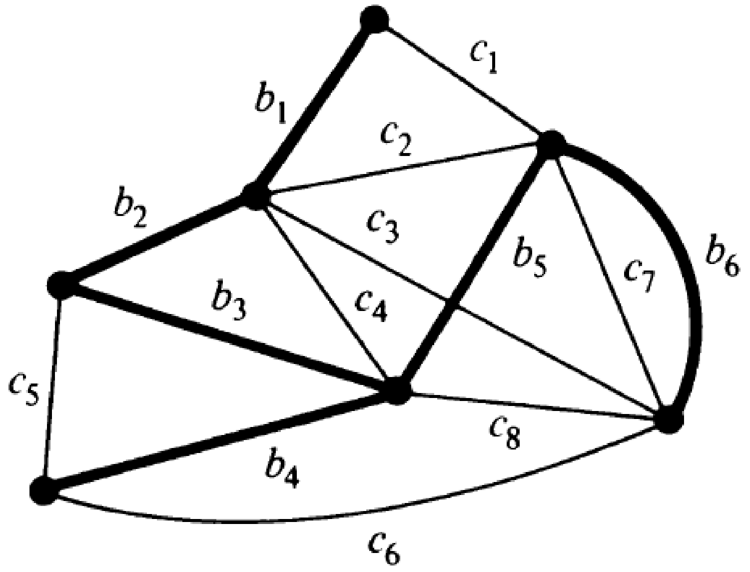
Fundamental Circuits

Fundamental Circuits

Let us now consider a spanning tree T in a connected graph G. 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*.



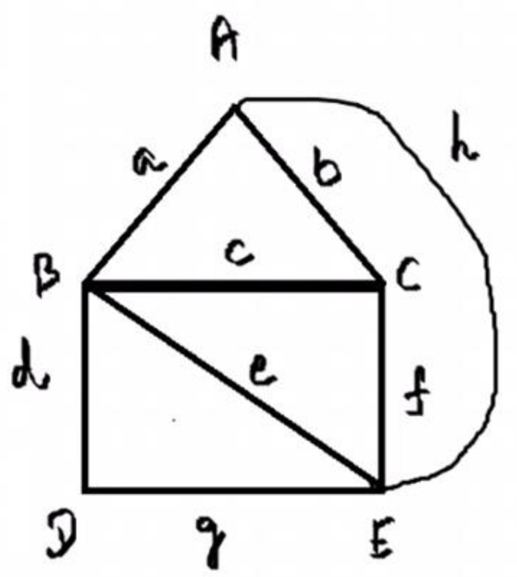
Let us look at the tree $\{b_1, b_2, b_3, b_4, b_5, b_6\}$

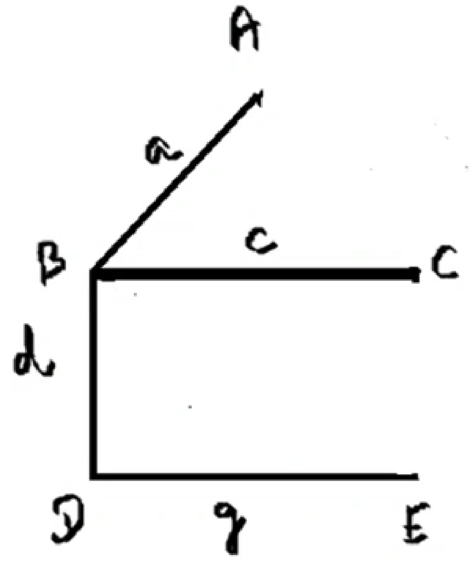
$\{b_1, b_2, b_3, b_4, b_5, b_6, c_1\}$ → $\{b_1, b_2, b_3, b_5, c_1\}$

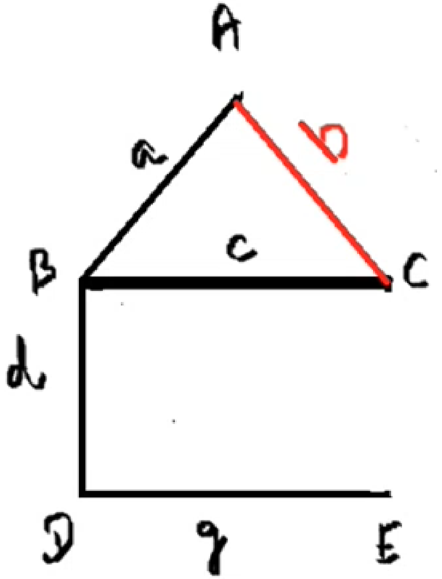
$\{b_1, b_2, b_3, b_4, b_5, b_6, c_1, c_2\}$ → $\{b_1, b_2, b_3, b_5, c_1\}$

