Reg. No.____________ Name:______________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE MODEL EXAMINATION,NOVEMBER 2017

Course Code: **CS 309**

Course Name: **GRAPH THEORY AND COMBINATORICS** (CS)

Max. Marks:100 Duration: 3 Hours

PART A

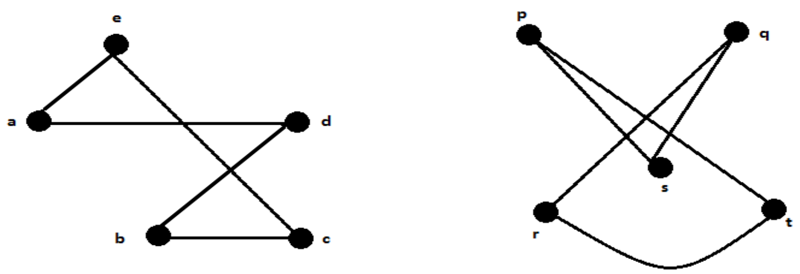
(Answer ALL Questions)

1. Prove that the number of vertices of odd degree in a graph is always even. (3)
2. Differentiate a walk, path and circuit in a graph. (3)
3. Define Hamiltonian circuit, Euler graph. Give one example for each. (3)
4. Define Directed graphs? Differentiate Symmetric digraphs and Asymmetric digraphs. (3)

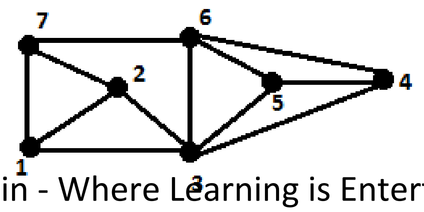
PART B

(Answer Any Two Questions)

5.
(a) Define Isomorphism of graphs. Verify whether the following graphs are isomorphic or not. (4.5)



(b) State Handshaking Theorem? Verify Handshaking Theorem for the following graph (4.5)



6. Prove that a connected graph G is an Euler graph if all vertices of G are of even degree. (9)

7. Prove the following theorems

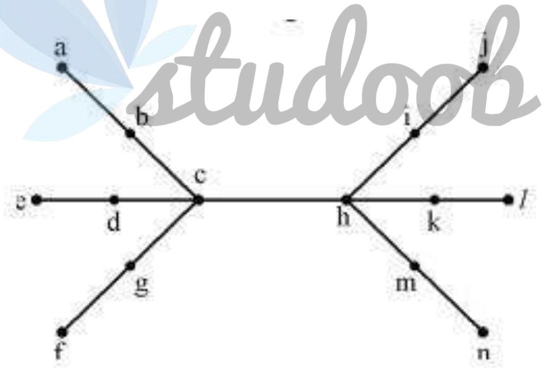
(a) The total number of different, not edge disjoint, Hamiltonian circuits in a complete graph of n vertices is $\frac{(n-1)!}{2}$ (4.5)

(b) A connected graph G is an Euler graph if and only if it can be decomposed into circuits. (4.5)

PART C

(Answer ALL Questions)

8. Define eccentricity of a vertex ? Using the property of eccentricity of a vertex, find every vertex that is the centre of the following tree (3)



9. Define Branch, Chord , Rank and Nullity in a spanning tree with an example. (3)

10. Prove that every cut set in a connected graph G must contain at least one branch of every spanning tree of G. (3)

11. Define edge connectivity,vertex connectivity and separable graph. Give one example for each. (3)

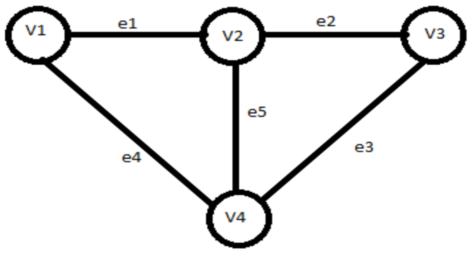
PART D

(Answer Any Two Questions)

12. Prove that a tree with n vertices has $n-1$ edges. (9)

13. State and prove Cayley's Theorem. (9)

14.
    (a) Define Cut-sets, Fundamental cut-sets? List any five cut-sets of the following graph and find all fundamental cut-sets ? (4.5)



    (b) Prove that every circuit has an even number of edges common with any cut-set. (4.5)

PART E

(Answer Any Four Questions)

15. Write an algorithm to determine the connectedness and components of a given graph. (10)

16. Write an algorithm to find spanning tree or spanning forest of a given graph. (10)

17. Write an algorithm for finding shortest path between two specified vertices. (10)

18. Derive the relationship between incidence matrix, fundamental circuit matrix and fundamental cut-set matrix of a connected graph. (10)

19. Write any two matrix representations of a graph. (10)

20. Prove the following theorems

    (a) Prove that the rank of cut-set matrix is equal to the rank of the incidence matrix, which equals the rank of graph G (5)

    (b) Prove that if B is a circuit matrix of a connected graph G with e edges and n vertices, rank of B $= e - n + 1$ (5)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE MODEL EXAMINATION,NOVEMBER 2017

Course Code: **CS 309**

Course Name: **GRAPH THEORY AND COMBINATORICS** (CS)

PART A

1. The sum of the degrees of all vertices in graph G can be expressed as the sum of two sums , each taken over vertices of even and odd degrees

   That is $\sum_{i=1}^{n} d(v_i) = \sum_{even} d(v_j) + \sum_{odd} d(v_k)$ ....(1)

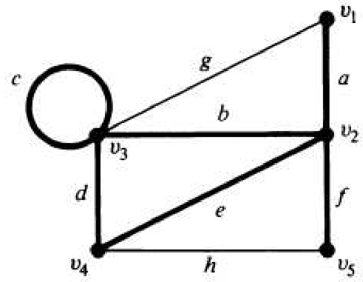
   Since the left hand side of the above equation is even ($\sum_{i=1}^{n} d(v_i) = 2e$), and the first expression on the right hand side is even (being a sum of even numbers),the second expression must also be even.

