Qp - October - 2022

Data Structure.

Q.1 Attempt any Eight of the following.

a) How to measure performance of an algorithm

→ Performance analysis of an algorithm is the process of Calculating Space required by that algorithm and time required by that algorithm. Performance analysis of an algorithm is performed by Using the Following measure.

1) Space required to Complete the task of that algorithm (Space Complexity) includes program Space and data Space

2) Time required to Complete the task of that algorithm (Time Complexity).

b) What is polynomial ? How is it differ from Structure.

→ A polynomial is a Mathematical expression involving a Sum of powers in One or more Variables Multiplied by Coefficients.

The manipulation of Symbolic polynomials is an application of the list processing The polynomial is $A(x) = a_1 x^{em} + a_2 x^{e(m-1)} + \dots + a_n x^{e1}$, where $em-1 > \dots > e_1 \geq 0$ and $a_1 \dots a_n$ are Coefficients. A node of the Linked list will represent each term. A node having three Fields represents the Coefficient, exponent of a term and a pointer to the next item.

c) What is balance Factor ? How is it Calculated ?

→ A Balance Factor of a Node will be defined as height of the Left Subtree of the node, height of the right Subtree of the node

i.e Balance Factor = $h_L - h_R$

for example. A node when balance Factor is 0, we Conclude that the height of the left and right Subtree are equal. A node in the AVL tree is known as balanced if the longest path in left and right Subtree are identical.

d) What are abstract Data types ?

→ An Abstract Data types (ADT) is a data declaration packaged together with the Operations that are meanful for data type. We encapsulate the data and the Operations on data and we hide them from a User. They are not Concerned With the Implementation details like Space and time Compte effiency.

1) Domain. A Collection of Data.
2) functions. A Set of Operation on the data or Subset of Data
3) Axioms. A Set of Axioms, or rules of behavior governing the interaction of Operations.

e) What is Ancestor of Node ?

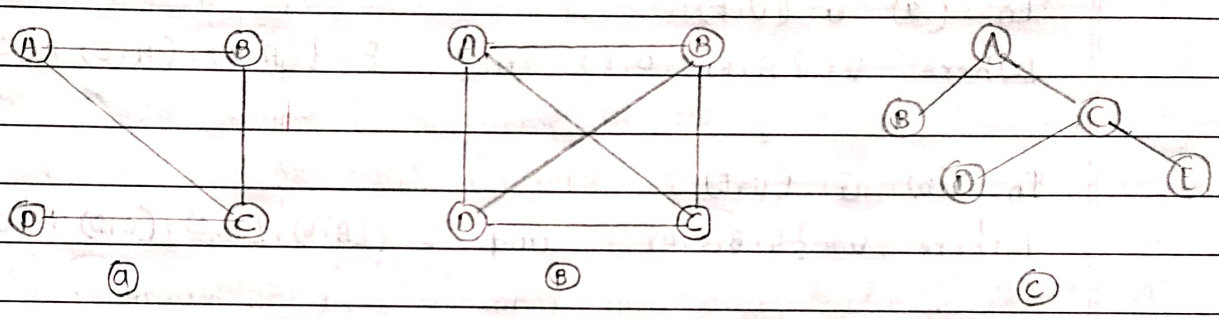
→ An ancestor of a Node is any other node on the path from the node to the root. A descedant is the Inverse relationship of ancestor. A node p is a descendant of

a node q if and only if q is an ancestor of p.
a path from one node to another.

f] Q. State the types of graph.

→

1] Undirected Graph - In an undirected graph, the pair of vertices representing any edge is unordered i.e the pairs (V_1, V_2) and (V_2, V_1) represent the same edge. In other words the edges have no direction in undirected graph.



