School of Computer Engineering
KIIT deemed to be University
Laboratory Lesson Plan – Autumn 2023 (3rd Semester)

Discipline: B.Tech (All branches)

Course name and Code: Data Structure Laboratory (CS29001)

L-T-P-Cr : 0-0-2-1

Faculty Name: Dr. Chittaranjan Pradhan, Email: chittaranjanfcs@kiit.ac.in, Mob: 9861397865

Faculty Chamber: Faculty Room no. F-108, Block-A, Campus-15

Tentative Consulting hours: Tue (5PM-6PM), Wed (5PM-6PM), Thu (5PM-6PM)

Technical Assistants Names:

- TA 1(Mr. Deepak Ku. Das, deepak.das@kiit.ac.in, Mob. 9861586668)
- TA 2(Mrs. Pooja Allamsetti, pooja.allamsetti@kiit.ac.in, Mob. 9494493714)

Course Contents:

The course focuses on basic and essential topics in data structures and algorithms, including:

- Introduction (Structure, pointer, Dynamic Memory allocation and de-allocation)
- Arrays (following operations to be performed)
    - Insert, Delete, Linear search, Traversal, smallest and largest element, reverse the array element, sorting and searching an element in a dynamic array.
    - Sparse Matrix (3-tuple format, transpose, addition operation)
- Single Linked list
    - Traversal of the list.
    - Check if the list is empty.
    - Insert a node at the certain position (at beginning/end/any position).
    - Delete a node at the certain position (at beginning/end/any position).
    - Delete the node for a given key.
    - Count the total number of nodes.
    - Search for an element in the linked list.
    - sort the list in ascending order
    - Reverse the list
    - Polynomial representation (single variable)
- Double and Circular Linked list

    - Insert a node at the certain position (at beginning/end/any position).
    - Delete a node at the certain position (at beginning/end/any position).
    - Traversal of the list.
- Stacks (using Array and single linked list)
    - Push
    - Pop
    - check stack is empty or not
    - check stack is full or not
    - display stack elements
    - infix to postfix expression
- Linear Queues (using Array and single linked list)
    - Enqueue
    - Dequeue
    - IsEmpty
    - IsFull
    - Traverse
- Circular Queue (using array)
    - Enqueue
    - Dequeue
    - IsEmpty
    - IsFull
    - Traverse
- Deques (both Input restricted and output-restricted using static array)
    - Enqueue
    - Dequeue
    - IsEmpty
    - IsFull
- Priority Queue (using linked list)
    - Enqueue
    - Dequeue
    - Traverse
- Trees (Binary search tree using linked list)
    - preorder traversal
    - postorder traversal
    - inorder traversal
    - search an element
    - insert an element to the BST
    - display the largest element
    - display the smallest element
    - height of a node
    - count number of leaf nodes
- Graph (un-directed graph using Adjacency Matrix)
    - Display degree of each vertex
    - BFS traversal

    - DFS Traversal
- Sorting
    - Insertion sort
    - Selection sort
    - Merge sort
    - Quick sort
    - Heap sort
- Searching
    - Binary search

List of Experiments (Day wise):

Lab 1 (Introduction)

1.1. WAP to store one student's information (i.e. student's roll no, name, gender, marks etc) of an educational institute and display all the data, using structure (Access the structure members using pointer).

Input: 120045 Rahul Male 77
Output: Rollno.-120045, Name-Rahul, Gender-Male, Marks-77

1.2. WAP to store n book's data (title, author, publication, price etc.) using structures with dynamic memory allocation. Display all the books information of a particular author.

