[Total No. of Questions: 7]

Registration No.:

[Total No. of Pages: 2]

G H Raisoni Institute of Engineering & Technology, Pune

(An Autonomous Institute affiliated to SP Pune University)

Second Year B.Tech. End Semester Examination **Winter - 2020**

Electrical Engineering

Data Structures and algorithms (19EL204)

Duration: **2 Hr 30 Min** Maximum Marks: **60**

Instructions to the Candidates:

i. ***Q.1 is compulsory***
ii. *Attempt **Q.2 or Q. 3, Q.4 or Q.5 and Q.6 or Q.7***
iii. *Figures to the right indicate full marks*
iv. ***BL** indicates **Bloom's Taxonomy level, CO** indicates **Course Outcome & PO** indicates **Program Outcome (PO)***
v. *Use of logarithmic table, drawing instruments and non programmable calculator is permitted*
vi. *Diagrams / sketches should be given wherever necessary.*
vii. *Assume suitable data wherever necessary and state the assumptions made.*

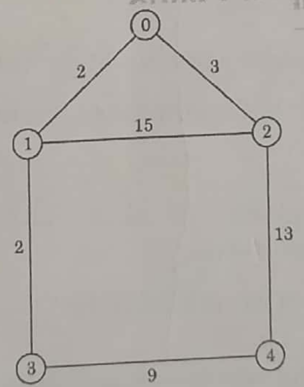
Q. No.		Question	Marks	BL	CO	PO
Q. 01		Attempt any ***Five*** of following				
	a	Explain function parameter passing by reference with example.	[04]	L2	1	
	b	Describe in detail algorithm for selection sorting technique.	[04]	L3	2	
	c	Explain linear search with example. State its time complexity	[04]	L1	2	
	d	Write C function for PUSH operation to implement stack using array.	[04]	L2	3	
	e	What is time complexity of algorithm? Comment on time complexity of insertion sorting.	[04]	L1	2	
	f	Sort the following data using bubble sort 29, 11, 15, 27, 33, 18, 16, 30	[04]	L3	2	
	g	Convert the given infix expression to postfix expression. (a * b) – (c – f)) / (e + g)	[04]	L3	3	
Q. 02	a	Differentiate between static and dynamic data structures with examples.	[04]	L2	5	
	b	Define structure declaration for node in singly linked list with detailed explanation about members of structure.	[05]	L1	5	
	c	Write a 'C' function to delete a number from singly linked list.	[04]	L2	5	

OR

Q. 03	a	State the advantages of circular linked list over singly linked list.	[04]	L2	5
	b	What is pointer ? What are the advantages using pointer ? Explain pointer declaration and its initialization with an example.	[05]	L1	5
	c	What is a linked list .What are its various types.	[04]	L1	5
Q. 04	a	Describe tree. Explain any 5 key terms associated with tree.	[04]	L1	6
	b	Differentiate between complete binary tree and perfect binary tree.	[05]	L1	6
	c	Construct binary search tree for the following 35 , 22 , 64 , 15 , 18 , 42 , 39 , 77	[04]	L3	6

OR

Q. 05	a	What is Binary Search Tree (BST)? Explain with example.	[04]	L1	6
	b	State the advantages of linked implementation of a binary tree over array implementation.	[05]	L2	6
	c	What are the different types of tree traversal? Explain any one with example.	[04]	L1	6
Q. 06	a	Define graph. Explain it with example.	[04]	L1	6
	b	State the different methods of representing graph.	[05]	L2	6
	c	Give the adjacency matrix and adjacency list for the graph shown in figure	[05]	L3	6



OR

Q. 07	a	What is an adjacency list. Explain with example.	[04]	L2	6
	b	Define Depth Search First (DSF) and Breadth Search First (BSF) terms of graph with example.	[05]	L2	6
	c	Explain in detail with suitable example Prim's Algorithm for finding out minimal spanning tree.	[05]	L3	6

Blooms Taxonomy level 1: Remember, 2: Understand,3: Apply, 4: Analyse, 5: Evaluate, 6: Create

2021/3/16 10:44

[Total No. of Questions: 7]

Registration No.:

[Total No. of Pages: 2]

Marking Scheme

	G H Raisoni Institute of Engineering & Technology, Pune *(An Autonomous Institute affiliated to SP Pune University)*
	Second Year B.Tech. End Semester Examination Winter – 2020
	Electrical Engineering
Data Structures and algorithms (19EL204)	
Duration: 2 Hr. 30 Min	Maximum Marks: 60

Instructions to the Candidates:

i. Q.1 is compulsory
ii. Attempt Q.2 or Q. 3, Q.4 or Q.5 and Q.6 or Q.7
iii. Figures to the right indicate full marks
iv. BL indicates Bloom's Taxonomy level, CO indicates Course Outcome & PO indicates Program Outcome (PO)
v. Use of logarithmic table, drawing instruments and non programmable calculator is permitted
vi. Diagrams / sketches should be given wherever necessary.
vii. Assume suitable data wherever necessary and state the assumptions made.