(a) (B) (C)

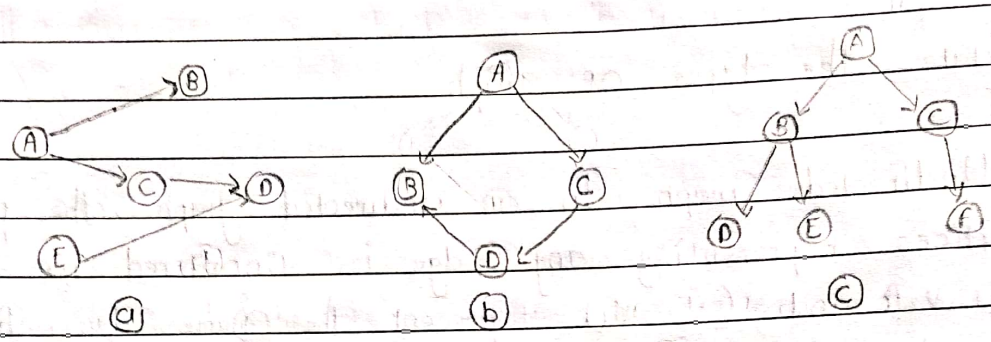
in (a) G = (V, E)
where V = {A, B, C, D} and E = {(A, B), (B, C), (B, C) (C, D)}

in (b) G = (V, E)
where V = {A, B, C, D} and E = {(A, B), (A, C) (A, D), (B, D), (B, C), (D, C)}

in (c) G = (V, E)
where V = {A, B, C, D, E} and E {(A, B), (A, C), (C, D), (C, E)}
This undirected graph is called Tree. Tree is a special case of graph.

2] Directed graph - In a directed graph each edge is represented by a directed pair (V_1, V_2), V_1 is the tail and V_2 is head of the edge i.e the pairs (V_1, V_2) and (V_2, V_1) are different

edges. In other words the edges have direction in directed graph is also called as digraph.



In (a) G = (V, E)
Where V = {A, B, C, D, E} and E = {(A, B), (A, C), (C, D), (A, E)(E

In (b) G = (V, E)
Where V = {A, B, C, D} and E = {(A, B), (A, C), (C, D), (D, B)}

In (c) G = (V, E)
Where V = {A, B, C, D) and E = {(A, B), (A, C) (B, D) (B, E) (C, F)}

9] Differentiate array and structure.

→

Array	Structure.
1. An Array is a Collection of related data element of same type	1. Structure can have elements of different types.
2. An Array is derived data type	2. A structure is a programmer defin
3. Array name works as a pointer to the first element of it.	3. But it is not in case of structure.

h) What is Space and time Complexity?

→ "Total amount of Computer Memory required by an algorithm to Complete its execution is Called as Space Complexity of that algorithm.

1) Instruction Space. It is the amount of Memory Used to Store. Compiled Version of instruction.

2) Environmental Stack. It is the amount of Memory Used to Store information of partially executed functions at the time of function Call.

3) Data Space - It is the amount of Memory Used to Store all the Variables and Constants.

Time Complexity. "The time Complexity of an algorithm is the total amount of time required by an algorithm to Complete its execution." The total time taken by the algorithm or program is Calculated Using the Sum of the time taken by each of executable Statement in algorithm or program.

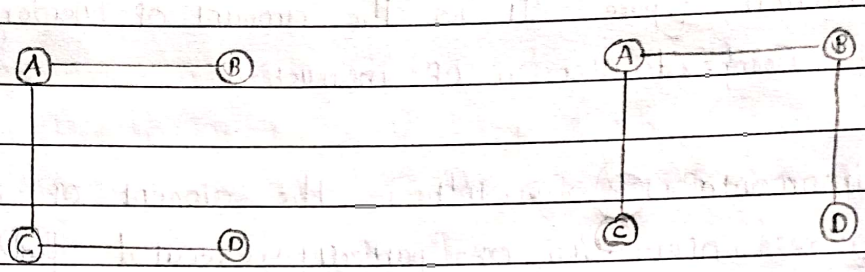
1) Best Case. 2) Worst Case 3) Average Case.

i) What is pointer to pointer?

