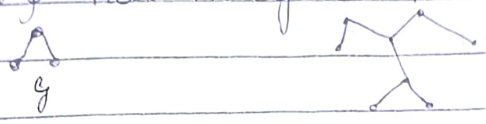
M-02

Tree

A graph 'g' is said to be tree if it is connected & has no cycles in it.

Eg: 

The tree has to be simple graph because loops & parallel edges form cycles. The tree possess at least 2 pendent vertices. A tree is called leaf.

A graph which is the tree is usually denoted by P.

Each connected component of a tree is called as forest.

Theorem 1 - In a tree there is only 1 path b/n every pair of vertices.

proof: Let t be a tree. Then t is a simple connected graph. Since t is connected there must be atleast 1 path b/n every 2 vertices. If there are 2 paths b/n a pair of vertices of t, the union of the path will become an cycle & t cannot be a tree. Thus b/n every pair of vertices in a tree their most exist one & only one path.

Theorem 2 - In a graph 'g' there is only 1 path b/n every pair of vertices, then g is a tree.

proof - Since there is a path b/n every pair of vertices in 'g' it is obvious that 'g' is connected. Since there is only 1 path b/n every pair of vertices, g cannot have a cycle. Because if there is a cycle then there with two path b/n 2 vertices on the cycle. Thus g is connected graph containing no cycles.

This means g is a tree.

Sapna
PAGE NO.
DATE / /

Theorem - A tree with n vertices has n-1 edges.

proof - We prove the theorem by induction on n.

The theorem is obvious for n=1, n=2, n=3

Assume that the theorem holds for all trees with 'k' vertices where k≥n. for a specified +ve integer K. consider a tree t with k+1 vertices. In this 't' let 'e' be an edge with 'n' vertices u & v. Since 't' is a tree, it has no cycles & therefore there exist no other edges or path b/n u & v. Hence deletion of 'e' from 't' will disconnect the graph & t-e consist of exactly 2 components say t_1 & t_2. Since t doesn't contain any cycle the components t_1 & t_2 also not contain any cycle. Hence t_1 & t_2 are trees on their own. Both of this trees have less than k+1 vertices each & therefore by assumption the theorem holds for t_1 & t_2 therefore, the total no. of vertices in t_1 & t_2 are taken together is k+1, the total no. of edges in t_1 & t_2 taken together is (k+1)-2 = k-1. But t_1 & t_2 taken together is t-e. Thus t-e contains k-1 edges. Consequently t has exactly k edges.

Thus, the theorem is true for k+1 vertices Hence by induction the theorem is true for all +ve integers n.

Directed tree - Let 'D' be a directed graph & 'G' be its underlying graph. We say that 'D' is a directed tree whenever 'G' is a tree.

Sapna
PAGE NO
DATE / /

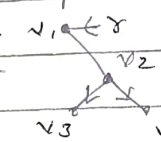
Thus directed tree is a directed graph whose underlying graph is a tree.

A directed tree 'T' is called a rooted tree if (i) T contains the unique vertex, called the root whose indegree = 0.
(ii) The indegree of all other vertices of T are equal to 1

Eg: 

In a rooted tree, we denote the root by 'r' & draw the diagram of the tree downward from an upper level to lower level, so that the arrows can be dropped.

A vertex of a rooted tree is said to be at the kth level are has level no. k if the path from r to V is of length k.

Eg: 

V_1 → 1st level - parent
V_2 → 2nd level - child
V_3, V_4 → 3rd level

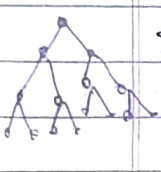
In a rooted tree a vertex whose outdegree is 0 is called a leaf & a vertex which is not a leaf is called an internal vertex.

m-ary tree - A rooted tree 'T' is called an m ary tree if every internal vertex of T is of outdegree ≤ m, i.e., if every internal vertex of T has atmost m children.