   $\sum_{odd} d(v_k)$ =**an even number** ....(2)

   Because in equation (2) each $d(v_k)$ is odd , the total number of terms in the sum must be even to make the sum an even number. Hence the theorem.

2. A **walk** is defined as a finite alternating sequence of vertices and edges beginning and ending with vertices, such that each edge is incident with the vertices preceding and following it.No edge appears more than once in a walk. A vertex may appear more than once .

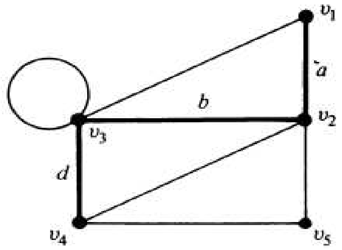
   <u>Example</u>



V_1 a V_2 b V_3 c V_3 d V_4 e V_2 f V_5 is a walk

Path-An open walk in which no vertex appears more than once is called a path.A path does not intersect itself. The number of edges in a path is called the length of a path

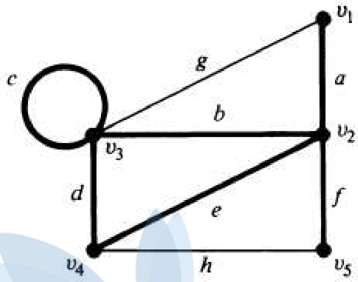
Example



V_1 a V_2 b V_3 d V_4 is a path

Circuit- A closed walk in which no vertex (except the initial and the final vertex) appears more than once is called a circuit.Every vertex in a circuit is of degree two.

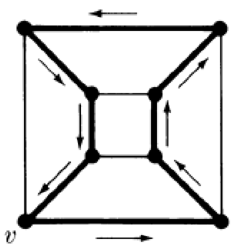
Example



V_2 b V_3 d V_4 e V_2 is a circuit

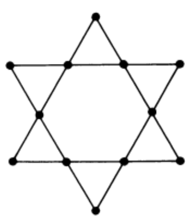
3. **Hamiltonian circuit** in a connected graph is defined as a closed walk that traverses every vertex of G exactly once,except the starting vertex at which the walk also terminates.

Example



Euler graph –If some closed walk in a graph contains all the edges of the graph , then the walk is called an Euler line and the graph is called Euler graph.

Example



4.

A *directed graph* (or a *digraph* for short) G consists of a set of vertices $V = \{v_1, v_2, \ldots\}$, a set of edges $E = \{e_1, e_2, \ldots\}$, and a mapping Ψ that maps

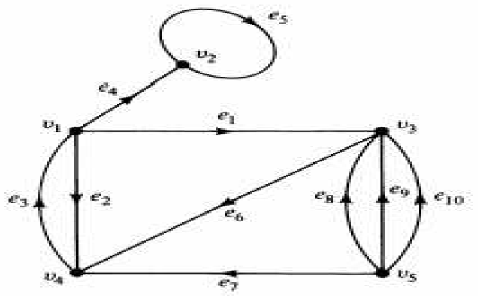


Fig. 9-1 Directed graph with 5 vertices and 10 edges.

every edge onto some *ordered* pair of vertices (v_i, v_j). As in the case of undirected graphs, a vertex is represented by a point and an edge by a line segment between v_i and v_j with an arrow directed from v_i to v_j. For example, Fig. 9-1 shows a digraph with five vertices and ten edges. A digraph is also referred to as an *oriented graph*.†

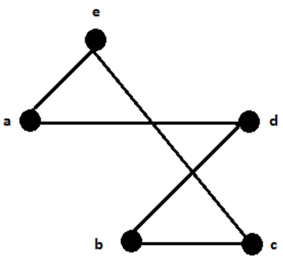
Asymmetric Digraphs: Digraphs that have at most one directed edge between a pair of vertices, but are allowed to have self-loops, are called *asymmetric* or *antisymmetric*.

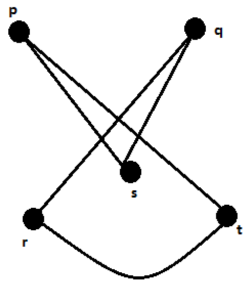
Symmetric Digraphs: Digraphs in which for every edge (a, b) (i.e., from vertex a to b) there is also an edge (b, a).

5.

(a) Two graphs G and G' are said to be isomorphic if there is a one to one correspondence between their vertices and between their edges such that the incidence relationship is preserved.In other words,suppose that edge e is incident on vertices v1 and v2 in G .then the corresponding edge e' in G' must be incident on the vertices v_1' and v_2' that correspond to v1 and v2 respectively. Two isomorphic graphs must have

- The same number of vertices
- The same number of edges
- An equal number of vertices with a given degree





The number of vertices in both graph is 5. The number of edges in both graph is 5.

In the first graph,

Degree of vertex a is, d(a)=2

Degree of vertex b is, d(b)=2

Degree of vertex c is, d(c)=2

Degree of vertex d is, d(d)=2

Degree of vertex e is, d(e)=2

In the second graph,

Degree of vertex p is, d(p)=2

Degree of vertex q is, d(q)=2

Degree of vertex r is, d(r)=2

Degree of vertex s is, d(s)=2

Degree of vertex t is, d(t)=2

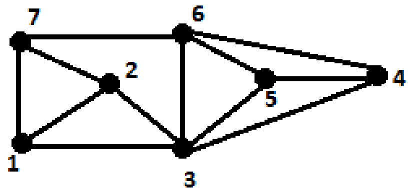
Both graphs have equal number of vertices with degree 2.

Therefore the two graphs are isomorphic.

