SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Department of Artificial Intelligence and Data Science

23ITB201 – Data Structures and Algorithms

Unit 1- Question Bank

Part - A

1. What is Data Types?
2. What are the Primitive Data Types?
3. What is Data Structure?
4. How data structures are classified?
5. Differentiate linear and non-linear data structure.
6. What is an Abstract Data Type(ADT)?
7. Mention the features of ADT.
8. What is an Algorithm?
9. Why analysis of algorithm?
10. How performance of algorithm will be evaluated?
11. What is rate of growth?
12. Write the commonly used rate of growth.
13. What are the types of Analysis?
14. What is Asymptotic Notations?
15. Write about Big Oh notation.
16. Write about the Omega notation.
17. Write about the Theta notation.
18. What is recursion?
19. What is the need of recursion?
20. Write factorial program with recursion concept.
21. Write few examples for recursion concept.
22. What is back tracking?
23. How list can be implemented?
24. What is Linked List?
25. What are the List ADT operations?
26. What are the disadvantages of array implementation?
27. Draw the structure of singly Linked List.
28. How the doubly linked list can be represented?
29. How a node can be created in the C program?
30. When singly linked list can be represented as circular linkedlist?
31. When doubly linked list can be represented as circular linkedlist?.

32. List down the applications of List.
33. What are the advantages of linked list?
34. What are the operations performed in list?
35. What are the merits and demerits of array implementation of lists?
36. List out the different ways to implement the list?
37. Difference between singly and doubly linked list

Part – B

1. Explain the various operations of the list ADT with examples
2. Write about asymptotic notations
3. Write a program for array implementation of lists.(Include all the operations)
4. Discuss the steps in mathematical analysis for recursive algorithm. Do the same for finding the factorial of a number.
5. Explain the operations of singly linked lists (All operations)
6. Explain the operations of doubly linked lists (All operations)
7. Explain the operations of circularly linked lists (All operations)
8. Explain the steps involved in inserting and deleting the first element in singly anddoubly linked list.

Part – C

1. The system involves developing a music player application that allows users to manage a playlist represented by numbers. Users can add a song by appending a number to the playlist, delete a song by removing a number, search for a specific song by its number, and simulate playing all songs sequentially. Identify the application uses what kind of data structure for the playlist, where each song is represented by a unique number. Implement the above operations with suitable Data structure.
2. The system involves developing a student enrollment system using a list data structure to manage student records. The system allows administrators to add new students by appending their details (ID, name, course) to the list, delete students by removing them from the list based on their ID, search for specific students by ID to view their details, and display all enrolled students in the order they were added. The implementation is straightforward, using basic list operations such as `append` for adding and `remove` for deleting. Justify which list implementation is suitable for the above scenario and use the same to implement the above mentioned operations.
3. Compare the efficiency of Linear Search and Binary Search algorithms using asymptotic notations. Discuss their time complexities in terms of Big O, Big Omega, and Theta notations, and explain under what conditions each algorithm would be preferable based on the size of the dataset and whether the data is sorted.
4. Consider the Linear Search and Binary Search algorithms. For each algorithm, describe the following scenarios in terms of their time complexity:

i. **Best Case:** Explain the best-case time complexity for both Linear Search and Binary Search. Provide examples of input conditions that represent the best-case scenarios.
ii. **Worst Case:** Discuss the worst-case time complexity for both algorithms. Describe the conditions under which these worst-case scenarios occur.
iii. **Average Case:** Analyze the average-case time complexity for Linear Search and Binary Search. Illustrate how the time complexity is computed on average and what typical input conditions might be.

Compare and contrast the efficiency of both algorithms in these scenarios, and discuss the impact of input size and data sorting on their performance.