DEPARTMENT OF INFORMATION TECHNOLOGY

SUBJECTNAME: DATASTRUCTURES

SUBJECT CODE: U20EST356

UNIT I

TWO MARK QUESTIONS WITH ANSWER

1. **Define Data Structure.**

A data structure is logical way of storing and organizing data either in computer's memory or on the disk storage so that it can be used efficiently.

2. **Define the classification of data structures.**

There are two types of data structures:

- Linear Data Structure
- Non-linear data structure

3. **Define linear data structure?**

Linear data structures are data structures having a linear relationship between its adjacent elements. Eg. Linked List.

4. **Define Non Linear data structure?**

Non Linear data structures are data structures don't have a linear relationship between its adjacent elements, but have the hierarchical relationship.

Eg. Graph,Tree.

5. **List some common examples of data structure**.

Some common examples of data structure includes arrays, linked lists, queues, stacks, binary trees, and hash tables

6. **List out the various application areas of data structures**.

- Compiler design
- Operating system
- DBMS
- Simulation
- Artificial Intelligence
- Graphics

7. **Define ADT and give an example (or) Write short notes on ADT**.

The Abstract Data Type (ADT) is a set of operations .They is mathematical abstractions and defines how the set of operations are implemented. Eg: Integer ,real number

8. **What are the various operations done under list ADT?**

- Print list
- Insert
- Make empty
- Remove
- Find Next
- Find Previous
- Find k^{th} element from some position

9. **What are the various implementation of List ADT?**

- Array Implementation
- Linked List Implementation
- Cursor Implementation

10. **What do you mean by array implementation of List?**

- Array is collection of similar data elements in contiguous memory location.
- Insertion and deletion are more expensive and takes more time
- Find and print list operation takes constant time.
- Memory space is wasted

11. **What is Linked List? (or) What does the Linked List contain?**

In a linked list, each element is called a node. Every node in the list points to the next node in the list. Therefore, in a linked list every node contains two types of information:

- The data stored in the node
- A pointer or link to the next node in the list

12. **List the basic operations carried out in a linked list?**
- Creation of list
- Insertion of list
- Deletion of list
- Modification of list
- Traversal of List

13. **What are the various types of linked lists?**

The various types of linked lists are:
- singly linked lists
- circularly linked lists
- doubly-linked lists

14. Difference between array and Linked List.

Array	Linked list
Size of any array is fixed	Size of 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
Insertion and deletions are difficult and Costly	Insertions and deletions are done in less time
It occupies less memory	It occupies more memory

15. Why you need a data structure?

A data structure helps you to understand the relationship of one data element with the other and organize it within the memory. Sometimes the organization might be simple and can be very clearly visioned. Eg) List of names of months in a year – Linear Data Structure, List of historical places in the world - Non-Linear Data Structure. A data structure helps you to analyze the data, store it and organize it in a logical and mathematical manner.

16. Difference between Static and Dynamic data structures

a) Static data structure: The data structure is created using static memory allocation is called static data structure or fixed data structure. In static data structure, the no of data items are known in advance. E.g., Array

b) Dynamic data structure: The data structure is created using dynamic memory allocation is called dynamic data structure or variable size data structures. In Dynamic data structure, the no of data items are not known in advance. E.g., linked list.

17. What are the two basic types of Data structures?

- Primitive Data structure
- Non Primitive Data Structure
- Linear Data structure
- Non linear Data structure

17. **Difference between linear and non-linear data structures**

Linear data structures, data elements are organized sequentially and therefore they are easy to implement in the computer's memory. **Eg:** Stacks, Queues, Linked Lists, etc.

Non-Linear data structure is that if one element can be connected to more than two adjacent element then it is known as non-linear data structure. **Eg:** Trees, Graphs, etc.

18. **What is an Algorithm**

- Algorithm is a step by step procedure to solve a problem.
- Developing a program for an algorithm first we should select data structures.

19. **Mention some of the properties of an Algorithm**.

Finiteness: algorithm stops after few steps

Definiteness: each step in algorithm has some particular work to do

Input: algorithm accepts 0 or more i/p's

Output: algorithm produces atleast 1 output

Effectiveness: at short time algorithm can process all the steps

20. **What are the types of algorithm efficiencies?**
    a. Time Complexity
    b. Space Complexity

21. **List some algorithm design technique.**
    a. Divide and conquer algorithm design
    b. Brute force
    c. Dynamic programming
    d. Greedy algorithm

22. **What do asymptotic notation means?**

Asymptotic notations are terminology that is introduced to enable us to make meaningful statements about the time and space complexity of an algorithm. The different notations are

a. Big – Oh notation (O)
b. Omega notation(Ω)
c. Theta notation(Θ).