(b) Handshaking theorem

Consider a graph G with e edges and n vertices $v_1, v_2 .... v_n$.Since each edge contributes two degrees, the sum of the degrees of all vertices in G is twice the number of edges in G. That is

$$\sum_{i=1}^{n} d(v_i) = 2e$$



The total number of edges , e =12

Degree of vertex 1, d(1)=3

Degree of vertex 2, d(2)=3

Degree of vertex 3, d(3)=5

Degree of vertex 4, d(4)=3

Degree of vertex 5, d(5)=3

Degree of vertex 6, d(6)=4

Degree of vertex 7, d(7)=3

d(1)+d(2)+d(3)+d(4)+d(5)+d(6)+d(7)=3+3+5+3+3+4+3

=24

$$\sum_{i=1}^{n} d(v_i) = 24$$

$$2 \times e = 2 \times 12 = 24$$

Therefore

$$\boldsymbol{\sum_{i=1}^{n} d(v_i) = 2e}$$

6. *Proof:* Suppose that G is an Euler graph. It therefore contains an Euler line (which is a closed walk). In tracing this walk we observe that every time the walk meets a vertex v it goes through two "new" edges incident on v—with one we "entered" v and with the other "exited." This is true not only of all intermediate vertices of the walk but also of the terminal vertex, because we "exited" and "entered" the same vertex at the beginning and end of the walk, respectively. Thus if G is an Euler graph, the degree of every vertex is even.

To prove the sufficiency of the condition, assume that all vertices of G are of even degree. Now we construct a walk starting at an arbitrary vertex v and going through the edges of G such that no edge is traced more than once. We continue tracing as far as possible. Since every vertex is of even degree, we can exit from every vertex we enter; the tracing cannot stop at any vertex but v. And since v is also of even degree, we shall eventually reach v when the tracing comes to an end. If this closed walk h we just traced includes all the edges of G, G is an Euler graph. If not, we remove from G all the edges in h and obtain a subgraph h' of G formed by the remaining edges. Since both G and h have all their vertices of even degree, the degrees of the vertices of h' are also even. Moreover, h' must touch h at least at one vertex a, because G is connected. Starting from a, we can again construct a new walk in graph h'. Since all the vertices of h' are of even degree, this walk in h' must terminate at vertex a; but this walk in h' can be combined with h to form a new walk, which starts and ends at vertex v and has more edges than h. This process can be repeated until we obtain a closed walk that traverses all the edges of G. Thus G is an Euler graph. ■

7.

(a) **Proof:** Let G be a complete graph with n vertices. Starting from any vertex we have n-1 edges to choose from first vertex, (n-2) edges to choose from second vertex ,(n-3) edges to choose from third vertex and so on.These being independent choices we get (n-1)! Possible number of choices.This number is, however, divided by 2 , because each Hamiltonian circuit has been counted twice.

Therefore The total number of different, not edge disjoint, Hamiltonian circuits in a complete graph of n vertices is $\frac{(n-1)!}{2}$.

(b)

Proof: Suppose graph G can be decomposed into circuits; that is, G is a union of edge-disjoint circuits. Since the degree of every vertex in a circuit is two, the degree of every vertex in G is even. Hence G is an Euler graph.

Conversely, let G be an Euler graph. Consider a vertex v_1. There are at least two edges incident at v_1. Let one of these edges be between v_1 and v_2. Since vertex v_2 is also of even degree, it must have at least another edge, say between v_2 and v_3. Proceeding in this fashion, we eventually arrive at a vertex that has previously been traversed, thus forming a circuit Γ. Let us remove Γ from G. All vertices in the remaining graph (not necessarily connected) must also be of even degree. From the remaining graph remove another circuit in exactly the same way as we removed Γ from G. Continue this process until no edges are left. Hence the theorem. ■

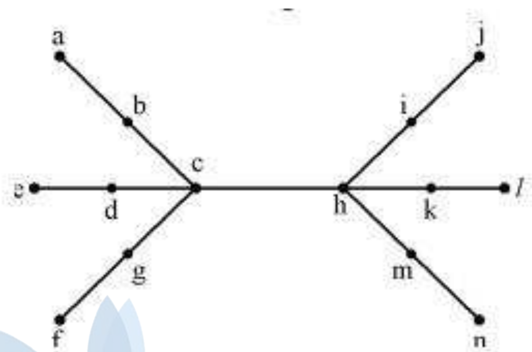
PART C

(Answer ALL Questions)

8. The eccentricity $E(v)$ of a vertex v in a graph G is the distance from v to the vertex farthest from v in G. That is

$$E(v) = \max_{vi \, \varepsilon \, G} d(v, v_i)$$

A vertex with minimum eccentricity in a graph G is called a **center** of G.



For the above graph,

Eccentricity of vertex a, E(a)=5

Eccentricity of vertex b, E(b)=4

Eccentricity of vertex c, E(c)=3

Eccentricity of vertex d, E(d)=4

Eccentricity of vertex e, E(e)=5

Eccentricity of vertex f, E(f)=5

Eccentricity of vertex g, E(g)=4

Eccentricity of vertex h, E(h)=3

Eccentricity of vertex i, E(i)=4

Eccentricity of vertex j, E(j)=5

Eccentricity of vertex k, E(k)=4

Eccentricity of vertex l, E(l)=5

Eccentricity of vertex m, E(m)=4

Eccentricity of vertex n, E(n)=5

Vertices with minimum eccentricity are vertex c, and vertex h.

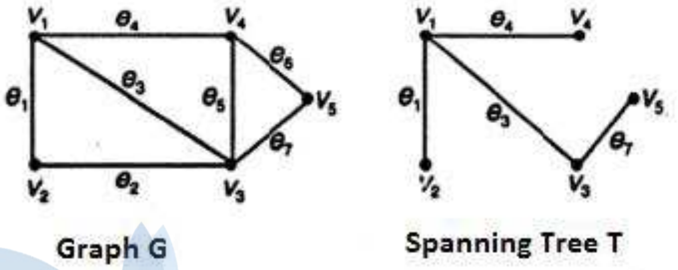
E(c)=3

E(h)=3

Therefore vertices c and h are the two centers of the graph.

9. A tree T is said to be a spanning tree of a connected graph G if T is a sub graph of G and T contains all vertices. An edge in a spanning tree T is called a **branch** of T. An edge of G is not in a given spanning tree T is called a **chord** (tie or link)

EXAMPLE :



The edges e_1, e_3, e_4, e_7 are called branches of T.
The edges e_2, e_5, e_6 are called chords with respect to T.