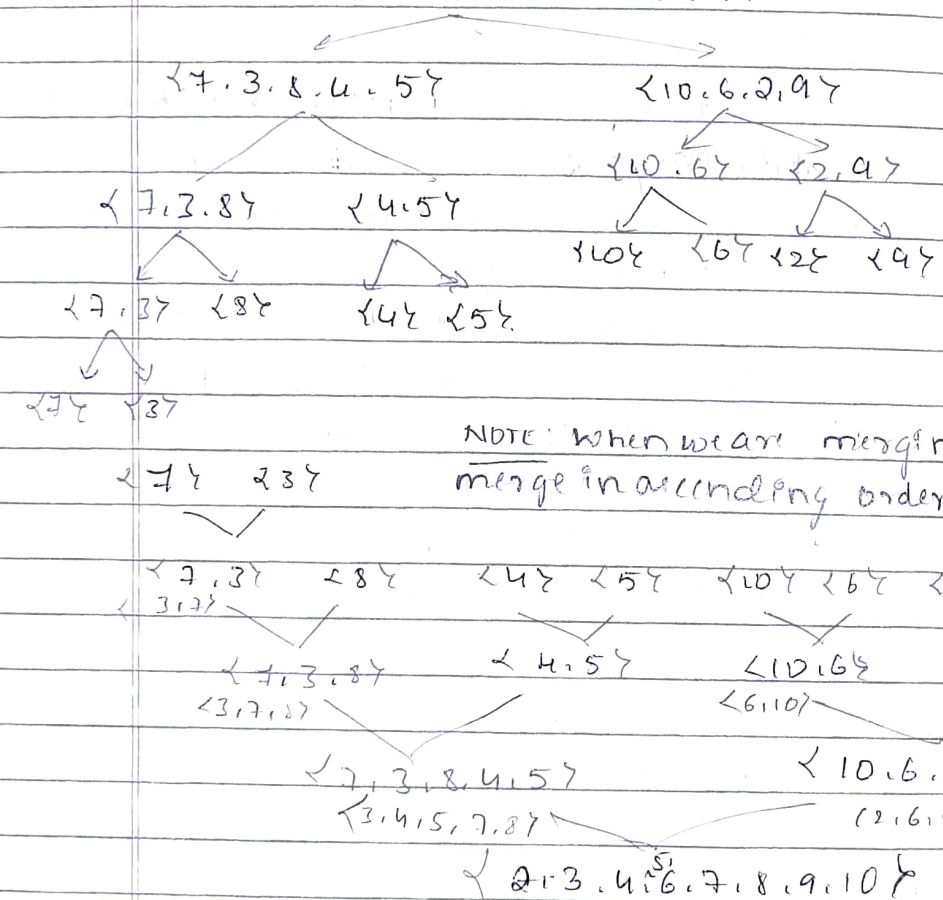
A rooted tree T is called an complete m-ary tree if every internal vertex of T is of outdegree m i.e., every internal vertex of T has exactly m children.

Sapna
PAGE NO.
DATE / /

Binary tree – An m-ary tree for which m equal to 2 is called a binary tree. In other words the rooted tree T is called a B.T. If every vertex of T is of outdegree ≤ 2, i.e., every vertex has atmost m childrens.



1. Using the mergesort method, sort the list 7, 3, 8, 4, 5, 10, 6, 2, 9



NOTE: When we are merging, merge in ascending order

2. {-1, 7, 4, 11, 5, -8, 15, -3, -2, 6, 10, 3}

Sapna
PAGE NO
DATE / /

Binary sequence - A sequence consist of only 0 & 1 is called a binary sequence or binary string.
Eg: 01, 001, 1101
The no. of element contained in a sequence is called it length. thus, Eg: 2, 3, 4 for 01, 001, 1101.

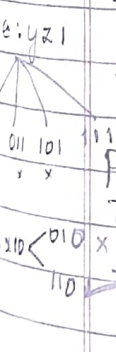
prefix code - Let 'P' be a set of binary sequence that represent a set of symbols. Then 'P' is called a prefix code, if no sequence in P is the prefix of any other sequence in P.
Eg: $P_1 = \{10, 0, 1101, 111, 1100\}$
$P_2 = \{01, 0, 101, 10, 1\}$ → This is not a prefix code. because 10 is replating. i.e. 10 & 101.

1. Consider the prefix code a: 111, b: 0, c: 1100, d: 1101, e: 10 using this code. decode the following sequence.
i) 1001111101 - e b a d.
ii) 101111001100011101 - e a e b c b d
iii) 11011111100010 - d a c e.

soln: The given seq is →
keeping the given code we split the given sequence into appropriate no of parts so that sequence can be decoded.

2. A code for a, b, c, d, e is given by a: 00, b: 01, c: 101, d: x10, e: yz1 where x, y, z ∈ {0, 1}. Determine x, y, z so that the given code is a prefix code.

