**Internal Assignment**

 **Set- 1st**

**1(a). What is a linked list? Discuss the algorithms for insertion and deletion of values in the end of a linked list?**

**Ans.** A linked list is asequence of data structures, which are connected together via links. Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array.

**Inserting Elements to a Linked List:**

**Insert a Node at the end of a Linked list**

Take a look at the linked Its Half solved only

Buy Complete from our online store

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

MUJ Fully solved assignment available for**session Jul/Aug 2021,**

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.

Contact no is +91 87-55555-879

**1(b). Define stacks and what are the applications of Stack?**

**Ans.** A Stack is a widely used linear data structure in modern computers in which insertions and deletions of an element can occur only at one end, i.e., top of the Stack. It is used in all those applications in which data must be stored and retrieved in the last.

An everyday analogy of a stack data structure is a stack of books on a desk, Stack of plates, table tennis, Stack of bootless,

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

Ans. A binary tree is an important type of tree structure which occurs very often. It is characterized by the fact that any node can have at most two branches, i.e., there is no node with degree greater than two. For binary trees we distinguish between the subtree on the left and on the right,

**2(b). What is a List Structure? Explain Adjacency list and Incidence list?**

**Ans.** A list structure consists of a set of lists and an optional lock table of exclusive locks, which you can use to serialize the use of lists, list entries, or other resources in the list structure. Each list is pointed to by a header and can contain a number of list entries. A list entry consists of list entry controls and

**3. Discuss the types of directed graphs and matrix representation of Digraph?**

**Ans.** The different types of directed graphs are discussed below:

**Definition:** A directed graph (or) a digraph D consists of a non-empty set V (the elements of V are normally denoted by v1, v2, …) and a set E (the elements of E are normally denoted by e1, e2, ….) and a mapping  that maps every element of E onto an ordered pair (vi, vj) of elements from V.

 **Set - II**

**4. a. Explain the algorithms of Bubble sort and Merge sort?**

**Ans. Bubble Sort**: I would

**4(b). What are the characteristics Building Blocks of an Algorithm?**

**Ans.** The process of executing the individual statements in a given order is called control flow.

The control can be executed in three ways

1. sequence

2. selection

**5(a). How is the Efficiency of an Algorithm measured?**

**Ans. Efficiency of an Algorithm measured:-**

* **Back to the Basics:** Put simply, algorithms are a way of solving a class of problems. They are a ‘finite sequence’ made to work with any programming language. Think of these as instructions, just like a recipe! It includes step-by-step directions that can be applied to return the desired

**6. Discuss about All Pair Shortest Paths and Travelling Salesman Problem?**

**Ans. All Pair Shortest Paths:** Let G = (V, E) be a directed graph with n vertices and let cost be a cost adjacency matrix for G such that cost (i, i) = 0, 1  i n. Then cost (i,j) is the length (or cost) of edge <i,j> if <i,j> E(G) and cost (i,j) =  if i  j and < i, j > E(G). The all pairs shortest path problem is to determine a matrix A such that A (i,j) is the length of a shortest path from i to j.

Let us examine a shortest i to j path in Gi  j. This path originates at vertex I and goes through some intermediate vertices (possibly none) and terminates at vertex j. We can assume that this path contains no cycles for if there is a cycle, and then this can be deleted without increasing the path length. If