A graph G with n number of vertices, e number of edges, and k number of components with the following constraints $n - k \geq 0$ and $e - n + k \geq 0$.

Rank $r = n - k$

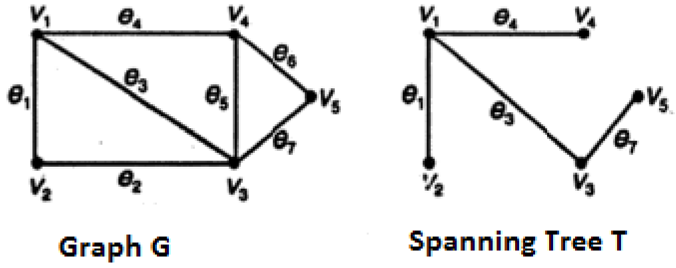
Nullity $\mu = e - n + k$

Rank of G = number of branches in any spanning tree of G

Nullity of G = number of chords with respect to spanning tree of G

Rank+Nullity=e, Total number of edges

Example



Rank, $r = n - k$

$= n - 1$

$= 5 - 1$

$= 4$, Number of branches

Nullity, $\mu = e - n + k$

$= 7 - 5 + 1$

$= 2 + 1$

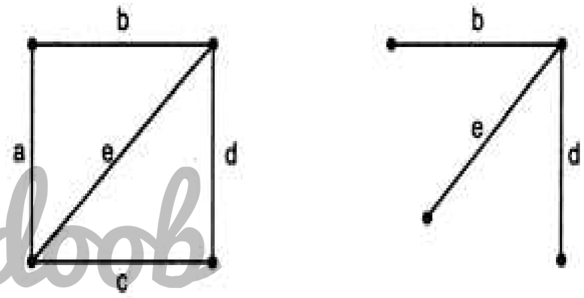
$=3$, Number of chords

Rank+Nullity$= 4 + 3 = 7$, Total number of edges

10.

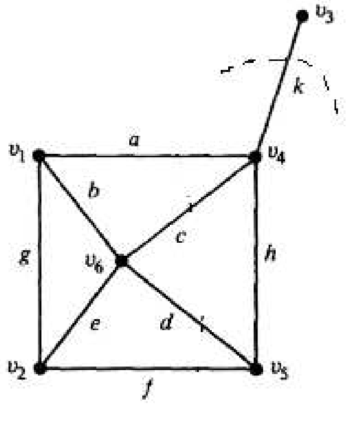
Proof. Let S be a cut-set of G. Let T be a spanning tree of G. Suppose S does not contain any branch of T. Then all the edges of T present in G-S. It means G-S is a connected graph. It means S is not a cut-set. Hence a cut-set must contain at least one branch of a spanning tree of G.

$[a, c]$ is not a cut-set so it should contain one branch of T to become a cut-set.



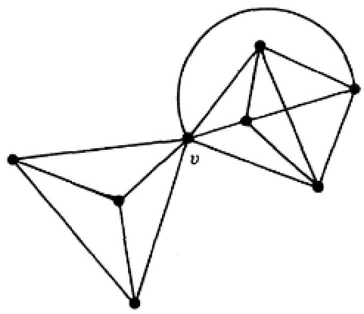
11.

Edge Connectivity: Each cut-set of a connected graph G consists of a certain number of edges. The number of edges in the smallest cut-set (i.e., cut-set with fewest number of edges) is defined as the *edge connectivity* of G. Equivalently, the edge connectivity of a connected graph† can be defined as the minimum number of edges whose removal (i.e., deletion) reduces the rank of the graph by one. The edge connectivity of a tree, for instance, is one.



Edge connectivity=1

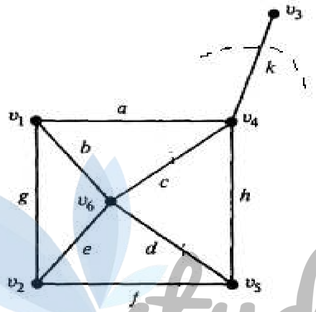
Vertex connectivity of a connected graph G is defined as the minimum number of vertices whose removal from G leaves the remaining graph disconnected.



Vertex Connectivity=1

Separable graph:A connected graph is said to be separable if its vertex connectivity is one.All other connected graphs are called nonseparable.In a separable graph ,a vertex whose removal disconnects the graph is called articulation point(cut vertex).

Example



V_4 is a cut vertex

PART D

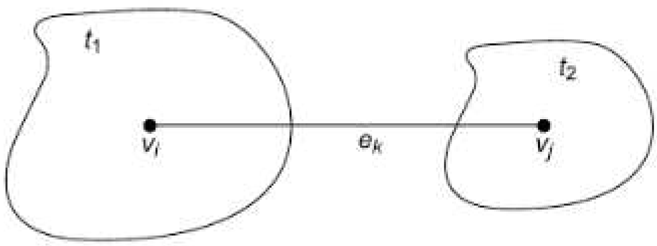
12. The theorem will be proved by induction on the number of vertices

Basic step: A tree with n vertices has n-1 edges is true for n=1

Induction Hypothesis:Assume that the theorem holds for all trees with fewer than n vertices.

Induction step:

Now consider a tree with n vertices. Let e_k be any edge in T where end vertices are v_i and v_j. Since T is a tree, there is no other path between v_i and v_j except e_k.Therefore the deletion of e_k from T will disconnect the graph.



T- e_k consists of exactly two components t1 and t2. Since there were no circuits in T to begin with each of these component is a tree. Both these trees t1 and t2 have fewer than n vertices each.

Therefore by induction hypothesis, each contains one less edge than the number of vertices in it.

$$|E(T_1)| = |V(T_1)| - 1$$

$$|E(T_2)| = |V(T_2)| - 1$$

T- e_k consists of n-2 edges. Hence T has exactly n-1 edges

13. **Cayley's Theorem**

There are n^{n-2} labeled trees with n vertices ($n \geq 2$).

