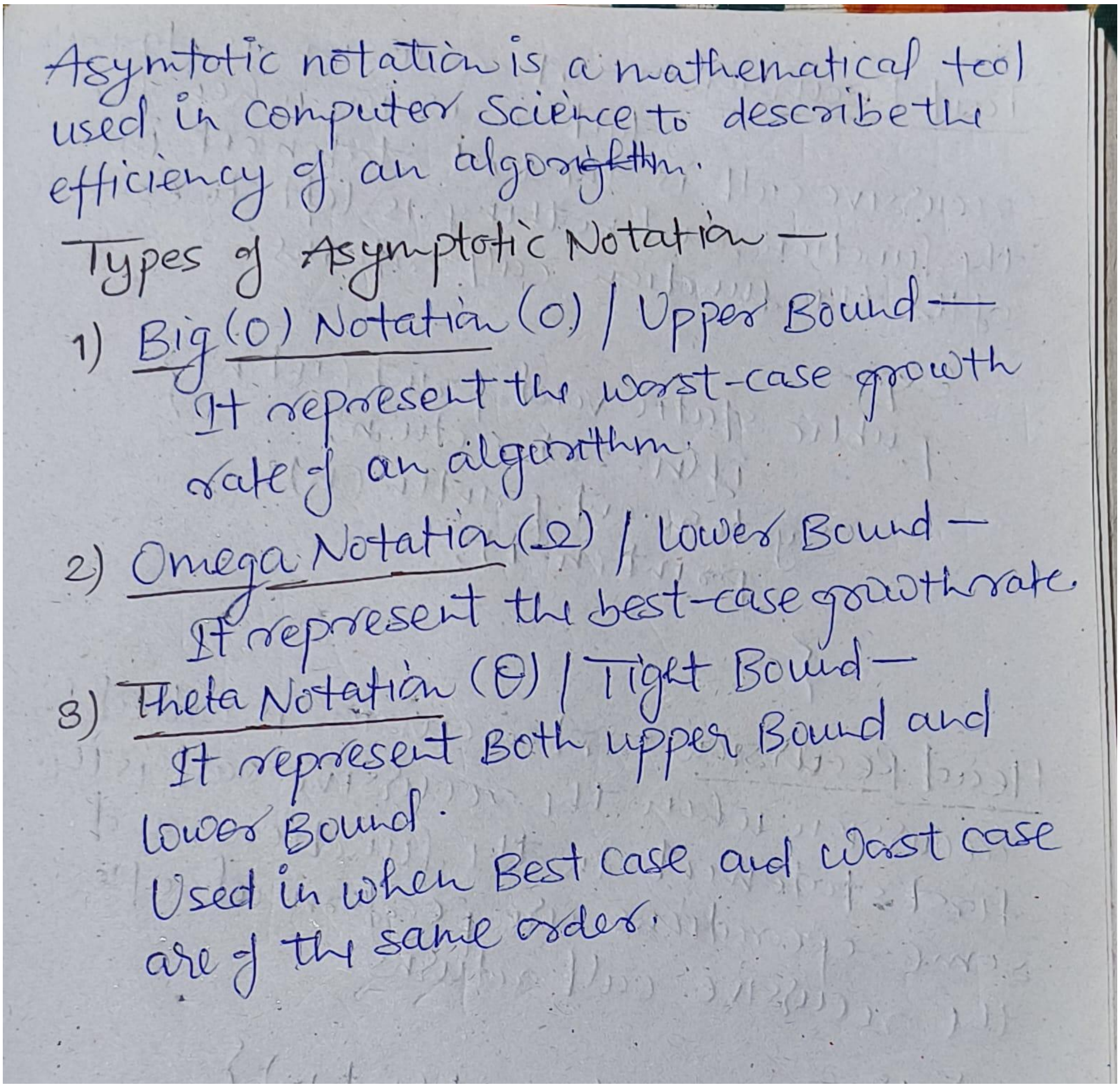
1. Define asymptotic notation and explain its importance in analyzing algorithm efficiency. (Low, K1, 3 marks)



Asymtotic notation is a mathematical tool used in Computer Science to describe the efficiency of an algorigthm.

