Sri Sai Vidya Vikas Shikshana Samithi ®

SAI VIDYA INSTITUTE OF TECHNOLOGY

Approved by AICTE, New Delhi, affiliated to VTU, Recognized by Govt. of Karnataka.
Accredited by NBA, New Delhi (CSE, ECE, ISE), NAAC – "A" Grade

DEPARTMENT OF CSE(Artificial Intelligence & Machine Learning)

RAJANUKUNTE, BENGALURU – 560 064, KARNATAKA
Phone: 080-28468191/96/97/98 * E-mail: hodaiml@saividya.ac.in * URL www.saividya.ac.in

Data Structures & Application (BCS304)

MODULE WISE QUESTION BANK

MODULE-1 (Introduction to Data Structures)

1. Define Data Structure. With a neat diagram, explain the classification of data structure with example**.** What are the basic operations that can be performed on data structure**?** **Jan/Feb 2021, (July/August 2022), Feb/Mar 2022, (Jan/Feb 2023), June/July 2023, Dec 2023/Jan 2024, June/July 2024, Model Paper-1(2023-24)**
2. Explain the dynamic memory allocation functions supported by c with syntax and examples. **(Jan/Feb 2021), Feb/Mar 2022, (July/August 2022), June/July 2023, June/July 2019, Model Paper-1(2023-24)**
3. Define Stack. Explain the different operation that can be performed on the stack with suitable 'C' function. **(Jan/Feb 2021), Feb/Mar 2022, (Jan/Feb 2023), (July/August 2022), June/July 2023, Dec 2023/Jan 2024, Model Paper-1(2023-24)**
4. Write functions in C for the following operations without using built in functions

   i) Compare two strings. ii) Concatenate two strings. iii) Reverse a string? **June/July 2019, (July/August 2022), Model Paper-1(2023-24)**
5. Convert the following infix to postfix expression using the stack (A+(B-C)*D). **(July/August 2022), (Jan/Feb 2023), June/July 2024, Dec 2023/Jan 2024**
6. Outline the algorithm for infix to postfix. Apply the same algorithm convert the following infix expression into postfix expression. a) ((A + (B - C) * D) ^ E + F). **Jan/Feb 2021, June/July 2023**
7. Difference Between Structure and Union. Show examples for both. **(July/August 2022), Dec 2023/Jan 2024**
8. What do you mean by pattern matching? Outline the Knuth Morris Pratt (KMP) algorithm and illustrate it to find the occurrences of the following pattern. **Jan/Feb 2021, Feb/Mar 2022, June/July 2023, Model Paper-(23-24)**

   P: ABCDABD

   S: ABC ABCDAB ABCDABCDABDE
9. Define sparse matrix. Explain the flowing matrix in triple form find its transpose. **Jan/Feb 2021, Jan/Feb 2023, June/July 2023**

$$A \begin{bmatrix} 3 & 0 & 0 & 0 \\ 5 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 8 \\ 0 & 0 & 9 & 0 \end{bmatrix}.$$

MODULE-2 (Queues & Linked List)

1. Develop a 'C' program to implement insertion, deletion and display operation of a circular queues. **(July/August 2022) ,Model Paper-1(2023-24)**Define queue. Write a C program for QInsert() and QDelete() operation on a circular queue ? **June/July 2019, , Jan/Feb 2021, Feb/Mar 2022, (Jan/Feb 2023)**
2. Define linked list. List and explain the different types of linked list with an example. **(Jan/Feb 2023), Dec 2023/Jan 2024**
3. Write the C function to add two polynomials. Show the linked representation of the below two polynomials and their addition using a circular singly linked list

   P1: 5x3 + 4x2 +7x + 3

   P2: 6x2 + 5 Output: add the above two polynomials and represent them using the linked list**. June/July 2023 Dec 2023/Jan2024, Model Paper-1(2023-24)**
4. Develop a 'C' function to implement the following using singly linked list **(Jan/Feb 2023), June/July 2023**
   a. Insert an Item at the front end in the SSL
   b. Insert an item at the rear end of the SSL
   c. Delete an item from the rear end in the SSL
   d. Delete an item from the front end of the SSL