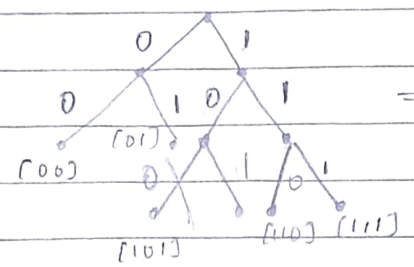
soln: If we set x=0, then the code for d is 010 which contains the code 01 of b. which is a prefix code already. ∴ we cannot take x=0. This implies x=1, & then d is 110. checking, for e. is taking y=0 & z=0 then, e has the code 001 which contains the code 00 of a i.e. a: 00 which is prefix. this implies e is not a prefix code. cont'd write i) & ii) & iv)



Sapna
PAGE NO
DATE / /

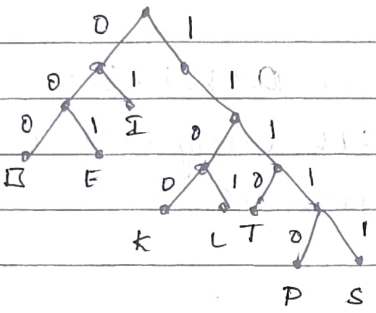
∴ the value of $a=1, y=1$ & $z=1$

3. Construct the Binary tree that represent the prefix code $P=\{00, 01, 101, 110, 111\}$.



4. Obtain a prefix code for the letters B, E, I, K, L, T, P, S. If the coding scheme is given by.

8M



i) find the codes for the word BEST & PIPE.

ii) Decode the string 11111 11111 011 0101 1110

Soln: B: 000, E: 001 I: 01, K: 1100

L: 1101 T: 1110 P: 11110 S: 11111

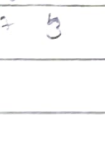
i) The code for BEST : 000001 11111 1110

PIPE : 11110 01 11110 001

ii) SPLIT

Weighted trees - Consider a set of 'n' +ve integers $w_1, w_2, w_3 \dots w_n$ where $w_1 \le w_2 \le w_3 \le \dots \le w_n$. Suppose we assign these integers to the 'n' leaves of a complete B-T in any 1-1 manner. the resulting tree is called complete weighted B-T with $w_1, w_2 \dots w_n$ weight. If $l(w_i)$ then $W(T) = \sum_{i=1}^{n} w_i\, l(w_i)$ is called the weight of the tree.

Ex:



> (l is the level no. of the leaf T which has the weight w_i is assigned

$W(T) = (2\times 2) + (2\times 3) + (1\times 4) = 14 + 6 + 4 = 24$

Sapna
PAGE NO
DATE / /

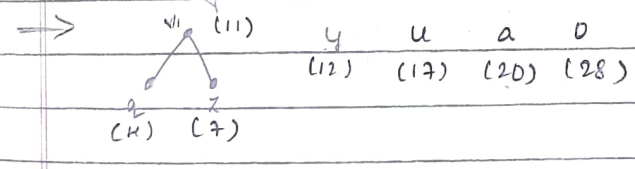
Optimal Tree - Given a set of weights, suppose we consider a set of all complete B-T to whose leaves, these weights are assigned. Here is tree in this set which carries the minimum weight is called an **optimal tree** for the weights.

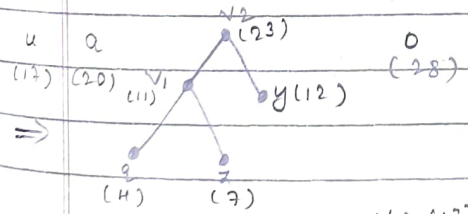
Optimal prefix code - All the vertices of the tree can be identified by binary sequence. The binary sequence through which the leaf are identified yield a prefix code for the symbol representing these leaves. this prefix code is known as **optimal prefix code**.

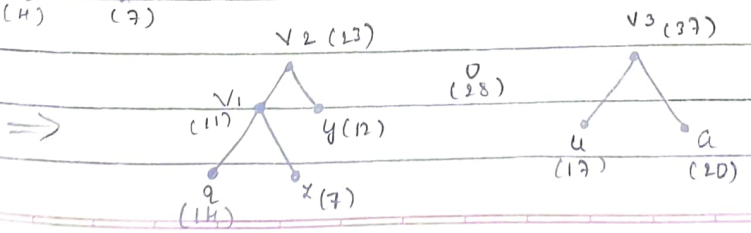
Construct a optimal prefix code for the symbols a, o, q, u, y, z that occurs with frequencies 20, 28, 4, 17, 12, 7.

