1 What is data structure and explain different type of data structure with example.

Ans 1.) There are two type of data structure based on how it is arranged:-

(i) Linear data structure:
Here the data is stored in linear pattern or sequence. Array, stack, queue, link list are example of linear data structure.

(ii) Non linear data structure:-
Here the data is stored in non-linear pattern
Ex. Tree, Graph.

2.) There are two types of data structure based on memory type used.

(i) Static data structure:
The data structure whose memory occupation is fixed i.e. memory cannot be increased or decreased during run time is known as Static Data Structure.
Ex. Array

(ii) Dynamic data structure:
The data structure whose memory occupation is not fixed i.e. memory can be increased or decreased during run time is known as dynamic data structure
Ex. Link list

(iii) Other data structure like stack, queues, tree, graph can be static or dynamic depends on how they are implemented (using array or link list)

3.) There are two types of data structure based on which type of data it is holding.

(i.) Homogenous data structure: The data structure which stored data of similar type is known as homogenous data structure.

Ex. Array

(ii.) Non homogenous data structure.

a.) The data structure which stores data of different types is known as non-homogenous data structure.

b.) Link list is an example

(iii.) Other data structure like stack, queues, tree, graph can be homogenous or non-homogenous depends on how they are implemented using array or link list.

2 Define Algorithms, program and flow chart with respective example.

Ans. 1) Algorithm:

(i) An algorithm is a finite step-by-step procedure of solving particular task in a limited amount of time

(ii) It is a mathematical description.

(iii) It is the computational model

for example Algorithm for addition of two number can be written as.

Step 1: Start

Step 2: Take two numbers num1 and num2 from the user

Step 3: Add two numbers and store it in the variable sum

Step 4: Display sum

Step 5: Stop

2) Program:

(i) A program is a group of instructions that performs a specific given task when executed by a computer

(ii) Program is the implementation of an algorithm

(iii) It is directly associated with machine.

3) properties of algorithm.

(i) Non ambiguity: The algorithm should be simple and clear. There should not be any ambiguous statements in the algorithm.

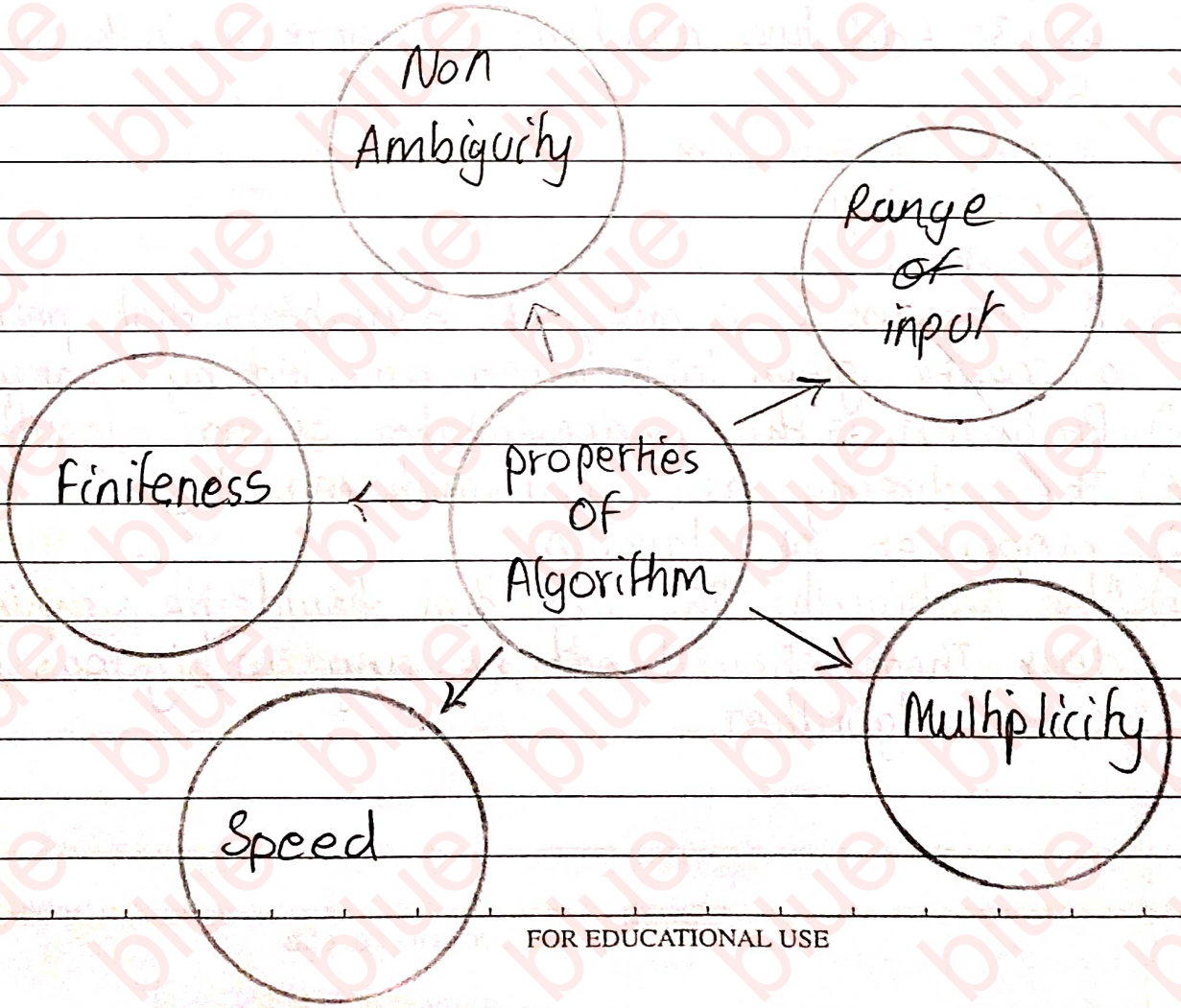
(ii) Range of input: The range of input should be specified for the algorithm, otherwise it will go to infinite state if not specified in advance.

