CS6109 – GRAPH THEORY

Module – 4

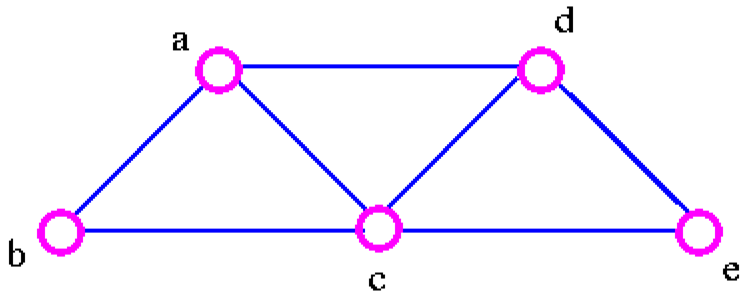
Presented By
Dr. S. Muthurajkumar,
Assistant Professor,
Dept. of CT, MIT Campus,
Anna University, Chennai.

Module - 4

- Spanning Tree
- Fundamental Circuits
- Cut Sets
- Cut Sets – Properties
- Connectivity - Separability
- Network Flows
- 1 – Isomorphism
- 2 – Isomorphism
- Related Theorems

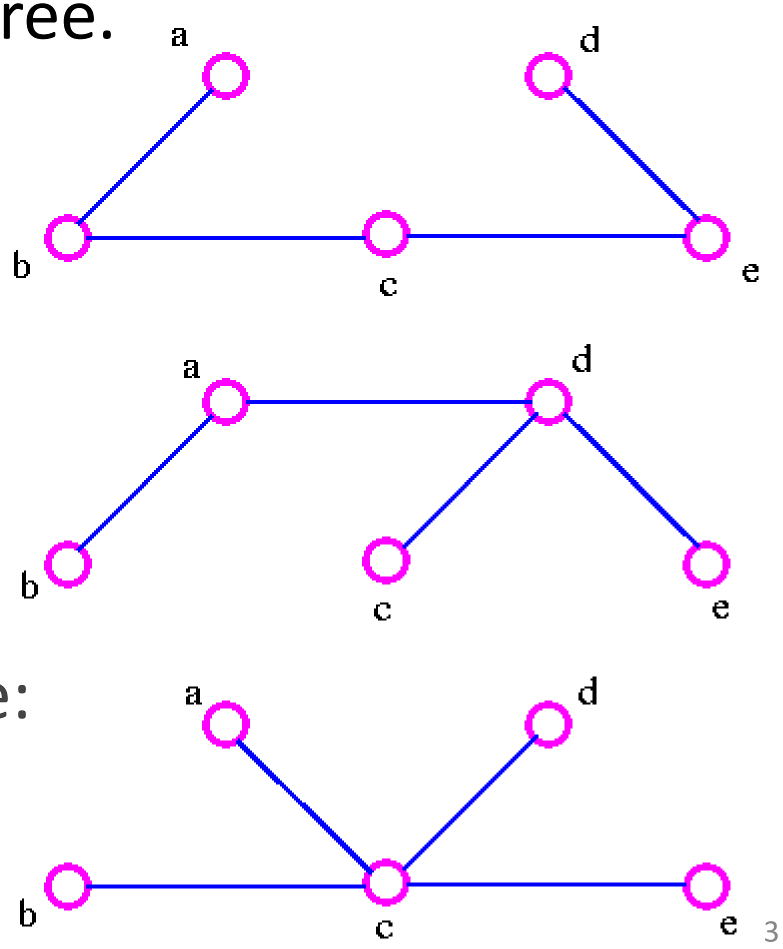
Spanning Tree

- A tree *T* is said to be a *spanning tree* of a connected graph *G* if *T* is a subgraph of *G* and *T* contains all vertices of *G*.
- Let G be a connected graph. A spanning tree in G is a subgraph of G that includes all the vertices of G and is also a tree.
- The edges of the trees are called branches.



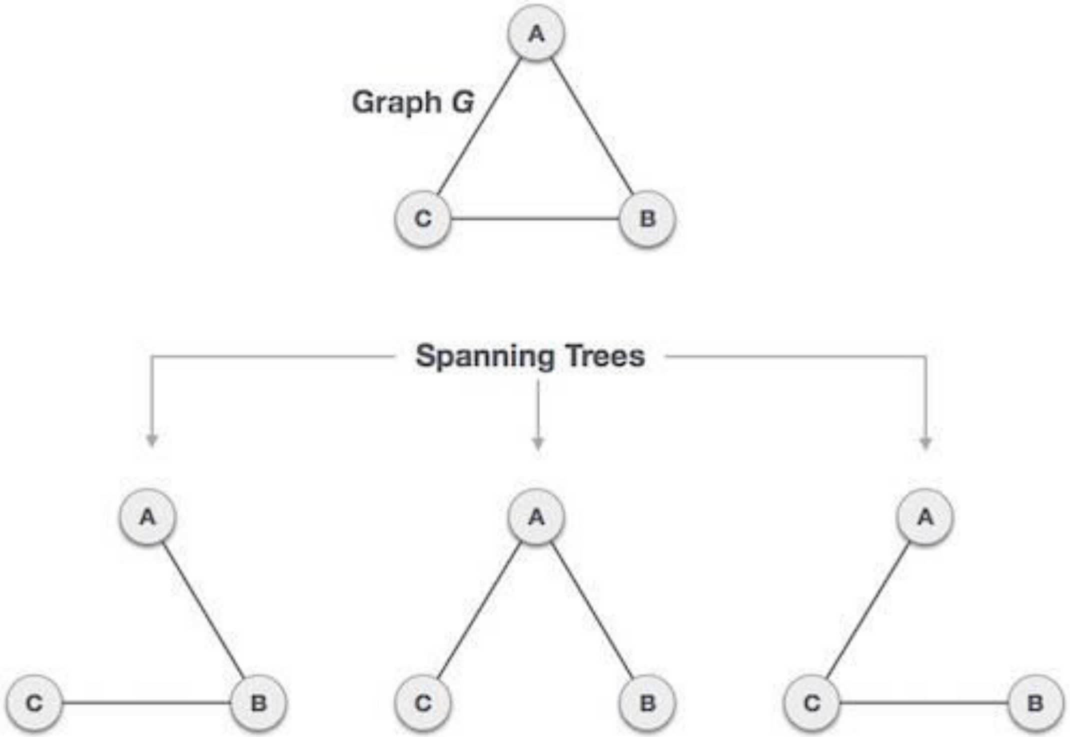
Graph *G*

The three spanning trees G are:



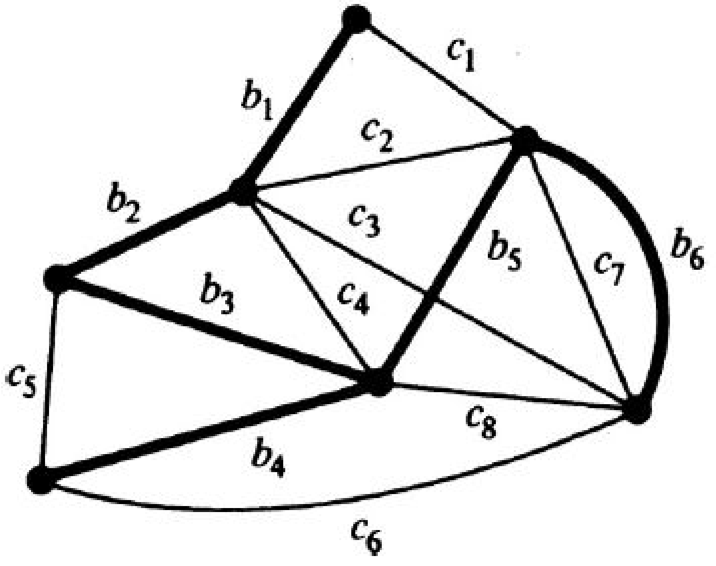
Spanning Tree

Graph G

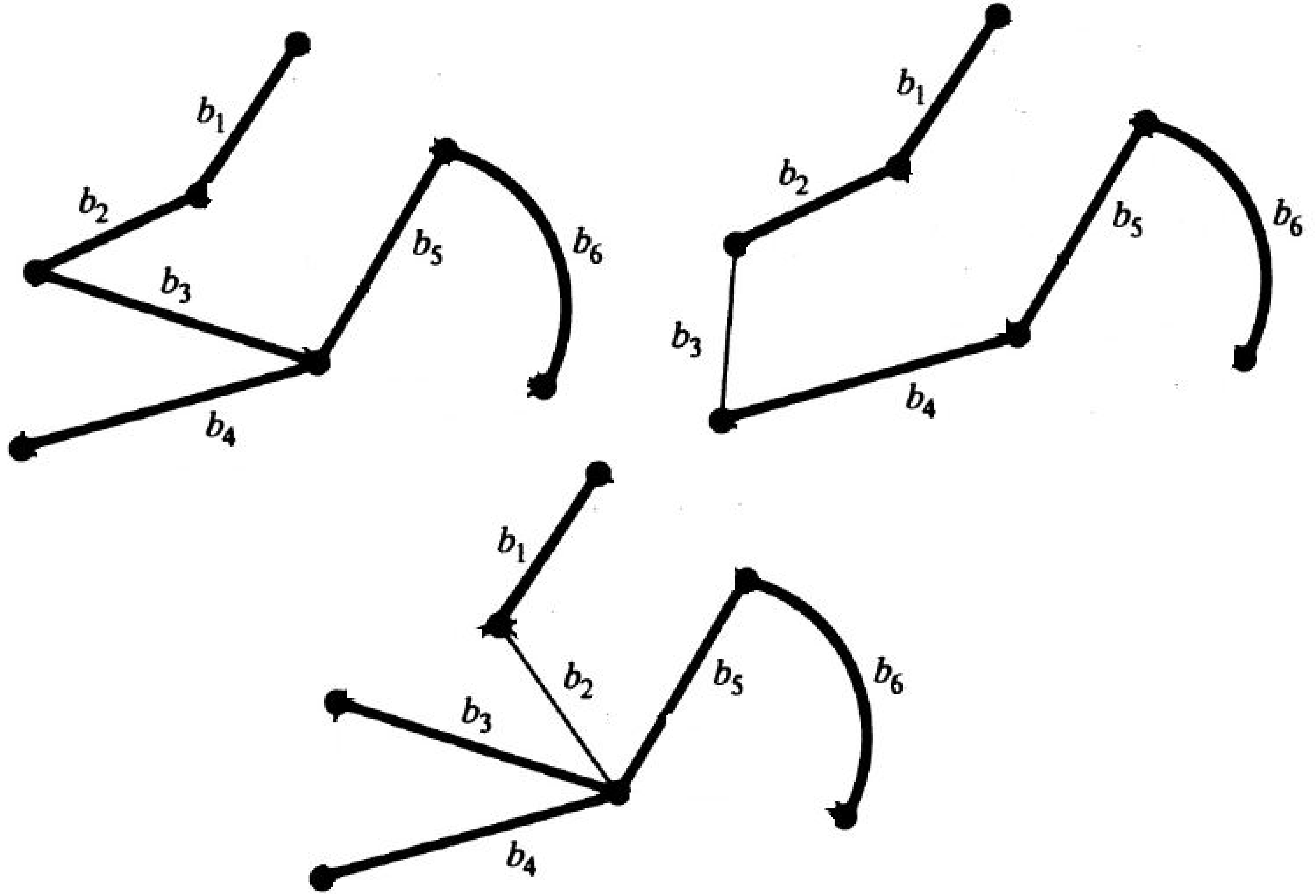


Spanning Tree

Graph *G*



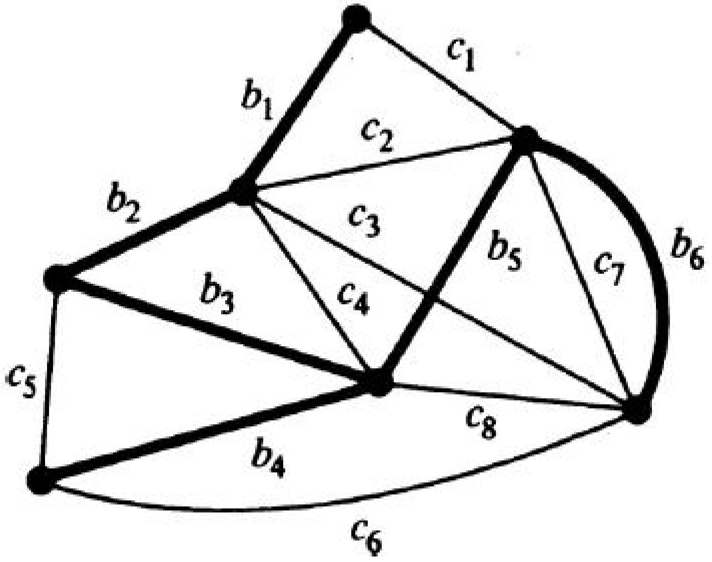
Spanning trees G are:



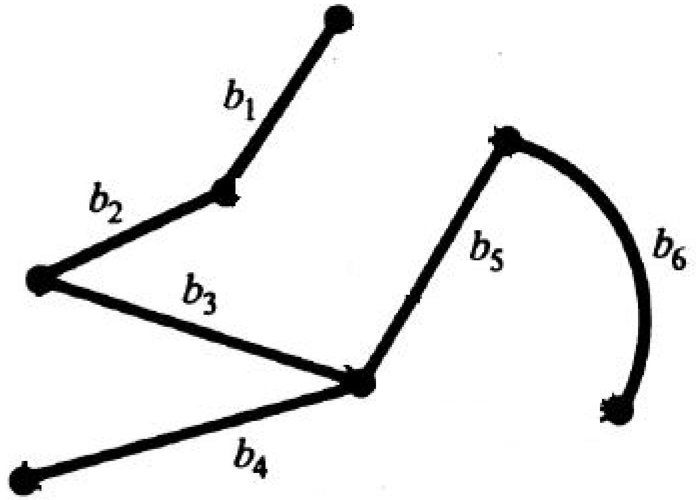
Spanning Tree - Branch of tree

An edge in a spanning tree T is called a branch of T.