Q. 01		Attempt any *Five* of following	Marks	BL	CO	PO
	a	Explanation – Syntax – 02 M C Program Code -02 M	[04]	6	CO1	PO1 PO3
	b	Data representation and Explanation- 02 M Steps for sorting -02 M	[04]	2	CO2	PO1
	c	Steps with data set -03M Time Complexity -01M	[04]	5	CO3 CO4	PO2
	d	Push function declaration -01M User defined push function prototype- 01 M Main function -02M	[04]	2	CO1	PO3
	e	Time Complexity -02 M Time Complexity of Insertion sort- 02M	[04]	2	CO1	PO3
	f	Explanation - 02 M Steps for Sorting – 02 M	[04]	6	CO3	PO3
	g	Explanation -02 M Steps – 02 M	[04]	6	CO1	PO3
Q. 02	a	Differentiate points -03M Example – 01M	[04]	2	CO5	PO1
	b	Explanation of SLL – 02M Members of structures- 03M	[05]	2	CO5	PO1
	c	Program body intilization -01 M Prototype – 01M	[04]	6	CO5	PO3

		Main Programm -02M				
		OR				
Q. 03	a	Advantages – 04 points – 04M	[04]	2	CO5	PO1
	b	Definition -01M Advantages -02 M Declaration -01M Initilization -01M	[05]	2	CO5	PO1
	c	Linked List Explanation -03m Varrous types-01m	[04]	2	CO5	PO1
Q. 04	a	Definition of Tree -01 M 05 key Terms- 03M	[04]	2	CO6	PO1
	b	Differentiating 03 Points – 03 M Each graphs- 02M	[05]	2	CO6	PO1
	c	All steps – each steps -01 M	[04]	6	CO6	P
		OR				
Q. 05	a	Explanation -03M Example -01m	[04]	2	CO6	PO1
	b	Advantages -5 points – 05 M	[05]	2	CO6	PO1
	c	Types – 01m Explanation of any one -03M	[04]	2	CO6	PO1
Q. 06	a	Definition -01M Explanation with graphs example - 03 M	[04]	3	CO6	PO4
	b	Four methods with explanation -04M Graphs -01M	[05]	2	CO6	PO4
	c	Adjacency Matrix-03 M Adjacency List -02M	[05]	2	CO6	PO4
		OR				
Q. 07	a	Definition -02 M Explanation with Example -03m	[04]	3	CO6	PO4
	b	Four Point each -03M Graphs of each -02M	[05]	2	CO6	PO4
	c	Graphs – 01M Explanation -02M Minimum Spanning Tree -2M	[05]	2	CO6	PO4

Blooms Taxonomy level 1: Remember, 2: Understand,3: Apply, 4: Analyse, 5: Evaluate, 6: Create

2021/3/16 10:47

16/12/2020

End Sem Examination, Winter-2020

BE. - Data Structure & Algorithms - 19EL204

2 Hr. 30 min

Maximum Marks - 60

Q.-1

(a) - Explain function parameter passing by reference with example.

- Explanation - syntax - 2M
  code - C program - 2M

Solⁿ: C allows passing of parameters by:

(i) By Value. (ii) By Reference

→ When a parameter is passed by value, the calling program variable remains unaffected by whatever the function does to its local variables

→ When a parameter is passed by reference, the function can change the contents stored at the address.

The Sample program

There are two variations of the function increment. In the first variation incr1() a parameter n is passed by value, whereas in the second variation incr2() a parameter n is passed by reference function incr1() cannot change the value of n declared in the calling program

Function incr2() is able to change the value of n declared in the calling program.

Code:

```
#include<stdio.h>
void incr1(int n)
 {
   n++;
 }
void incr2(int *n)
 {
   *n = *n +1
 }
void main()
 {
   int n=10;
 printf("\n Value of n before calling incr1()
                        = %d", n);
 incr1(n)
 printf("\n Value of n after Calling incr1()
                        = %d", n);
 incr2(&n)
 printf("\n Value of n after calling incr2()
                        = %d", n);
```

output : Value of n before Calling incr1() = 10

— " — after — " — = 10

— " — — " — — incr2() = 11

Q.1 (b) Describe in detail algorithm for selection Sorting techniques

- data representation & Explation 2 M
- step - 1, - 2, - 3 for selection, - sorted & unsorted - 2 M

selection sorting is a simple sorting algorithm. This sorting algorithm is an in-place comparsion based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. initially the sorted part is empty and the unsorted part is the entire list.

The smallest element is selected form ~~unsorted~~ unsorted array and swapped with the leftmost element and that element becomes a part of the sorted array. This process continues moving unsorted array boundary by one element to the right

50	5	78	45	70	4

4	5	78	45	70	50

4	5	78	45	70	50

4	5	45	78	70	50

4	5	45	50	70	78

4	5	45	50	70	78

Sorted array list with selection sort

Q.1. (c) Explain linear search with example. State its time complexity

→ linear search steps with given data set for searching particular element - 03 Marks

→ Time Complexity - 01 Marks

Soln :- Linear Search is very Simple Search Algorithm. A Sequential Search is made over all items one by one. in linear Search Every element is checked and if a match is found. then that particular element is returned. Otherwise. the search Continues till the end of the data Collection.

10	14	19	26	$\boxed{33}$	31	44

33

10	26	44
best case	average case	worst case

Time complexity - Time for Searching the element in given data is depends upon the location of that Searching element. If the element is at the start of data set then Time complexity is less. if it is the middle the time Complexity is average and if it is at the end end then time Compelixity is worst.

