SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

COIMBATORE – 35

DEPARTMENT OF COMPUTER SIENCE AND ENGINEERING (UG & PG)

First Year, 2nd Semester

2 Marks Question and Answer

Subject Code & Name: 19ITT102 & Data Structures and Algorithms

Prepared by : Mrs.G.DEVI /AP / CSE

UNIT – II

Linear Data Structure

1. Define Data Structures

Data Structures is defined as the way of organizing all data items that consider not

only the elements stored but also stores the relationship between the elements.

2. Define Linked Lists

Linked list consists of a series of structures, which are not necessarily adjacent in memory.

Each structure contains the element and a pointer to a structure containing its successor. We call

this theNext Pointer. The last cell'sNext pointer points to NULL.

3. State the different types of linked lists

The different types of linked list include singly linked list, doubly linked list and

circular linked list.

4. List the basic operations carried out in a linked list

The basic operations carried out in a linked list include:

• Creation of a list

• Insertion of a node

• Deletion of a node

• Modification of a node

• Traversal of the list

5. List out the advantages of using a linked list

• It is not necessary to specify the number of elements in a linked list during its declaration

• Linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list

• Insertions and deletions at any place in a list can be handled easily and efficiently

• A linked list does not waste any memory space

6. List out the disadvantages of using a linked list

• Searching a particular element in a list is difficult and time consuming

• A linked list will use more storage space than an array to store the same number of elements

7. List out the applications of a linked list

Some of the important applications of linked lists are manipulation of polynomials, sparse matrices, stacks and queues.

8. State the difference between arrays and linked lists

Arrays	Linked Lists
Size of an array is fixed	Size of a list is variable
It is necessary to specify the number of elements during declaration.	It is not necessary to specify the number of elements during declaration
Insertions and deletions are somewhat difficult	

Insertions and deletions are carried

out easily

It occupies less memory than a

linked list for the same number of

elements

It occupies more memory

9. Define an Abstract Data Type (ADT)

An abstract data type is a set of operations. ADTs are mathematical abstractions; nowhere

in an ADT's definition is there any mention of how the set of operations is implemented. Objects

such as lists, sets and graphs, along with their operations can be viewed as abstract data types.

10. What are the advantages of modularity?

• It is much easier to debug small routines than large routines

• It is easier for several people to work on a modular program simultaneously

• A well-written modular program places certain dependencies in only one

routine, making changes easier

11. What are the objectives of studying data structures?

• To identify and create useful mathematical entities and operations to determine what

classes of problems can be solved using these entities and operations

• To determine the representation of these abstract entities and to implement the

abstract operations on these concrete representation

12. What are the types of queues?

• Linear Queues – The queue has two ends, the front end and the rear end. The rear end is

where we insert elements and front end is where we delete elements. We can traverse in a

linear queue in only one direction ie) from front to rear.

• Circular Queues – Another form of linear queue in which the last position is connected

to the first position of the list. The circular queue is similar to linear queue has two ends,

the front end and the rear end. The rear end is where we insert elements and front end is where we delete elements. We can traverse in a circular queue in only one direction ie) from front to rear.

• Double-Ended-Queue – Another form of queue in which insertions and deletions are made at both the front and rear ends of the queue.

13. List the applications of stacks

• Towers of Hanoi

• Reversing a string

• Balanced parenthesis

• Recursion using stack

• Evaluation of arithmetic expressions

14. List the applications of queues

• Jobs submitted to printer

• Real life line

• Calls to large companies

• Access to limited resources in Universities

• Accessing files from file server

4. Define a stack

Stack is an ordered collection of elements in which insertions and deletions are restricted to one end. The end from which elements are added and/or removed is referred to as top of the stack. Stacks are also referred as piles, push-down lists and last-in-first-out (LIFO) lists.

15. List out the basic operations that can be performed on a stack

The basic operations that can be performed on a stack are

• Push operation

• Pop operation

• Peek operation

• Empty check

• Fully occupied check

16. State the different ways of representing expressions

The different ways of representing expressions are

• Infix Notation

• Prefix Notation

• Postfix Notation

17. State the rules to be followed during infix to postfix conversions

• Fully parenthesize the expression starting from left to right. During parenthesizing, the operators having higher precedence are first parenthesized

• Move the operators one by one to their right, such that each operator replaces their corresponding right parenthesis

• The part of the expression, which has been converted into postfix is to be treated as single operand

18. Mention the advantages of representing stacks using linked lists than arrays

• It is not necessary to specify the number of elements to be stored in a stack during its declaration, since memory is allocated dynamically at run time when an element is added to the stack

• Insertions and deletions can be handled easily and efficiently

• Linked list representation of stacks can grow and shrink in size without wasting memory space, depending upon the insertion and deletion that occurs in the list

• Multiple stacks can be represented efficiently using a chain for each stack

19. Mention the advantages of representing stacks using linked lists than arrays

• It is not necessary to specify the number of elements to be stored in a stack during its declaration, since memory is allocated dynamically at run time when an element is added to the stack

• Insertions and deletions can be handled easily and efficiently

• Linked list representation of stacks can grow and shrink in size without wasting

memory space, depending upon the insertion and deletion that occurs in the list

• Multiple stacks can be represented efficiently using a chain for each stack

20. Define a queue

Queue is an ordered collection of elements in which insertions are restricted to one end

called the rear end and deletions are restricted to other end called the front end. Queues are also

referred as First-In-First-Out (FIFO) Lists.

21. Define a priority queue

Priority queue is a collection of elements, each containing a key referred as the priority for

that element. Elements can be inserted in any order (i.e., of alternating priority), but are arranged

in order of their priority value in the queue. The elements are deleted from the queue in the order

of their priority (i.e., the elements with the highest priority is deleted first). The elements with the

same priority are given equal importance and processed accordingly.

22. State the difference between queues and linked lists

The difference between queues and linked lists is that insertions and deletions may occur

anywhere in the linked list, but in queues insertions can be made only in the rear end and

deletions can be made only in the front end.

23. Define a Deque

Deque (Double-Ended Queue) is another form of a queue in which insertions and deletions

are made at both the front and rear ends of the queue. There are two variations of a deque,

namely, input restricted deque and output restricted deque. The input

restricted deque allows insertion at one end (it can be either front or rear) only. The

output restricted deque allows deletion at one end (it can be either front or rear)

24. What is the need for Priority queue?

In a multiuser environment, the operating system scheduler must decide which of the several processes to run only for a fixed period of time. One algorithm uses queue. Jobs are initially placed at the end of the queue. The scheduler will repeatedly take the first job on the queue, run it until either it finishes or its time limit is up and place it at the end of the queue if it does not finish. This strategy is not appropriate, because very short jobs will soon to take a long time because of the wait involved in the run.

Generally, it is important that short jobs finish as fast as possible, so these jobs should have precedence over jobs that have already been running. Further more, some jobs that are not short are still very important and should have precedence. This particular application seems to require a special kind of queue, known as priority queue. Priority queue is also called as Heap or Binary Heap.