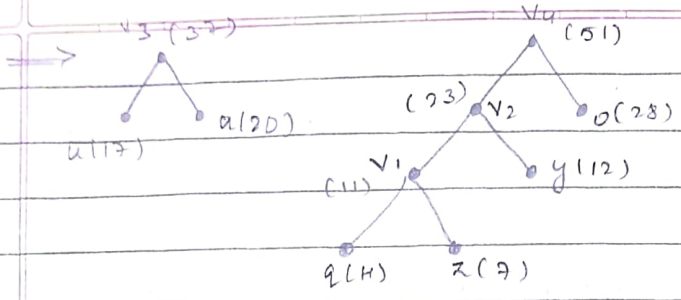
First arrange the give frequencies in ascending order.

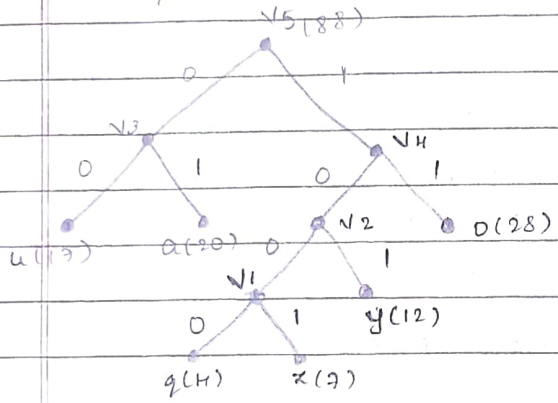
q z y u a o
(4) (7) (12) (17) (20) (28)











∴ The prefix codes are

u : 00 y : 101

a : 01 o : 11

q = 1000

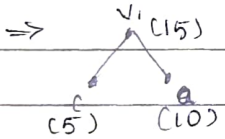
z = 1001

2) a b c d e f

10 30 5 15 20 15

c a d f e b

5 10 15 15 20 30

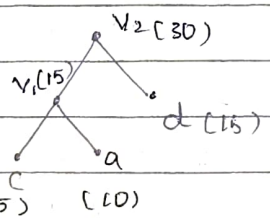


d f e b

15 15 20 30

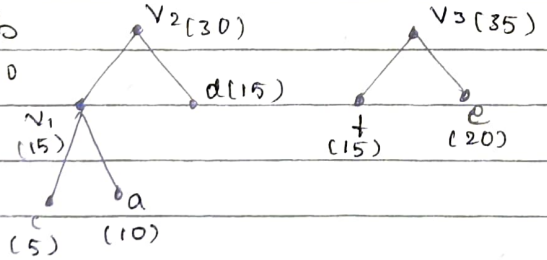
=> f e b

15 20 30

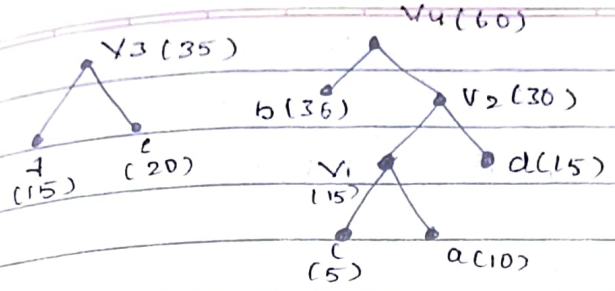


=> b

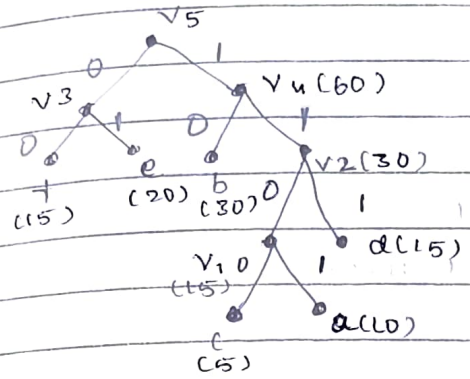
30



=>



=>



a : 1101 d : 111
b : 10 e : 01
c : 1100 f : 00

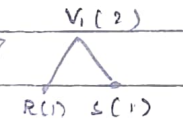
3. construct an optimal prefix code for the letters of the word PROPOSAL ACCEPTED.