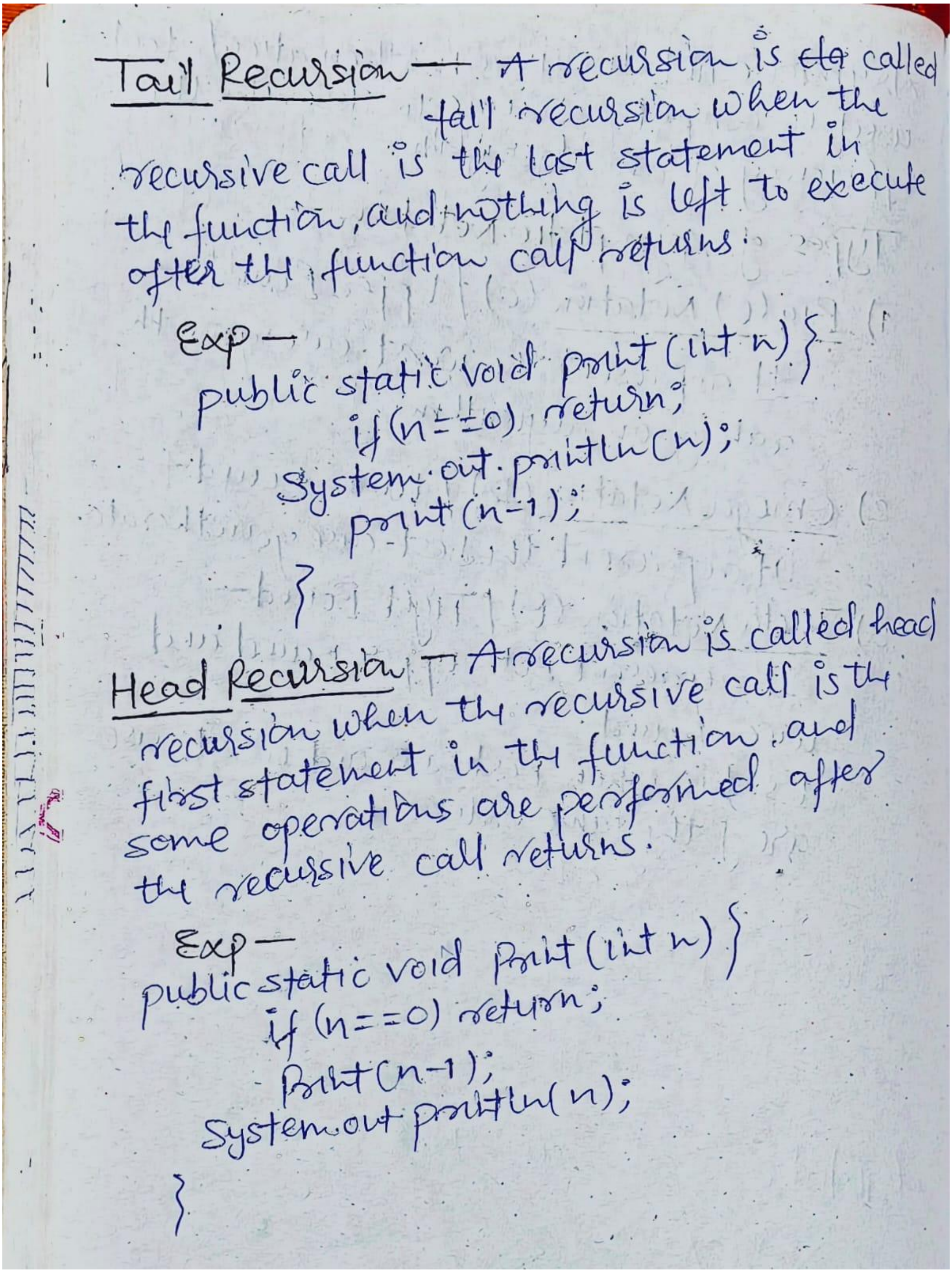
Types of Asymptotic Notation —

1) Big (O) Notation (O) / Upper Bound —
It represent the worst-case growth rate of an algorithm.