Graph *G*



Spanning trees G are:



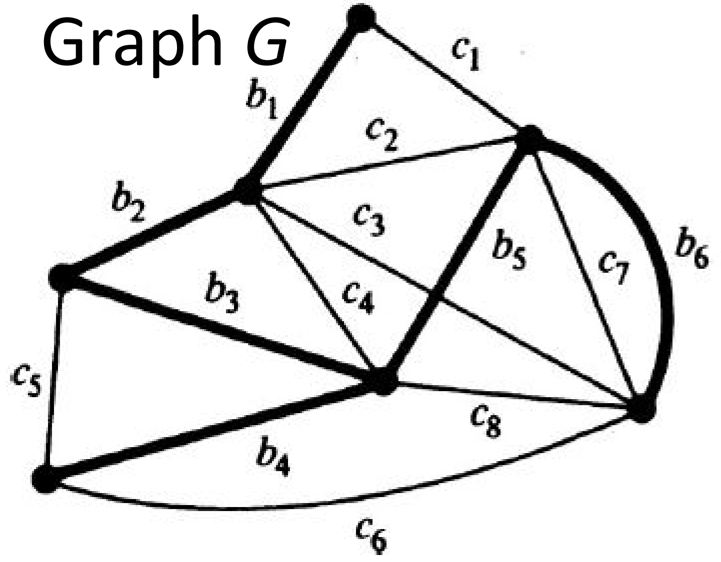
Spanning Tree - Chord

An edge of G that is not in a given spanning tree is called a chord.

Note :

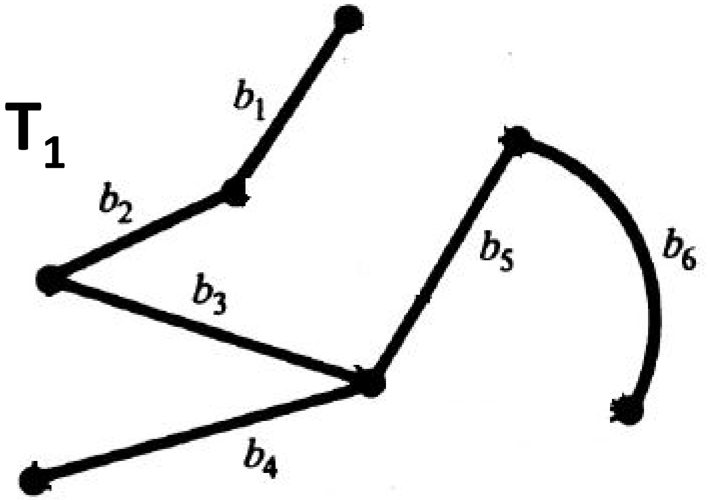
1. The branches and chords are defined only with respect to a given spanning tree.
2. An edge that is a branch of one spanning tree T_1 (in a graph G) may be chord, with respect to another spanning tree T_2.

Graph *G*

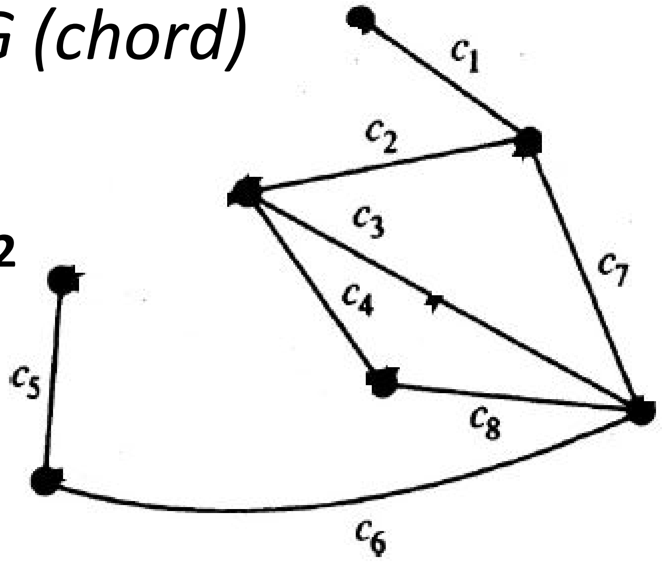


Spanning Tree T of *G (chord)*

T_1

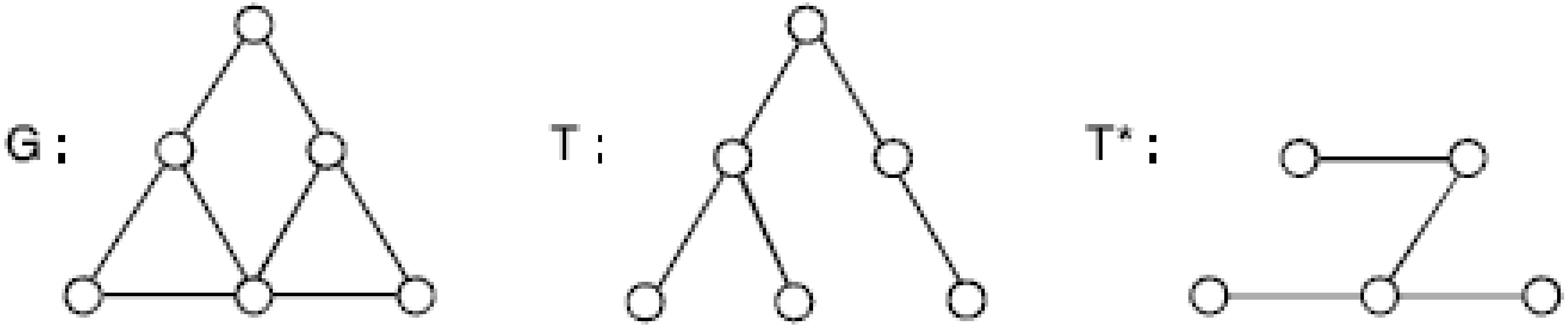


T_2



Spanning Tree - Cotree

- The cotree T* of a spanning tree T in a connected graph G is the spanning subgraph of G containing exactly those edges of G which are not in T.
- The edges of G which are not in T* are called its twigs.



Spanning Tree – Forest, Spanning Forest

Forest

- A collection to tree in a graph G is called a forest in the graph.

Spanning Forest

- A forest that contains every vertex of a graph *G* such that two vertices are in the same tree of the forest when there is a path in *G* between these two vertices.
- In other words, a spanning forest of a graph *G* is a collection of exactly one spanning tree from each of its connected components.

Spanning Tree – Rank, Nullity

Rank

- If in a graph G there are total n vertices and k compo- nents then the rank, generally denoted by r, is defined as
  $r = n - k$.

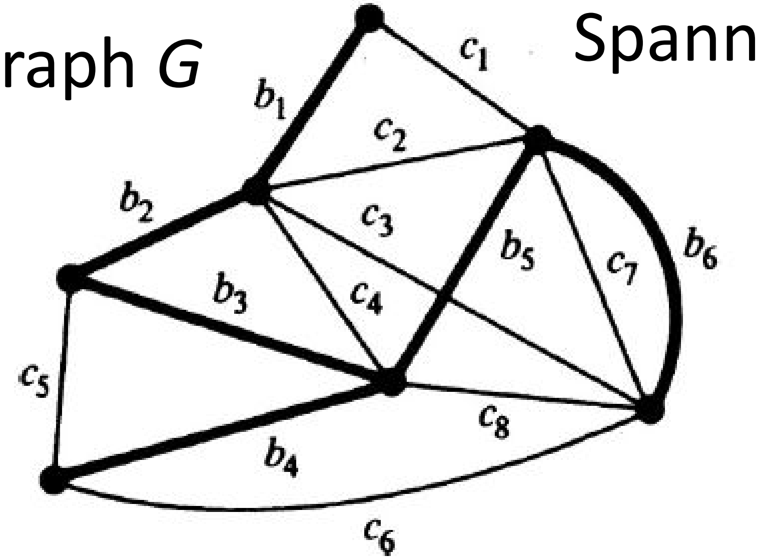
Nullity

- If in a graph G there are total n vertices, k components and e edges then the nullity of G, generally denoted by μ, is defined as $\mu = e - n + k$

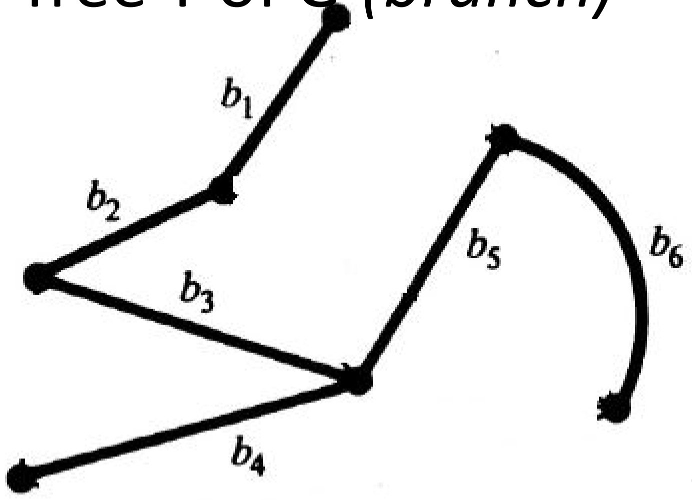
THEOREM 3-11: Every connected graph has at least one spanning tree.

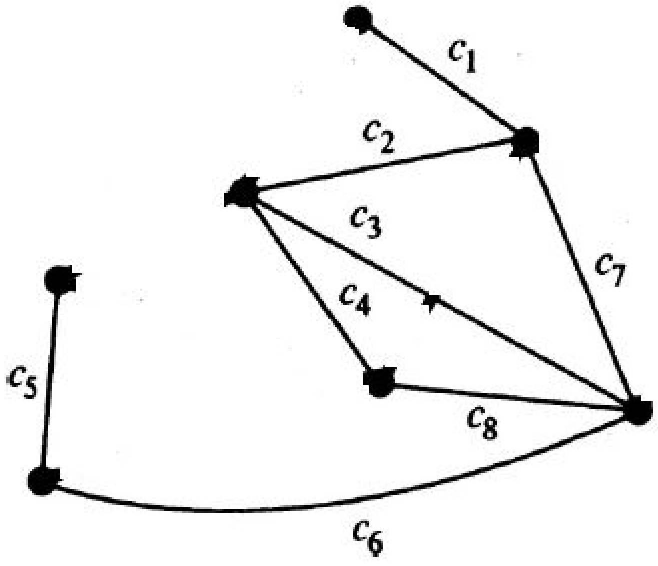
- An edge in a spanning tree *T* is called a *branch* of *T*.
- An edge of *G* that is not in a given spanning tree *T* is called a *chord*.
- An edges b_1, b_2, b_3, b_4, b_5, and b_6 are branches of the spanning tree.
- An edges c_1, c_2, c_3, c_4, c_5, c_6, c_7, and c_8 are chords.
- It must be kept in mind that branches and chords are defined only with respect to a given spanning tree.
- An edge that is a branch of one spanning tree T_1 (in a graph *G*) may be a chord with respect to another spanning tree T_2.