2021/3/16 10:4

Q.1 (d) Write C function for PUSH operation to implement stack using array

- PUSH operation explanation — 01 M
- User defined push function — 01 M
- Main function push operation — 03 M

Sol^n

```
#include <stdio.h>
#include <stdlib.h>
#define size 20.
int stack [size], top-of-stack;
void push[int [], int);
            // to insert the element in the
               stack.
void pop [int[]);
void display (int []);
void main( )
{  int num = 0;
   int option = 0;
   top-of-stack = -1;
printf("\n Enter the item you want to
        insert into stack:");
scanf("\n %d", ~~&num~~ &num);
push(stack, num);
}
```

```
Void PUSH(int stack[], int num)
{
    if(top-of-stack == size-1)
    {
     printf("\n Cannot add element, stack is
          full Now\n");
         return;
    }
    top-of-stack++;
   stack[top-of-stack] = num;
}
```

Q.1. e. What is time Complexity of Algorithm? Comment on Time complexity of insertion Sorting.

Def of Time Complexity - 02 M
Time Complexity of insertion sorting - 02 M.

Time Complexity of an algorithm is the representation of amount of time required defined as a numerical function t(N)

This depends upon the frequency count/ iteration/ loop. Time required to ~~run~~ make one comparision is T and if there are

2021/3/16 10:50

iteration/comparision then time Complexity is equal to
= T x n

Time Complexity in insertion Sort

The array is searched sequentially and unsorted items are moved and inserted into the sorted sub list. its average and worst case Complexity are of $O(n^2)$ where n is the number of items.

14	33	27	10	35	19	42

14 - 33 — Allready in Sorted manner.

33 - 27 — unsorted manner Swap.

27 - 33

33 — 10 unsorted maner Swap

and so on.

Q-1-f. Sort the following data using bubble sort.

29	11	15	27	33	18	16	30

Explation — Bubble sort — 02

Sorting - Technigues — 2

The bubble sorting is Comparison-based algorithm in which each pair of adjacent elements is Compared and elements are Swapped if they are not in order. This algorithm is suitable for large data sets as its average and worst case. complexity are. $O(n^2)$ where n is is the. number of elements in data set.

29	11

29 > 11 so swapped

11	29	15

29	15

29 > 15 so swapped

11	15	29	27

29	27

29 > 27 so swapped

11	15	27	29	33	18

33 > 18 so swapped

11	15	27	29	18	33	16	30

18 is less than 27, 29.

So ~~Ith~~ it has to shift.

11	15	18	27	29	33	16	30

11	15	16	18	27	29	30	33

This is final sorted array of given data.

1. (g). Convert the given infix expression to ~~pre~~ postfix expression

$$[(a*b) - (c-f)] / (e+g)$$

2(a) Differentiate between static and dynamic data structure with examples

- Differentiation of ~~Two~~ point — 03
  03
  Example — 01 point — 01.

Static	Dynamic
1. Memory allocated at the time of loading	1. Memory allocated at the time of running
2. Static data structure may cause overflow due to under allocation of memory	2. No such problem in dynamic data structure
3. under utilization of memory in case of over allocation of memory	3. No such problem.
4. Access speed is higher no re usability of memory	4. Access speed is slow re-usable of memory is possible.
int a, int c[10],	malloc() calloc() realloc(), free()
5 less efficient	5 More efficient
6 Allocation is done before program execution	(6) Allocation is done during program execution.
7. Variable get allocated permanently	7 → System defined function are used
8 Stack memory	(8) Heap Memory

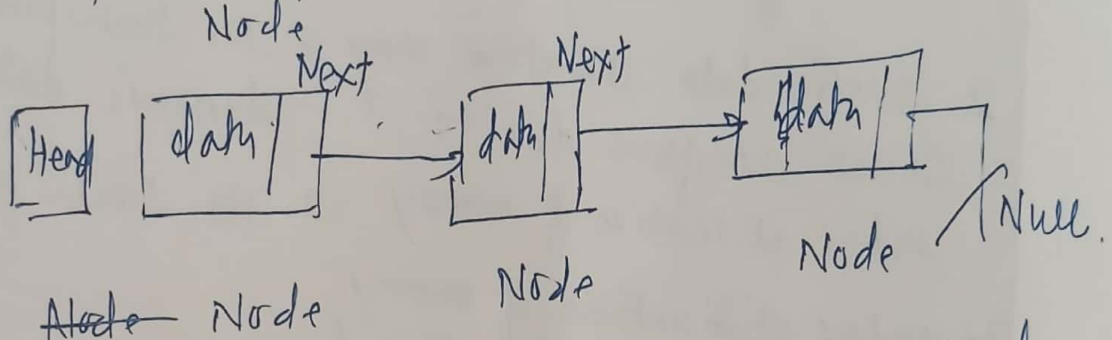
2021/3/16 10:52

Q.2 (b) Define structure declaration for node in singly linked list with detailed explanation about members of structures

Explation of singly linked list - 02

member of structure in singly link list & explanation ~~list~~ - 03

→ A linked list is a chain of node, where every node points to the next node



* linked list contains a link element called first.
* Each link carries a data field(s) and a link field called next
* Each link is linked with its next link using its next link.
* Last link carries a link as null. to mark the end of the list

```
struct LinkedList
{
  int data;
  struct LinkedList *next;
};
```

above given code is structure code for Singly linked list. The list of member in Singly linked list are integer type data, in place of a data type, struct LinkedList is written befor next. It indicates that it is self-referencing pointer. It means it is a pointer that points to whatever it is a part of. Next is a part of a node and it will point to the next node.