MODULE-3 (Linked List & Trees)

1. Develop a 'C' function to implement the following using doubly linked list

   a. Insert an node at the front end & rear end b. Delete an node at the front end & rear end

   b. Concatenation of two DLL d. Search for the given key element

   June/July 2019, Feb/Mar 2022, (July/August 2022), June/July 2024, Model Paper-1(2023-24)
2. What is linked list? List and explain the different types of linked list with examples. **Dec.Jan 2020 Jan/Feb 2023,**
3. Define Binary tree with an example Write C recursive routine to traverse the given tree using inorder, preorder and postorder. **Jan/Feb 2021, Model Paper-1(2023-24)**
4. Define tree? with any six terminology i) Binary tree ii) Complete binary tree

   iii) Strictly binary tree iv) Almost complete binary tree.

   Feb/Mar 2022, (July/August 2022), Jan/Feb 2023, June/July 2024
5. What is the advantage of threaded binary tree over binary tree? Construct the threaded binary tree for 10,20,30,40 and 50. : **Dec. Jan 2020,(July/August 2022), Jan/Feb 2023, Dec 2023/Jan 2024, Model paper-1 (2023-2024)**
6. Write a C function for: i) inserting a node at the end of single linked list

   ii) Inserting a node at the end of single linked list **June/July 2023**

7. Define Sparse matrix. For the given sparse matrix, give the linked list representation:

June/July 2019, , Jan/Feb 2021, Feb/Mar 2022, (July/August 2022), June/July 2023, June/July 2024, Dec 2023/Jan 2024, Jan/Feb 2023, Model Paper-1(2023-24)

$$A = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

MODULE-4 (Trees & Graphs)

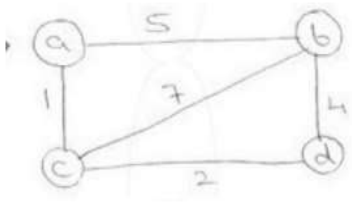
1. Construct a binary Search tree (BST) for the following data: 23, 67, 100, 2, 11, and 56, 90, 34, 99. Perform all traversals of the constructed binary tree. **(July/August 2022), June/July 2023, Jan/Feb 2023, June/July 2024, Model Paper-1(2023-24)**

2. Construct a BST by using the following in-order & post order traversals: **Dec. Jan 2020**

   Inorder : **BCAEDGHFI** Preorder : **ABCDEFGHI**

3. Write algorithm for DFS and BFS traversal for the given graph G = (V, E). **June/July 2023**

4. Define Graph. For the given graph, show the adjacency matrix and adjacency list representation of the graph. **Dec. Jan 2020, Jan/Feb 2023, Dec 2023/Jan 2024.**



MODULE-5 (Hashing & Priority Queues)

1. Define hashing. Explain different types hashing functions with examples. Discuss the properties of a good hash function. **, Jan/Feb 2021, Feb/Mar 2022, (July/August 2022), Jan/Feb 2023, June/July 2024, Dec 2023/Jan 2024, Model Paper-1(2023-24)**

2. What is collision? Explain the method of resolve collision with suitable algorithm of liner probing. Insert keys: 72, 27, 36, 24, 63, 81, 92, 101 into table [size=10].

   (July/August 2022), June/July 2019, June/July 2023, June/July 2024, Dec 2023/Jan 2024

3. Define priority queue. Explain Leftist trees with their two varieties.

4. Explain Static and Dynamic hashing in detail. **Feb/Mar 2022, June/July 2024, Dec 2023/Jan 2024, Model Paper-1(2023-24)**

Course Coordinator

Ms. Pooja A

Assistant Professor