

Course Code: BCA-305

Course Title: Lab Course I (Data Structure)

Total Contact Hours: 48 hrs.
(60 Lectures)

Total Credits: 04 Total Marks: 100

Teaching Scheme: Theory-05 Lectures/ Week

Assignment	Title of the Experiment	Remark
1	Non-Recursive Sorting Techniques 1. Bubble Sort 2. Insertion Sort	
2	Recursive Sorting Techniques 1. Quick Sort 2. Merge Sort	
3	Searching Techniques 1. Linear search 2. Binary Search	
4	Linked List 1. Singly Linked List 2. Singly Circular Linked List 3. Doubly Linked List 4. Doubly Circular Linked List 5. Static Linked List (Using Array)	
5	Stack 1. Static Stack Implementation 2. Dynamic Stack Implementation	
6	Queue 1. Static Queue Implementation 2. Dynamic Queue Implementation	
7	Binary Search tree (Dynamic)	
8	Graph 1. Adjacency Matrix Representation 2. Adjacency List Representation	

Remarks and their meaning:

0: Not done [] -Assignment is not completed by the student.

1: Incomplete [] – All exercises are not completed by the student.

2: Late Complete [] – All exercises are completed but after deadline.

3: Needs Improvement [] – Assignments are completed but the concepts are not clear.

4: Complete [] – All compulsory exercises are completed, but not case study.

5: Well done [] – All compulsory exercises are completed including CASE STUDY

Lab Course I

Based on course contents of Data Structure using C programming is expected. To conduct practical's following is the list of sample assignments. Note that, these are only the sample assignments. Teacher may conduct practicals by preparing similar type of examples. **Sample Assignments for LAB I**

Assignment 1:

Non-recursive Sorting Techniques

1. Write a C program to accept n integers from user and sort them using bubble sort in ascending order. (Improve bubble sort algorithm is expected). Also print time complexity.
2. Write a C program to accept n integers from user and sort them using insertion sort in ascending order. Print time complexity.

Case Study

1. Banks often record transactions on an account, in order of the times of the transactions, but many people like to receive their bank statements with cheques listed in order by cheque number. People usually write(use) cheques in order by cheque number, and merchants usually cash them with reasonable dispatch. Thus few cheque numbers are usually out of order. Use an appropriate sorting algorithm for converting time of transaction ordering to cheque number ordering. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

Assignment Evaluation

0:NotDone[]

1:Incomplete []

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 2:

Recursive Sorting Techniques

1. Write a C program to accept n integers from user and sort them using Quick Sort in ascending order. Print time complexity.
2. Write a C program to accept n integers from user and sort them using Merge Sort in ascending order. Print time complexity.

Case Study

1. Write a program to choose any random text file from within your computer and sort all words present in the file in ascending order using any of the above sorting methods.

Assignment Evaluation

0:NotDone[]

1:Incomplete []

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 3:

Searching techniques

1. Write a C program to accept n integers from user and search any integer using linear search algorithm. Count the number of comparisons and display.

2. Write a C program to accept n integers from user and sort them using any algorithm of your choice and then modify linear search algorithm for sorted data. Count the number of comparisons.

Compare both versions of linear search algorithm and write your findings in assignment sheet.

3. Write a C program to accept n integers from user and sort them using any algorithm of your choice and use binary search to find any integer. Print number of comparisons.

Case Study

1. A spell checker is a program that looks at a document and compares each word in the document towards stored in a dictionary. If it finds words in the dictionary, it moves on to the next word, If it does not find the word, it reports the user about the misspelled (possibly) word. Formulate the problem and write a C Program to solve the problem by using appropriate data structures and algorithms.

Assignment Evaluation

0:NotDone[]

1:Incomplete[]

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 4:

Linked List

1. Write a menu driven C program to implement singly linked list of integers with following operations:
 1. Append
 2. Insert
 3. Delete
 4. Search
 5. Display
2. Write a menu driven C program to implement singly circular linked list of integers with following operations:
 1. Append
 2. Insert
 3. Delete
 4. Search
 5. Display
3. Write a menu driven C program to implement doubly linked list of integers with following operations:
 1. Append
 2. Insert
 3. Delete
 4. Search
 5. Display

4. Write a menu driven C program to implement doubly circular linked list of integers with following operations:
 1. Append
 2. Insert
 3. Delete
 4. Search
 5. Display

5. Write a menu driven program to implement static linear linked list using array. Provide all menu options as mentioned above.

Case Study

1. A file manager identifies each file using an inode-no. A directory is also a file identified by an inode-no. Directory is a list of file search represented by an inode-no . The list has the header containing the inode –no of the parent that is the directory itself .A separate table indexed by inode-no maintains attributes of files including the name of the file. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. The union operation of two disjoint sets takes two disjoint sets S1 and S2 and returns ad is joint sets consisting of all the elements of S1and S2 and the original sets S1and S2 are destroyed by the union operation. How to implement union in O(1) time using a suitable list data structure for representing a set?

OR

3. Linked list can be used to store single variable polynomial of any number of terms. Write a program to accept two polynomials with different terms to calculate their sum and multiplication.

Assignment Evaluation

0:NotDone[]	1:Incomplete []	2:LateComplete[]
3:NeedsImprovement[]	4:Complete[]	5:WellDone[]

Assignment 5:

Stack

1. Write a C program to implement static stack of integers with following operations: Push(),Pop(), Empty(), StackTop()/Peek()
2. Write a C program to implement dynamic stack of integers with the same operations as mentioned above.