Q.2 (C) Write a 'C' function to delete a number from Singly linked List

```
# include <stdio.h>
# include <string.h>        ⇒ header files
# include <stdlib.h>
# include <stdbool.h>

 struct node
     {
         int data
         int key
     struct node *next;
     };
```

// Function for deleting the number from Singly linked list

```
struct node * deletefirst()
{
struct node *tempLink = head;
head = head -> next;
return tempLink;
}

struct node * delete(int key)
{ // start from the First link
struct node * Current = head;
struct node * previous = Null;
if(head == Null)
{
Return Null;
}
While (Current -> key != Key)
{
if(current -> next == Null)
{ Return Null;
```

2021/3/16 10:53

```
else
 {
   previous = current;
   current = current -> next.
   }
 }
  if(current = = head)
     {
      head = head -> next
      }
   else
   previous -> next = current -> next;
   }
return current;
}.
```

Q.3 (a) State advantages of circular linked list over singly linked list

- linked list, each node in the list stores the contents and a pointer or reference to the next node in the list.

1. In CLL end node will points to the first node - It does not contain a Null pointer. In SLL it ~~will~~ won't point to the first node

ii. Circular list is very useful in case of Game play to turn for each player without any failure (due to it's circular connectivity)

iii Circular linked list can be used for ~~impetse~~ implementation of Queue

iv.) It saves time when we have to go to first node form last node it can be done in single step because there is no need to traverse the in between nodes.

Q.3 (b) What is pointer? What are the advantages using pointer. Explain pointer declaration and it's initialization with an example.

pointer definition – 1M
Advantage – 02
declaration – 01
initialization – 01

Point is a variable whose value is the address of another variable i.e direct address of the memory location.

int var = 10;
int *p;
p = & var;

P	Var.
0X7FFF0	10
0X7FFF1	0X7FFF0

p is the pointer that stores the address of variable var, The data type of pointer p and variable var should match because an integr pointer can only hold address of integr. variable.

Q.3 (c) ~~What is~~

Advantages of pointer

i. It allows management of structures which are allocated memory dynamically.

ii. It allows passing of arrays and strings to functions more efficiently.

iii. It makes possible to pass address of structures instead of entire structure to the functions.

pointer declaration & its initialization

```
#include<stdio.h>

Void main()
    int Var=10;
    int *ptr=&var.
      *ptr= 20
    int **ptr = &ptr.
      **ptr= 30;
```

```
#include<stdio.h>
 Void main()
 {   int a=10;
     int * ptr; //declaration
     ptr= &a; // initialization
 }
```

Q.4. (a) Describe tree. Explain any 5 Key terms associated with tree:

Definition tree - 02

5 Key terms - 02

A tree is nonlinear data structure compared to array linked lists, stack and queues which are linear data structures. A tree can be empty with no nodes or a tree is a structure consisting of one node called the root and zero or one or more subtrees.

Terms associated with tree

Root: The node at the top of the tree is called root. There is only one root per tree and one path from the root node to any node.

Parent: Any node except the root node has one edge upward to a node called parent.

Path: path refers to the sequence of nodes along the edges of a tree.

Child: The node below a given node connected by its edge downward is called its child node.

Leaf: The node ~~below a given node~~ which does not have any child node is called the leaf node.

Subtree: Subtree represents the descendants of a node.

Key: Key represent a value of a node based on which a search operation is to be carried out

Visiting: Visiting refers to the checking the value of a node when control is on the node.

Traversing: Traversing means passing through nodes in a specific order

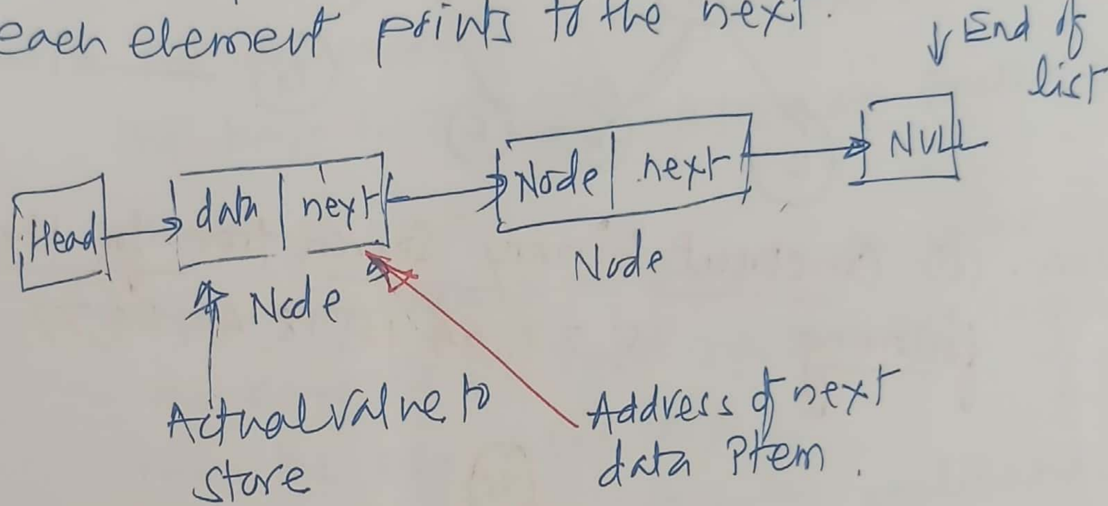
Level – 0 Level, 1 level, 2 level
Parent, Child, grandchild

3. (c) What is linked list: What are its various types:

linked list explanation — 03 M

Various Type — 01 M

linked list is a linear collection of data elements whose order is not given by their physical placement in memory. Instead each element points to the next.