Graph *G*



Spanning Tree T of *G (branch)*



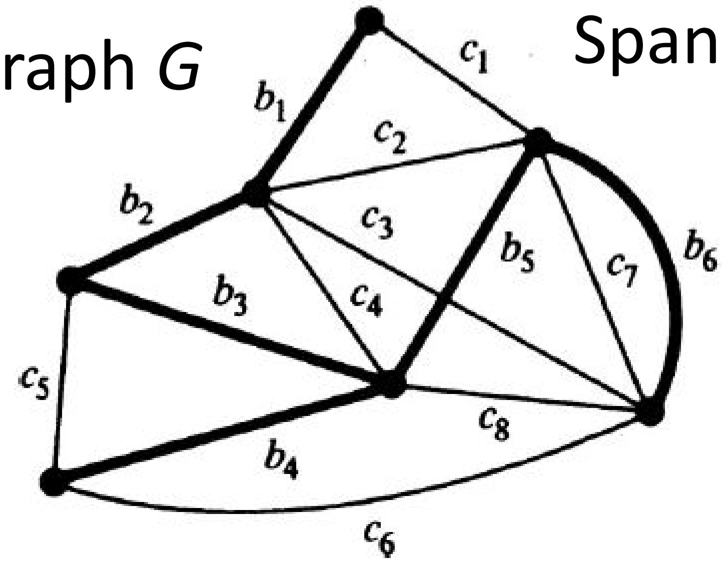


Spanning Tree T of *G (chord)*

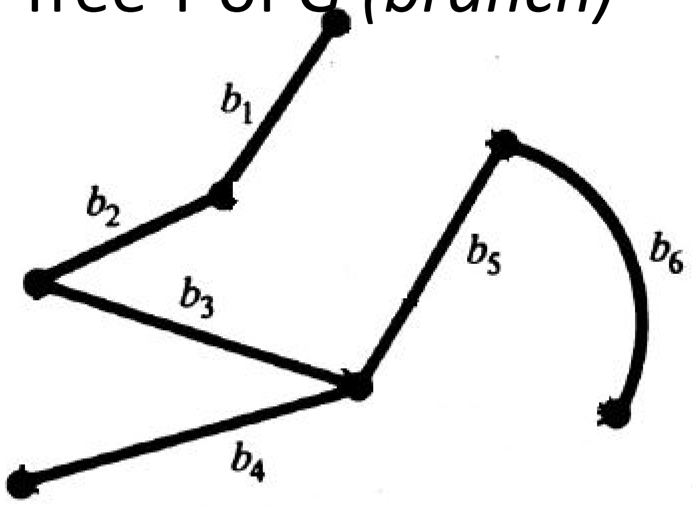
THEOREM 3-11: Every connected graph has at least one spanning tree.

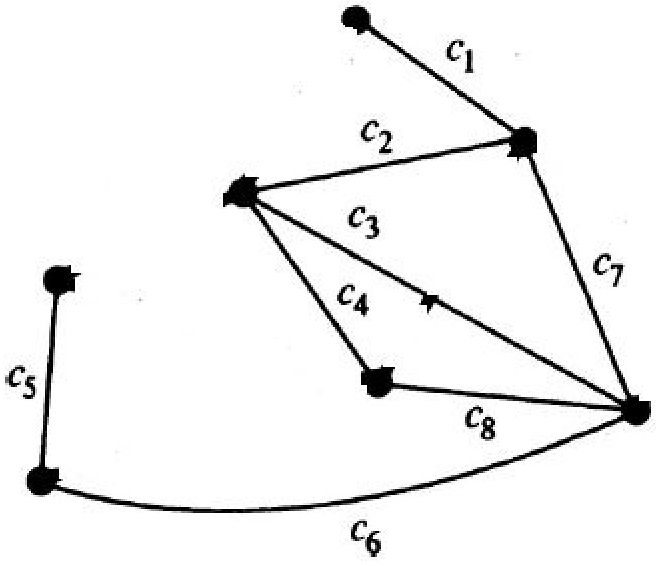
- It is sometimes convenient to consider a connected graph *G* as a union of two sub-graphs, *T* and ; that is, $T \cup \bar{T} = G$.
- where *T* is a spanning tree, and $\bar{T}$ is the complement of T in *G*. Since the sub-graph is the collection of chords, it is quite appropriately referred to as the *chord set* (or *tie set* or *cotree*) of *T*. From the definition, and from Theorem 3-3(A tree with *n* vertices has *n* − 1 edges.), the following theorem is evident.

Graph *G*



Spanning Tree T of *G (branch)*



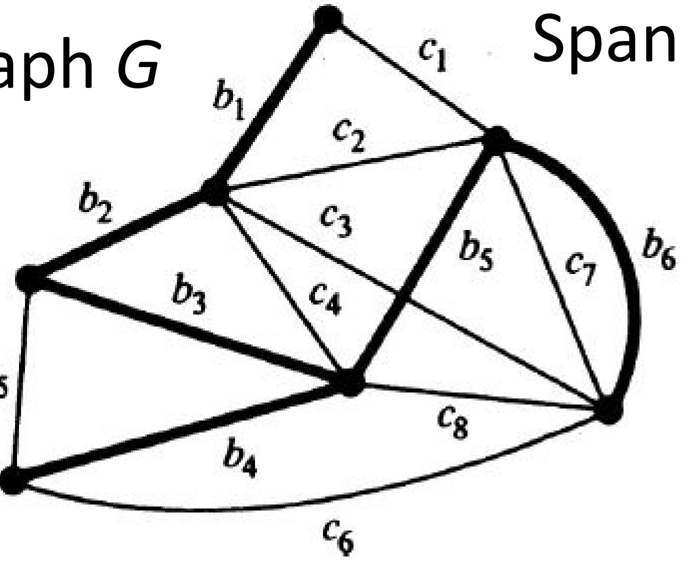


Spanning Tree T of *G (chord)*

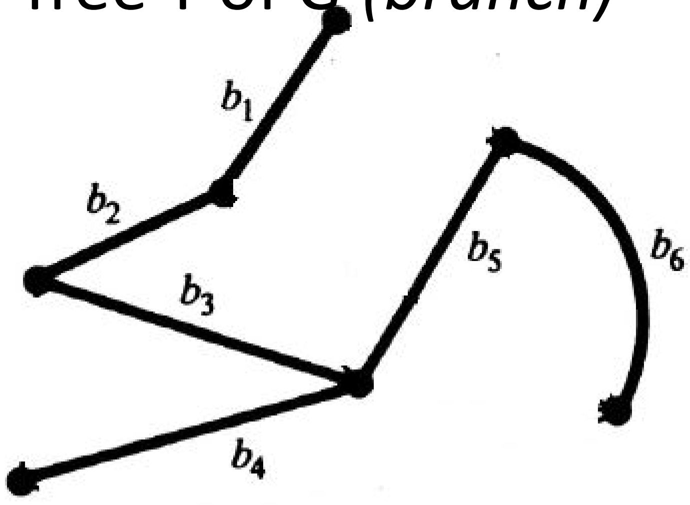
THEOREM 3-12: With respect to any of its spanning trees, a connected graph of *n* vertices and *e* edges has *n* − 1 tree branches and *e* − *n* + 1 chords.

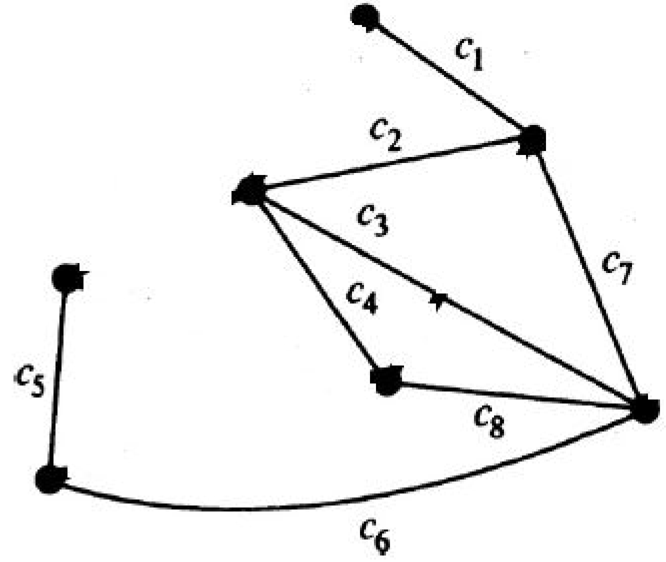
- n = 7
- e = 14 (6+8) (6 branch, 8 chord)
- Branch = n – 1 ➔ 7 – 1 ➔ 6 branch
- Chord = e – n + 1 ➔ 14 – 7 + 1 ➔ 8 chord

Graph *G*



Spanning Tree T of *G (branch)*

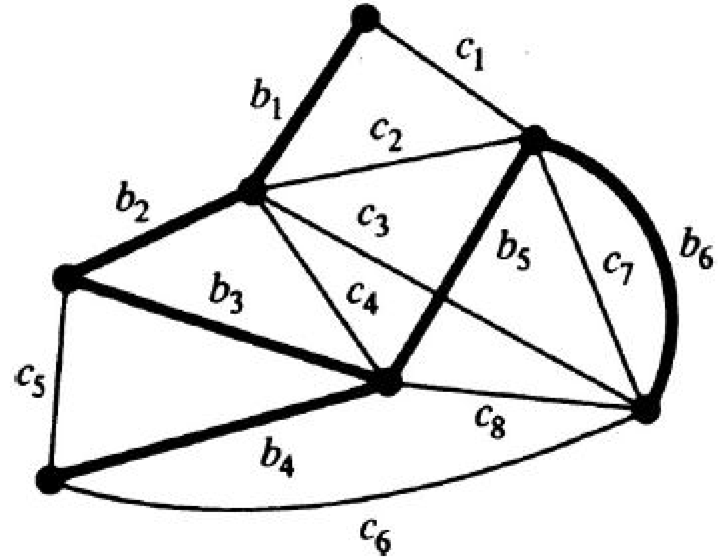


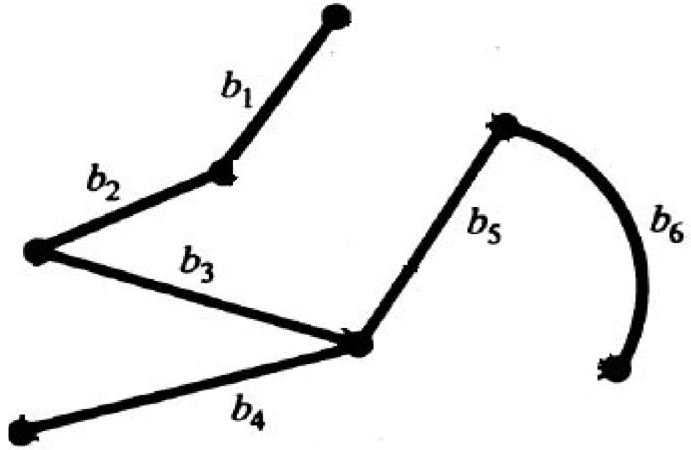


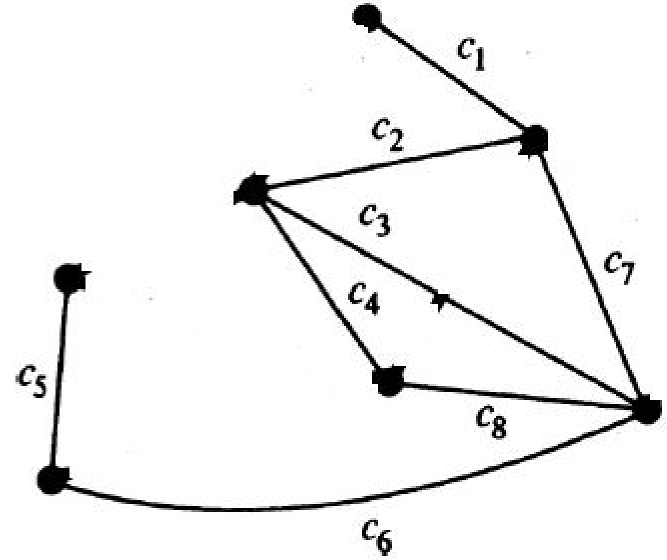
Spanning Tree T of *G (chord)*

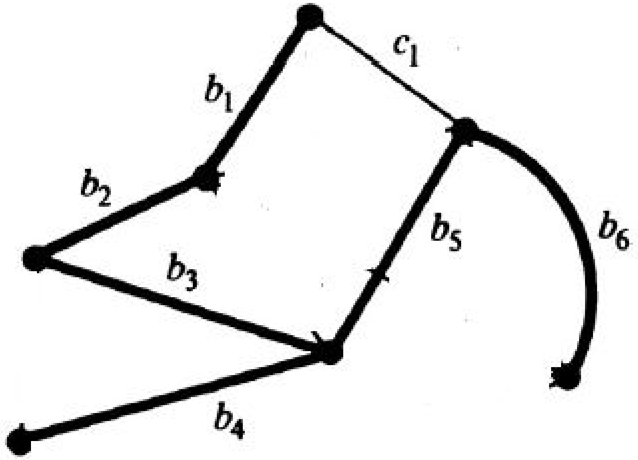
Fundamental Circuit

- Adding any one chord to *T* will create exactly one circuit. Such a circuit, formed by adding a chord to a spanning tree, is called *a fundamental circuit*.