=> P R O S A L C E T D.
3 1 2 1 2 1 2 2 1 1

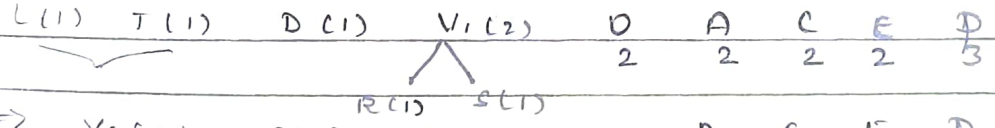
The given message consist of letters P,R,O,S,A,L,C,E,T,D with frequencies 3,1,2,1, 2,1,2,2,1,1

R S L T D O A C E P
1 1 1 1 1 2 2 2 2 3

=>

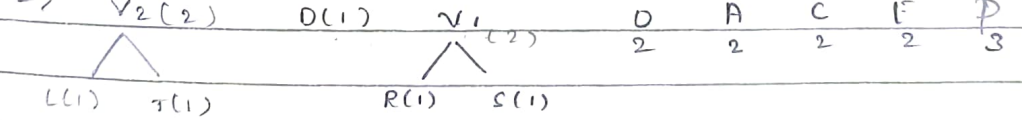


L T D O A C E P
1 1 1 2 2 2 2 3



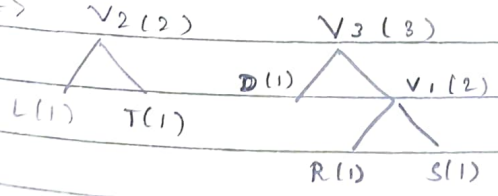
O A C E P
2 2 2 2 3

=>



O A C E P
2 2 2 2 3

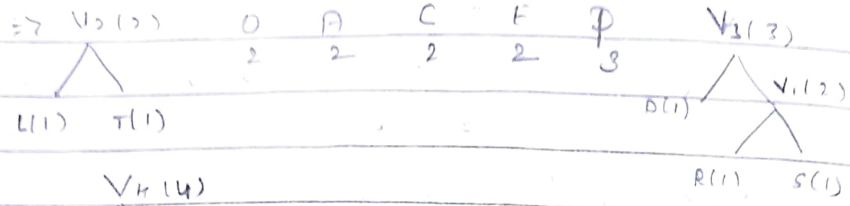
=>

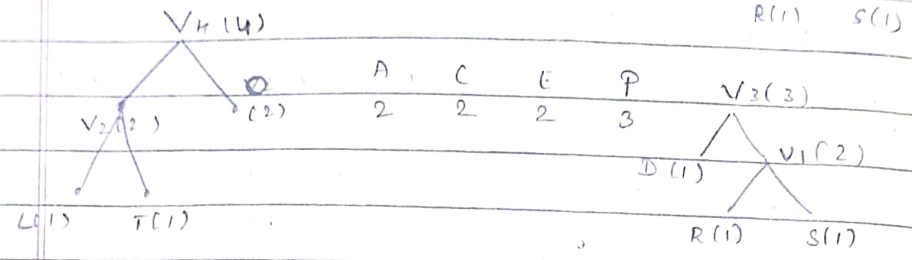


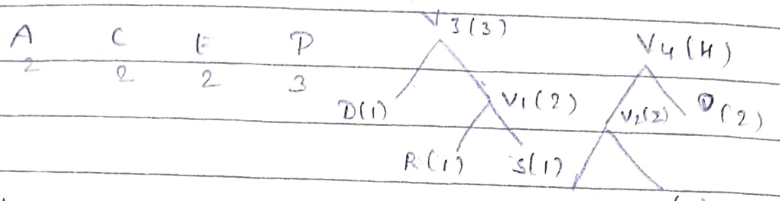
O A C E P
2 2 2 2 3

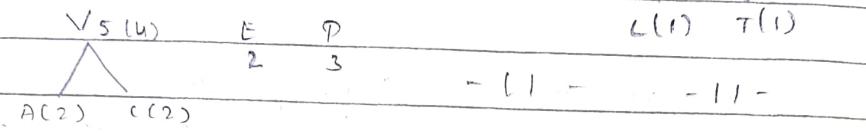
R : 110 S : 00 N : 10
G : 111 E : 01

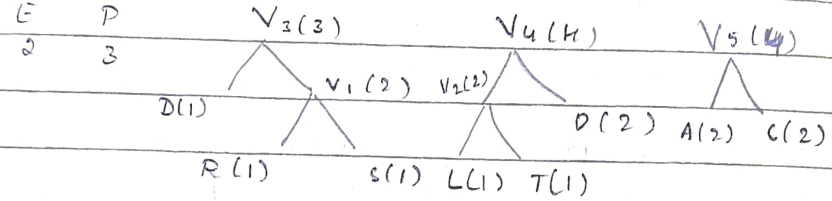
Sapna
PAGE NO
DATE / /

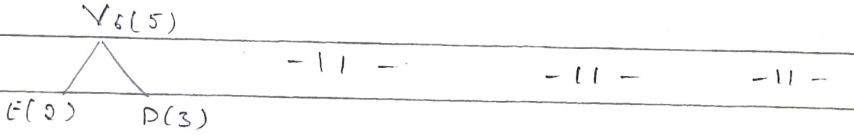


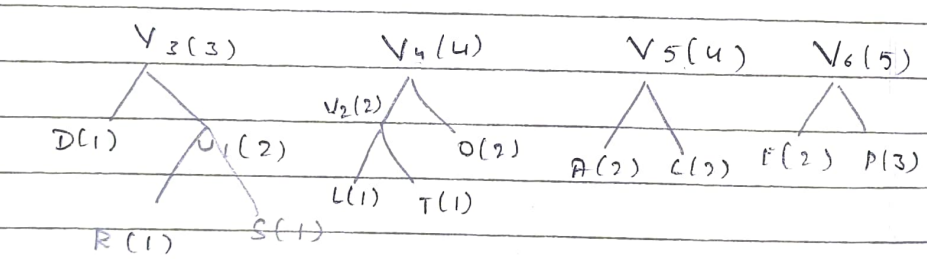


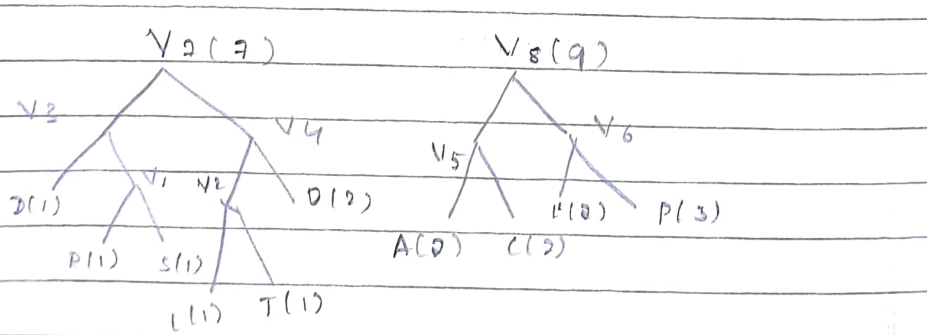