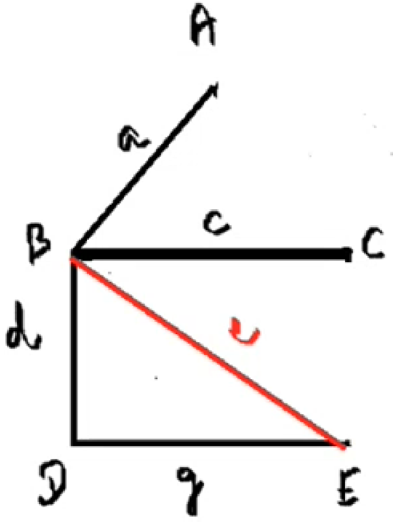
→ $\{b_2, b_3, b_5, c_2\}$

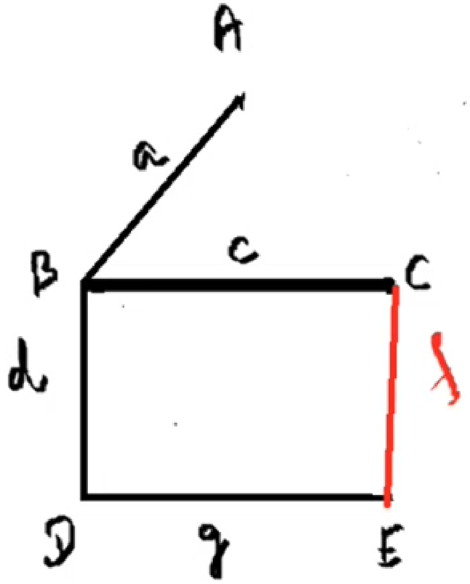
→ $\{b_1, c_1, c_2\}$ ✗

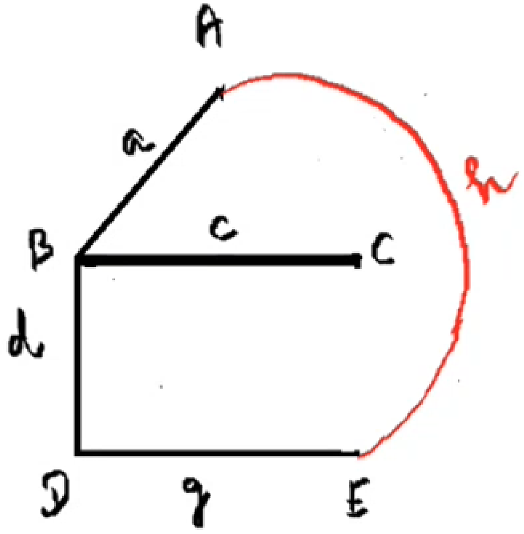

A
a
b
h
B
c
C
d
e
f
D
g
E


A
a
B
c
C
d
D
g
E


A
a
b
B
c
C
d
D
g
E

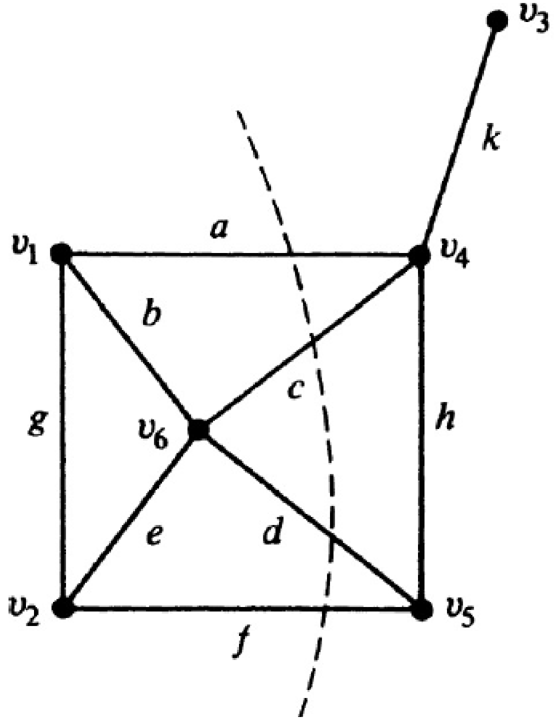

A
a
B
c
C
d
e
D
g
E

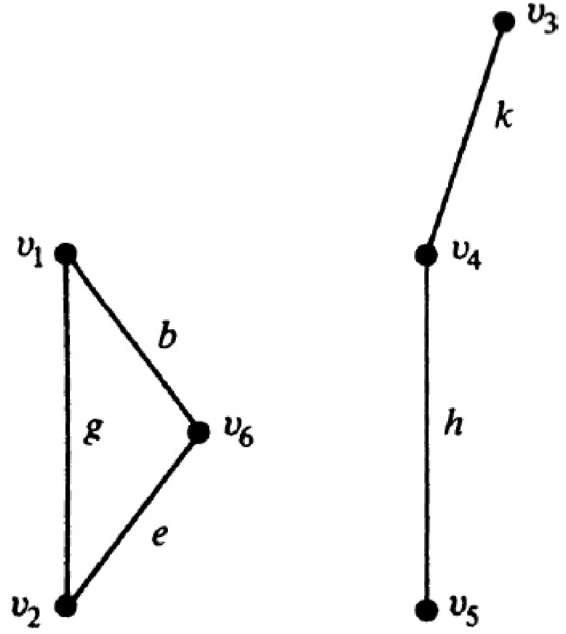

A
a
B
c
C
d
f
D
g
E


A
a
h
B
c
C
d
D
g
E

Cut-sets or edge cut

- 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.
- For instance, in Fig. the set of edges {a, c, d, f} is a cut-set.
- There are many other cut-sets, such as {a, b, g} , {a, b, e, f} , and {d, h, f}. Edge {k} alone is also a cut-set.
- The set of edges {a, c, h, d} , on the other hand, is not a cut-set, because one of its proper subsets, {a, c, h}, is a cut-set.