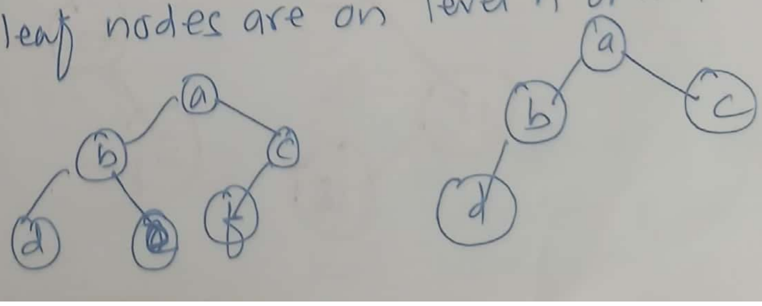
Types of linked list.

1. Singly linked list
2. Doubly linked list
3. Circular linked list.

Q.04. (b) Differentiate between complete binary tree and perfect binary tree..

A complete binary tree is defined as a binary tree where:

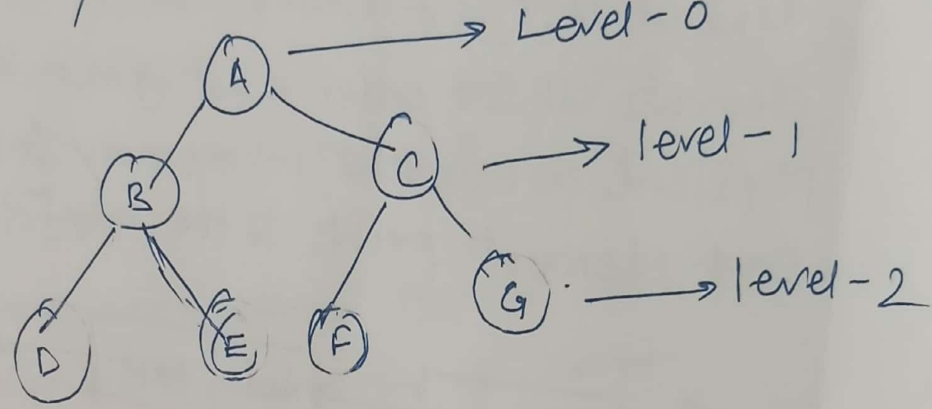
(i) All leaf nodes are on level n or n-1



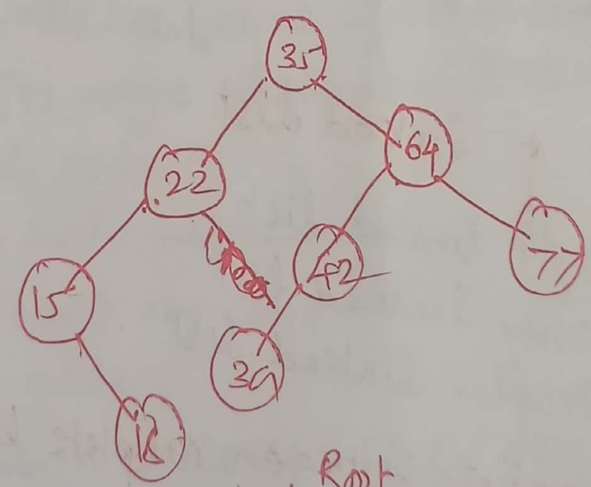
Perfect Binary Tree

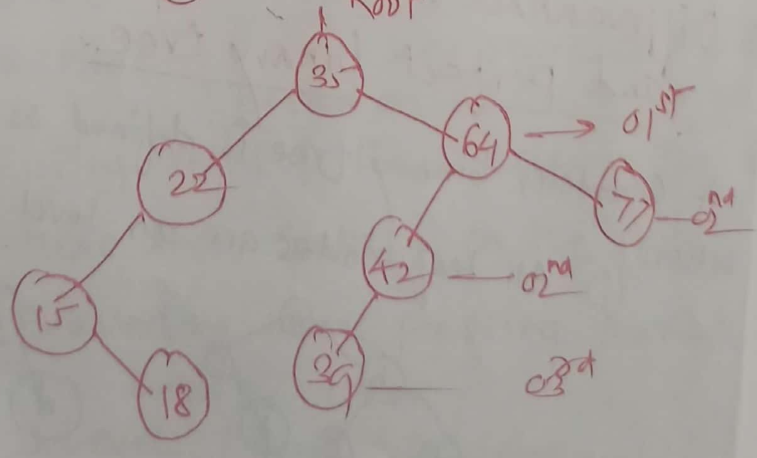
A binary tree is said to be full binary tree if each of its node has either two children or no child at all. Number of node at any level i in a full binary is given by 2^i

A. full binary tree is shown below.



Q 04. (C) Construct binary search tree for the following - 35, 22, 64, 15, 18, 42, 39, 77





5 (a) What is Binary Search Tree (BST) Explain with examples

Explation - 03

Example - 01

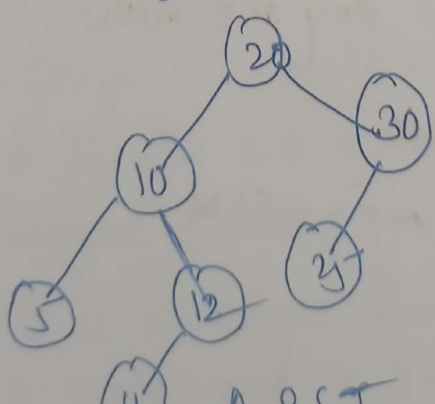
A binary search tree is a binary tree, which is either empty or in which each node contains a key that satisfies the following conditions:

(i) All keys are distinct

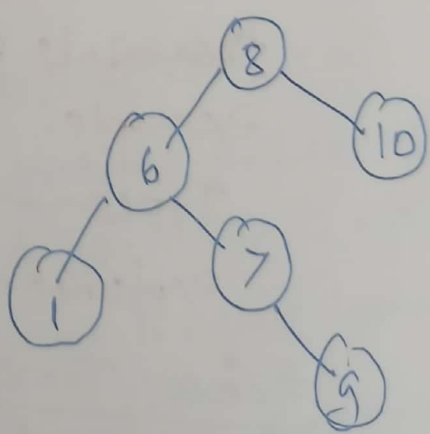
ii. For every node x in the tree, the values of all the keys in its ~~right~~ left subtree are smaller than the values in x.

iii For every node x in the tree the values of all the keys in its right subtree are larger than the key value in x

Binary search tree finds it application in searching. In below figures. The tree shown in fig (a) is BST and the tree shown in fig (b) is not BST.



A BST



left subtree of the node with key 8 has a key ...

Q.5(b) State the advantages of linked implementation of a binary tree over array implementation.

5-point advantages - 5 Marks

i It has dynamic size allocation

ii It has ease in insertion and deletion.

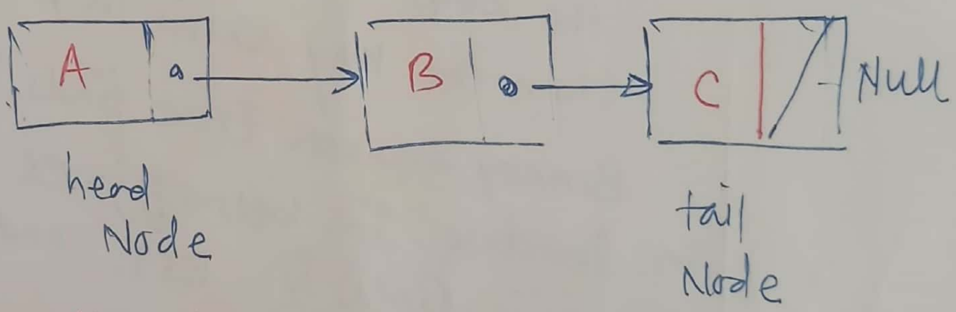
iii Dynamic data structure.

linked list is a dynamic data structures so it can grow and shrink at runtime by allocating and deallocating memory.

iv) No memory wastage.

(V) Easy implementation using linked list

Vi



Q.5.(C) What are the different types of tree traversals? Explain any one with example.

Types — 01 M

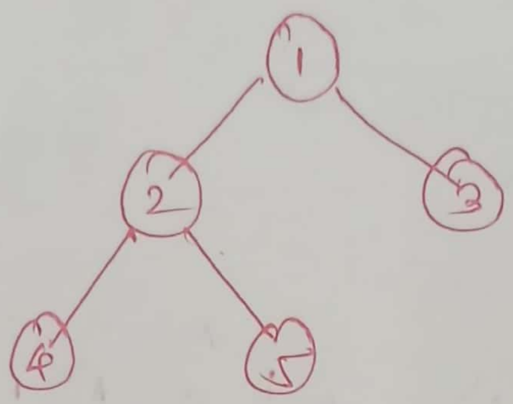
explaination - any one — 03 M

Tree Traversals

① Inorder

II Pre order

III Post order



Depth First Traversals

(1) Inorder — Left - Root - Right

4, 2, 5, 1, 3

(2) pre-order - Root - left - Right

1, 2, 4, 5, 3.

(3) post order - Left - Right - Root

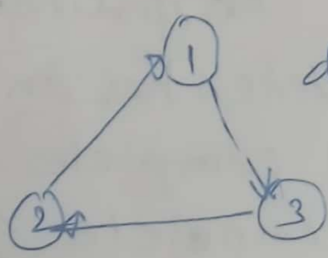
4, 5, 2, 3, 1

Q-6 (a) Define graph. Explain it with example.

— definition — 01 Marks

explaination with example — 03 Marks

A graph is a data structures that consists of a finite set of nodes or vertices and set of edges connecting them A pair (x, y) is refered to as an edge which connect the line between x vertex and y vertex



directed graph

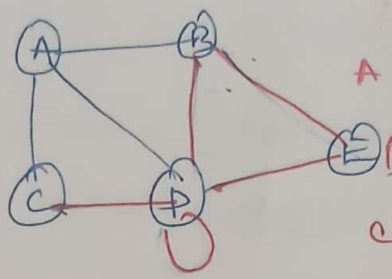
Vertices are - 1, 2, 3

edges are (2,1) (3,2) (1,3)

Q.6 (b) State the different methods of representing graph.

* Types — of terms w
with representati

(i) Undirected graph

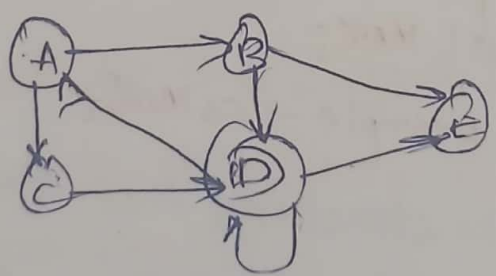


	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	0	0
D	0	1	1	1	1
E	0	1	0	1	0

A graph containing unordered pair of vertices is called an undirected graph.