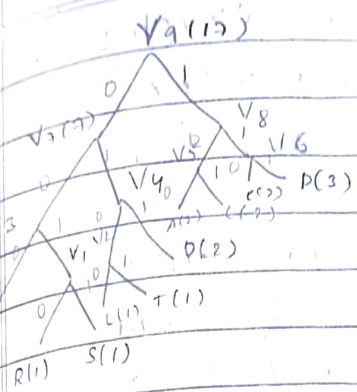












P: 111 L: 0101
R: 0010 C: 101
O: 011 E: 110
S: 0011 T: 0101
A: 100 D: 000

ENGINEERING

Sapna
PAGE NO
DATE / /

Theorem - Any connected graph with 'n' vertices & n-1 edges is a tree.

proof - Let 'G' be a connected graph with 'n' vertices & n-1 edges. Assume that 'G' is not a tree. Then 'G' contains a cycle say 'C'. Let 'E' be an edge in 'C'. The graph 'G' will not become disconnected if 'E' is deleted.

Then G-E is a connected graph. But on the other hand G-E has 'n' vertices & n-2 edges. Therefore it can be connected. This is a contradiction to G. Hence, G most not have a cycle. This means 'G' most be a tree.

Minimally connected graph - A connected graph is said to be minimally connected if the removal of any edge from it disconnects the graph.