Proof of Theorem 3-10: Let the n vertices of a tree T be labeled $1, 2, 3, \ldots, n$. Remove the pendant vertex (and the edge incident on it) having the smallest label, which is, say, a_1. Suppose that b_1 was the vertex adjacent to a_1. Among the remaining $n - 1$ vertices let a_2 be the pendant vertex with the smallest label, and b_2 be the vertex adjacent to a_2. Remove the edge (a_2, b_2). This operation is repeated on the remaining $n - 2$ vertices, and then on $n - 3$ vertices, and so on. The process is terminated after $n - 2$ steps, when only two vertices are left. The tree T defines the sequence

$$(b_1, b_2, \ldots, b_{n-2}) \tag{10-3}$$

uniquely. For example, for the tree in Fig. 10-1 the sequence is (1, 1, 3, 5, 5, 5, 9). Note that a vertex i appears in sequence (10-3) if and only if it is not pendant (see Problem 10-2).

Conversely, given a sequence (10-3) of $n - 2$ labels, an n-vertex tree can be

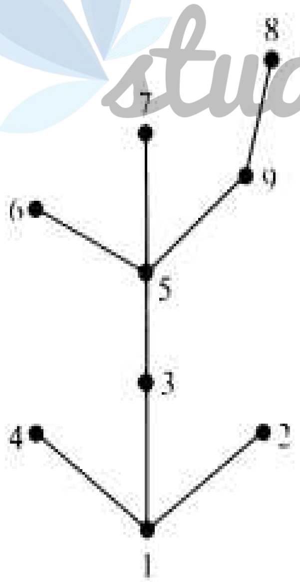


Fig. 10-1 Nine-vertex labeled tree, which yields sequence (1, 1, 3, 5, 5, 5, 9).

constructed uniquely, as follows: Determine the first number in the sequence

$$1, 2, 3, \ldots, n \tag{10-4}$$

that does not appear in sequence (10-3). This number clearly is a_1. And thus the edge (a_1, b_1) is defined. Remove b_1 from sequence (10-3) and a_1 from (10-4). In the remaining sequence of (10-4) find the first number that does not appear in the remainder of (10-3). This would be a_2, and thus the edge (a_2, b_2) is defined. The construction is continued till the sequence (10-3) has no element left. Finally, the last two vertices remaining in (10-4) are joined. For example, given a sequence

$$(4, 4, 3, 1, 1),$$

we can construct a seven-vertex tree as follows: (2, 4) is the first edge. The second is (5, 4). Next, (4, 3). Then (3, 1), (6, 1), and finally (7, 1), as shown in Fig. 10-2.

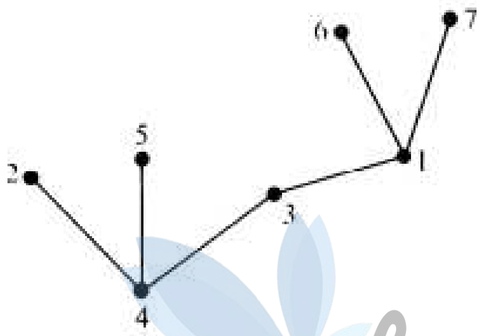


Fig. 10-2 Tree constructed from sequence (4, 4, 3, 1, 1).

For each of the $n - 2$ elements in sequence (10-3) we can choose any one of n numbers, thus forming

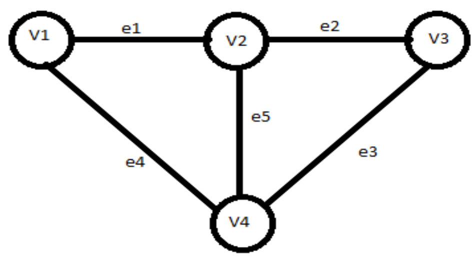
$$n^{n-2} \tag{10-5}$$

$(n - 2)$-tuples, each defining a distinct labeled tree of n vertices. And since each tree defines one of these sequences uniquely, there is a one-to-one correspondence between the trees and the n^{n-2} sequences. Hence the theorem. ■

14.

(a) Cut is a partition of a graph into two subsets S and T. **Cut-set** of a cut is the set of edges that have one end point in S and the other endpoint in T.Cut-set is a set of edges whose removal from G leaves G disconnected provided removal of no proper subset of these edges disconnects G.

A cut-set S containing exactly one branch of a spanning tree T, is called **fundamental cut set** with respect to T.



The cut-sets of the above graph are

$$\{e1, e4\}$$

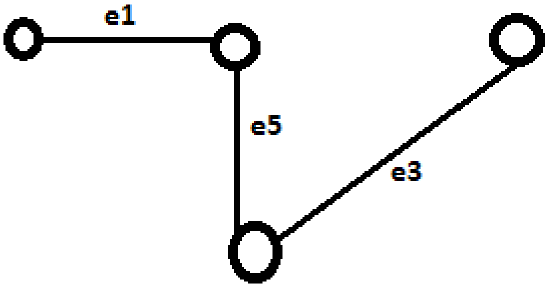
$$\{e2, e3\}$$

$$\{e1, e5, e3\}$$

$$\{e2, e5, e4\}$$

$$\{e4, e5, e3\}$$

Spanning tree of the above graph is



Fundamental cut-sets are

$$\{e1, e4\}$$

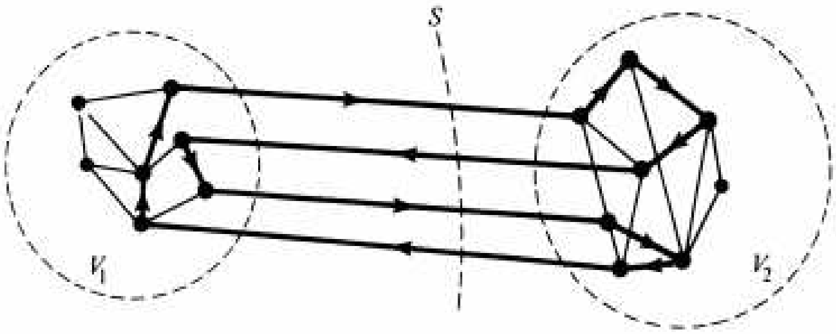
$$\{e2, e3\}$$

$$\{e2, e5, e4\}$$

(b)

