

## DSA ASSIGNMENT 2

**Deadline: 11:59 PM, 06/04/2025**

---

**Total Marks : 60 Marks**

**Instructions:**

1. **Do any 3 of the following 4 questions.**
2. **Questions 1 and 2 are of theory while Questions 3 and 4 are programming.**
3. Assignments are to be attempted individually.
4. Submit the assignment as a single zipped folder (**A2\_<RollNumber>.zip**) containing a pdf file for all the theory questions (**must also contain the explanation of programming questions and screenshots of relevant test cases**) and “C” files (code\_<QuesNum>\_<RollNumber>.c) for programming questions, for example “code\_3\_1\_<RollNumber>.c” for programming question 3.1.
5. Please read the instructions given in the questions carefully. In case of any ambiguity, post your queries on Google Classroom at least 3 days before the deadline. **No TA will be responsible for responding to the queries after this.**
6. All the TAs will strictly follow the rubric provided to them. **No requests will be entertained related to the scoring strategy.**
7. **The use of generative tools (such as ChatGPT, Gemini, etc.) is strictly prohibited.** Failure to comply may result in severe consequences related to plagiarism.
8. **Extension and Penalty clause:**
  - Even a 1-minute late submission on Google Classroom will be considered late. Please turn in your submissions at least 5 minutes before the deadline to avoid any hassle.
  - Not explaining the answers properly will lead to zero marks.

## 1 Lists [20 marks]

Consider the following list of numbers:

**432, 235, 129, 543, 602, 764, 230, 987, 564, 345**

**1.1.** Using Radix Sort, show how these numbers would be sorted step by step and show your working. (15 marks)

**1.2.** After sorting, explain why Radix Sort is more efficient than Comparison-based Sorting algorithms for sorting large numbers of integers. Also, does Radix sort work for smaller integers? Why or why not? (5 marks)

## 2 Decision Trees [20 marks]

A warehouse manager is in charge of organizing inventory deliveries efficiently. The warehouse receives five different shipments (A, B, C, D, E) each day, but the order in which they are processed affects overall efficiency. The manager follows a decision-based approach to determine the order of processing based on the following rules:

1. Larger shipments (those with more items) should ideally be processed first.
2. If two shipments have the same size, perishable items should be processed first.
3. If both shipments have the same size and perishability, the decision is made alphabetically (A before B, B before C, etc.).

**2.1.** Draw a decision tree for the five shipments based on the manager's decision rules. Demonstrate how your decision tree would give an optimal arrival order for the shipments using a descriptive example (assume some size and perishability properties for each shipment). (15 marks)

**2.2.** At the end of each month, the warehouse receives double the load of shipments, adding up to a total of 10. If every order of the shipments arrival is valid, estimate how many leaves the decision tree must have to represent all possible processing orders using **Stirling's approximation**. Also compute the minimum height  $h$  of this end-of-month decision tree. Show your working. (5 marks).

*Hint: Estimate the number of permutations of 10 shipments as  $10!$  using Stirling's approximation. The minimum height of a decision tree with  $N$  possible orderings is approximately  $\lg(N!)$ .*

### **3 Stack/Queues (20 Marks)**

#### **3.1 Parcel Tracking System**

You are working on a parcel tracking system for a logistics company. The company handles priority-based parcel delivery, where urgent parcels are processed first. You must implement a stack using a linked list to simulate the parcel-handling process (12 Marks).

Each parcel contains the following information: Tracking ID (a unique integer), Sender Name (string, max 50 characters), Receiver Name (string, max 50 characters), Weight (kg) (float), Priority Level (1 = High, 2 = Medium, 3 = Low).

Write a C program that performs the following operations:

Add Parcel → Push a new parcel onto the stack. Parcels with higher priority (lower value) should always be on top;

Dispatch Parcel → Pop and display the most urgent parcel (top of the stack);

View Next Parcel → Display the details of the next parcel to be dispatched without removing it;

Display All Parcels → Print all parcels in order of dispatch (top to bottom), showing their details;

Count Parcels by Priority → Count and display the total number of parcels for each priority level;

Exit → Terminate the program.

This tracking system now requires a search feature that allows staff to quickly find the parcel by their ID (8 Marks); Search parcel by Tracking ID → Search for a parcel by their ID and display their details.

### **4 Hashing (20 Marks)**

#### **4.1 The Lost Treasure Map**

Captain Drake, a legendary pirate, was known for hiding his treasures on multiple islands. He created a map with encrypted coordinates, but to keep it safe from his rivals, he stored the coordinates in a hashed format. However, during a storm, part of the map was torn apart, leaving only some of the hashed values. (20 Marks)

To find the treasure, you have to:

- Insert the remaining coordinates into a hash table.
- Handle collisions using chaining.
- Search for a specific treasure coordinate by matching its hashed value.

- Display the map after insertion.

You need to write this program in **C**:

- Implement a hash table using chaining (linked lists).
- Define a hash function to convert coordinates into hash keys.
- Example: Insert the following treasure coordinates into the table:
  - (12, 45)
  - (34, 89)
  - (23, 78)
  - (12, 56)
  - (45, 89)
- Search for the coordinate (23, 78) and display whether it is found or not.
- Print the final hash table with all inserted coordinates.