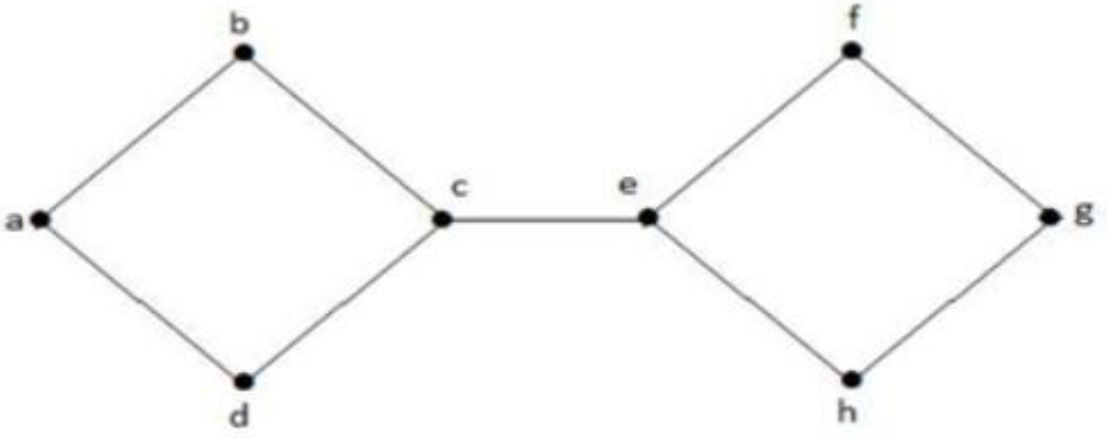


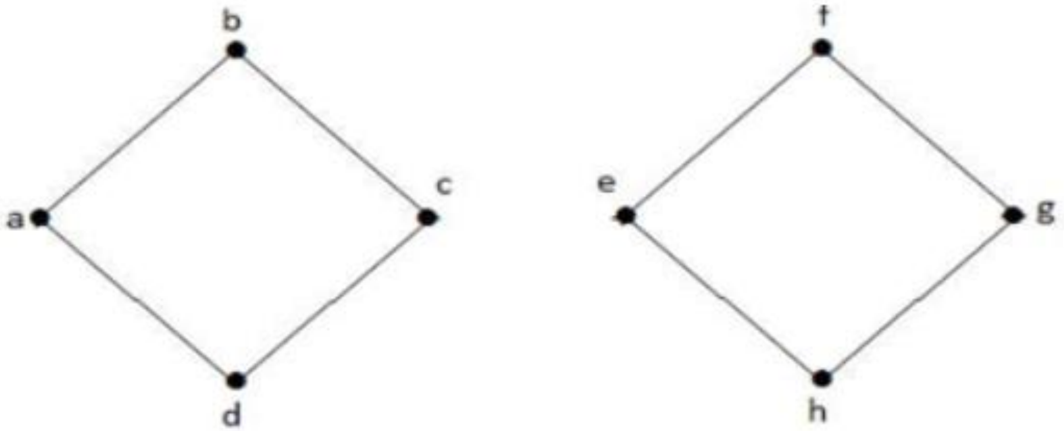


- To emphasize the fact that no proper subset of a cut-set can be a cut-set, some authors refer to a cut-set as a minimal cut-set, a proper cut-set, or a simple cut-set.
- A cut-set always “cuts” a graph into two.
- Therefore, a cut-set can also be defined as a minimal set of edges in a connected graph whose removal reduces the rank of the graph by one.
- If we partition all the vertices of a connected graph G into two mutually exclusive subsets, a cut-set is a minimal number of edges whose removal from G destroys all paths between these two sets of vertices.
  - For example, in Fig. cut-set {a, c, d, f} connects vertex set {v1, v2 , v6} with {v3, v4, v5}.
- Since removal of any edge from a tree breaks the tree into two parts, every edge of a tree is a cut-set.
- Edge connectivity, K’(G), is the size of the smallest edge cut.

Bridge or cut edge

- A **cut- Edge or bridge** is a single edge whose removal disconnects a graph.
- Let G be a connected graph. An edge e of G is called a cut edge of G, if G-e results a disconnected graph.