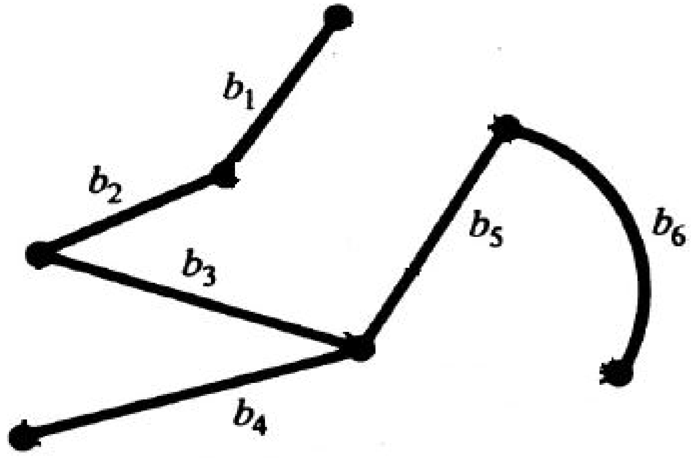


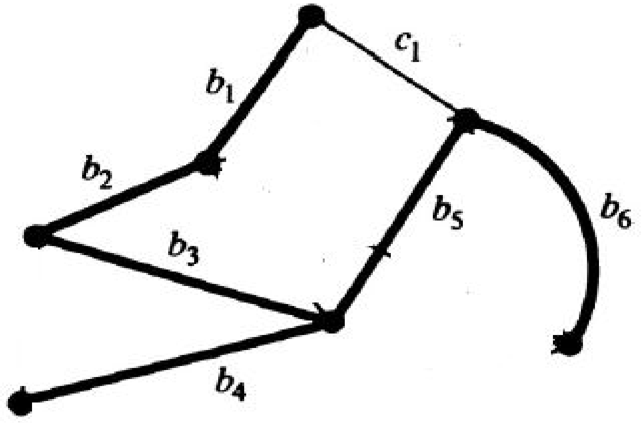


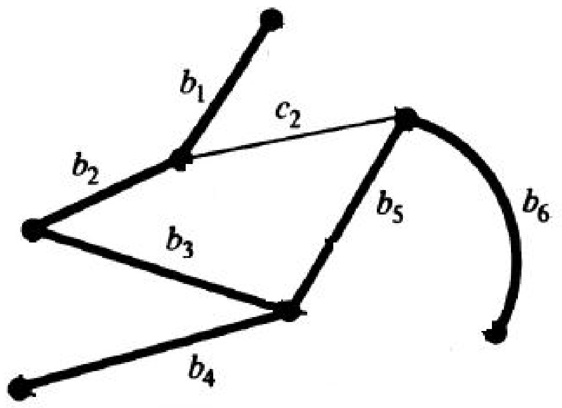


Fundamental Circuit

- Let us look at the tree $\{b_1, b_2, b_3, b_4, b_5, b_6\}$.
- Adding c_1 to it, we get a subgraph $\{b_1, b_2, b_3, b_4, b_5, b_6, c_1\}$, which has one circuit (fundamental circuit), $\{b_1, b_2, b_3, b_4, b_5, c_1\}$. Had we added the chord c_2 (instead of c_1) to the tree, we would have obtained a different fundamental circuit, $\{b_2, b_3, b_5, c_2\}$.

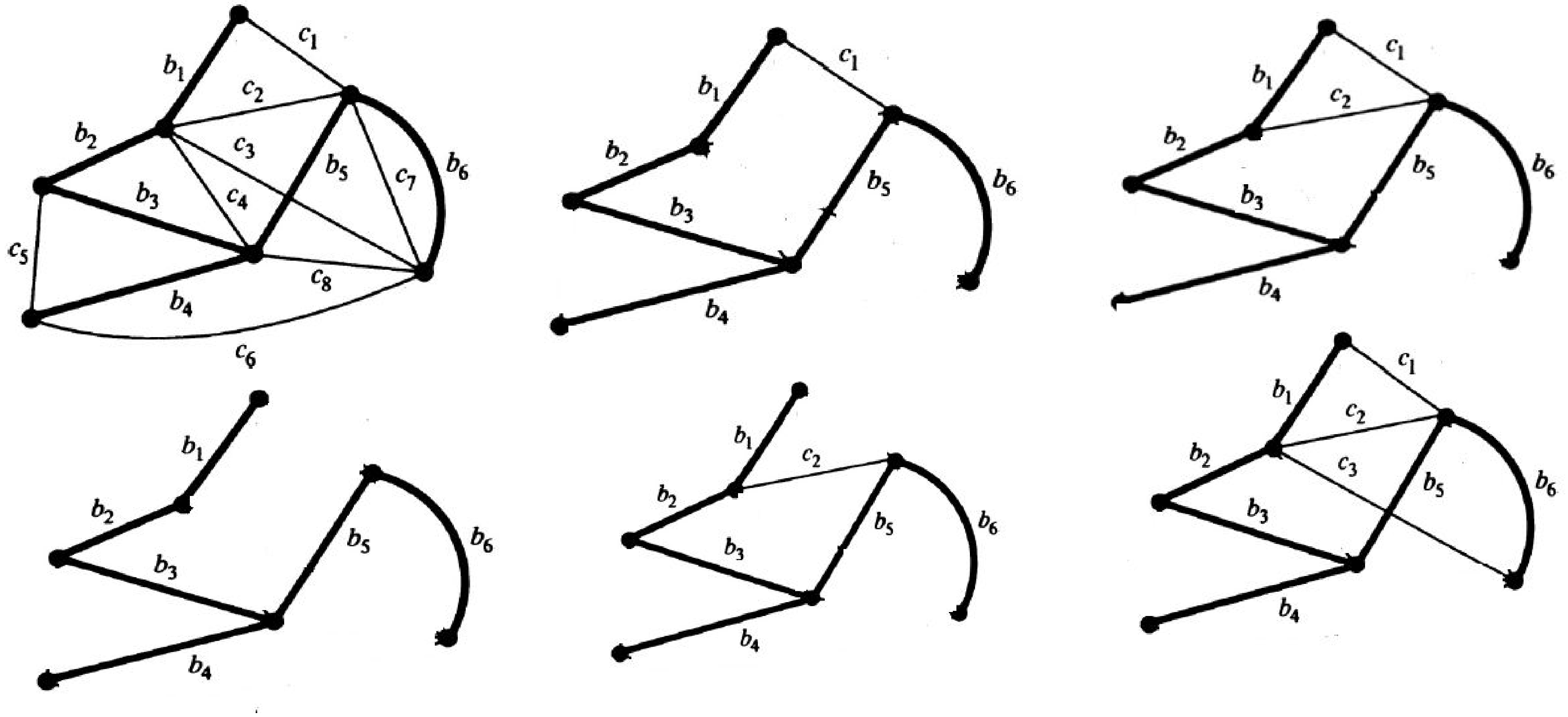




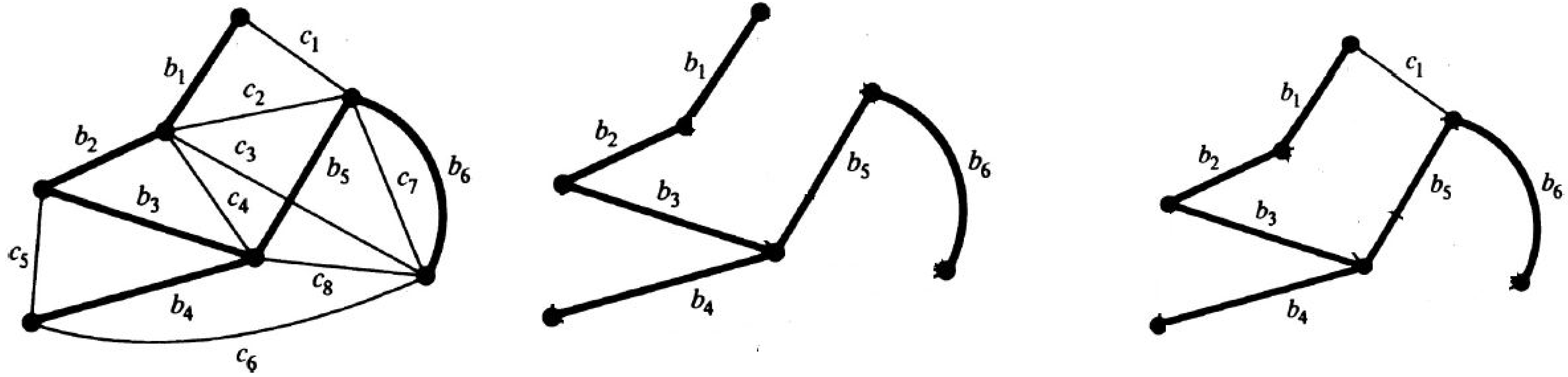


Fundamental Circuit

- Now suppose that we add both chords c_1 and c_2 to the tree. The subgraph $\{b_1, b_2, b_3, b_4, b_5, b_6, c_1, c_2\}$, has not only the fundamental circuits we just mentioned, but it has also a third circuit, $\{b_1, c_1, c_2\}$, which is not a fundamental circuit. Although there are 75 circuits in below Figure (enumerated by computer), only eight are fundamental circuits, each formed by one chord (together with the tree branches).



THEOREM 3-13: A connected graph G is a tree if and only if adding an edge between any two vertices in G creates exactly one circuit.



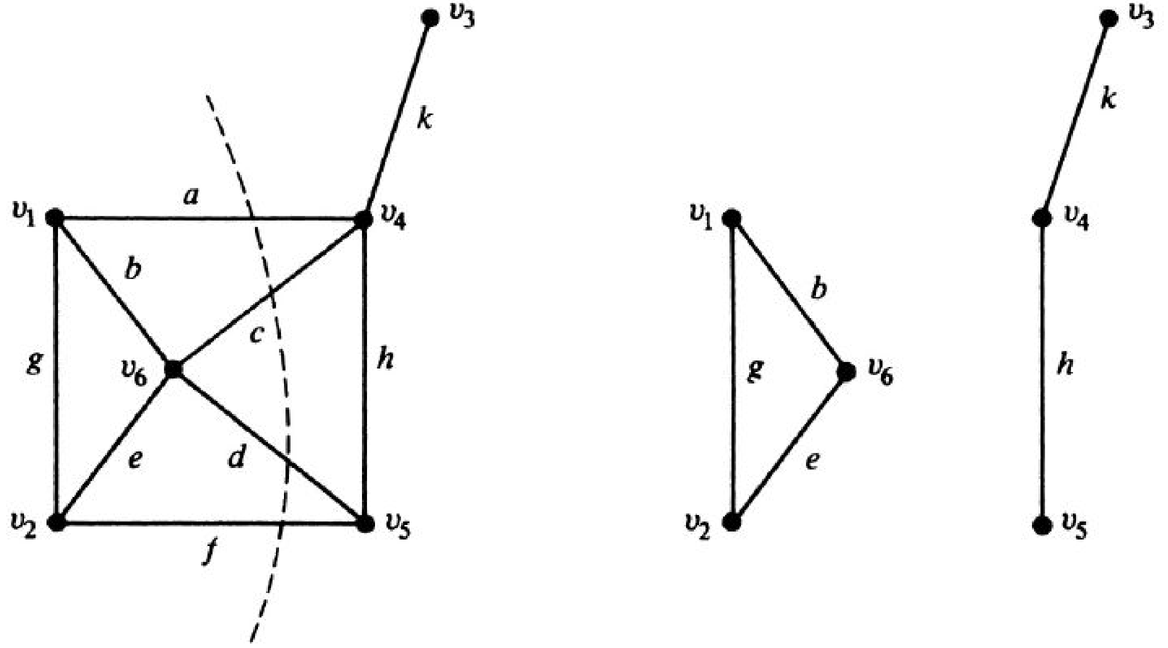
Cut-sets

Cut-sets:

In a connected graph G, a *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.

The set of edges {*a*, *c*, *d*, *f*} is a cut-set.