(iii) Multiplicity: The same problem statement can be solved using different ~~all~~ algorithm. for example

Ex. searching can be done using binary or sequential search.

(iv) Speed: The logic of algorithm should be written in such a way that it requires less amount of time for giving output.

(v) Finiteness: The algorithm should be finite and can be terminated after certain instructions.



3 Explain Asymptotic Analysis and Notation

Ans:) (i:) Asymptotic analysis is use to find out which algorithm is better for the same problem statement It is use to find out how different algorithms works for different output.

Ex If there are two algorithm, algorithm 1 and algorithm 2 then for the same input algorithm 1 may perform better than algorithm 2 or it may be possible the performance of the algorithm 1 is good on one a machine as compared to algorithm 2.

(ii:) Asymptotic notation are the way to tell how algorithm behave for the different input value. It also tells how fast the function grows with the increase in input size.

(a.) Big-O notation.

(i.) It is denoted by 'O'

(2.) It is used to represent upper bound of the algorithm running time.

(3.) Using this we can compute the longest amount of time taken by any algorithm.

(b.) Big omega Notation.

1.) It is denoted by 'Ω'

2.) It is used to represent lower bound of the algorithm running time

3.) Using this we can compute the shortest amount of time taken by any algorithm

(c.) Big theta Notation

1.) It is denoted by 'Θ'.

2.) Using this we can compute the average amount of time taken by any algorithm.

3.) It lies between upper bound and lower bound of an algorithm.

4.) It is used to represent the upper and lower bound of a function

4 Define array with examples traversing operation of array

Ans) (i) Array is the most common data structure use for storing and retrieving a data.
It is used as a building block for other data structure.

Ex.

48	13	54	34	89	77	45	23

(ii) Traversing means visiting elements. Now if we want to print each element of an array the we need traversing to visit each element and print the element one by one.

Ex.

4	10	12	15	19
0	1	2	3	4

Algorithm for traversing an array:
Step 1: Start
Step 2: Repeat from lower bound (lb) to upper bound (ub)
for i = lb to ub
Step 3: Print element
print a[i]
Step 4: End

5 Define Array with examples insertion operation of array

Ans (i) Array is the most common data structure use for storing and retrieving a data. It is used as a building block for other data structure.

Ex.

