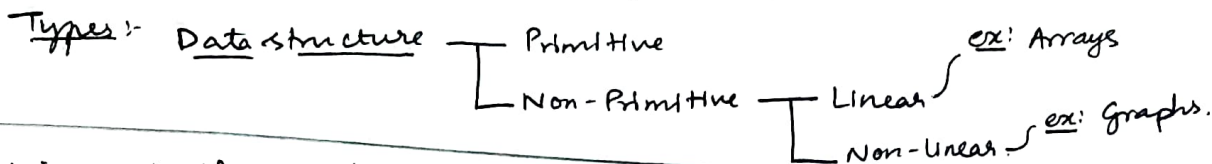


## Part-B

### 1. Types of data structure:-

⇒ Data structure :- Particular way of organizing data in a computer so that it can be used effectively.



### 2. List out the various operations on data structures:-

- ~~Traversing~~ - access each element
- Searching - Find any element
- Inserting - Insert any element
- Deleting - Delete any element
- Sorting - Sort the elements in desired fashion
- Merging - Merging arrays of elements

### 3. Define abstract data type with one example.

- ⇒ It is a type/class for objects whose behaviour is defined by a set of values & operations
- ⇒ It is easier to understand and its implementations can be changed with requiring changes to the program.

ex:) Stacks and queues. (LIFO & FIFO).

### 4. What is time complexity of an algorithm?

Illustrate with an example:-

- ⇒ Time complexity is the amount of time taken by an algorithm to run, as a function of the length of the input.

ex:) Time complexity for Linear search  $\rightarrow O(n)$   
" " " Binary search  $\rightarrow O(\log n)$ .

### 5. ~~Significance~~

Significance & Limitations of Big O:-

- ⇒ Big O notation is used to express the upper bound of the runtime of an algorithm.

∴ It measures the worst-case time complexity of an algorithm.

- It analyses & calculates the time & memory required for the execution of an algorithm for an input value.

### Limitations

- Sometimes ignores the important constants
- Although it shows growth of algorithm w.r.t size, it does not show the related efficiency.

7) Discuss the complexity of Insertion sort.

⇒ Best case :- Occurs when array is already sorted.

- Linear running time i.e.,  $O(n)$

→ As, in each iteration, first element of unsorted list is compared to ~~last~~ last element of sorted list.

Worst Case :- Quadratic running time i.e.,  $O(n^2)$

→ As, in each iteration of inner loop, elements of sorted array must be shifted to insert new element.

Average Case :- Quadratic running time.

- At least  $(k-1)/2$  comparisons are made.

8) Discuss the complexity of bubble sort :-

⇒ Total passes ⇒  $N-1$

1<sup>st</sup> pass → comparisons are made to place highest element in correct position. [ $N-1$  comparisons are made.]

2<sup>nd</sup> pass →  $N-2$  comparisons are made and second highest element is placed in its position.

∴ ~~To calculate~~ ∴ To find complexity :-

$$F(n) = (n-1) + (n-2) + \dots + 3 + 2 + 1.$$

$$\Rightarrow F(n) = \frac{n(n-1)}{2}$$

$n \rightarrow$  Total no. of elements in array.

$$\Rightarrow F(n) = \frac{n^2}{2} + O(n) = \underline{O(n^2)}.$$

[Time required to execute it is proportional to  $n^2$ ]

9) Discuss the complexity of binary search :-

⇒ Complexity →  $F(n)$

$n \rightarrow$  no. of elements in array.

⇒ Complexity depends upon no. of comparisons made.

∴ In each comparison, size of search segment is halved.

$$\text{i.e., } 2^{F(n)} > n$$

$$\text{or } F(n) = \log_2 n.$$

10) What is the need of using data structures?

⇒ (i) Helps in management of huge amounts of data.

(ii) Helps in efficient data search and retrieval.

(iii) Helps in storing data in a specific manner in memory.

(iv) They are used in every program/software system to arrange data.