There are many other cutsets, such as {*a*, *b*, *g*} , {*a*, *b*, *e*, *f*} , and {*d*, *h*, *f*}. Edge {*k*} alone is also a cut-set.

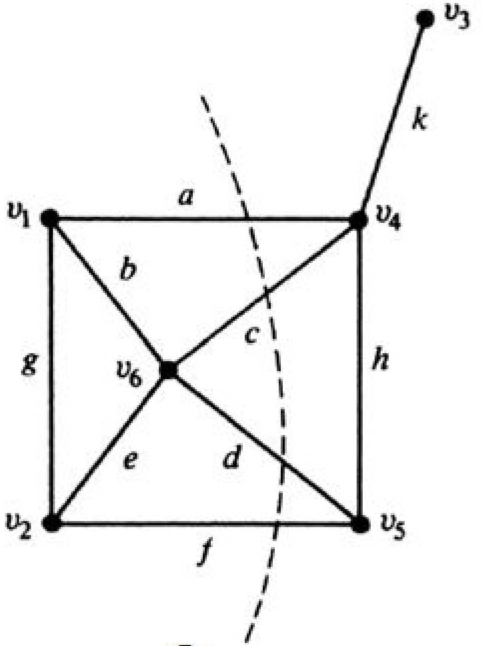


PROPERTIES OF A CUT-SET

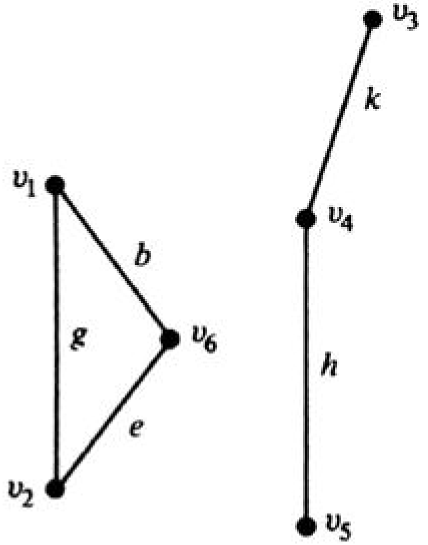
- Every cut-set in a connected graph G must contain at least one branch of every spanning tree of G.
- In a connected graph G, any minimal set of edges containing at least one branch of every spanning tree of G is a cut-set.
- Every circuit has an even number of edges in common with any cut set.

THEOREM 4-1: Every cut-set in a connected graph *G* must contain at least one branch of every spanning tree of G.

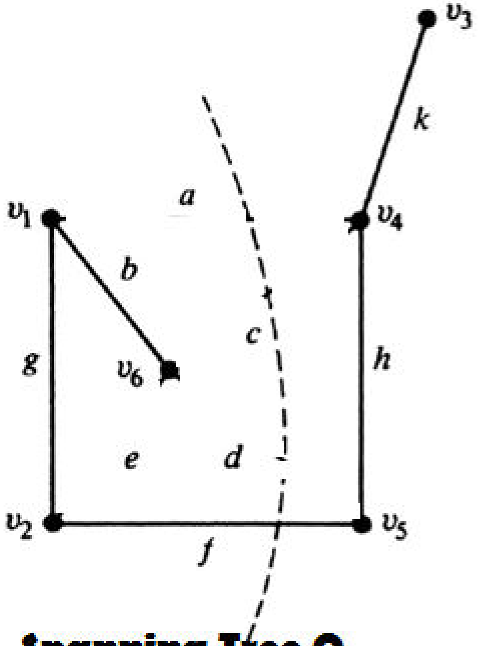
- In a given connected graph *G*, let *Q* be a minimal set of edges containing at least one branch of every spanning tree of *G*.
- Consider *G* − *Q*, the subgraph that remains after removing the edges in *Q* from *G*.
- Since the subgraph *G* − *Q* contains no spanning tree of G, *G* − *Q* is disconnected (one component of which may just consist of an isolated vertex).



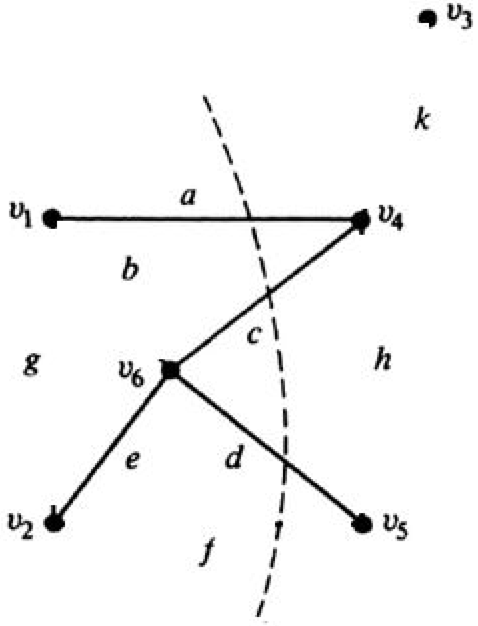
G



cut sets



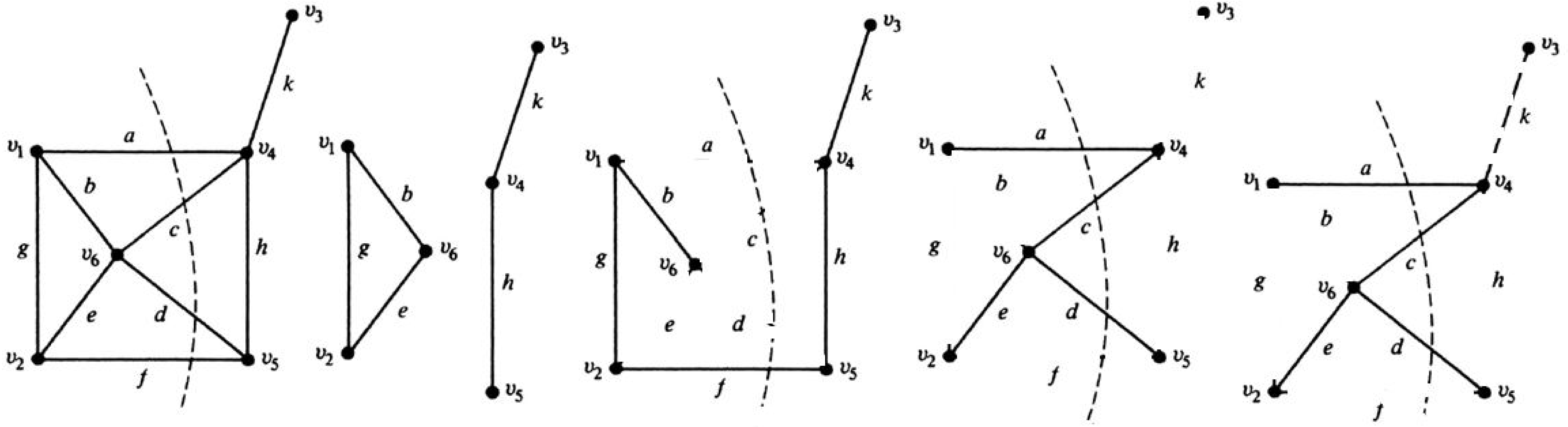
Spanning Tree Q



G - Q

THEOREM 4-1: Every cut-set in a connected graph *G* must contain at least one branch of every spanning tree of G.

- Also, since *Q* is a minimal set of edges with this property, any edge *e* from *Q* returned to *G* − *Q* will create at least one spanning tree.
- Thus the subgraph *G* − *Q* + *e* will be a connected graph.
- Therefore, *Q* is a minimal set of edges whose removal from *G* disconnects G. This, by definition, is a cut-set. Hence

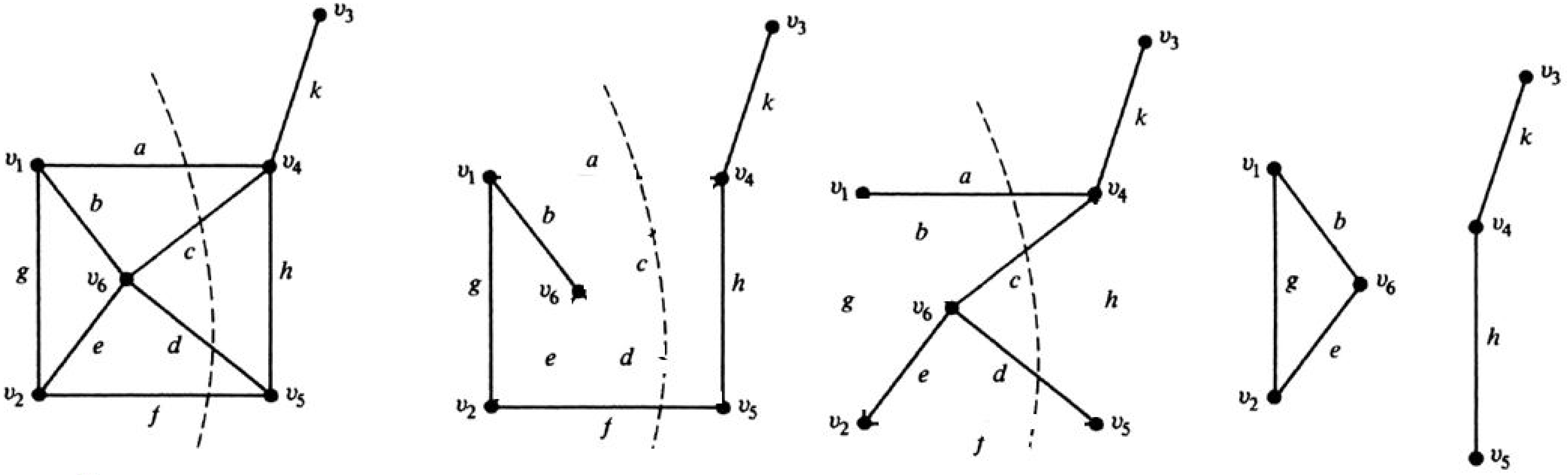


G **Cut Sets** **Spanning Tree Q** **G - Q** **G - Q + e**

PROPERTIES OF A CUT-SET

THEOREM 4-2:

In a connected graph *G*, any minimal set of edges containing at least one branch of every spanning tree of *G* is a cut-set.



G

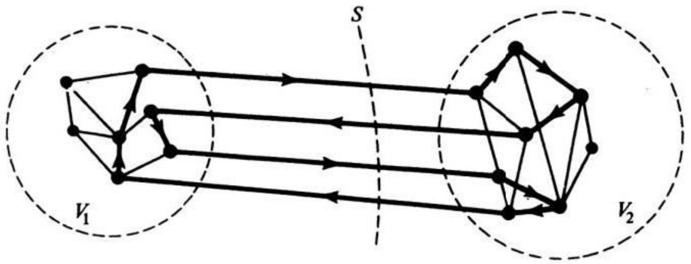
Spanning Tree Q

Cut Sets

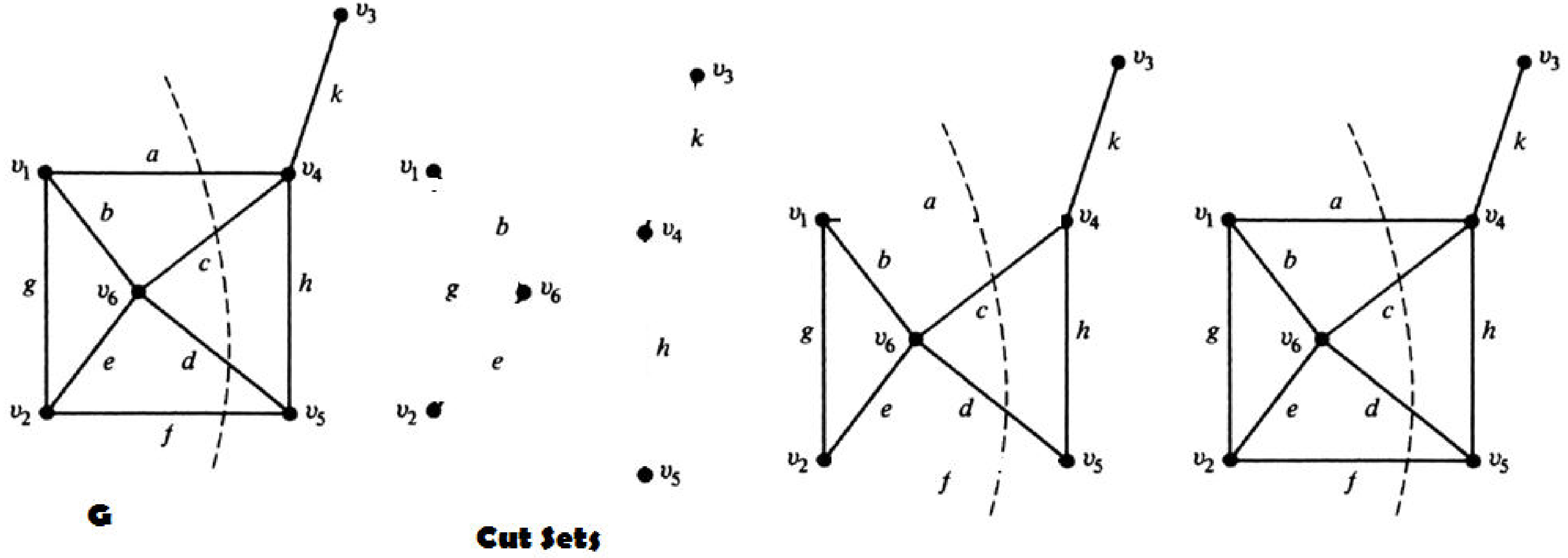
THEOREM 4-3: Every circuit has an even number of edges in common with any cut-set.