→ A pointer in C is used to allocate memory dynamically i.e at run time. Normal Variable Stores the Value Whereas pointer Variable Store the address of the Variable. The Content of the C pointer always be a Whole number i.e. address. Always C pointer is initialized to Null, i.e. int *p = null.

j) What is Spanning tree?

→ A Spanning tree of a graph G = (V, E) is a Subgraph of G having all vertices of G and Containing only those edges that are necessary to join all the vertices in the Graph. A Spanning tree does not Contain Cycle in it.



Q.2 Attempt any Four of the following.

a) Explain Insertion Sort technique with an example.

→ In Insertion Sort the element is Inserted at an appropriate place. Here the array is divided into two parts Sorted and Unsorted Sub-array. In each pass, the first element of Un Sorted Sub array is picked up and moved into the Sorted Sub array by Inserting it Suitable position.

Example.

```
#include<stdio.h>
#define MAX 20
Void insertionSort (int A[MAX], int n)
{
    int i, j, key;
    for (i=1; i<=n; i++)
    {
```

```
        key = A[i]
    for (j=i-1; (j>=0) && (key<A[j]); j--)
            A[j+1] = A[j]
            A[j+1] = key;
    }
}
Void main ()
{
    int A[MAX], i, n;
    printf("How many element you want to Sort? \n");
    Scanf("%d", &n);
    printf("\n Enter the element into an array: \n");
    For (i=0; i<n; i++)
        Scanf("%d", &A[i]);
    Insertion Sort (A, n); /* Function Call */
    printf("\n Elements after Sorting: \n");
    for (i=0; i<=n; i++)
        printf("%d\t", A[i]);
}
```

Output:

How many Element you want to Sort ?

5

Enter the element into an array:

6 4 8 -9 0

Elements after Sorting

-9 0 4 6 8

b) What is circular queue? How it is differ from static queue

→ The technique which essentially allows the queue to

Wrap around upon reaching the end of array, eliminates these drawbacks. Such a technique which allows queue to wrap around from end to start is called Circular queue. Which is more efficient queue representation Usign array Q as Circular.

Suppose an array Q is of size n. Now is we go on adding elements in queue, we may reach the location n-1. If queue is not Circular, we cannot add more element when end is reached, even through there are free locations at the front side of array. In such case, these slots would be filled by new elements added to the queue, rather than signaling a error as queue is full. Hence Circular queue allows us to continue adding elements, even through rear = n-1

The queue is said to be full only when there are n elements in the queue. The

5] What is stack? What are the various application of stack. List operations performed on stack.

→ A stack is an ordered collection of Homogeneous data elements where the insertion and deletion operation take place at only one end; called as top of the stack.

1) Expression Conversion- An expression can be represented in prefix, postfix or infix notation. Stack can be used to convert one form of expression to another.

2) Expression Evaluation - A stack can be used to evaluate prefix postfix and infix expression.

3] Syntax parsing - Many Compilers use a Stack for parsing the Syntax of expressions, program blocks etc. before translating into Low Level Code.

4] String Reversal - Stack is Used to reverse a String. We push the Character of String one by one into Stacks and then pop Character from Stack.

5] Parenthesis Checking - Stack is used to Check the proper Opening and Closing of parenthesis.

6] function Call - Stack is Used to keep information about the active Functions or Subordinates.

