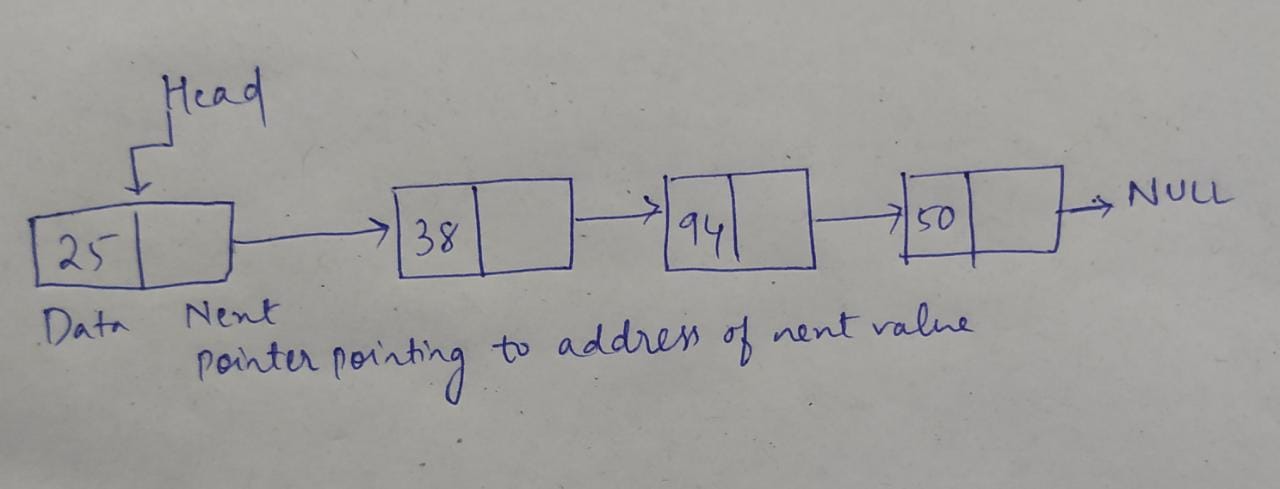
**Data Structures and Algorithms Assignment 3**

1. What is a linked list?

ANS:

A linked list is a data structure made of a chain of nodes in which each node contains a value and a pointer to the next node in the chain. The elements are not stored at contiguous memory locations.



2. What are the different forms of linked lists?

ANS:

The different forms of linked lists are:

a) Singly Linked List

b) Doubly Linked List

c) Circular Linked List

d) Doubly Circular linked list

e) Header Linked List

3. What is a linked list's purpose?

ANS:

a) Implementation of stacks and queues

b) Implementation of graphs: Adjacency list representation of graphs is most popular which is uses linked list to store adjacent vertices.

c) Dynamic memory allocation: We use linked list of free blocks.

d) Maintaining directory of names

e) Performing arithmetic operations on long integers

4. What are the advantages of linked lists over arrays?

ANS:

The advantages of linked lists over arrays are:

a) No Memory Wastage:

As the size of a linked list can grow or shrink at runtime, so there is no memory wastage. Only the required memory is allocated.

In arrays, we have to first initialize it with a size which we may or may not be required, hence there is wastage of memory.

b) Insertion and Deletion Operation:

In a Linked List, insertion and deletion operations are quite easy, as there is no need to shift every element after insertion or deletion. Only the address present in the pointers needs to be updated.

While in an array, we have to shift elements. Suppose we have an array that is sorted, and now we need to insert some element in the array in a sorted way.

c) Dynamic Data Structure:

Linked List being a dynamic data structure can shrink and grow at the runtime by deallocating or allocating memory, so there is no need for an initial size in linked list.

5. What is the purpose of a circular linked list?

ANS:

a) Any node can be a starting point. We can traverse the whole list by starting from any point. We just need to stop when the first visited node is visited again.

b) Useful for implementation of queue. Unlike this implementation, we don’t need to maintain two pointers for front and rear if we use circular linked list. We can maintain a pointer to the last inserted node and front can always be obtained as next of last.

c) Circular lists are useful in applications to repeatedly go around the list. For example, when multiple applications are running on a PC, it is common for the operating system to put the running applications on a list and then to cycle through them, giving each of them a slice of time to execute, and then making them wait while the CPU is given to another application. It is convenient for the operating system to use a circular list so that when it reaches the end of the list it can cycle around to the front of the list.

6. How will you explain Circular Linked List?

ANS:

Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list.