Proof:

Consider a cut-set S in graph G. 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.

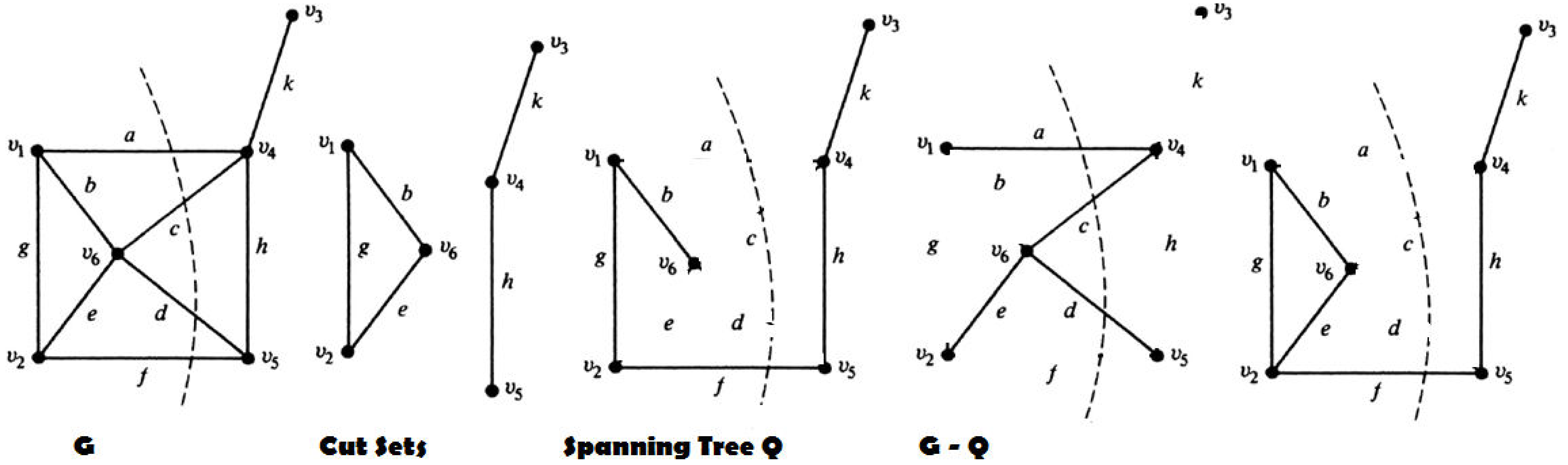


Circuit Γ shown in heavy lines, and is traversed along the direction of the arrows



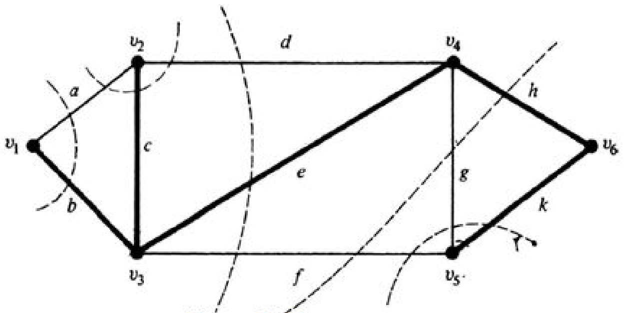
Fundamental Circuit and CUT-SETS

Consider a spanning tree T in a given connected graph G. Let c_i be a chord with respect to T, and let the fundamental circuit made by c_i be called Γ, consisting of k branches $b_1, b_2, \ldots, b_k$ in addition to the chord c_i; that is, $\Gamma = \{c_i, b_1, b_2, \ldots, b_k)$ is a fundamental circuit with respect to T.

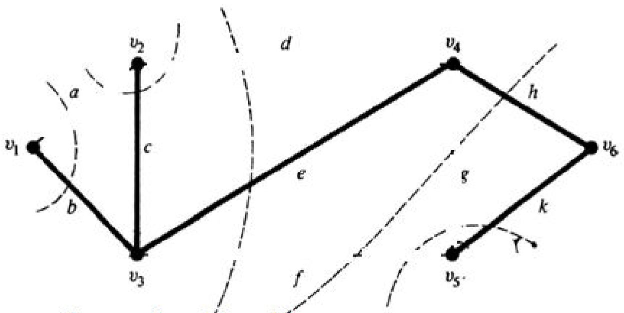


THEOREM 4-5: With respect to a given spanning tree *T*, a chord c_i that determines a fundamental circuit Γ occurs in every fundamental cut-set associated with the branches in Γ and in no other.

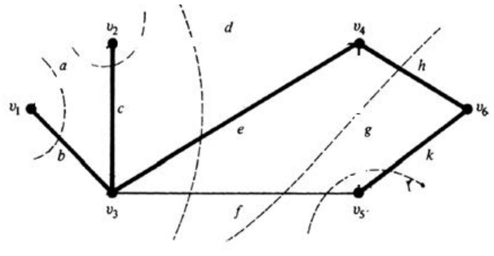
- consider the spanning tree {*b*, *c*, *e*, *h*, *k*}
- The fundamental circuit made by chord *f* is {*f, e, h, k*}.
- The three fundamental cut-sets determined by the three branches *e, h*, and *k* are
  - determined by branch *e*: {*d*, *e*, *f*},
  - determined by branch *h*: {*f*, *g*, *h*},
  - determined by branch *k*: {*f*, *g*, *k*}.
- Chord *f* occurs in each of these three fundamental cut-sets, and there is no other fundamental cut-set that contains *f*. The converse of Theorem 4-5 is also true.



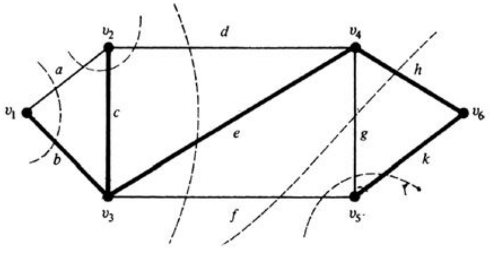
Graph G



Spanning Tree T



Fundamental Circut using f

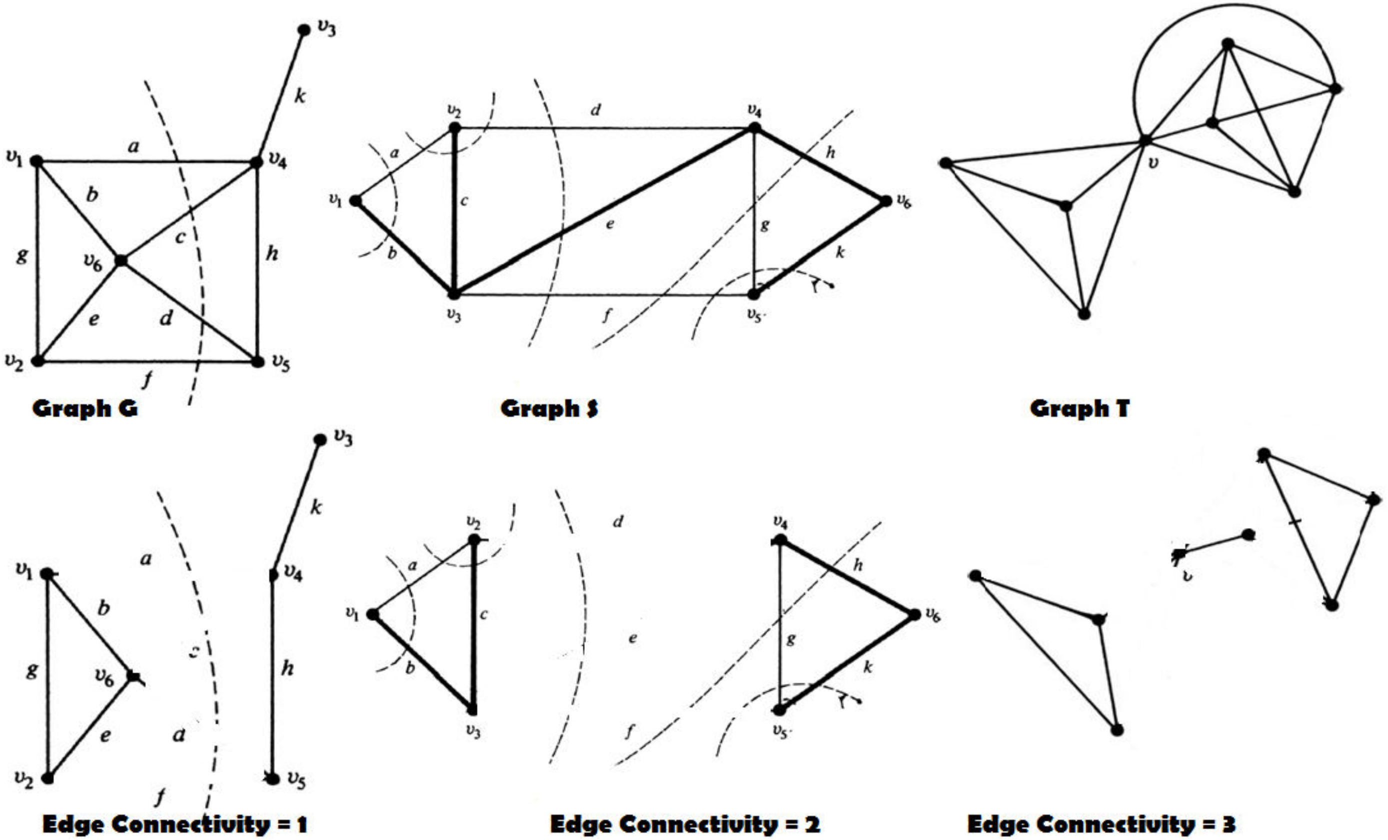


Fundamental cutsets of a graph.

CONNECTIVITY AND SEPARABILITY

Edge Connectivity

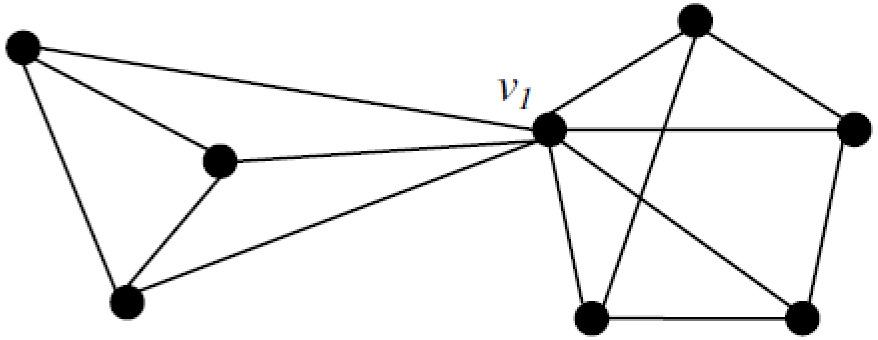
- Each cut-set of a connected graph G consists of certain number of edges.
- The number of edges in the smallest cut-set is defined as the **edge Connectivity of G.**
- The **edge Connectivity** of a connected graph G is defined as the minimum number of edges whose removal reduces the rank of graph by one.
- The edge Connectivity of a tree is one.