Proof: Consider a cut-set S in graph G (Fig. 4-2). Let the removal of S partition the vertices of G into two (mutually exclusive or disjoint) subsets V_1 and V_2. Consider a circuit Γ in G. If all the vertices in Γ are entirely within vertex set V_1 (or V_2), the number of edges common to S and Γ is zero; that is, $N(S \cap \Gamma) = 0$, an even number.†

If, on the other hand, some vertices in Γ are in V_1 and some in V_2, we traverse



Circuit Γ shown in heavy lines, and is traversed along the direction of the arrows

Fig. 4-2 Circuit and a cut-set in G.

back and forth between the sets V_1 and V_2 as we traverse the circuit (see Fig. 4-2). Because of the closed nature of a circuit, the number of edges we traverse between V_1 and V_2 must be even. And since very edge in S has one end in V_1 and the other in V_2, and no other edge in G has this property (of separating sets V_1 and V_2), the number of edges common to S and Γ is even. ■

PART E

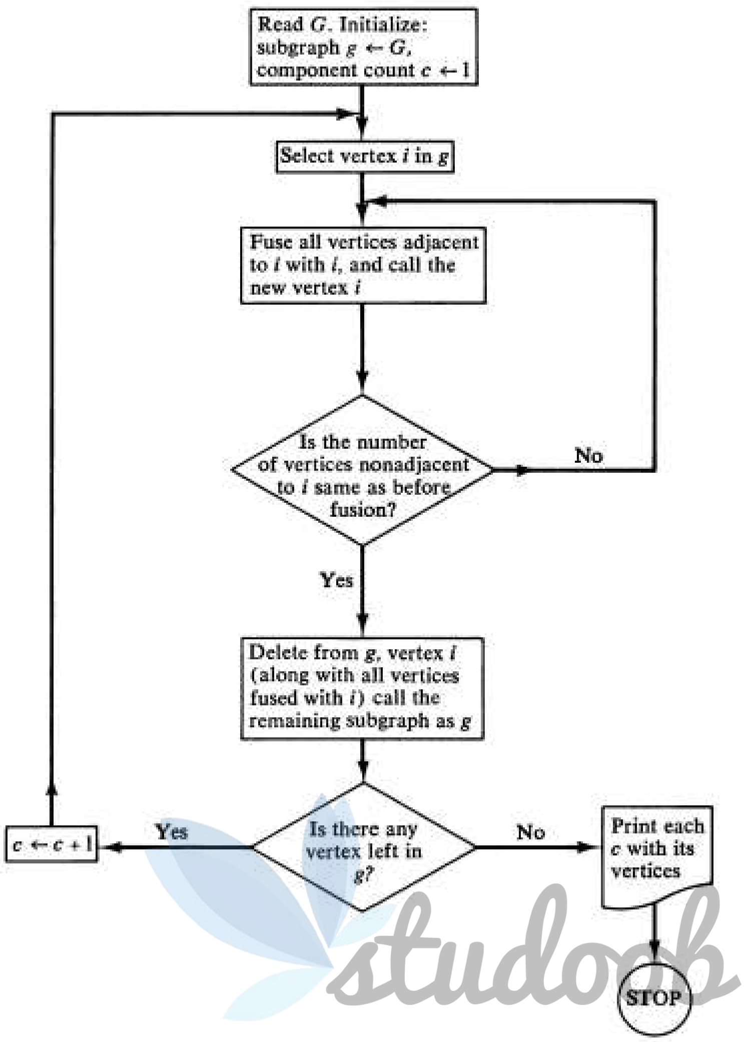
15.

Description of the Algorithm: The basic step in this algorithm is the fusion of adjacent vertices (recall Section 2-7). We start with some vertex in the graph and fuse all vertices that are adjacent to it. Then we take the fused vertex and again fuse with it all those vertices that are adjacent to it now. This process of fusion is repeated until no more vertices can be fused. This indicates that a connected component has been "fused" to a single vertex. If this exhausts every vertex in the graph, the graph is connected. Otherwise, we start with a new vertex (in a different component) and continue the fusing operation.

In the adjacency matrix the fusion of the jth vertex to the ith vertex is accomplished by OR-ing, that is, logically adding the jth row to the ith row as well as the jth column to the ith column. (Remember that in logical adding $1 + 0 = 0 + 1 = 1 + 1 = 1$ and $0 + 0 = 0$.) Then the jth row and the jth column are discarded from the matrix. (If it is difficult or time consuming to discard the specified rows and columns, one may leave these rows and columns in the matrix, taking care that they are not considered again in any fusion.)

Note that a self-loop resulting from a fusion appears as a 1 in the main diagonal, but parallel edges are automatically replaced by a single edge because of the logical addition (or OR-ing) operation. These, of course, have no effect on the connectedness of a graph.

The maximum number of fusions that may have to be performed in this algorithm is $n - 1$, n being the number of vertices. And since in each fusion one performs at most n logical additions, the upper bound on the execution time is proportional to $n(n - 1)$.



Algorithm 1: Components of G.

16.

Description of the Algorithm: Let the given undirected self-loop-free (if the graph has any self-loops, they may be discarded) graph G contain n vertices and e edges. Let the vertices be labeled $1, 2, \ldots, n$, and the graph be described by two linear arrays F and H [i.e., in the form (d) of Section 11-2] such that $f_i \in F$ and $h_i \in H$ are the end vertices of the ith edge in G.