2) Omega Notation (Ω) / Lower Bound —
It represent the best-case growth rate.

3) Theta Notation (Θ) / Tight Bound —
It represent Both upper Bound and lower Bound.
Used in when Best case and Worst case are of the same order.

2. Explain the difference between tail recursion and head recursion with examples. (Low, K1, 3 marks)



Tail Recursion — A recursion is ~~ela~~ called tail recursion when the recursive call is the last statement in the function, and nothing is left to execute after the function call returns.

Exp —

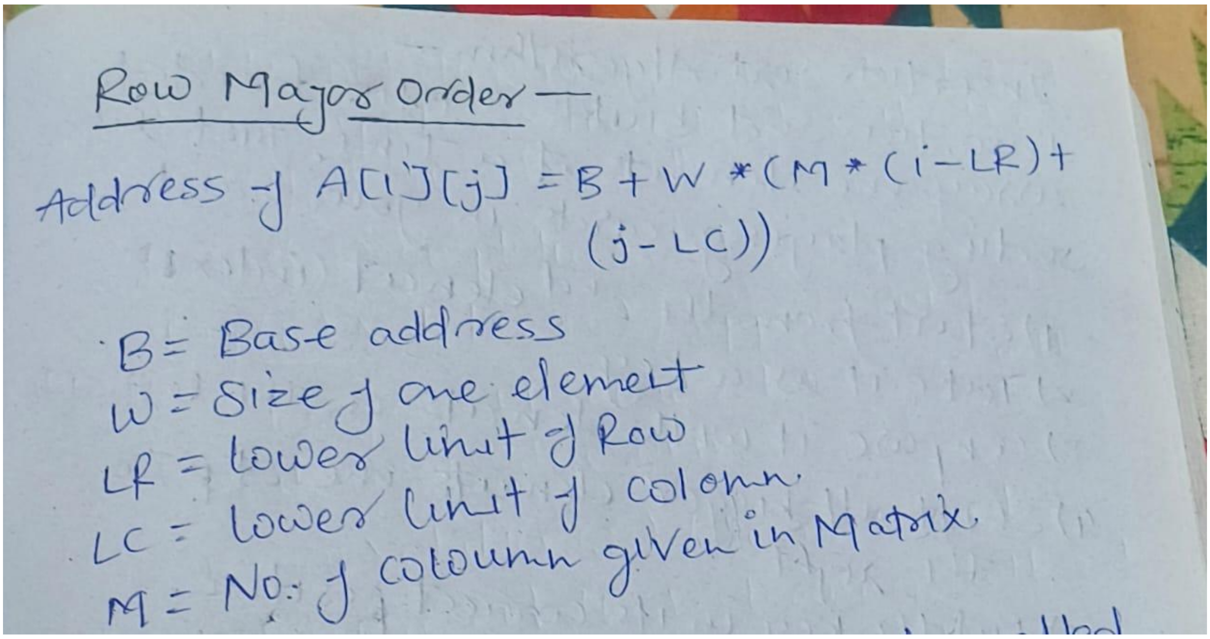
```
public static void print (int n) {
        if (n==0) return;
    System.out.println(n);
        print (n-1);
    }
```

Head Recursion — A recursion is called head recursion when the recursive call is the first statement in the function, and some operations are performed after the recursive call returns.