Graph G Graph S Graph T

Edge Connectivity = 1 Edge Connectivity = 2 Edge Connectivity = 3

Edge Connectivity

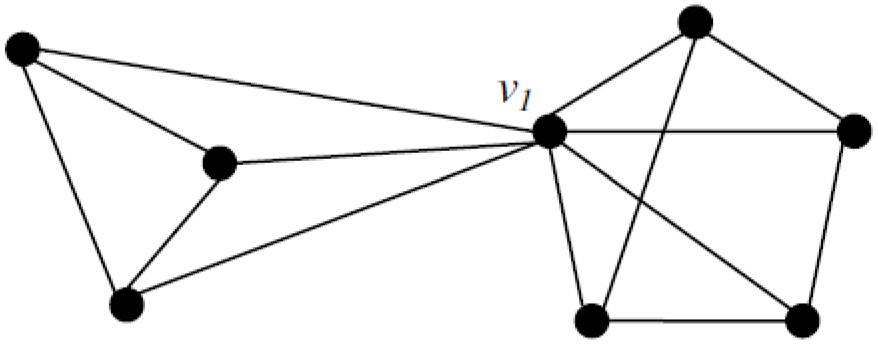


The edge Connectivity of the above graph G is three.

CONNECTIVITY AND SEPARABILITY

Vertex Connectivity

- The **vertex Connectivity** of a connected graph G is defined as the minimum number of vertices whose removal from G leaves the remaining graph disconnected.
- The vertex Connectivity of a tree is one.



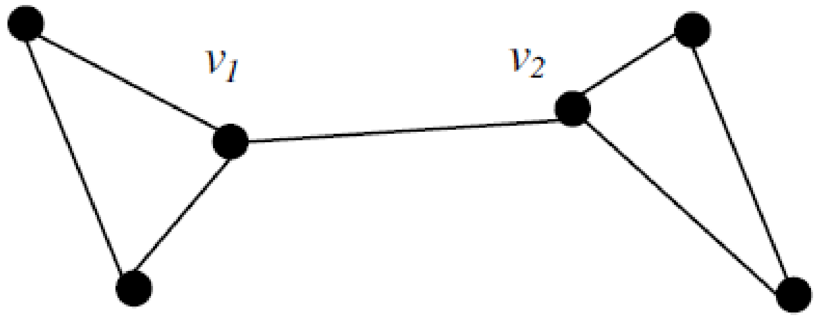
The vertex Connectivity of the above graph G is one.

SEPARABILITY

Separable and non-separable graph

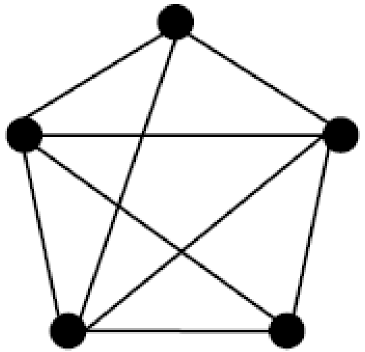
- A connected graph is said to be separable graph if its vertex connectivity is one.
- All other connected graphs are called non-separable graph.

Separable Graph G:



articulation point.

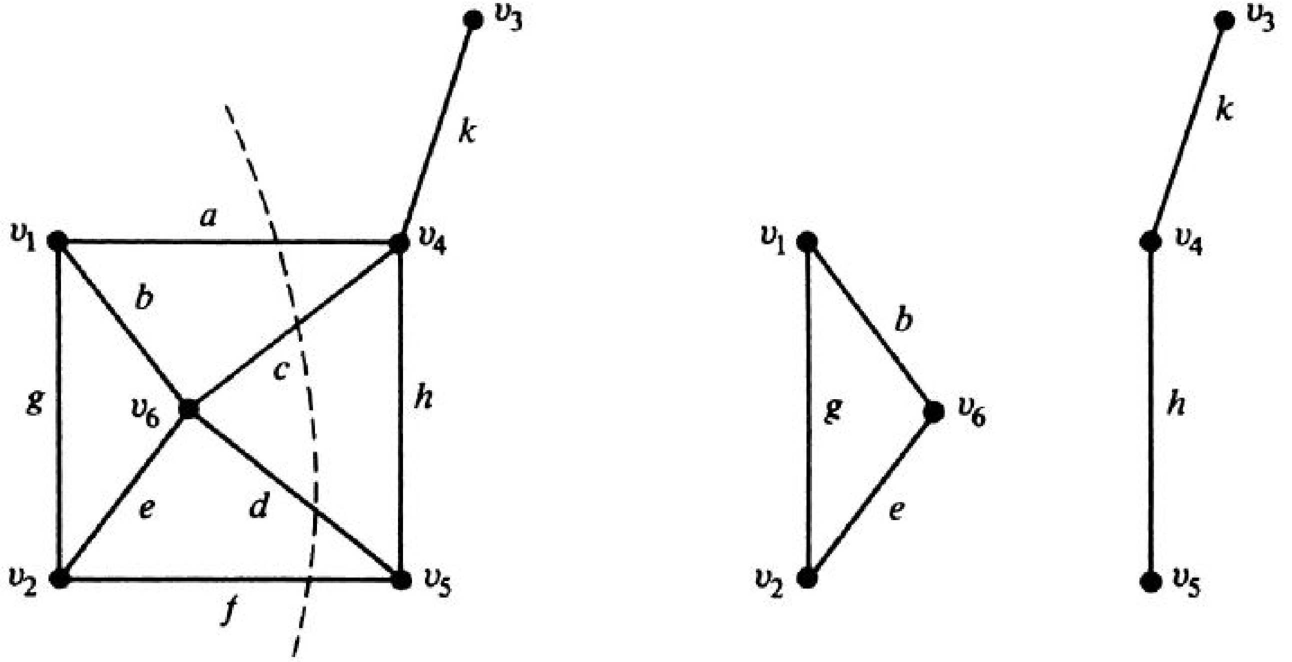
Non-Separable Graph H:



THEOREM 4-8: The edge connectivity of a graph *G* cannot exceed the degree of the vertex with the smallest degree in *G*.

Proof:

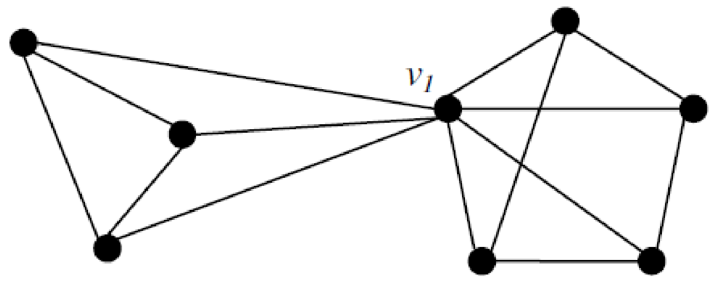
- Let vertex v_i be the vertex with the smallest degree in G. Let $d(v_i)$ be the degree of v_i. Vertex v_i can be separated from *G* by removing the $d(v_i)$ edges incident on vertex v_i. Hence the theorem.



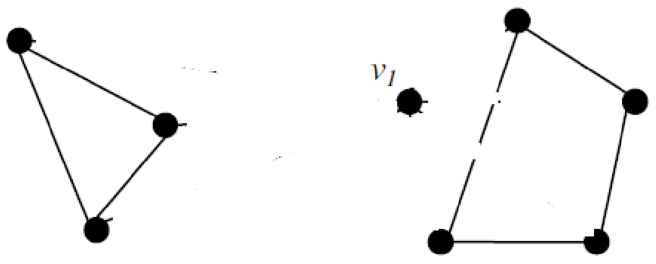
THEOREM 4-9: The vertex connectivity of any graph *G* can never exceed the edge connectivity of *G*.

Proof:

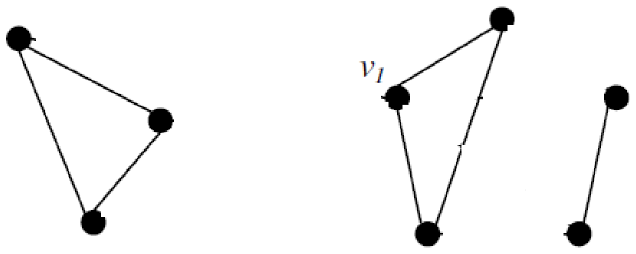
- Let α denote the edge connectivity of *G*. Therefore, there exists a cut-set *S* in *G* with α edges. Let *S* partition the vertices of *G* into subsets V_1 and V_2. By removing at most a vertices from V_1 (or V_2) on which the edges in *S* are incident, we can effect the removal of *S* (together with all other edges incident on these vertices) from G. Hence the theorem.



Graph *G*



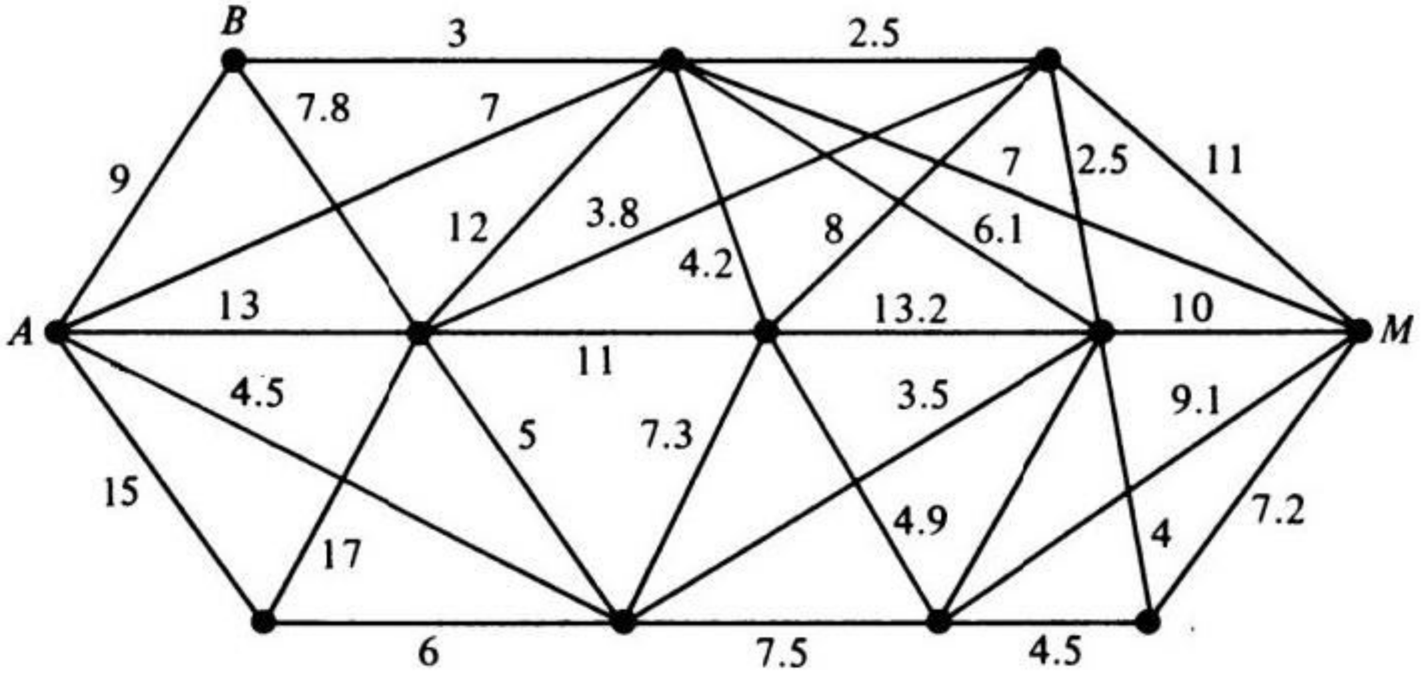
Vertex Connectivity = 1



Edge Connectivity = 3

NETWORK FLOWS

- A flow network (also known as a transportation network) is a graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge.

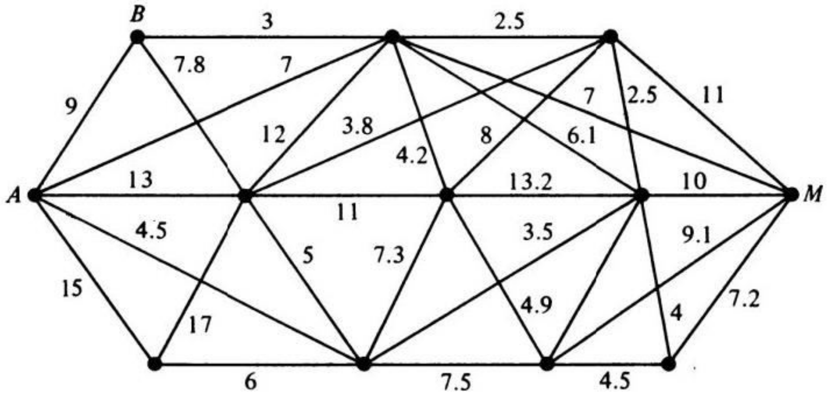


Graph of a flow network.

THEOREM 4-13: The maximum flow possible between two vertices *a* and *b* in a network is equal to the minimum of the capacities of all cut-sets with respect to *a* and *b*.

