$. We need to design an algorithm to create our representative reservoir such that the probability of a query to be part of the reservoir is $\frac{K}{J}$ after observing J queries. (3)

Here is a simple plan to develop an algorithm to implement the above strategy. You have to complete it and write the full algorithm and analyze its time complexity. Keep first K queries in the reservoir R . After observing $J - 1$ queries, you need to update the reservoir by replacing one query in the reservoir with the J th query randomly. Get a random number r between 1 and J , and if $r < K$, substitute query r in the reservoir R with the J th query in the stream.

Bonus 1 mark if you can show that the probability of any query in the stream of J queries to be part of the reservoir of size K is $\frac{K}{J}$.

2. Implement stack using queues. You have to implement the storage, and important operations associated to stacks, such as *push* and *pop* using the storage and basic operations of queue *enqueue* and *dequeue*. You can assume that operation *enqueue* inserts an element into a queue, and operation *dequeue* removes an element from a queue. Note that, in a stack, the element inserted last gets removed first, whereas in a queue the element inserted first gets removed first. (3)
3. Consider a set of drugs D . Write an algorithm to compare similarity s_{ij} between two drug molecules D_i and D_j . Scores should be between 0 and 1. Note that, each drug molecule D_i is a graph G_i , such that each node v_{ia} denotes some atom and each edge e_{iab} denotes bond between two atoms v_{ia} and v_{ib} . Each node v_{ia} and each edge e_{iab} have features ϕ_{ia} and ψ_{iab} of dimension d . Your algorithm should return 0 for two same molecules, and the score should monotonically increase based on the dissimilarity between the molecules, and near 1 for two very dissimilar molecules in the set D . (4)