(ii) Directed graph

A graph containing ordered pairs of vertices is called a directed graph. If an edge is represented using a pair of vertices $(V_1 V_2)$ then the edge is said to be directed from V_1 to V_2



	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	1	0
D	0	0	0	1	1
E	0	0	0	0	0

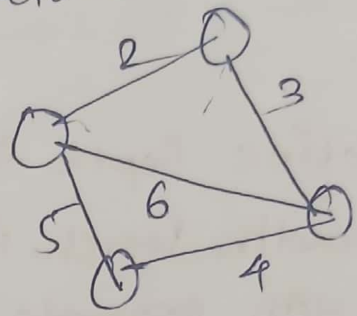
(iii) A Competete graph

An undirected graph in which every vertex has an edge to all vertices is called a complete graph

A complete graph with N vertices has $\frac{N(N-1)}{2}$ edges

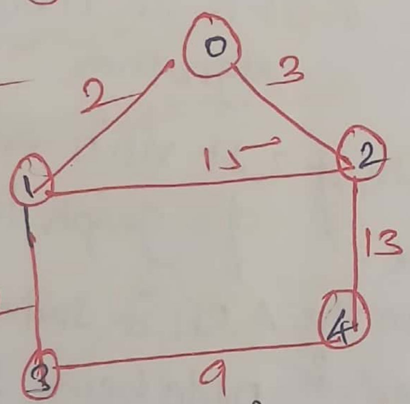
(iv) Weighted Graph

A weighted graph is a graph in which edges are assigned some value. Most of the physical situation are shown using weighted graph. An edge may represent a highway link between two cities. The weight will denote the distance between two connected cities using highway



Q. 6 (c) Give the adjacency matrix and adjacency list for the graph shown below.

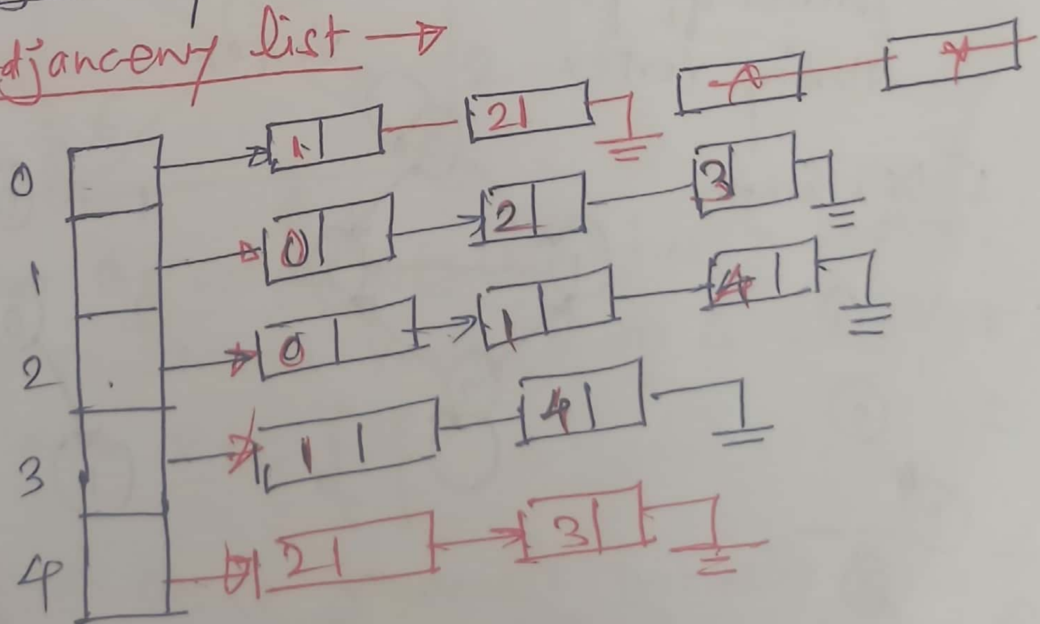
Matrix - 02
list - 02



Adjancency Matrix →

	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

Adjancency list →



Q. 07. (a) What is an adjacency list. Explain with example

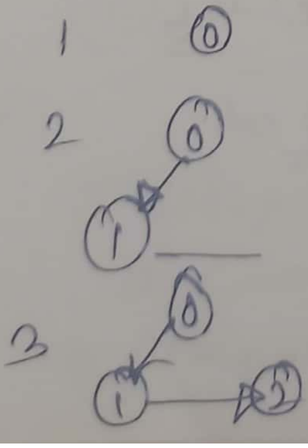
Soln - definition - 02 Mark
Explanation with example - 02 - Marks

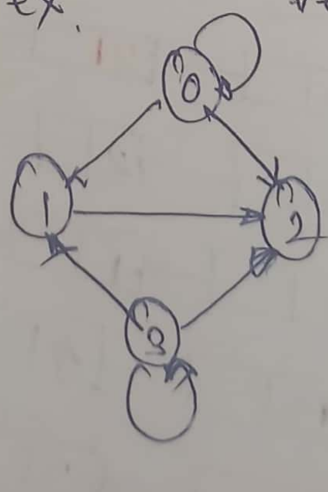
Refer Q. 06. (C) - definition as well as example.

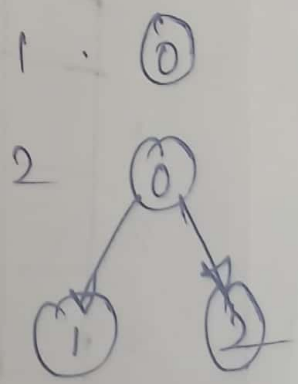
Q. 07 (b) Define Depth Search First (DFS) and Breadth Search First (BFS) terms of graph with example.

