Set -1

1. An algorithm is a finite set of well-defined instructions for accomplishing a specific task. Its importance in efficiency is that it helps to ensure that algo uses minimal computational resources (such as time and memory to solve a problem and is more scalable.

2. Row-Major representation :-
Address(A[i][j]) = BA + (i x No. of cols + j) x size
Column-Major :-
Address (A[i][j]) = BA + (j x No. of rows + i) x size

3. For a 2D array A[i][j] with base address B, R rows, C cols, size S, the Address is :-
A[i][j] = B + (i x C + j) x S

So, General Formula :-

$$\text{Address}(A[d_1][d_2][d_3]\ldots[d_n]) = B + (d_1 \times \prod_{k=2}^{n} D_k + d_2 \times \prod_{k=3}^{n} D_k + \ldots d_{n-1} \times D_n + d_n) \times$$

4)

Insertion Sort	Selection Sort	
$O(n)$	$O(n^2)$	Best Case
$O(n^2)$	$O(n^2)$	Average case
$O(n^2)$	$O(n^2)$	Worst Case

5) Merge Sort Algorithm :-
It is a divide-and-conquer algo. It works by recursively dividing the array into two halves until each sub-array consists of only one element. Then, it merges the sorted sub-arrays to produce a single sorted array.

6) Already done

7) Insertion in a SLL :-

It is a process that can be performed at three posn : at the beginning, at the end or at a specific posn. To insert a new node, you create a new node, set its data, and then adjust the ptrs.

Example :
Insert 4 after the node 2 in a list
1 → 2 → 3 → null
Create a newNode, new, with data 4.

Traverse it into the list and find the node with data 2, let's call it curr.

Set new.next to curr.next
Set curr.next to new

The list becomes
1 → 2 → 4 → 3 → null

Q) Already done

Q) A sparse matrix is a matrix in which the no. of zero elements is much larger than the no. of non-zero elements.

example -

0	0	3	0	=
2	0	0	0	
3	0	0	0	

Advantages :-

i) Saves memory by storing only the non-zero el
ii) Faster computation for operations that only involv non-zero elements.

Limitations :-

i) Accessing an element is slower than in a standard 2-D array.
ii) Requires more complex code to manage the representation.
iii) The overhead of storing row and column indices can be significant if the matrix is not sufficiently sparse.

Set - 5

1) Algorithm complexity is the measure of the resources required to run an algorithm. The different types of complexities are :-
1) Time complexity - Measure of time.
2) Space complexity - Measure of space required

2) Already done

3) Already done

4) Searching is a process of finding a specific element within a collection of data. Types :-
1) Linear Searching
2) Binary Searching
3) Hashing

5) Already done

6) Already done

7) Already done

8) Recursion and iterative approaches are two ways to solve problems, but they differ significantly in their use of the call stack.

	Iterative	Recursive
Stack Usage	Does not use it for repeated function calls, so it avoids stack overflow issues. Uses a loop instead.	Each recursive call adds a new frame to the call stack. This can lead to a stack overflow error if the recursion depth is too large.
Memory code	Uses less memory can be complex code	Uses more memory. Often results in more concise and elegant code

9) Merge Sort Algo - It is a sorting algorithm that follows the divide and conquer paradigm

Approach :-

Divide - The problem of sorting a list is divided into two smaller subproblems of sorting two halves of the list.

Conquer - The two subproblems are solved recursively by calling merge sort on each half.

Combine - The sorted halves are then merged back together to form a single sorted list

Merge sort has a time C. of $O(n \log n)$ in all cases, making it highly efficient and predictable for sorting large datasets, unlike quick sort which has a worst-case complexity of $O(n^2)$.