Input: Number of Books: 4

Book1: C, Balagurswamy, TMH. 220

Book 2: Database Systems, Silberschatz Korth, McGraw Hill, 750

Book3:Computer Architecture, Zaky Hammacher, TMH. 320

Book4: Compiler Design, Aho Ullman Shetty, Pearson, 430

Particular Author: Balaguruswamy

Output: C, Balagurswamy, TMH. 220

1.3. WAP to calculate the length of a string using a pointer.

Input: Orange
Output: The length of String: 6

1.4. WAP using dynamic memory allocation to find out the smallest and largest element stored in an array of n integers.

Input: Enter the value of n: 7

Enter the elements: 10, 20, 15, 25, 90, 45, 80

Output: Smallest: 10 and Largest: 90

1.5. WAP using pointer, dynamic memory allocation to search an element in an array of n numbers.

Input: Enter the value of n: 5
Enter the elements: 10 20 50 35 22
Enter the element to be search: 50
Output: 50 exists at index 2

1.6. WAP to read two numbers and compare the numbers using function call by address.

Input: Enter two numbers: 50 80
Output: 50 is smaller than 80

Input: Enter two numbers: 40 20
Output: 40 is greater than 20

Lab 2 (Array)

2.1. WAP to reverse the contents of a dynamic array of n elements.

Input: Enter the value of n: 5
Enter the elements: 10 20 35 45 30
Output: 30 45 35 20 10

2.2. WAP that takes an array as input and removes any duplicate elements, keeping only the first occurrence of each element.

Input: Enter the array elements: 1 2 2 3 3 4 4 5
Output: Array with distinct elements: 1 2 3 4 5

2.3. WAP a menu driven program to create an array and perform the following operations:

a. Traverse elements
b. Insert an element
c. Delete an element
d. Merging elements of two arrays

Input: Enter the value of n: 5

Enter the elements: 10 20 35 45 30
1. Traverse
2. Insert
3. Delete
4. Merge
5. Exit
Enter choice: 1

Output: 10 20 35 45 30

2.4. WAP to create a nXn matrix and perform the following:

e. Find the number of non zero elements
f. Sum and average of elements row wise
g. Sum and average of elements column wise

Input: Enter value of n: 3

Enter the elements: 1 2 3

4 2 0

6 5 4

Output:

Number of non zero elements: 8

Row1, Sum=6, average=2

Row 2, Sum=6, average=2

Row 3, Sum=15, average=5

Column 1, Sum=11, average=3

Column 2, Sum=9, average=3

Column 3, Sum=7, average=2

Lab 3 (Array)

3.1. WAP to perform transpose of a given sparse matrix in 3-tuple format.

Sample Input:

Enter sparse matrix in 3-tuple format
4 5 4
0 2 33
1 1 17
2 3 46
3 4 51

Sample Output:

Transpose of sparse matrix:
R C Element
5 4 4
1 1 17
2 0 33
3 2 46
4 3 51

3.2. WAP to perform addition of two given sparse matrix in 3–tuple format.

Sample Input:

Enter sparse matrix-1 in 3-tuple format
4 5 4
0 3 30
1 1 10
2 3 40
3 4 21
Enter sparse matrix-2 in 3-tuple format
4 5 5

0 2 65
1 1 12
2 3 45
3 3 71

Sample Output:

Resultant Matrix in 3-tuple format
4 5 5
0 2 65
0 3 30
1 1 22
2 3 85
3 3 71
3 4 21

3.3. WAP to represent the polynomial of single variable using 1-D array and perform the addition of two polynomial equations.

Input: Enter maximum degree of x: 2
Enter Polynomial-1 from lowest degree to highest degree : 4 2 3 (Hint: 4+2x+3x^2)
Enter Polynomial-2: 6 5 2

Output: Resultant Polynomial: 5x^2+7x^1+10x^0

Lab 4 (Linked List)

4.1. Write a menu driven program to perform the following operations in a single linked list (using self-referencing structure) by using suitable user defined functions for each case.

a. Create a list (Node Creation)
b. Display the list
c. Insert a node (at beginning/ at end / at any position)
d. Delete a node (at beginning/ at end / at any position)
e. Count the total number of nodes
f. Search for an element in the linked list

Input: Elements 10, 20, 30

Output:

1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit

Enter your choice: 1
Enter the data item: 10
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit

Enter your choice: 1
Enter the data item: 20
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit
Enter your choice: 1
Enter the data item: 30
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit

Enter your choice: 2
The Elements of the list are: 10 20 30
Do you want to Continue (Y/N): N

NB: Like wise other options can test

4.2. Write a menu driven program to perform the following operations in a double linked list (using self-referencing structure) by using suitable user defined functions for each case.

a. Create a list (Node Creation)
b. Display the list
c. Insert a node (at beginning/ at end / at any position)
d. Delete a node (at beginning/ at end / at any position)
e. Count the total number of nodes
f. Search for an element in the linked list

Input: Elements 10, 20, 30

Output:

1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit

Enter your choice: 1
Enter the data item: 10
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit

Enter your choice: 1
Enter the data item: 20
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit
Enter your choice: 1
Enter the data item: 30
Do you want to Continue (Y/N): Y
1.Create a Node
2.Display the list
3.Insert the element at beginning
4.Insert the element at the end
5.Insert the element at specific position
6.Delete the element from the beginning
7.Delete the element from the end
8.Delete the element from specific position
9.Count the total number of nodes
10.Search an element in the linked list
11.Exit
Enter your choice: 2
The Elements of the list are: 10 20 30
Do you want to Continue (Y/N): N

NB: Like wise other options can test

Lab 5 (Linked List)

5.1. Write a program to represent the polynomial equation of single variable using single linked list and perform the addition of two polynomial equations.

Input:

Polynomial-1:
Enter the Maximum power of x: 2
Enter the coefficient of degree 2: 4
Enter the coefficient of degree 1: 3

Enter the coefficient of degree 0:2
Polynomial-2:
Enter the Maximum power of x: 3
Enter the coefficient of degree 3: 5
Enter the coefficient of degree 2: 4
Enter the coefficient of degree 1:6
Enter the coefficient of degree 0:10

Output: Sum: 5x^3+8x^2+9x^1+12x^0.

5.2. Write a program to create a circular linked list and display the elements of the list.

Input:
Enter no. of nodes: 5
Enter info of node1: 30
Enter info of node2: 50
Enter info of node3: 40
Enter info of node4: 20
Enter info of node5: 70

Output: Circular linkedlist: 30 50 40 20 70

5.3. Write a program to represent the given sparse matrix using header single linked list and display it.

Input: Enter size of the sparse matrix: 4 5
Enter elements of sparse matrix: 0 0 33 0 0 0 17 0 0 0 0 0 0 46 0 0 0 0 0 51

Output: sparse matrix in 3-tuple format

4	5	4
0	2	33
1	1	17
2	3	46
3	4	51

Lab 6 (Stack)

6.1. Write a menu driven program to demonstrate the operations performed on stack using array and suitable user defined functions for each case.

Operations: PUSH, POP, DISPLAY, EXIT

Input: Size :3 , Elements: 3,5,7
Output:
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1

Enter the element to be inserted: 3
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1
Enter the element to be inserted: 5
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1
Enter the element to be inserted: 7
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 2
The deleted element is: 7
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 3
The Elements of the stack are: 5 3
Do you want to continue (Y/N): N

NB: Like wise other options can test

6.2. Write a menu driven program to demonstrate the operations performed on stack using linked list and suitable user defined functions for each case.

Operations: PUSH, POP, DISPLAY, EXIT

Input: Size :3 , Elements: 3,5,7
Output:
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1
Enter the element to be inserted : 3

Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1
Enter the element to be inserted: 5
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 1
Enter the element to be inserted: 7
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 2
The deleted element is: 7
Do you want to continue (Y/N): Y
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter Your Choice: 3
The Elements of the stack are: 5 3
Do you want to continue (Y/N): N

NB: Like wise other options can test

6.3. Write a Program to convert an infix expression into its equivalent postfix expression using suitable user defined function.

Input : a+b
Output:
Enter the infix expression
a+b
Equivalent Postfix Expression is:
ab+

6.4. Write a Program to convert an infix expression into its equivalent prefix expression using suitable user defined function.