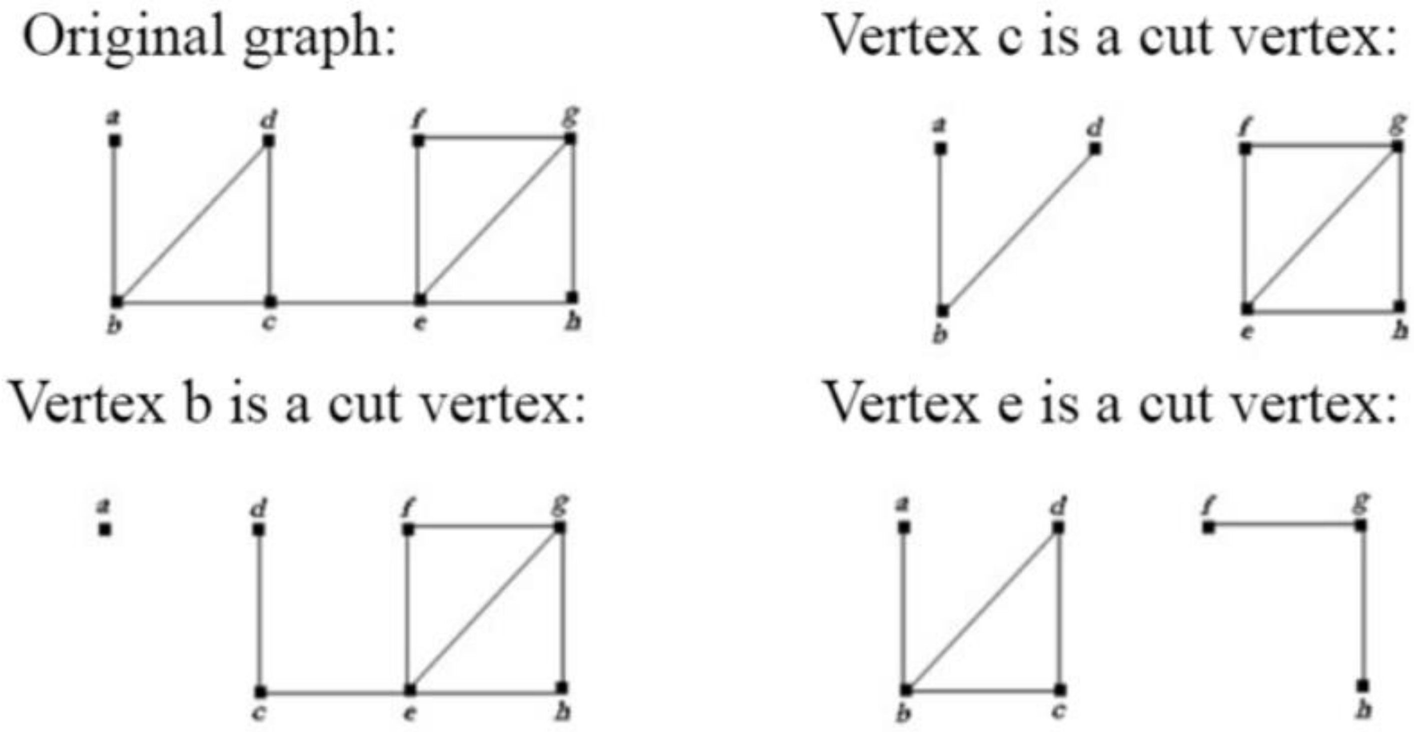




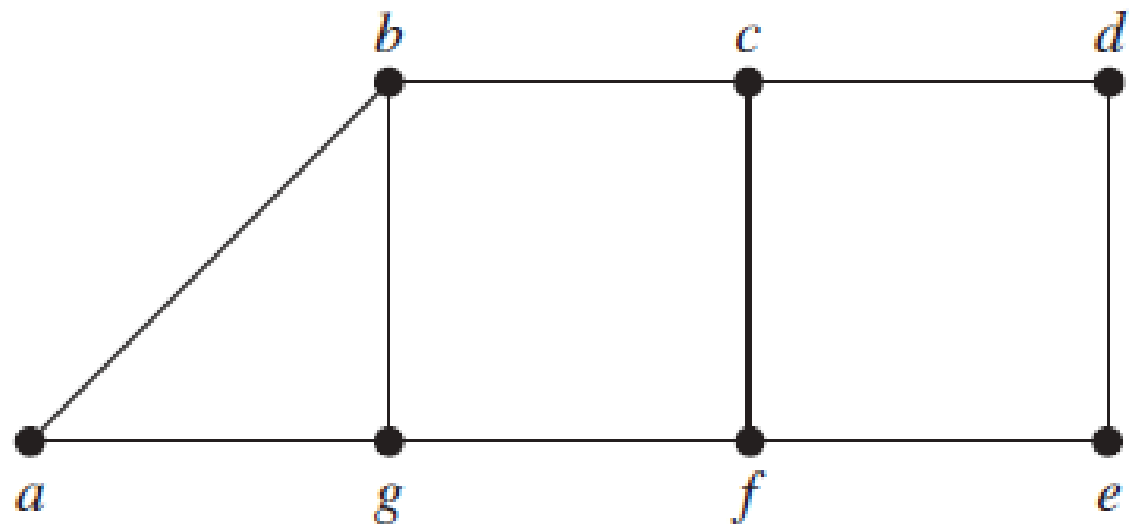
- Theorem: An edge *e* ϵ *E(G)* is a bridge if and only if $\exists u, w \in V(G)$, $u \neq w$ such that *e* is on every *u*-*w* path of G

Vertex Cut

- A subset V’ of the vertex set V of G = (V, E) is a vertex cut, or separating set, if G − V’ is disconnected.
- A vertex v Є V(G) is called a cut vertex, a cut-node, or an articulation point if G-v has more connected components than G.



- A graph is k-connected if there does not exist a set of k-1 vertices whose removal disconnects the graph.
- The **vertex connectivity K(G)** of G is defined as the largest k such that G is k-connected. It is the minimum size of a vertex set $S \subseteq V \ such \ that \ G - S$ is disconnected or has only one vertex (in case of complete graph).
  - **vertex connectivity** of a noncomplete graph *G* is the minimum number of vertices in a vertex cut
  - When G is a complete graph, it has no vertex cuts, because removing any subset of its vertices and all incident edges still leaves a complete graph. Consequently, we cannot define κ(G) as the minimum number of vertices in a vertex cut when G is complete. Instead, we set $\kappa(K_n) = n - 1$, the number of vertices needed to be removed to produce a graph with a single vertex.



G_3

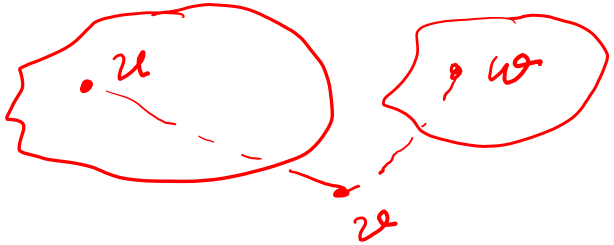
- *G*3 has no cut vertices, but that {*b, g*} is a vertex cut. Hence, κ(*G*3) = 2.
- There is no cut-vertex in this graph.

- Disconnected graphs and K_1 have $\kappa(G) = 0$
- Connected graphs with cut vertices and K_2 have $\kappa(G) = 1$
- Graphs without cut vertices that can be disconnected by removing two vertices and K_3 have $\kappa(G) = 2$, and so on
- If G is a k-connected graph, then G is a j-connected graph for all j with $0 \leq j \leq k$.

- Theorem: A vertex *v* ∈ *V(G)* is a cut vertex of G if and only if $\exists u, w \in V(G),\ u, w \neq v$ such that *v* is on every *u*-*w* path of G

Proof: Assume G is connected. (otherwise repeat the argument on connected components).

$\Rightarrow$ Let $v \in V(G)$ be a cut vertex. Then $G-v$ is disconnected. Let u, w be vertices in different components of $G-v$. So $\nexists$ any u-w path in $G-v$.



But G is connected so $\exists$ u-w path in G.

$\therefore$ all such paths went through v.

$\Leftarrow$ Suppose $\exists\, u, w \in V(G),\ u, w \neq v$ such that v lies on every u-w path.

