Summary of the Lecture 1 and Lecture 2:

1. What is Data Structure?

Data Structure is data collection, management, and storage format for easy access & modification. In other words, it is a collection of data values and the relationship between them and functions & operations that can be applied to these values.

2. What are the types of data structures?

Data Structures are mainly of the following two types:

1. **Linear Data structure** - A data structure is linear when its elements are in sequential order or are a linear list and every element is placed in a non-hierarchical manner.

   Few examples - Arrays, Stack, Strings, Queue, and Linked List.

2. **Non-Linear Data structure** - When the elements of the data structure are not arranged in a sequential manner and instead the element is attached with two or more elements in a non-linear arrangement.

   Few examples - Graphs & trees.

3. In what most areas the data structures are used?

Data structures are part of the following branch of computer science:

1. AI
2. DBMS - Database management systems
3. Numerical Analysis
4. Simulation
5. Statistical analysis package
6. Compiler design
7. Operating system
8. Compiler design

4. The linked list is an example of a linear or non-linear data structure?

A linked list can be a subset of both linear and nonlinear, it depends on its usage & application. If it is used for access strategies, then it is a part of linear structures but when used for data storage, it can be called a non-linear data structure.

3. What is the linked list data structure?

It is simply a sequence of data objects where elements of data structures are not stored in adjacent memory locations. Each element is inside such a setting is an individual object called a node. Every node has 2 items, one is a data field and the other is a reference to the next node. Head is the entry point in each linked list. When the list is empty, the head is a null reference and the last node has a reference to null.

6. What is the algorithm?

The algorithm is a defined procedure, which demonstrates the actions or steps to be executed in a certain order to get the desired outputs.

7. What different operations can be performed on any data structure?

- **Sorting** - To arrange the data in ascending or descending order, when there is numerical data or in dictionary order when the data is in alphanumeric form.
- **Searching** - To search the location of the data item if it is available in the given data items collection.
- **Deleting** - To delete no longer required data from the given data items collection.
- **Insertion** - To add new data in the given collection of data items.
- **Transversal** - To access each data item exactly as it is to help with further processing.

8. What is the difference between an Array and Linked list?

- Linked lists are dynamic size, whereas arrays have a fixed size.
- The linked list allows easy insertion & deletion of data, whereas performing insertion & deletion in arrays is quite expensive.
- You cannot have random access to the linked list.
- Arrays have great cache locality which adds to the great performance.
- In each element of the linked list, extra memory space is needed for the pointer.

9. What is a doubly-linked list? Mention some examples to it.

A doubly linked list is a complex type of linked list, further in which each node has two links, the first link connects to the next node pinned in the sequence, whereas the second link is connected to the previous node. It allows for easy transversal between the data elements in different directions.

Here are the few examples of doubly linked list:

1. The music or video playlist with next & previous buttons for navigation.
2. The browser cache having back-forward visited pages.
3. The undo & redo functions over a web browser that helps to get to the previous page.

11. What are dynamic data structures? Mention a few.

Dynamic data structures are a collection of data in memory that expands and shrinks as per the program. And allows the programmer to regulate memory utilization as per requirement.

The few examples of dynamic data structures are stack, queue, heap, linked list, and array.

12. Why is algorithm analysis required?

A problem can be solved in too many ways. To treat computational problems, the algorithm analysis provides an estimation of the required resources and also helps to determine the time and space these resources would require to execute.