Input : a+b
Output:
Enter the infix expression
a+b
Equivalent Postfix Expression is:
+ab

Lab 7 (Queue)

7.1. Write a menu driven program to implement the following queue operations :

a. Add a node in the queue
b. Delete a node in the queue
c. Traverse
d. Exit

Input: Size : 3, Elements: 4,5,6

Output:

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 1
Enter the Data 4

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 1

Enter the Data 5

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 1

Enter the Data 6

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 2

The Deleted element is: 4

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 3

No=5 no=6

a. Add
b. Delete
c. Traverse
d. Exit

Enter your Choice: 4

7.2. Write a menu driven program to implementation of dequeue operations using linked list. The operation are such as :-

1. Insert at rear end
2. Insert at front end
3. Delete at front end
4. Delete at rear end
5. Display
6. Quit

Input : Size 3, Elements: 11, 22,33

Output:

1. Insert at rear end
2. Insert at front end
3. Delete at front end
4. Delete at rear end
5. Display
6. Quit

Enter your Choice: 1

Enter the Element to be inserted: 11

1.Insert at rear end

2.Insert at front end

3.Delete at front end

4.Delete at rear end

5.Display

6. Quit

Enter your Choice: 1

Enter the Element to be inserted: 22

1.Insert at rear end

2.Insert at front end

3.Delete at front end

4.Delete at rear end

5.Display

6. Quit

Enter your Choice: 2

Enter the Element to be inserted: 33

1.Insert at rear end

2.Insert at front end

3.Delete at front end

4.Delete at rear end

5.Display

6. Quit

Enter your Choice: 5

The Elements in the Queue are: 33 11 22

1.Insert at rear end

2.Insert at front end

3.Delete at front end

4.Delete at rear end

5.Display

6. Quit

Enter your Choice: 6

NB: Like wise other options can test

7.3. Write a menu driven program to implement of Circular queue operations using array. The operation are such as :-

1. Insert
2. Delete
3. Display
4. Exit

Input : Array Size: 3, Elements: 5,6,7

Output:

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted: 5

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted: 6

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted: 7

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 2

Deleted Element is = 5

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 2

Deleted Element is = 6

Do you want to Continue (Y/N): N

NB: Like wise other options can test

7.4. Write a menu driven program to implement of Circular queue operations using linked list. The operation are such as :-

1. Insert
2. Delete
3. Display
4. Exit

Input : Array Size: 3, Elements: 5,6,7

Output:

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted: 5

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted: 6

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 1

Enter the Element to be inserted:7

Do you want to Continue (Y/N): Y

1.Insert

2.Delete

3.Display

4.Exit

Enter Your Choice: 2

Deleted Element is = 5

Do you want to Continue (Y/N): N

NB: Like wise other options can test

Lab 8 (Tree)

8.1. Write a menu driven program to perform the following operations in a binary search tree:

a. Insert in node in BST
b. Search a node in BST
c. Display (preorder) in BST
d. Display (inorder) in BST
e. Display (postorder) in BST
f. Find largest element
g. Find smallest element
h. Delete the element
i. Quit

Input: Enter your choice: 1

Enter items (Press 0 to quit):

82 77 90 346 35 0

Enter your choice: 2

Enter the number to be search: 35

Output: Number=: 35

NB: Likewise other options can test

Lab 9 (Graph)

9.1. WAP to create an un-directed graph using Adjacency Matrix Method and display the degree of each vertex.

Input:

Enter number of vertex: 5

Vertices 1 & 2 are Adjacent ? (Y/N) :y

Vertices 1 & 3 are Adjacent ? (Y/N) :n

Vertices 2 & 1 are Adjacent ? (Y/N) :y

Vertices 2 & 3 are Adjacent ? (Y/N) :y

Vertices 3 & 1 are Adjacent ? (Y/N) :n

Vertices 3 & 2 are Adjacent ? (Y/N) :y

Output:

Vertex	Degree
1	1
2	2
3	1

9.2. In addition to Q1, display DFS traversal sequence of the undirected graph.

Input:
Adjacency Matrix:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
Enter start vertex: 0

Output: Depth First Search: 0 1 4 2 3

9.3. In addition to Q1, display BFS traversal sequence of the undirected graph.

Input:

Enter number of vertex: 5
Enter Adjacency Matrix:
0 0 1 1 0
0 0 0 1 1
1 0 0 1 0
1 1 1 0 0
0 1 0 0 0

Enter start vertex: 0

Output: Breadth First Search: 0 2 3 1 4

Lab 10 (Sorting)

10.1. Write a program to sort array of n elements in ascending and descending order by bubble sort using function.

Input: Enter no. of elements: 5
Enter elements: 21 15 32 18 45

Output: Ascending order: 15 18 21 32 45

Descending order: 45 32 21 18 15

10.2. Write a program to sort array of elements in ascending and descending order by insertion sort using function.

Input: Enter no. of elements: 5
Enter elements: 22 55 33 88 44

Output: Ascending order: 22 33 44 55 88

Descending order: 88 55 44 33 22

10.3. Write a program to sort array of elements in ascending and descending order by selection sort using function.

Input: Enter no. of elements: 5
Enter elements: 11 55 22 66 33

Output: Ascending order: 11 55 22 33 66

Descending order: 66 55 33 22 11

10.4. Write a program to perform quick sort on array of elements to arrange it in ascending order using function.

Input: Enter no. of elements: 5

Enter elements: 70 50 40 20 10

Output: Ascending order: 10 20 40 50 70

10.5. Write a program to perform merge sort on array of elements to arrange it in ascending order using function.

Input: Enter no. of elements: 6

Enter elements: 10 30 40 60 20 80

Output: Ascending order: 10 20 30 40 60 80

Lab 11 (Sorting & Searching)

11.1. Write a program to perform heap sort on array of elements to arrange it in ascending order using function.

Input: Enter no. of elements: 8

Enter elements: 52 31 25 14 27 38 46 50

Output: 14 25 27 31 38 46 50 52

11.2. WAP to sort all even numbers in ascending order and all odd numbers in descending order.

Input: Enter number of elements: 6

Enter the elements: 1 3 5 4 7 10

Output: Sorted array: 7 5 3 1 4 10

11.3. WAP to search an element using linear search technique

Input: Enter no. of elements: 5
Enter numbers 1 2 3 4 5
Enter the no. to be search: 3
Output: 3 is found in position 3

11.4. WAP to search an element from list of elements using binary search technique

Input: Enter no. of elements: 6
Enter the numbers: 11 22 33 44 55 66
Enter the element to be search: 44
Output:44 is found at position 4

Grading Policies:

Continuous Evaluation components				
Sr#	**Area**	**Mark**	**#**	**Total**
1	**Internal Sending**			
1.1	Lab record evaluation	1	10	10
1.2	Quiz	5	2	10
1.3	Viva	1	10	10
1.4	Program Execution	2	10	20
1.5	Class Participation	1	10	10
Total				60
End semester evaluation				
2.1	Program Execution	10	2	20
2.2	Program Approach	10	1	10
2.3	Viva	10	1	10
Total				40

Practice Problem Sets:-

Reference Materials:-

Reference Book:

- Data Structures, Schaum's OutLines, Seymour Lipschutz, TATA McGraw Hill
- Data Structures using C by Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein. Pearson, 1st Edition
- Data Structures A Pseudocode Approach with C, 2nd Edition, Richard F. Gilberg, Behrouz Forouzan, CENGAGE Learning, India Edition
- Data Structures Using C, Second Edition, Reema Thereja, Oxford University Press
- Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education, 2nd Edition.

Reference Site:

- NPTEL - https://onlinecourses.nptel.ac.in/explorer
- Tutorials Point - https://www.tutorialspoint.com/data_structures_algorithms/
- Geeks for geeks - http://www.geeksforgeeks.org/