**SESSION SEPTEMBER 2022**

**PROGRAM BACHLER OF COMPUTER APPLICATIONS (BCA)**

**SEMESTER II**

**COURSE CODE &amp; NAME DCA1202 - Data Structure and Algorithm**

**Assignment Set –1**

**1 What are the characteristics and Building Blocks of an Algorithm? And what are Control Mechanism and Control structures?**

**Ans:** Next, we enumerate the basic actions and corresponding instructions used in a computer system based on Von Neumann architecture. We may recall that an instruction is a notation for an action and a sequence of instructions defines a program whereas a sequence of actions constitutes a process.

**Basic Actions & Instructions**

i) Assignment of a value

Its Half solved only

Buy Complete from our online store

<https://smuassignment.in/online-store/>

MUJ Fully solved assignment available for**session Jul/Aug 2022, Exam Sep 2022.**

Lowest price guarantee with quality.

Charges**INR 200 only per assignment.**For more information you can get via mail or Whats app also

Mail id is aapkieducation@gmail.com

Our website www.smuassignment.in

After mail, we will reply you instant or maximum

1 hour.

Otherwise you can also contact on our

whatsapp no 8791490301.

**2. A) What are Binary trees? How many types of Binary trees are there, discuss?**

**Ans: Definition**: A binary tree is a special case of tree where no node of a tree can have a degree of more than two. Therefore, a binary tree is a set of zero or more nodes T such that:

i) there is a specially designated node called the root of the tree

ii) the remaining nodes are partitioned into two disjointed sets, T1 and T2, each of which is a binary tree. T1 is called the left sub tree and T2 is

**B) Discuss the linked storage representation of binary tree.**

**Ans: Linked storage representation of binary tree** The most popular way to present a binary tree is Linked storage representation. Binary tree consists of nodes which can have at most two child, linked representation of such tree will be stored in the form shown Fig, where LPTR and RPTR denote the

**3. Explain the algorithms of Bubble sort and Merge sort.**

**Ans: Bubble sort Bubble sort** is a straightforward and simplistic method of sorting data that is used very commonly. The algorithm starts at the beginning of the data set. It compares the first two elements, and if the first is greater than the second, then it swaps them. It continues doing this for each pair of adjacent elements to the end of the data set. It then starts again with the first two

**Assignment Set – 2**

**4 A) What is dynamic memory storage and how is link list stored in memory? Write the algorithm for traversal of a singly link list.**

**Ans: Dynamic representation**

The efficient way of representing a linked list is using the free pool of storage. In this method, there is a memory bank (which is nothing but a collection of free memory spaces) and a memory manager (a program, in fact). During the creation of a linked list, whenever a node is required the request

**B) What are the different types of link list. Write an algorithm to create circular list.**

## Ans: Types of Linked Lists:-

**5. Write the Algorithm to find the maximum and minimum items in a set of n element. Also explain the working of the algorithm.**

**Ans:**

1. Write down an array of one element. Figure out how you would find the maximum and minimum of

**6. A) What is Stack? Discuss the Array implementation of a stack along with push () and pop () algorithms.**

**Ans: Stack**

A stack is a data structures in which insertion and deletion of items are made at the one end, called the top of the stack.

We have two basic operations in stack they are push and pop.

**Push Operation:**

**B) What is Queue? Discuss the Array implementation of a queue along with enqueue () and dequeue () algorithms.**

**Ans: Queue**

A queue is a linear list of elements in which deletions can take place only at one end, called the front and insertions can take place only at the other end, called the rear as referred in the figure. The terms “front” and “rear” are used in describing a linear list only when it is implemented as a