23. **What are the basic asymptotic efficiency classes?**

The various basic efficiency classes are

i. Constant :1
ii. Logarithmic : log n
iii. Linear : n
iv. N-log-n : nlog n
v. Quadratic : n^2
vi. Cubic : n^3
vii. Exponential : 2^n
viii. Factorial :n!

24. **What is best-case, worst-case and average-case efficiency?**

- Best case efficiency: The best-case efficiency of an algorithm is its efficiency for the best-case input of size n, which is an input or inputs for which the algorithm runs the fastest among all possible inputs of that size.
- Worst-case efficiency: The worst-case efficiency of an algorithm is its efficiency for the worst-case input of size n, which is an input or inputs of size n for which the algorithm runs the longest among all possible inputs of that size.
- Average-case efficiency: The average case efficiency of an algorithm is its efficiency for an average case input of size n. It provides information about an algorithm behavior on a "typical" or "random" input.

25. **Define Array. List out its types?**

- If we want to store a group of similar data together in one place. Array is mainly used for storing elements of same data type. The data types are float, integer, and characters hould be mentioned.
- . Types of Array
  - One Dimensional Arrays
  - Two Dimensional Arrays

- Multidimensional Arrays
- Pointer Arrays.

26. **What is searching?**

Searching refers to operation of finding the location of a given item in a collection of items. The search is said to be successful if ITEM does appears in the collection of items and unsuccessful otherwise.

27. **List out the types of searching techniques.**

Two searching Techniques:

1. Linear Search
2. Binary Search

28. **What is linear search?**

Linear search is the method for finding a particular value in a list, by checking everyone of its element, one at a time and in sequence, until the desired one is found.

Time Complexity is O(n).

29. **What is binary search?**

Binary search is a remarkably efficient algorithm for searching in a sorted array. It works by comparing a search key K with the arrays middle element A[m]. if they match the algorithm stops; otherwise the same operation is repeated recursively for the first half of the array if K < A[m] and the second half if K > A[m]. Time Complexity is O(log n).

30. **Give few examples for data structures.**

Stack, Queue, Circular Queue, Linked List, Doubly Linked list, Circular Linked list, Tree, Binary Search Tree, Graph, Map and others like searching and sorting.

31. **List down the limitation of Array implementation.**

- the dimension of an array is determined the moment the array is created, and cannot be changed later on;
- the array occupies an amount of memory that is proportional to its size, independently of the number of elements that are actually of interest;
- if we want to keep the elements of the collection ordered, and insert a new value in its correct position, or remove it, then, for each such operation we may need to move many elements (on the average, half of the elements of the array); this is very inefficient.

32. **State the difference between array and pointers.**

- An array name represents a fixed memory address that cannot be changed. You can index it, but you can't change the address it refers to.
- A pointer variable is a variable that contains the address of something. Because it's a variable, you can change the address stored in that variable to point to something else.

33. **What is Time Complexity?**

Time Complexity is a way to represent the amount of time required by the program to run till its completion. It's generally a good practice to try to keep the time required minimum, so that our algorithm completes it's execution in the minimum time possible

34. **What is Space Complexity?**

Its the amount of memory space required by the algorithm, during the course of its execution. Space complexity must be taken seriously for multi-user systems and in situations where limited memory is available

35. **Difference between Linear data structure and Non-Linear Data structure**

Linear Data Structure	Non-Linear Data Structure

A type of Data structure that arrange the data items in an orderly manner where the elements are attached adjacently	A type of data structure that arranges data in sorted order ,creating a relationship among the data elements
Memory utilization in inefficient	Memory utilization in efficient
Single level	Multi- level
Easier to implement	Difficult to implement
Ex : array ,linked list, queue , stack	Ex: Tree , Graph

36. **List the basic operations carried out in an Array?**

Array is a container which can hold a fixed number of items and these items should be of the same type. Most of the data structures make use of arrays to implement their algorithms. Following are the important terms to understand the concept of Array.

Element − Each item stored in an array is called an element.

Index − Each location of an element in an array has a numerical index, which is used to identify the element.

Following are the basic operations supported by an array.

Traverse − print all the array elements one by one.

Insertion − Adds an element at the given index.

Deletion − Deletes an element at the given index.

Search − Searches an element using the given index or by the value.

Update − Updates an element at the given index.