Exp —

```
public static void Print (int n) {
        if (n==0) return;
        Print(n-1);
    System.out.println(n);
}
```

3. Derive the index formula for accessing elements in a 2-D array stored in row-major order.



Row Major Order —

Address of $A[i][j] = B + W * (M * (i - LR) + (j - LC))$

B = Base address

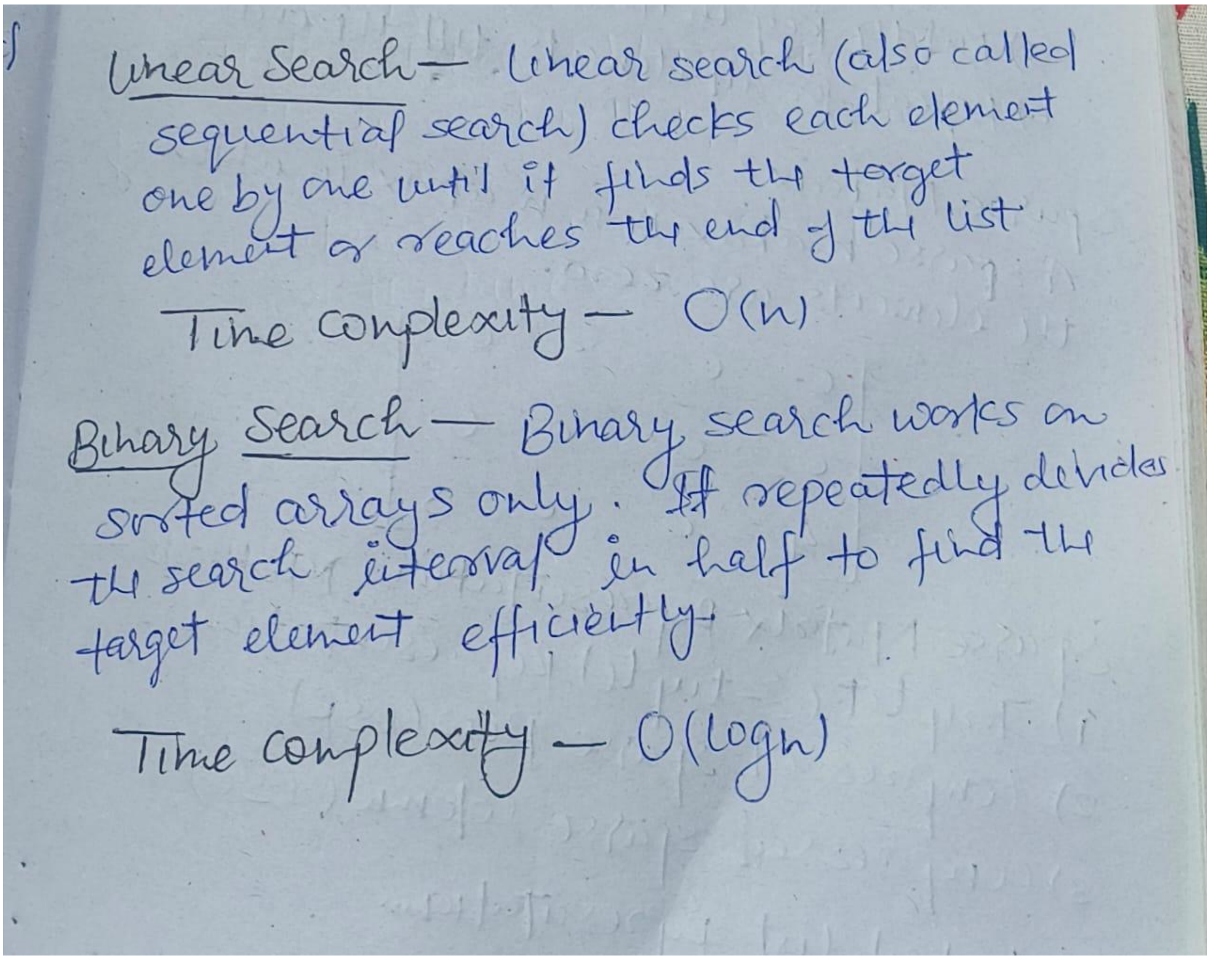
W = Size of one element

LR = lower limit of Row

LC = lower limit of colomn.

M = No. of coloumn given in Matrix.