Then removing v means $\nexists$ any u-w path in $G-v$.

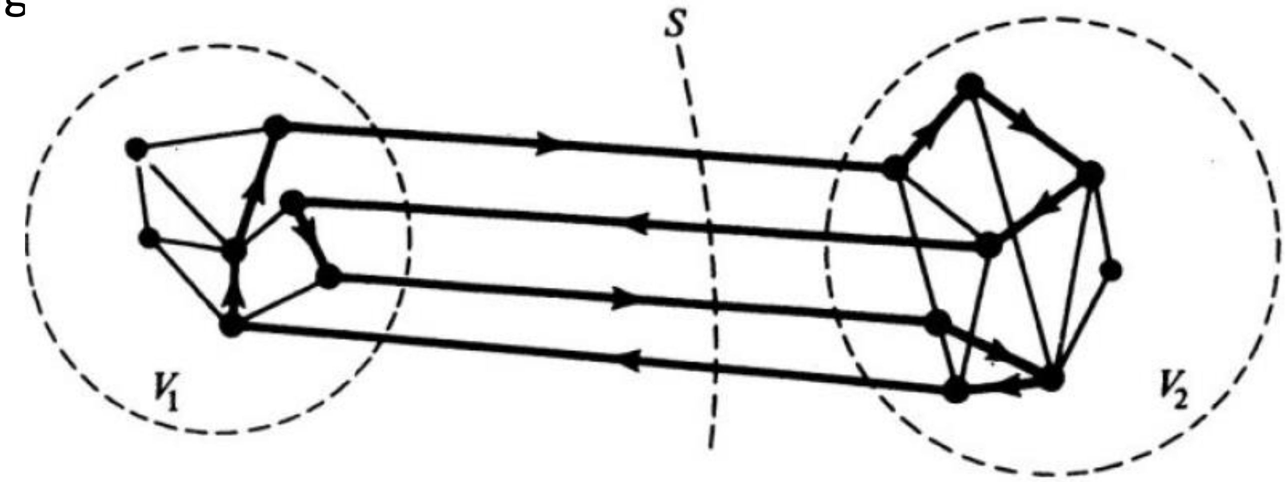
$\therefore$ $G-v$ is disconnected & v is a cut-vertex.

- Theorem 1: Every non-trivial connected graph contains at least two vertices that are not cut-vertices.
- Proof (by contradiction):
    - There exists a non-trivial connected graph with at most 1 non cut-vertex.
    - Let $u, v \in V(G)$ with distance, $d(u, v) = Diam(G)$.
    - At least one of u, v is a cut vertex, say v. So $\mathrm{G} - v$ is disconnected. Let $w \in V(G)$ be in a different component of $G - v$ than u. Then, every uw path v.
    - Therefore, $d(u, w) \; > d(u, v) = Diam(G)$ which is a contradiction.
    - Hence, the theorem is true.

- A non-trivial connected graph is called non-separable or 2-connected if it has no cut-vertices

- THEOREM 2: Every cut-set in a connected graph G must contain at least one branch of every spanning tree of G.
- THEOREM 3: 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.
- Proof:
  - 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. 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.

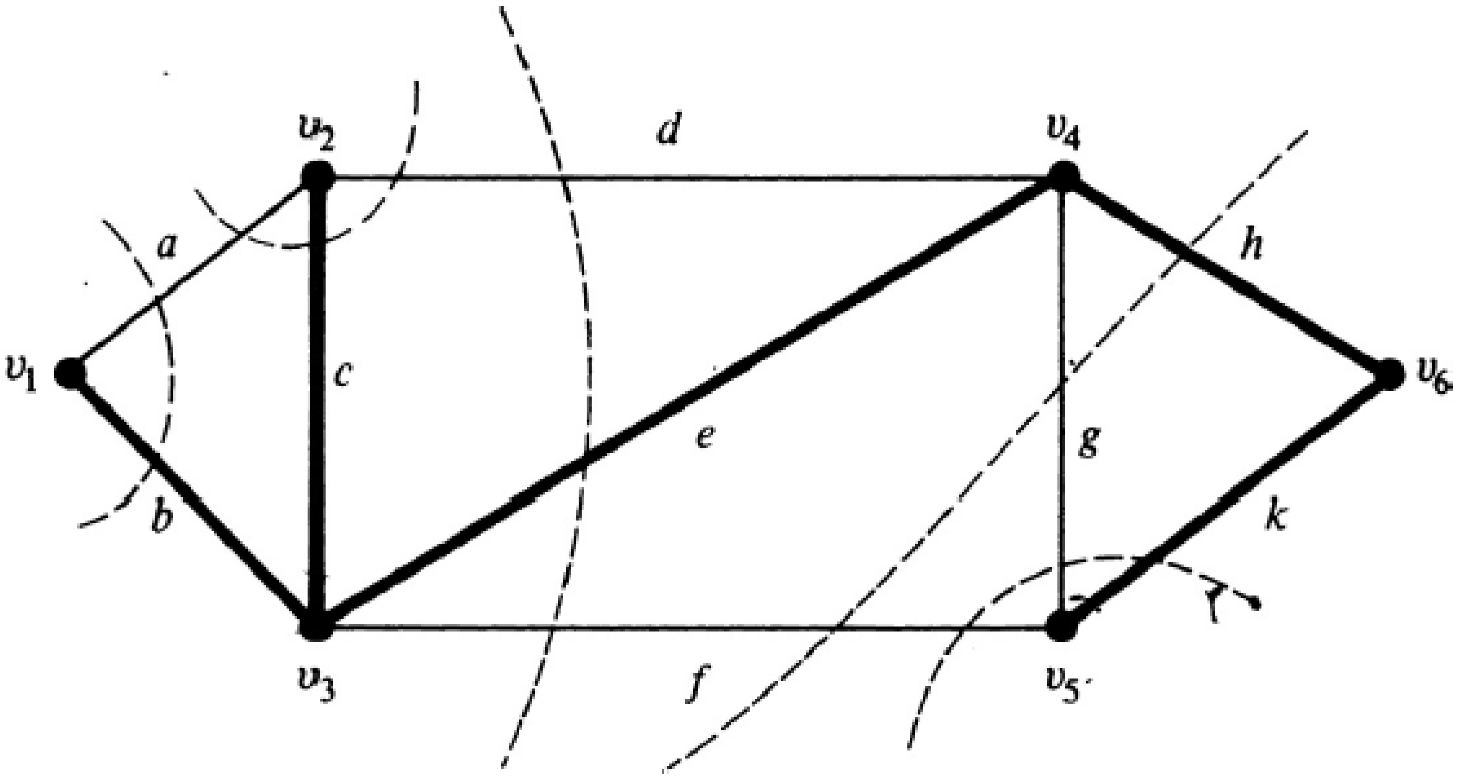
- THEOREM 4: Every circuit has an even number of edges in common with any cut-set.
- *Proof:*
  - Consider a cut-set *S* in graph *G*. Let the removal of *S* partition the vertices of *G* into two (mutually exclusive or disjoint) subsets *V*1 and *V*2. Consider a circuit Γ in *G*.
  - If all the vertices in Γ are entirely within vertex set *V1* (or *V*2), the number of edges common to *S* and Γ is zero; that is, *N*(*S* ⋂ Γ) = 0, an even number.
  - If, on the other hand, some vertices in Γ are in *V*1and some in *V*2, we traverse back and forth between the sets *V*1 and *V*2 as we traverse the circuit. 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*1and *V*2), the number of edges common to S and Γ is even.