At each stage in the algorithm a new edge is tested to see if either or both of its end vertices appear in any tree formed so far.† At the kth stage, $1 \leq k \leq e$, in examining the edge (f_k, h_k) five different conditions may arise:

1. If neither vertex f_k nor h_k is included in any of the trees constructed so far in G, the kth edge is named as a new tree and its end vertices f_k, h_k are given the component number c, after incrementing the value of c by 1.
2. If vertex f_k is in some tree T_i $(i = 1, 2, \ldots, c)$ and h_k in tree T_j $(j = 1, 2, \ldots, c$, and $i \neq j)$, the kth edge is used to join these two trees; therefore, every vertex in T_j is now given the component number of T_i. The value of c is decremented by 1.
3. If both vertices are in the same tree, the edge (f_k, h_k) forms a fundamental circuit and is not considered any further.

4. If vertex f_k is in a tree T_i and h_k is in no tree, the edge (f_k, h_k) is added to T_i by assigning the component number of T_i to h_k also.
5. If vertex f_k is in no tree and h_k is in a tree T_j, the edge (f_k, h_k) is added to T_j by assigning the component number of T_j to f_k also.

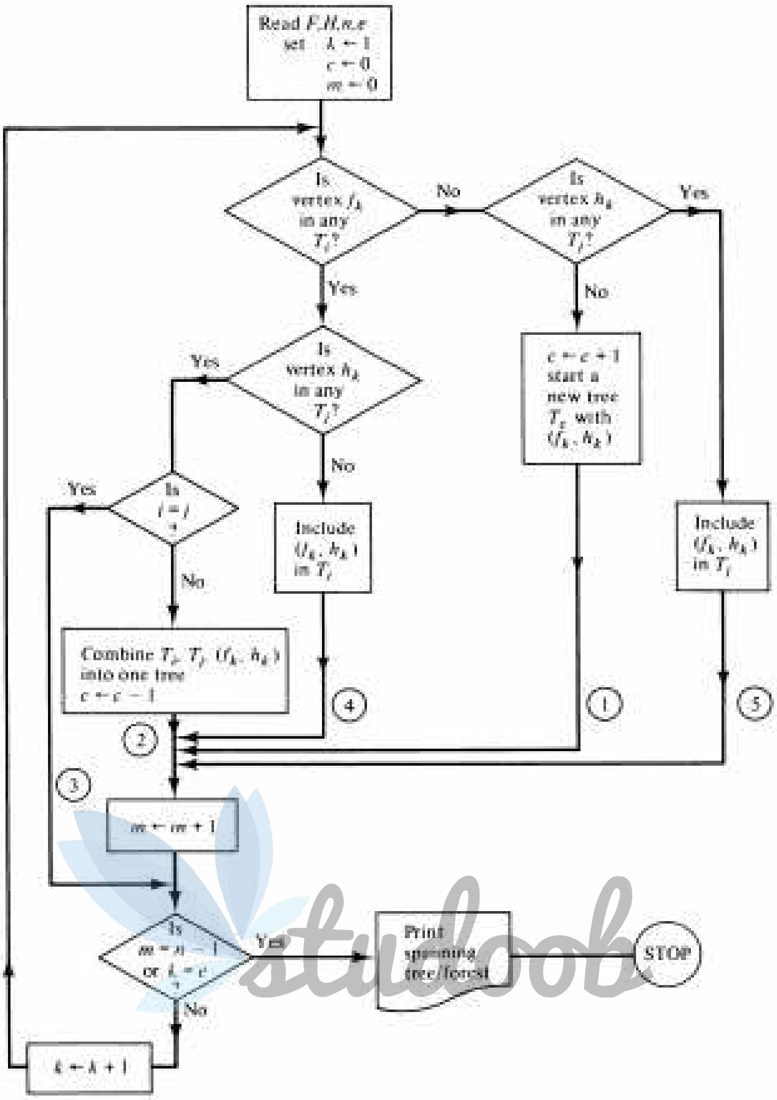


Fig. 11-3 Algorithm 2: Spanning tree/forest.

17. Algorithm for finding shortest path between two specified vertices.

Algorithm 6: Shortest Path from a Specified Vertex to Another Specified Vertex

The problem of finding the shortest path from a specified vertex s to another specified vertex t, can be stated as follows:

A simple weighted digraph† G of n vertices is described by an n by n matrix $\mathbf{D} = [d_{ij}]$, where

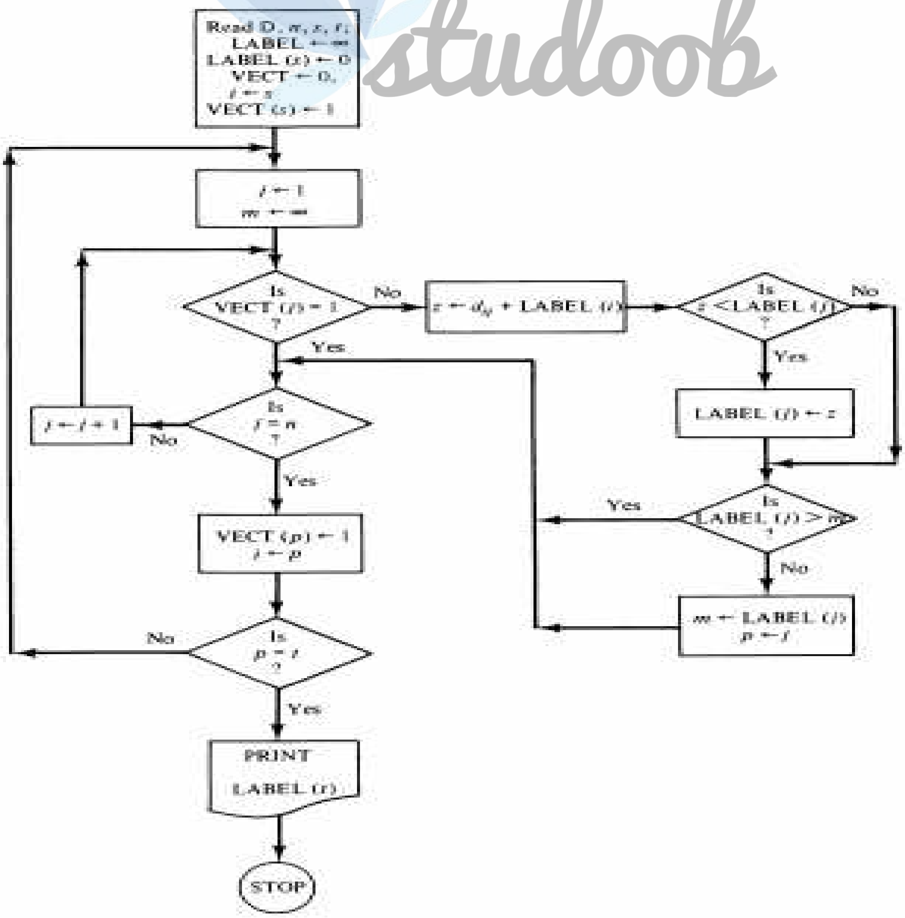
d_{ij} = length (or distance or weight) of the directed edge from vertex i to vertex j, $d_{ij} \geqslant 0$,