4. Explain the difference between linear search and binary search with their time complexities.



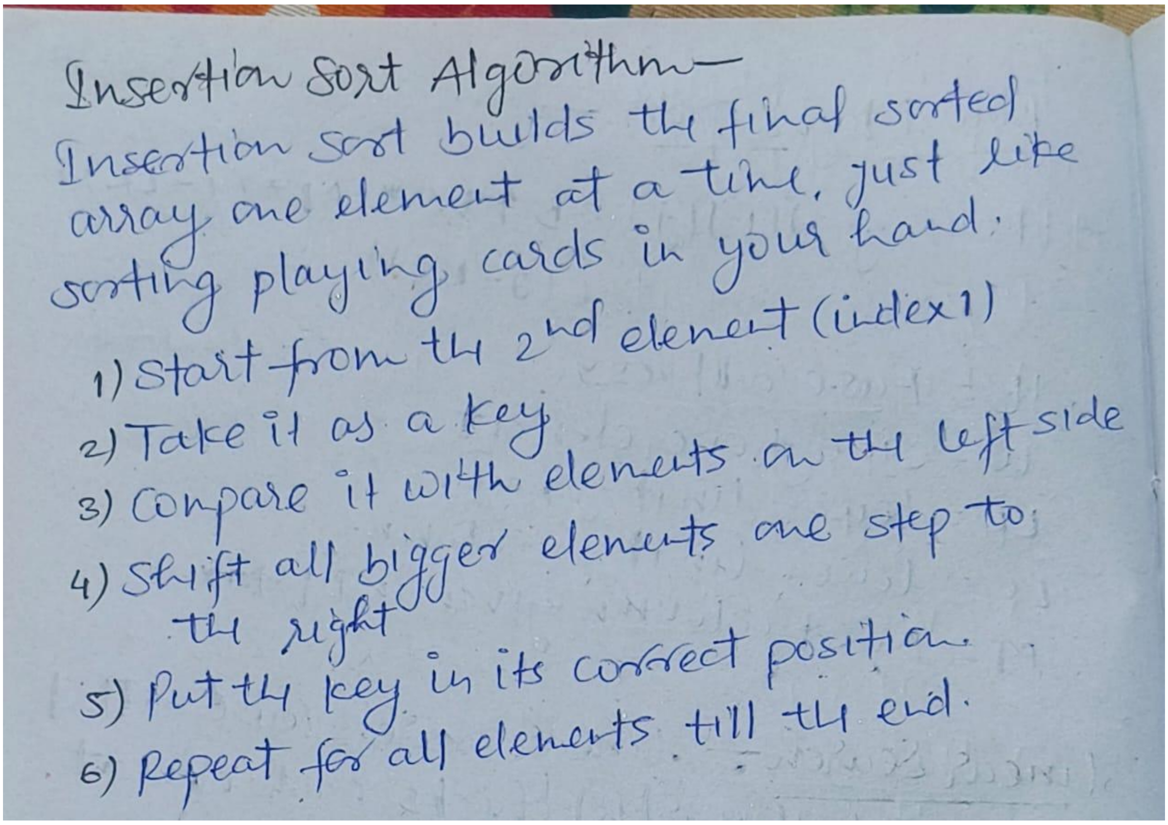
Linear Search — Linear search (also called sequential search) checks each element one by one until it finds the target element or reaches the end of the list.

Time complexity — $O(n)$

Binary Search — Binary search works on sorted arrays only. It repeatedly divides the search interval in half to find the target element efficiently.