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

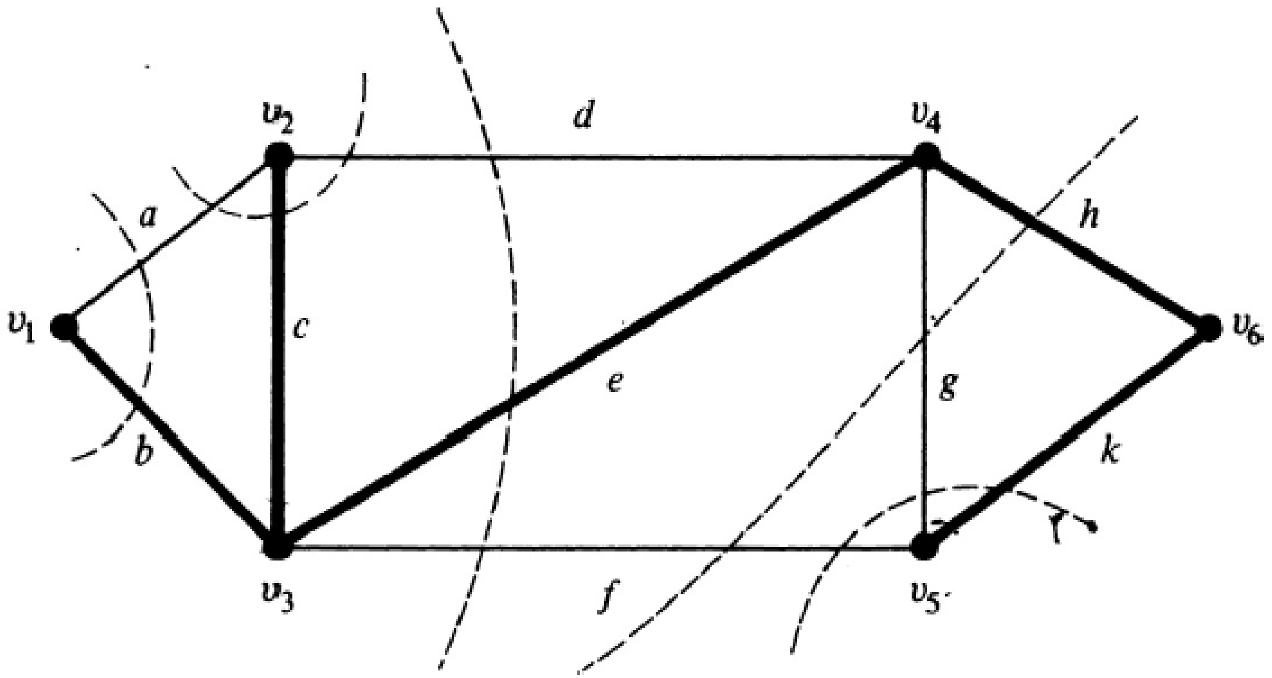
Fundamental Cut-Sets

- Consider a spanning tree T of a connected graph G. Take any branch b in T. Since $\{b\}$ is a cut-set in T, $\{b\}$ partitions all vertices of T into two disjoint sets—one at each end of b.
- Consider the same partition of vertices in G, and the cut set S in G that corresponds to this partition.
- Cut-set S will contain only one branch b of T, and the rest (if any) of the edges in S are chords with respect to T. Such a cut-set S containing exactly one branch of a tree T is called a *fundamental cut-set* with respect to T.
- Sometimes a fundamental cut-set is also called a *basic cut-set*.



- A spanning tree T (in heavy lines) and all five of the fundamental cut-sets with respect to T are shown (broken lines “cutting” through each cut-set).

- THEOREM 5: The ring sum of any two cut-sets in a graph is either a third cut-set or an edge-disjoint union of cut-sets.