10	20	30	40	50
0	1	2	3	4

(ii) Adding new element in the array is known as insertion operation, elements can be inserted at any location in the array provided the array is not full

(iii) Suppose, the capacity of the array is ten and all ten location are filled then it is not possible to insert an element. It is very easy to insert an item at the end of an array. But if we want to insert the element at the start or in between then we need to shift all the element to right by one position.

Ex. an array of capacity five and assume three element are filled

4	10	12		
0	1	2	3	4

(iv) Then in order to insert an element at position 1 we need to shift the element present at first position to right and shifting second position to right by 1, so that we can accommodate new element.

6 Define Array with examples merging of arrays with examples

Ans (i) Array shores data for use for storing and retrieving a data
It is used as a building block for other data structure

Ex.

10	20	30	40	50
0	1	2	3	4

(ii) combining the elements of the two linear array is known as merging of an array.
The array can be merged in two way.

(a) In First case:
we can merge two arrays without sorting

Ex.

Array 1				Array 2			
23	12	45	32	56	27	43	98

23 12 45 32 56 27 43 98

(b) In second case:
While combining the two lists we go on comparing the element, so that the resultant list will be in sorted order.

12	23	32	41		27	31	43	56
Array 1					Array 2			

12 23 27 31 32 41 43 56

7 Define Array will example with searching or array with example.

Ans (i) Searching means to find the element or the position of element in an array or given data There are searching ophon on Web browser or other website.

a) linear search

1. Searching in linear fashion is known as linear search, it is simplest form of search.
2. In this any random order list can be used for searching.
3. It does not mean that we can't do linear search when list is sorted.

Ex.

9 4 5 3 6 0 4 3 2 1

0 1 2 3 4 5 6 7 8 9

- It start from first number and compare if with the number which it wants to search
- it go on checking every number till it find the number equivalent to which it wants to search
- It can be done on sorted and unsorted sequence
- But, the time complexities in both the cases are nearly same

8 There is an array ARR[4..7, -1..3]. It requires 2 bytes to store each element. Calculate the address of [6][2] given base address 100.

Given: B=100, w=2, i=6, j=2, lbr=4 lbc=-1

Address of a[i][j] = B + w((i-lbr) + (j-lbc))

No. of column = ubc - lbc + 1 = 3-(-1)+1 = 5

Add(a[6][2]) = 100 + 2(5(6-4)+(2-(-1))

= 100 + 2(10+3) = 100+26 = 126

No. of rows = ubr - lbr + 1 = 7-4+1 = 4

Add[a[6][2]) = 100+2((6-4)+4(2-(-1)))

= 100+2(2+12) = 100+28 = 128

9 Consider a 3-D array M(2:8, -4:1, 6:10). Assuming the base adress is 100. and each element occupies 4 memory cell. calculate the adress of element M[5, -1, 8]

Given:

B = 200

$ub_1 = 8$ $ub_2 = 1$ $ub_3 = 10$

$lb_1 = 2$ $lb_2 = -4$ $lb_3 = 6$

$L_1 = ub - lb + 1 = 8 - 2 + 1 = 7$

$L_2 = ub - lb + 1 = 1 - (-4) + 1 = 6$

$L_3 = ub - lb + 1 = 10 - 6 + 1 = 5$

$i = 5$ $j = -1$ $k = 8$ $w = 4$

So, for row major order formula.

adress of Ca[i][j][k]:

base(A) + w[($L_2 L_3 (i - lb_1) + L_3 (j - lb_2) + (k - lb_3)$)]

= 200 + 4[6 * 5(5 - 2) + 5(-1 - (-4)) + (8 - 6)]

= 200 + 4[30(3) + 5(3) + 2]

= 200 + 4[90 + 15 + 2]

= 200 + 428 = 628

10 Write a short note on sparse Array

Ans 1.) The sparse array or sparse matrix is an array in which most of the elements are zero.

2) The sparse array is an array in which most of the elements have the same value the default value is zero or null.

3.) Sparse matrices are those array that has the majority of their elements equal to zero

4.) A sparse is an array in which elements do not have contiguous indexes starting at zero.

5.) Sparse arrays are used over arrays when there are lesser non-zero elements. Sparse arrays require lesser memory to store the elements and the computation time can be saved.

Ex.

$$\begin{bmatrix} 0 & 4 & 0 \\ 1 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 9 & 0 \end{bmatrix}$$

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