Proof:

- Consider any cut-set *S* with respect to vertices *a* and *b* in G. In the sub-graph *G − S* (the sub-graph left after removing *S* from *G)* there is no path between *a* and *b*. Therefore, every path in *G* between *a* and *b* must contain at least one edge of *S*. Thus every flow from *a* to *b* (or from *b* to *a*) must pass through one or more edges of *S*. Hence the total flow rate between these two vertices cannot exceed the capacity of *S*. Since this holds for all cut-sets with respect to *a* and *b*, the flow rate cannot exceed the minimum of their capacities.



Graph of a flow network.

1-ISOMORPHISM

A graph G_1 was 1-Isomorphic to graph G_2 if the blocks of G_1 were isomorphic to the blocks of G_2. Two graphs G_1 and G_2 are said to be 1-Isomorphic if they become isomorphic to each other under repeated application of the following operation.

Operation 1: "Split" a cut-vertex into two vertices to produce two disjoint sub-graphs.

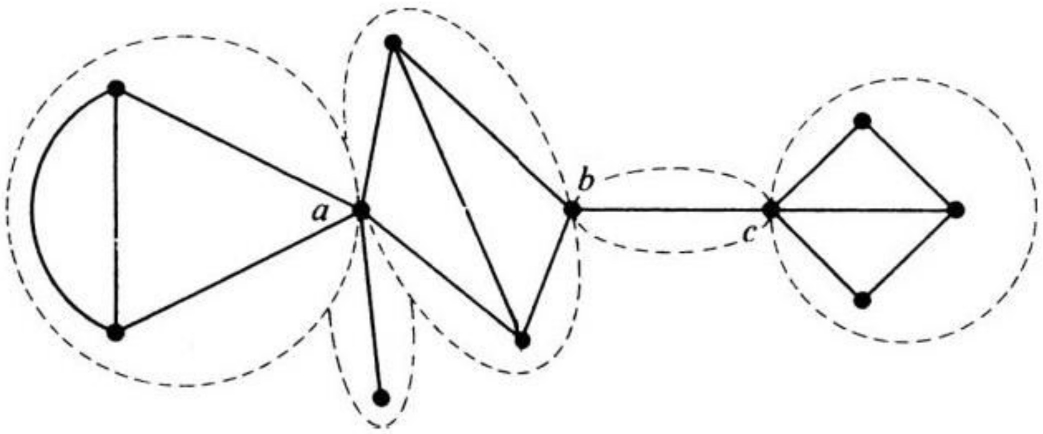


Fig. 4-8 Separable graph with three cut-vertices and five blocks.

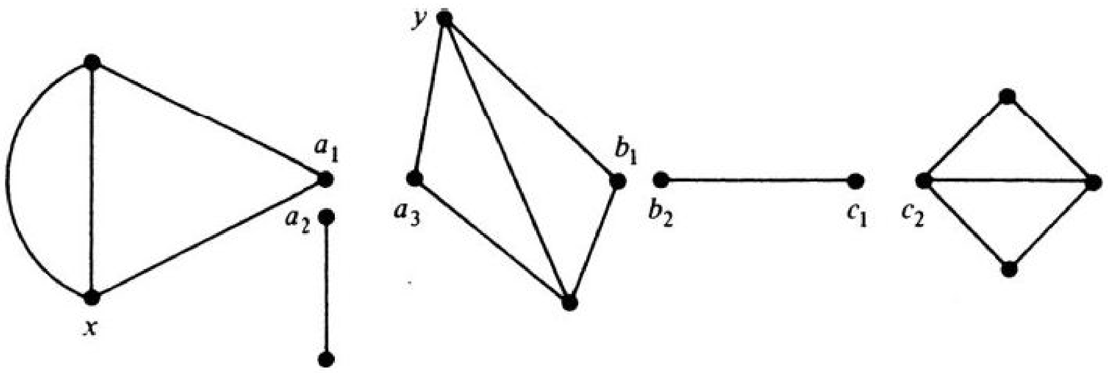


Fig. 4-9 Disconnected graph 1-isomorphic to Fig. 4-8.

THEOREM 4-14: If G_1 and G_2 are two 1-isomorphic graphs, the rank of G_1 equals the rank of G_2 and the nullity of G_1 equals the nullity of G_2.

Proof:

Under operation 1, whenever a cut-vertex in a graph G is "split" in to two vertices, the number of components in G increases by one. Therefore, the rank of G which is number of vertices in G − number of components in G remains invariant under operation 1. Also, since no edges are destroyed or new edges created by operation 1, two 1-isomorphic graphs have the same number of edges. Two graphs with equal rank and with equal numbers of edges must have the same nullity, because nullity = number of edges − rank.

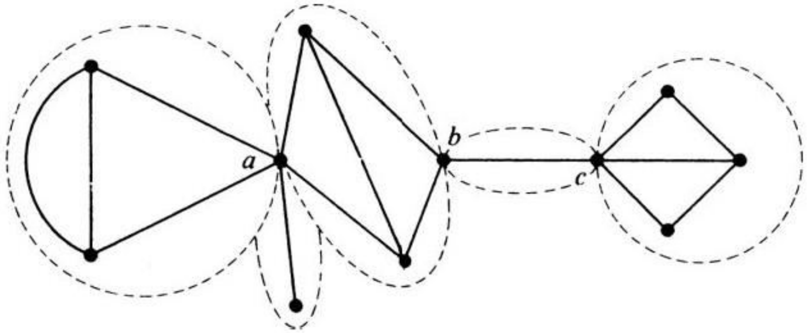


Fig. 4-8 Separable graph with three cut-vertices and five blocks.

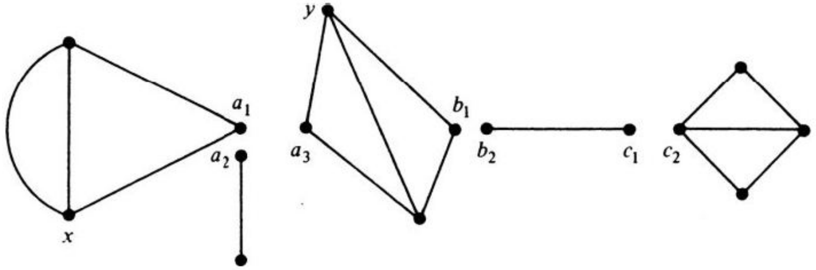


Fig. 4-9 Disconnected graph 1-isomorphic to Fig. 4-8.

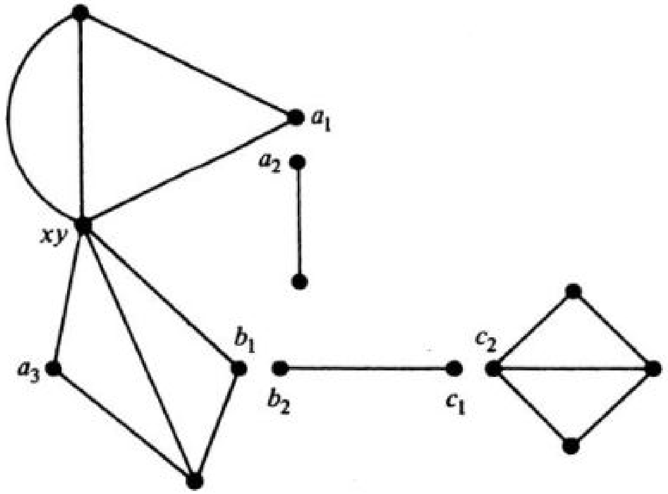


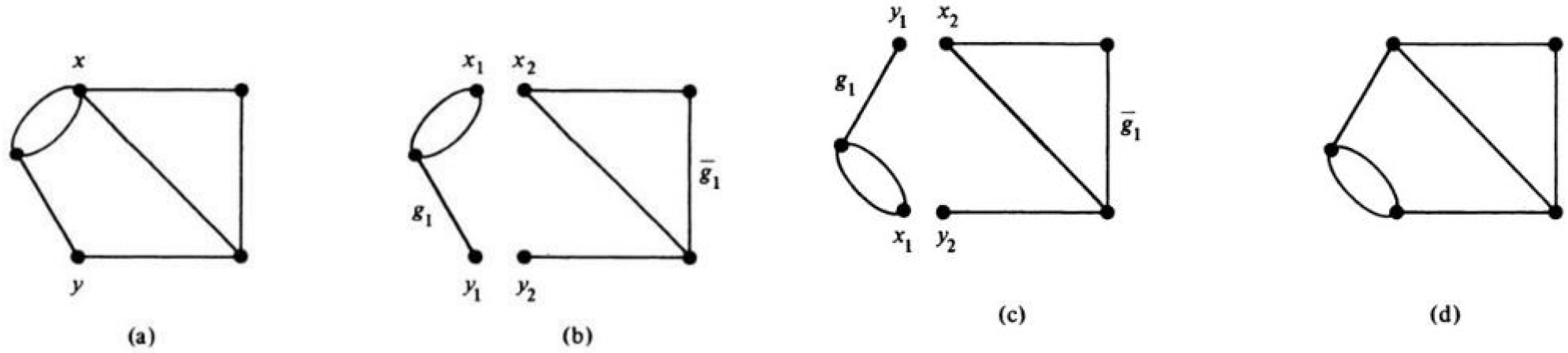
Fig. 4-10 Graph 1-isomorphic to Figs. 4-8 and 4-9.

2-ISOMORPHISM

Two graphs G_1 and G_2 are said to be 2-Isomorphic if they become isomorphic after undergoing operation 1 or operation 2, or both operations any number of times.

Operation 1: "Split" a cut-vertex into two vertices to produce two disjoint sub-graphs.

Operation 2: "Split" the vertex x into x_1 and x_2 and the vertex y into y_1 and y_2 such that G is split into g_1 and g_2. Let vertices x_1 and y_1 go with g_1 and vertices x_2 and y_2 go with g_2. Now rejoin the graphs g_1 and g_2 by merging x_1 with y_2 and x_2 with y_1.

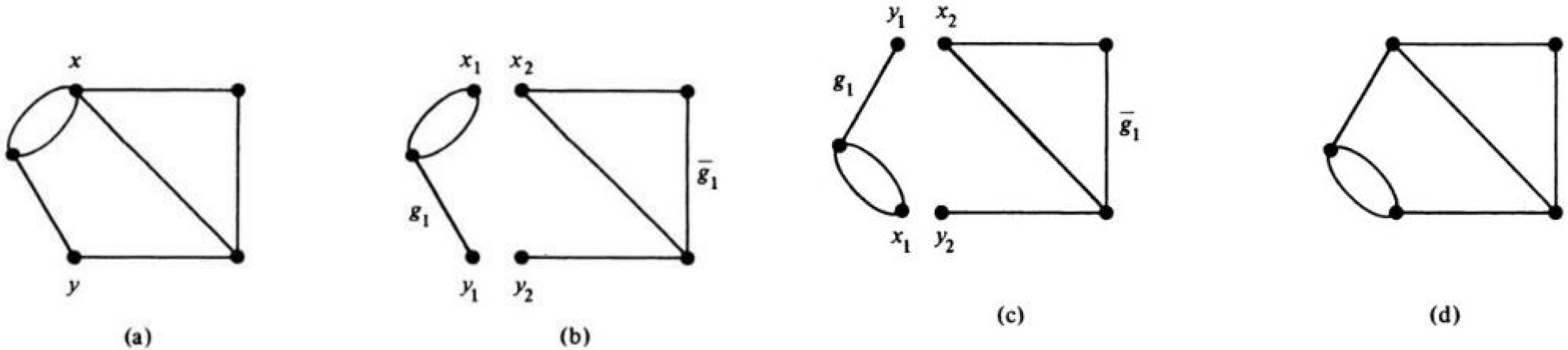


2-isomorphic graphs (a) and (d).

THEOREM 4-15: Two graphs are 2-isomorphic if and only if they have circuit correspondence.

Proof:

The "only if" part has already been shown in the argument preceding the theorem. The "if" part is more involved, and the reader is referred to Whitney's original paper [4-7].As we shall observe in subsequent chapters, the ideas of 2-isomorphism and circuit correspondence play important roles in the theory of contact networks, electrical networks, and in duality of graphs.



2-isomorphic graphs (a) and (d).

REFERENCES:

1. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice-Hall of India Pvt. Ltd, 2003.
2. S. Pirzada, “An Introduction to Graph theory”, University Press, 2012.
3. Frank Harary, “Graph Theory”, Narosa Publishing House, 2001.
4. West D. B., “Introduction to Graph Theory”, 2nd Edition, Pearson Education, 2001.
5. Diestel R, "Graph Theory", 5th Edition, Springer, 2017.

Thank you.