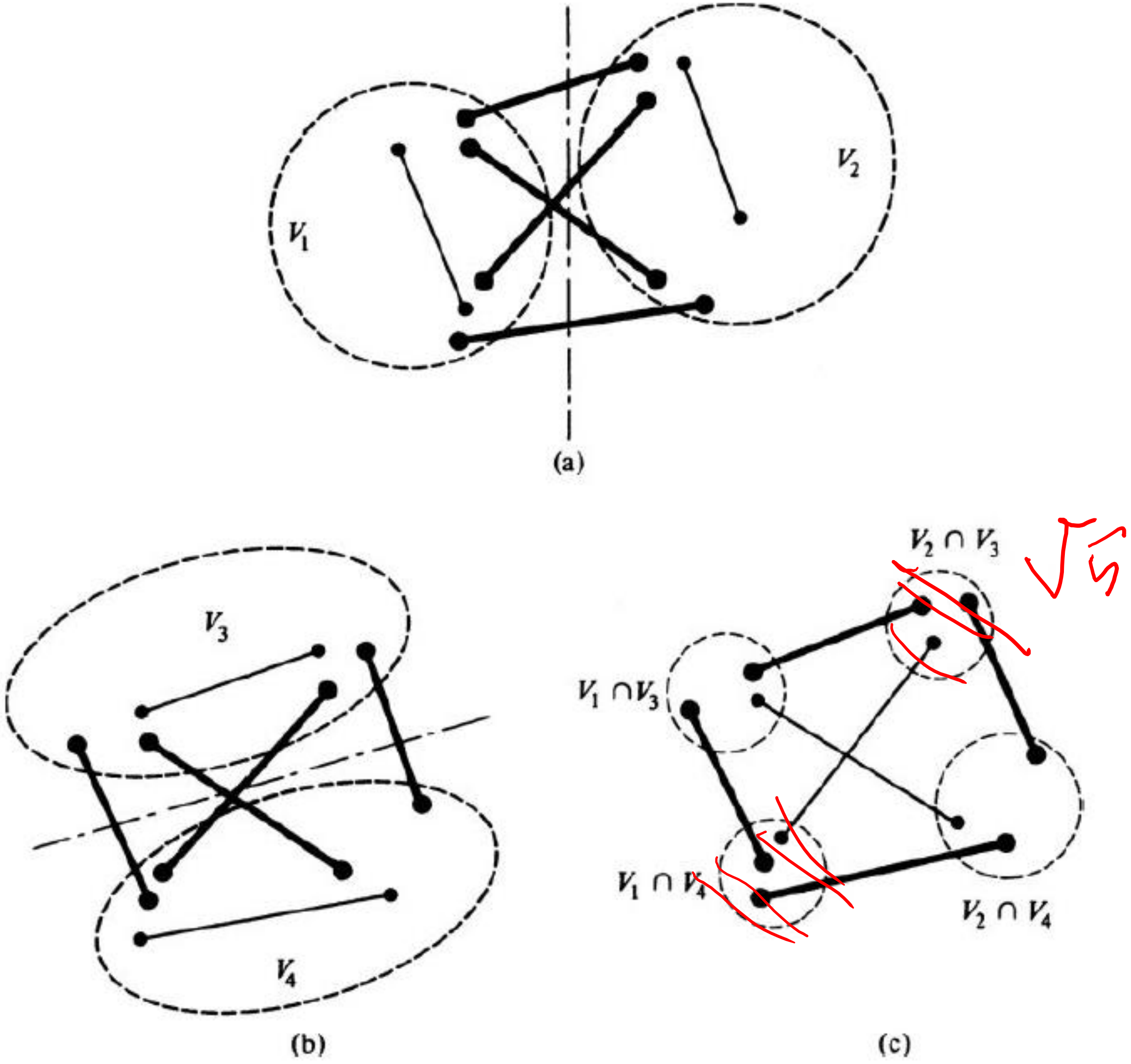


$$\{d, e, f\} \oplus \{f, g, h\} = \{d, e, g, h\}, \qquad \text{another cut-set,}$$

$$\{a, b\} \oplus \{b, c, e, f\} = \{a, c, e, f\}, \qquad \text{another cut-set,}$$

$$\begin{aligned}\{d, e, g, h\} \oplus \{f, g, k\} &= \{d, e, f, h, k\} \\ &= \{d, e, f\} \cup \{h, k\}, \text{ an edge-disjoint} \\ &\quad \text{union of cut-sets.} \quad \blacksquare\end{aligned}$$

- ***Proof:*** Let $S1$ and $S2$ be two cut-sets in a given connected graph G.
- Let $V1$ and $V2$ be the (unique and disjoint) partitioning of the vertex set V of G corresponding to $S1$. Let $V3$ and $V4$ be the partitioning corresponding to $S2$.
- $V1 \cup V2 = V$ and $V1 \cap V2 = \emptyset$,
- $V3 \cup V4 = V$ and $V3 \cap V4 = \emptyset$.
- Now let the subset $(V1 \cap V4) \cup (V2 \cap V3)$ be called $V5$, and this by definition is the same as the ring sum $V1 \oplus V3$. Similarly, let the subset $(V1 \cap V3) \cup (V2 \cap V4)$ be called $V6$, which is the same as $V2 \oplus V3$.
- The ring sum of the two cut-sets $S1 \oplus S2$ can be seen to consist only of edges that join vertices in $V5$ to those in $V6$. Also, there are no edges outside $S1 \oplus S2$ that join vertices in $V5$ to those in $V6$.
- Thus, the set of edges $S1 \oplus S2$ produces a partitioning of V into $V5$ and $V6$ such that
- $V5 \cup V6 = V$ and $V5 \cap V6 = \emptyset$.
- Hence, $S1 \oplus S2$ is a cut-set if the subgraphs containing $V5$ and $V6$ each remain connected after $S1 \oplus S2$ is removed from G. Otherwise, $S1 \oplus S2$ is an edge-disjoint union of cut-sets.