$d_{ii} = 0$,

$d_{ij} = \infty$, if there is no edge from i to j (in carrying out a program ∞ is replaced by a large number, say 9999999).

In general, $d_{ij} \neq d_{ji}$, and the triangle inequality need not be satisfied. That is, $d_{ij} + d_{jk}$ may be less than d_{ik}. [In fact, if the triangle inequality is satisfied, for every i, j, and k, the problem would be trivial because the direct edge (x, y) would be the shortest path from vertex x to vertex y.] The distance of a directed path P is defined to be the sum of the lengths of the edges in P. The problem is to find the shortest possible path and its length from a starting vertex s to a terminal vertex t.

Among several algorithms that have been proposed for the shortest path between a specified vertex pair, perhaps the most efficient one is an algorithm due to Dijkstra [11-16].



18.

Let T be a spanning tree of a graph G. Then

$$A_f \text{ (reduced incidence matrix)} = [A_c : A_t]$$
$$B_f \text{ (fundamental circuit matrix)} = [I_\mu : B_t]$$
$$C_f \text{ (fundamental cut-set matrix)} = [C_c : I_r]$$

where I denote the identity matrix. Suffices c and t denote the matrix corresponding to the chords and branches of T respectively μ are r are the nullity and rank of the graph.

Now consider the columns of A_f, B_f and C_f arranged in the same order, then

$$A_f \cdot B_f^T = 0 \pmod 2$$
$$[A_c : A_t]\,[I_\mu : B_t]^T = 0$$
$$[A_c : A_t]\begin{bmatrix} I_\mu \\ \hline B_t^T \end{bmatrix} = 0$$
$$A_c\, I_\mu + A_t \cdot B_t^T = 0$$
$$A_c + A_t \cdot B_t^T = 0 \quad \text{...(1)}$$

A_t is non-singular, its inverse A_t^{-1} exists. Multiplied by A_t^{-1} we get

$$A_t^{-1} \cdot A_c = -A_t^{-1} A_t\, B_t^T$$
$$A_t^{-1} \cdot A_c = -B_t^T \quad \text{...(2)}$$

Since in mod 2 arithmetic $-1 = 1$ So $-B_t^T = B_t^T$

Hence $B_t^T = A_t^{-1} A_c$

Similarly, $C_f \cdot B_f^T = 0 \pmod 2$

$$[C_c : I_r]\,[I_\mu : B_t]^T = 0$$
$$[C_c : I_r]\begin{bmatrix} I_\mu \\ \hline B_t^T \end{bmatrix} = 0$$
$$C_c\, I_\mu + I_r\, B_t^T = 0$$
$$C_c + B_t^T = 0$$
$$C_c = -B_t^T$$

However $-B_t^T = B_t^T$ in mod 2 operation.

Hence $\boxed{C_c = B_t^T = A_t^{-1} A_c}$

19. Matrix representations of a graph are

- Adjacency matrix
- Incidence matrix
- Circuit matrix
- Cut set matrix

Refer: **Graph Theory with Applications to Engineering and Computer Science** By Narsingh Deo (Chapter 7)

20.

(a) Prove that if B is a circuit matrix of a connected graph G with e edges and n vertices, rank of B $= e - n + 1$

Proof: If A is an incidence matrix of G, from Eq. (7-4) we have

$$A \cdot B^T = 0 \pmod 2).$$

Therefore, according to Sylvester's theorem (Appendix B),

$$\text{rank of A} + \text{rank of B} \leq e;$$

that is,

$$\text{rank of B} \leq e - \text{rank of A}.$$

Since $\text{rank of A} = n - 1$

we have $\text{rank of B} \leq e - n + 1.$

But $\text{rank of B} \geq e - n + 1.$

Therefore, we must have

$$\text{rank of B} = e - n + 1. \quad \blacksquare$$

(b)

Prove that the rank of cut-set matrix is equal to the rank of the incidence matrix, which equals the rank of graph G

The rank of cut-set matrix $\mathsf{C}(G)$ is equal to the rank of the incidence matrix $\mathsf{A}(G)$, which equals the rank of graph G.

As in the case of the circuit matrix, the cut-set matrix generally has many redundant (or linearly dependent) rows. Therefore, it is convenient to define a fundamental cut-set matrix, C_f, as follows:

A fundamental cut-set matrix C_f (of a connected graph G with e edges and n vertices) is an $(n - 1)$ by e submatrix of C such that the rows correspond to the set of fundamental cut-sets with respect to some spanning tree.

As in the case of a fundamental circuit matrix, a fundamental cut-set matrix C_f can also be partitioned into two submatrices, one of which is an identity matrix I_{n-1} of order $n - 1$. That is,

$$\mathsf{C}_f = [\mathsf{C}_c \mid \mathsf{I}_{n-1}], \tag{7-9}$$

where the last $n - 1$ columns forming the identity matrix correspond to the $n - 1$ branches of the spanning tree, and the first $e - n + 1$ columns forming C_c correspond to the chords.