Time complexity — $O(\log n)$

5. Write the algorithm and explain the working of insertion sort with an example.

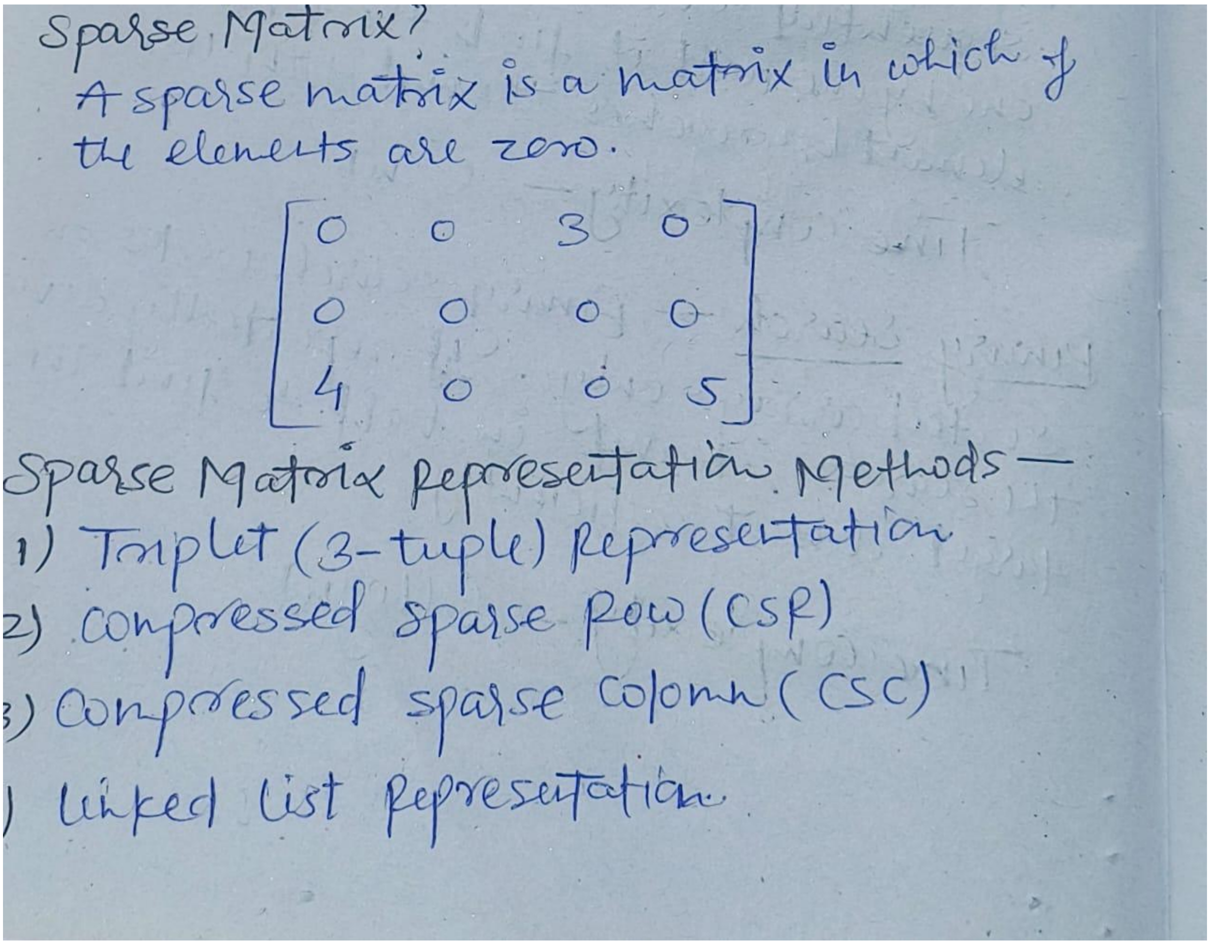


Insertion Sort Algorithm—

Insertion sort builds the final sorted array one element at a time, just like sorting playing cards in your hand:

1) Start from the 2nd element (index 1)
2) Take it as a key
3) Compare it with elements on the left side
4) Shift all bigger elements one step to the right
5) Put the key in its correct position.
6) Repeat for all elements till the end.

6. Discuss how sparse matrices are represented and explain any one representation method.



Sparse Matrix?

A sparse matrix is a matrix in which of the elements are zero.

$$\begin{bmatrix} 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 5 \end{bmatrix}$$

Sparse Matrix Representation Methods—

1) Triplet (3-tuple) Representation
2) Compressed Sparse Row (CSR)
3) Compressed Sparse Colomn (CSC)
) Linked List Representation