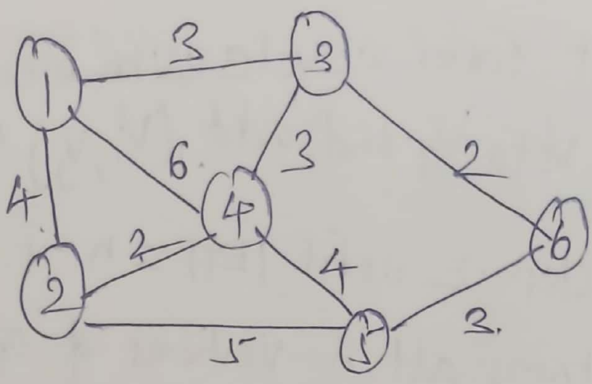
DFS	BFS
1. DFS uses stack	1. BFS uses a queue.
2. It is a recursive algorithm	2. It is an iterative algorithm.
3. It visits every node of a graph depth wise.	3. It visits every node of a graph level wise.
4. DFS is suited when the node being searched is farther off from the starting vertex.	4. BFS is suited when the node being searched is closer to the starting vertex.

5 DFS Graph







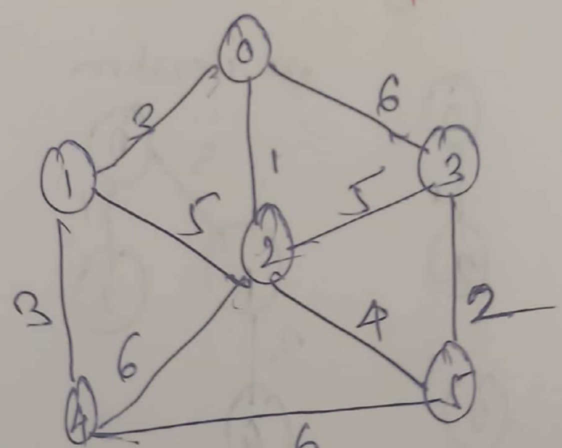


DFS — 1 → 2 → 4 → 3 → 6 → 5

BFS 1.
2, 3, 4
5, 6.

Q.7. (C) Explain in detail with suitable example prim's algorithm for finding out minimal spanning tree.

— Graph — 01M
explanation 02M
minimal spanning tree - 02



Let the graph G = (V, E) has n vertices

step - I choose vertex V_1 of G Let $V_T = (V_1)$
and $E_T = \{ \}$

step - II choose a nearest initially V_i of V_T that is adjacent to V_j $V_j \in V$ and for which

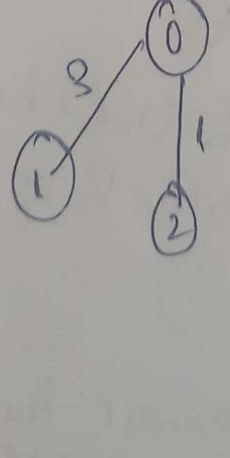
the edge
(V_i, V_j) doesn't form a cycle with member of E_T. Add V_i to V_T and add (V_i, V_j) to E_T

Step ③ - Repeat step 2 until $|E_T| = n-1$
Then V_T contains all n vertices of G and E_T contains the edges of the minimum cost spanning tree of Graph G.

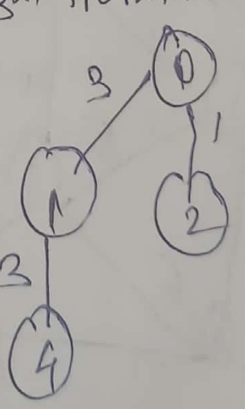
1st iteration

Select a new vertex with minimum distance from the spanning tree.
V = 2 from the distance[] array
distance[2] = 1 is minimum.
U = 0 as from[2] = 0 [i.e Node No 2 is at a distance 1 from Node number 0]
An edge (U, V) with weight 1 should be added to the tree.

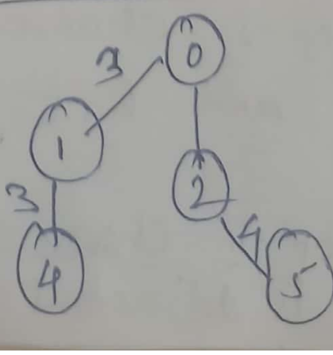
2nd iteration



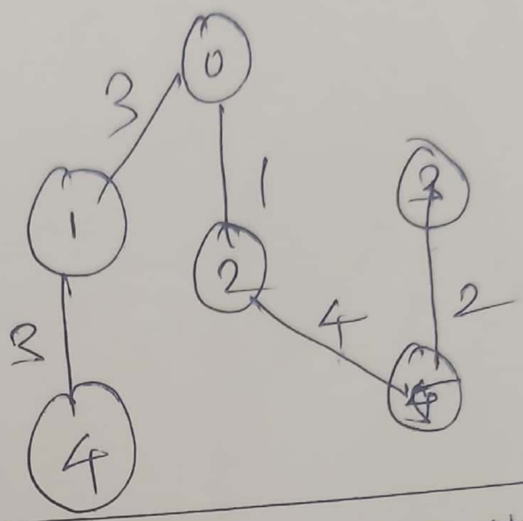
3rd iteration



4th iteration



5th iteration



Minimum spanning distance = 1 + 3 + 3 + 4 + 2
= 13

2021/3/16 10:58