Case Study

1. A postfix expression of the form $ab+cd-*ab/$ is to be evaluated after accepting the values of a, b, c and d . The values should be accepted only once and the same value is to be used for repeated occurrence of same symbol in the expression. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. An Infix expression of the form $a*(b+c)*((d-a)/b)$ need to be converted to postfix form using usual precedence's of operators. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

Assignment Evaluation

0:NotDone[]

1:Incomplete []

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 6:

Queue

1. Write a C program to implement static queue of integers with following operations: Push(),Pop(), Empty(), StackTop()/Peek()
2. Write a C program to implement dynamic queue of integers with the same operations as mentioned above.

Case Study

1. Suppose that we are selling the services of a machine. Each user pays a fixed amount peruse. However the time needed by each user is different. We wish to maximize the returns from this machine under the assumption that the machine is not to be kept idle unless no user is available. Whenever the machine becomes available, the user with the smallest time requirement is selected. When a new user requests the machine, he has to wait if there are pending requests. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. A factory has many machines and many jobs that require processing on some of the machines. Each job has a job card which contains all the attributes of a job including the machines on which it need to be processed and the processing time. For simplicity let us assume that there are ten machines numbered 0,1,2...9, each job is processed for fixed amount of time and the job card number is coded in such away that the machine requirements can be accessed. For example if job card number is 5438,it means job requires processing on machines 5,4,3 and 8.The job can be processed simultaneously on these machines. However each machine can process jobs one at a time and takes up the new job when it becomes idle. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

Assignment Evaluation

0:NotDone[]

1:Incomplete []

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 7

Binary Tree

1. Write a C program to implement binary search tree of integers with following operations:
 1. Inserting a new element
 2. Searching an element
 3. Inorder traversal
 4. Preorder traversal
 5. Postorder traversal
 6. Level-by-level traversal
 7. Counting leaf nodes
 8. Counting non-leaf nodes
 9. Creating mirror of the tree
 10. Checking whether two trees are mirrors

Case Study

1. Consider the database of books maintained in a library system. When a user wants to check whether a particular book is available, a search operation is called for. If the book is available and is issued to the user, a delete operation can be performed to remove this book from the set of available books. When the user returns the book, it can be inserted back into the set of available books. It is essential that we are able to support the above mentioned operations as efficiently as possible as since these operations are performed quite frequently. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

Assignment Evaluation

0:NotDone[]

1:Incomplete []

2:LateComplete[]

3:NeedsImprovement[]

4:Complete[]

5:WellDone[]

Assignment 8:

Graph

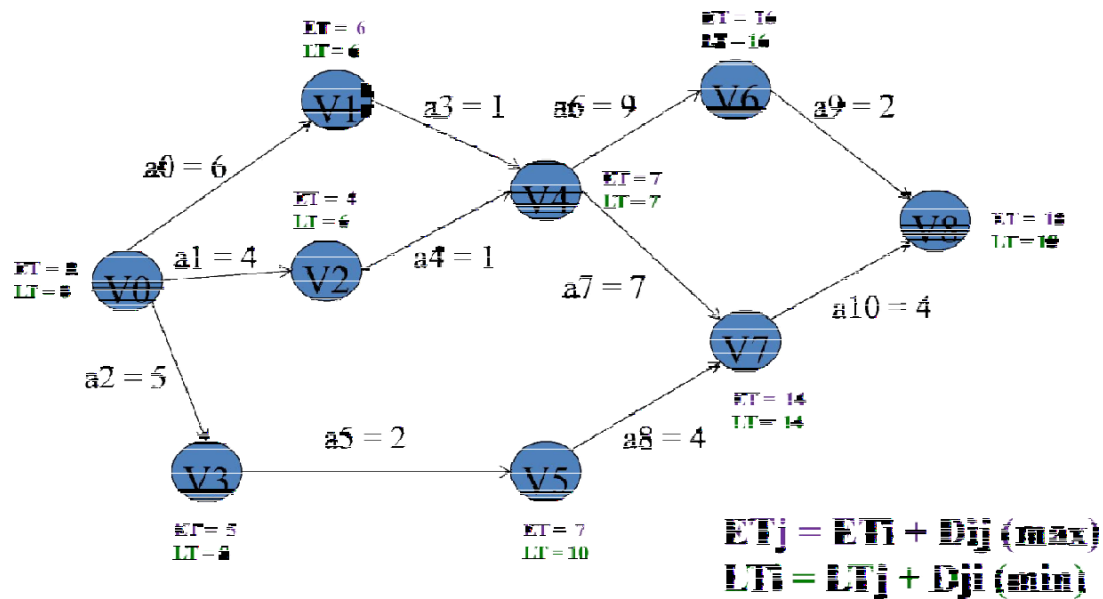
1. Write C program that accepts the vertices and edges for a graph and stores it as an adjacency matrix. Implement function to print In-Degree, Out-Degree and to display that adjacency matrix.
2. Write a C program that accepts the vertices and edges for a graph and stores it as an adjacency list. Implement function to print Out-Degree of any vertex i .
3. Write a C program that accepts the graph as an adjacency matrix and converts it to adjacency list representation. Write a function to display the graph in adjacency list form.

Case Study

An activity on edge network is a directed network in which each edge models an activity with the weight of the edge representing the time needed to complete the activity.

The vertex models significant events. An event represented by a vertex only happens when all of the activities modeled by edge leading into it have been completed. No activity can start until the event modeled by the vertex at its tail has occurred. The events modeled by the vertices are often project milestones such as “specification accepted by user”.

Normally, we include a start vertex with in-degree 0 to model the event “Project Begins”.



Write a program to find critical path for above network.

Assignment Evaluation

0: Not Done []

1: Incomplete [] 2: Late Complete [] 3: Needs Improvement [] 4: Complete [] 5: Well Done []