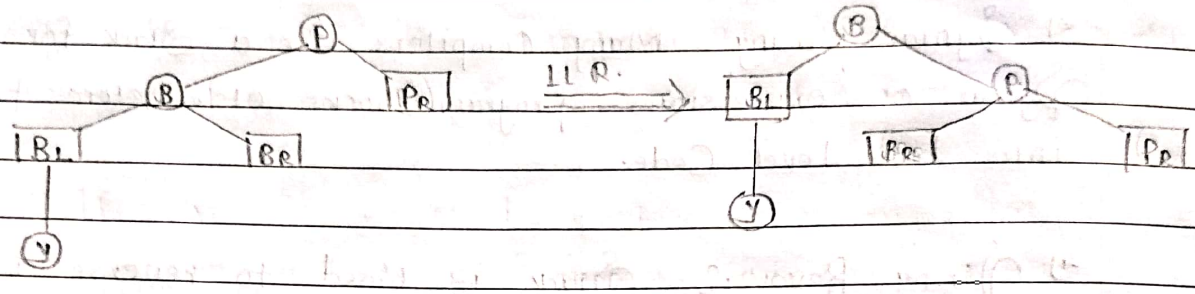
7] Backtracting - Backtracking is One of the algorithm designing technique. For that purpose, We dive into Some way, if that Way is not efficient, We Come back to the previous State and go into Some other paths. To get back From Current State, We need to Store the previous State. for that purpos We need Stack.

i] init ii] Push iii] Pop iv] Is empty v] Is full.

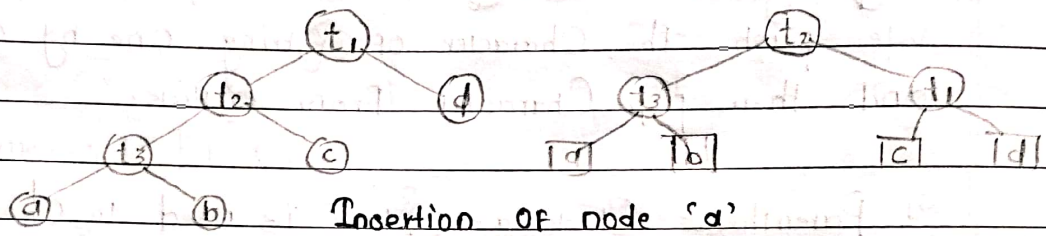
d] Explain different types of AVL rotations with an example.

→

1] LL Rotation - Left to left insertion. Unbalance Occured due to the insertion in the Left Subtree of the Left Child of the pivot node.