Theorem - A connected graph is a tree if & only if it is minimally connected.

proof - Suppose 'G' is a connected graph which is not a tree. Then 'G' contains a cycle 'C'. The removal of any ledge 'E' from this cycle will not make the graph disconnected. Therefore 'G' is not minimally connected. Thus, if a connected graph is not a tree then it is not minimally connected. This is equivalent to saying that if a connected graph is minimally connected then it is a tree.

Conversely, suppose 'G' is a connected graph which is not minimally connected. Then there exist an edge 'F' in 'G' such that G-F is connected. Therefore F most lie on some

'G' is not a tree means there will be a cycle.

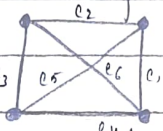
connected graph means there should be n-1 edges

Sapna
PAGE NO
DATE / /

cycle in G. This implies that G is not a tree. Thus, if a connected graph is not minimally connected then it is not a tree. This is equivalent to saying that if a connected graph is a tree then it is minimally connect-ed.

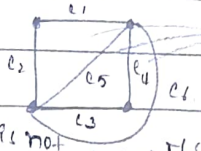
M-01

planar graph - A planar graph is one that can be drawn in a plane without any edges crossing. Eg:



K4

=>

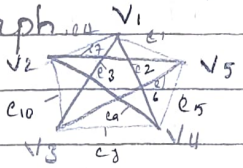


3 planes are there.

↳ This is not planar graph.

→ This is planar

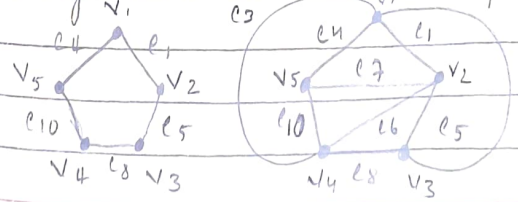
prove that complete graph K5 is a non-planar graph.



proof - The complete graph K5 there are 5 vertices & 10 edges in it. The vertices are named as V_1, V_2, V_3, V_4, V_5 & the edges are labelled as $e_1, e_2, \ldots e_{10}$.

The drawing of K5, the 5 edges e_1, e_5, e_8, e_{10} & e_4 form an pentagonal cycle. And the remaining 5 edges e_2, e_3, e_6, e_7, e_9 are all inside this cycle & intersect at point other than the vertices.

Let us try to draw a diagram of K5 in which edges meet only at the vertices. Let us start drawing our new K5 with a pentagonal cycle.

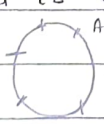


Consider the edge $e_7 = \{v_2, v_5\}$ can be drawn inside or outside the pentagonal cycle. Suppose we draw it inside as shown in fig. the other case follows similar. Now consider the edge $e_2 = \{v_1, v_3\}$ & $e_3 = \{v_1, v_4\}$. If we draw these edges also inside the pentagon, there will intersect e_7. i.e., they cross e_7 at point which are not vertices. Therefore, let us draw outside both of them outside the pentagon.

Consider, the edge $e_6 = \{v_2, v_4\}$ we draw this edge outside the pentagon it intersect the edge e_2. Therefore, let us draw e_6 inside the pentagon. lastly, the edge $e_9 = \{v_3, v_5\}$ draw outside the pentagon. it intersect the edge e_3. & if we draw it inside it intersect the edge e_6. this demonstrates that in every possible plane drawing of K_5, atleast 2 edges of K_5 intersect at a point which is not a vertex of K_5. This proves that K_5 is non-planar.

Theorem – P.T a connected graph 'G' remains connected after removing an edge 'E' from 'G' if & only if 'E' is a part of some cycle in 'G'.

Suppose 'E' is a part of some cycle in G. Then the 2 vertices of 'E' [say a & b] are joined by atleast 2 parts. 1 of which is E & the other is C – E.

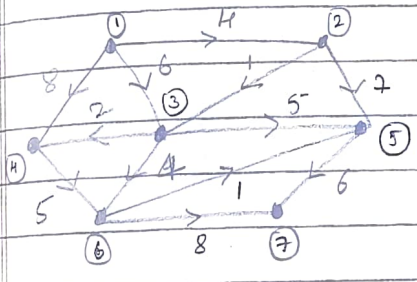


Hence the removal of 'E' from 'G' will not affect the connectivity of 'G' because

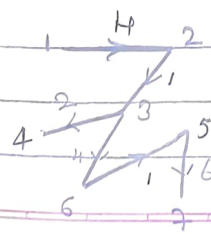
even after removal of 'E' the end vertices of 'E' remain connected through the path

conversely, suppose 'E' is not a part of any cycle in 'G'. Then the end vertices of 'E' are connected by utmost 1 part. Hence, the removal of 'E' from G disconnects these end points. This means G-E is a disconnected graph. Thus if 'E' is not a part of any cycle in 'G' then G-E is disconnected. this is equivalent to saying that if G-E is connected then E belongs to some cycle in G. Thus, the required result.