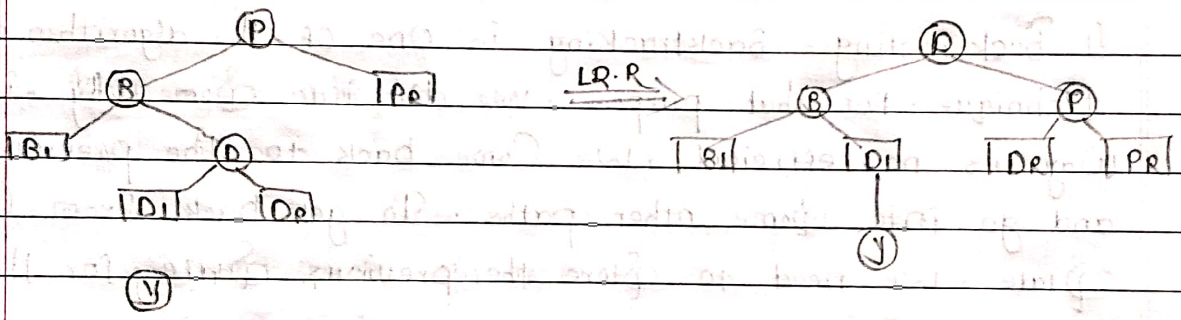


Ex -

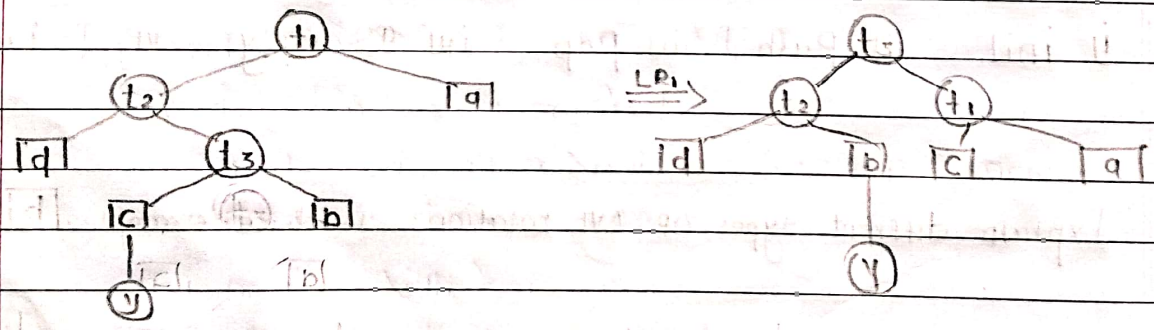


Insertion of node 'a'

3] LR Rotation Left to Right insertion - Unbalanced Occured due to the Insertion in the right Subtree of the left Child of the pivot node.

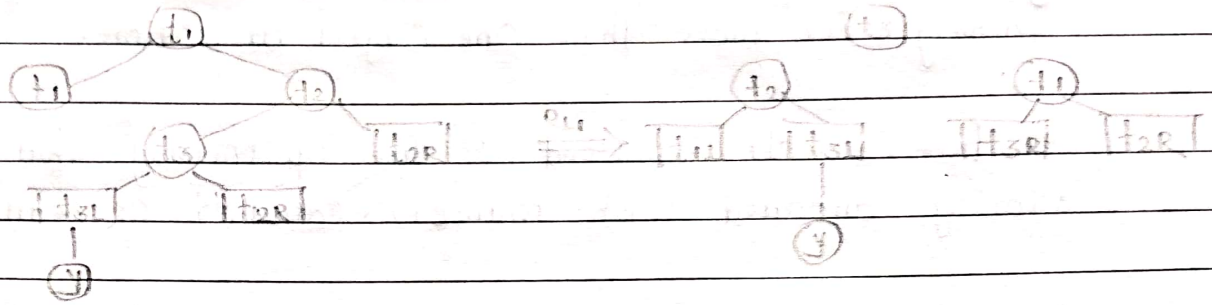


Ex



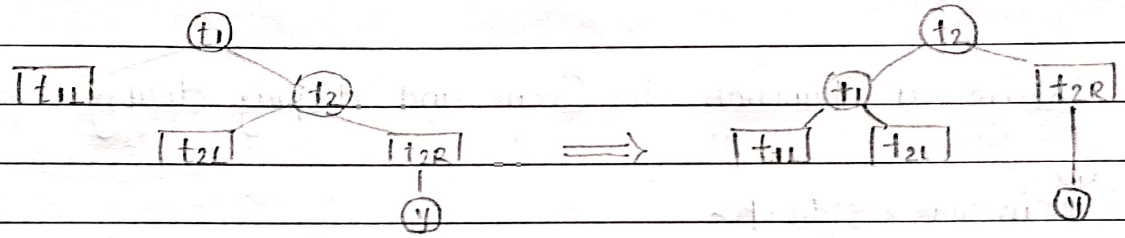
3] RL Rotation Right to Left insertion - Unbalanced Occured due to the insertion in the Left Subtree of right Child of the pivot node.

Ex -



4] RR Rotation Right to Right insertion - Unbalanced occured due to the Insertion in the right Subtree of the right Child of the pivot node.

Ex.



e] Explain various types of dynamic Memory Allocation functions.

→ The process of allocating memory at run time is known as dynamic Memory Allocation. Although C does not inherently have this facility, there are four library routines known as "Memory management functions" that can be used for allocating and freeing memory during programe execution.

1] malloc() - It is used to allocates a specified number of bytes in memory. Returns a pointer to the beggining of the allocated block.

2] Calloc() - It similar to malloc(), but initializes the alloca-

bytes to zero. This function also allows us to allocate Memory for more than One Object at a time.

3] free() - It is used to free up Memory that was previously allocated with malloc(), calloc(), or realloc().

4] realloc() - It is used to Change the Size of a previously allocated block.