⇒ Dijkstra's Algorithm.



Selected vertex	visited vertex	d[2]	d[3]	d[4]	d[5]	d[6]	d[7]
1	~~1~~ ⟨1,2⟩ [4]	6	8	∞	∞	∞	
2	⟨1,2,3⟩	[5]	8	11	∞	∞	
3	⟨1,2,3,4⟩		[7]	10	9	∞	
4	⟨1,2,3,6⟩			10	[9]	∞	
6	⟨1,2,3,6,5⟩			[10]		17	
5	⟨1,2,3,6,5,7⟩					[16]	

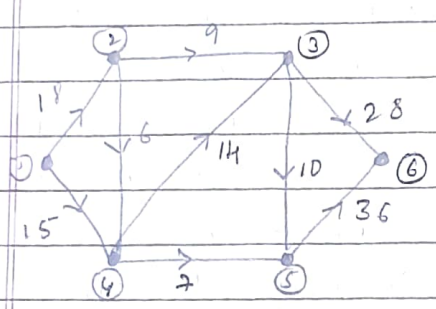


the following are the shortest path from vertex 1 to all other vertex.

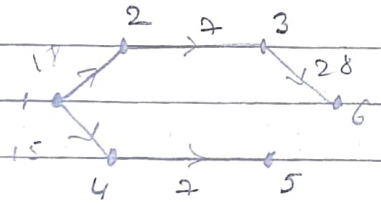
Sapna
PAGE NO.
DATE / /

i) path from {1,2} : 1 → 2 has weight 4
ii) path from {1,3} : 1 → 2 → 3 has weight 5
iii) path from {1,4} : 1 → 2 → 3 → 4 has weight 7
iv) path from {1,6} : 1 → 2 → 3 → 6 has weight 9
v) path from {1,7} : 1 → 2 → 3 → 6 → 5 → 7 has weight 11
vi) path from {1,5} : 1 → 2 → 3 → 6 → 5 has weight 10

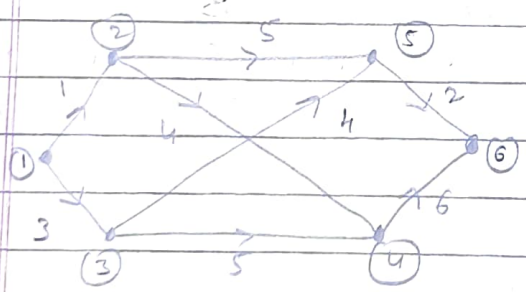
2.



selected vertex	visited vertex	d[2]	d[3]	d[4]	d[5]	d[6]
1	{1,4}	18	∞	[15]	∞	∞
4	{1,2}	[18]	29		22	∞
2	{1,4,5}		[27]		[22]	∞
5	{1,2,3}					58
3	{1,2,3,6}					[55]



3



Sapna
PAGE NO.
DATE / /

Selected vertex	visited vertex	d[2]	d[3]	d[4]	d[5]	d[6]
1	{1,2}	[1]	3	∞	∞	∞
2	{1,3}		[3]	5	6	∞
3	{1,2,4}			[5]	6	∞
4	{1,2,5}				[6]	11
5	{1,2,5,6}					[8]

1. P.T a complete graph K_n with 'n' vertices contains $n(n-1)/2$

Soln: A complete graph K_n is defined as a graph in which every pair of distinct vertices is connected by an unique edge.
A complete graph with 'n' vertices we want to count how many unique pair of vertices can be formed. Each pair of vertices corresponds to one edge. the no. of ways to choose '2' vertices from 'n' vertices is given by binomial co-efficient nC_2.

$$nC_2 = \frac{n!}{2!(n-2)!} = \frac{n(n-1)(n-2)!}{2(n-2)!} = \frac{n(n-1)}{2}$$

Spanning Trees - A spanning tree of a connected graph 'G' is a tree containing all the vertices of 'G'. A spanning tree of a graph 'G' is a maximal tree subgraph of that graph.