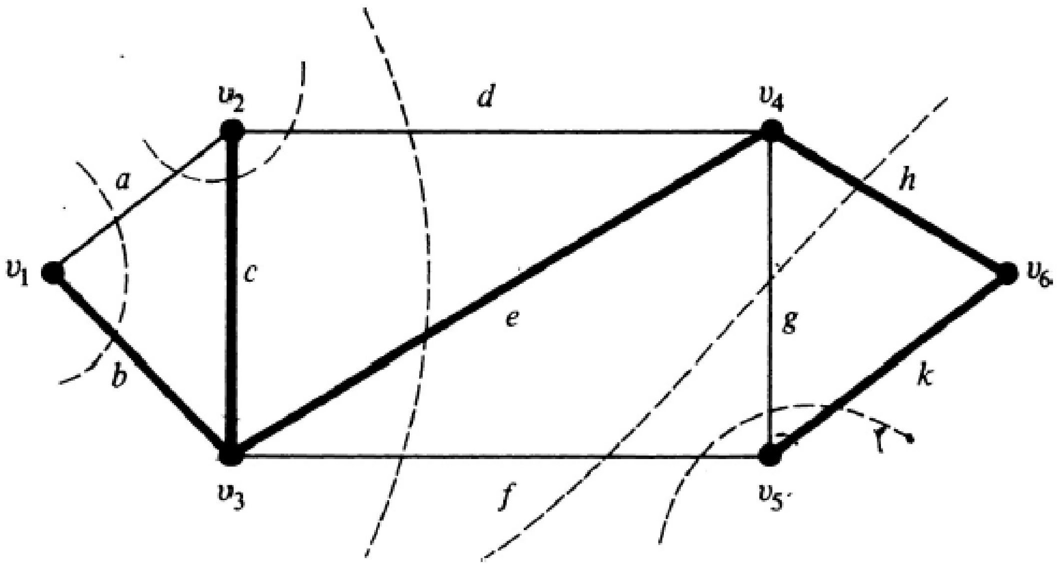


Two cut-sets and their partitioning's.

Fundamental Circuits and Cut-sets

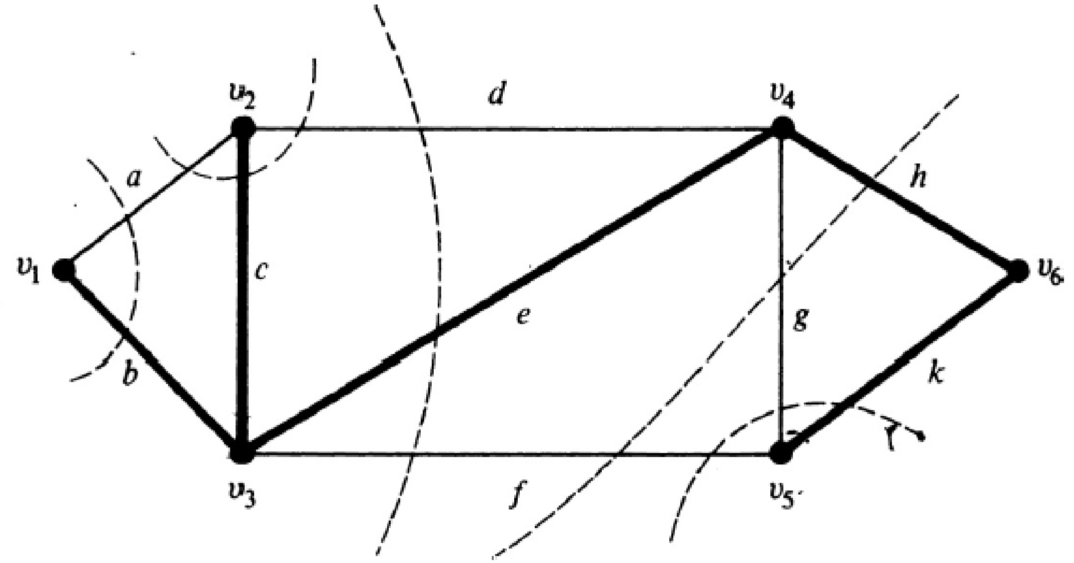
- THEOREM 6: With respect to a given spanning tree *T*, a chord c_i that determines a fundamental circuit Γ occurs in every fundamental cut-set associated with the branches in Γ and in no other.
- THEOREM 7: With respect to a given spanning tree *T*, a branch b_i that determines a fundamental cut-set *S* is contained in every fundamental circuit associated with the chords in *S*, and in no others.

- **Proof of Theorem 6:** 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 $b_1, b_2, \dots, b_k$ in addition to the chord c_i; that is,
  - $\Gamma = \{ci, b1, b2, \dots, bk)$ is a fundamental circuit with respect to T
- Every branch of any spanning tree has a fundamental cut-set associated with it. Let $S1$ be the fundamental cut-set associated with b_1, consisting of q chords in addition to the branch b_1; that is,
  - $S1 = \{b_l, c_l, c_2, \dots, c_q\}$ is a fundamental cut-set with respect to T
- Because of Theorem 4, there must be an even number of edges common to Γ and $S1$. Edge $b1$ is in both Γ and $S1$, and there is only one other edge in Γ (which is c_i) that can possibly also be in $S1$. Therefore, we must have two edges b_1 and c_i common to $S1$ and Γ. Thus, the chord c_i is one of the chords $c_1, c_2, \dots c_q$.
- Exactly the same argument holds for fundamental cut-sets associated with $b_2, b_3, \dots, b_k$. Therefore, the chord c_i is contained in every fundamental cut-set associated with branches in Γ.
- It is not possible for the chord c_i to be in any other fundamental cut-set S' (with respect to T) besides those associated with $b_1, b_2, \dots$ and b_k. Otherwise (since none of the branches in Γ are in S'), there would be only one edge c_i common to S' and Γ, a contradiction to Theorem 4.



- 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*.

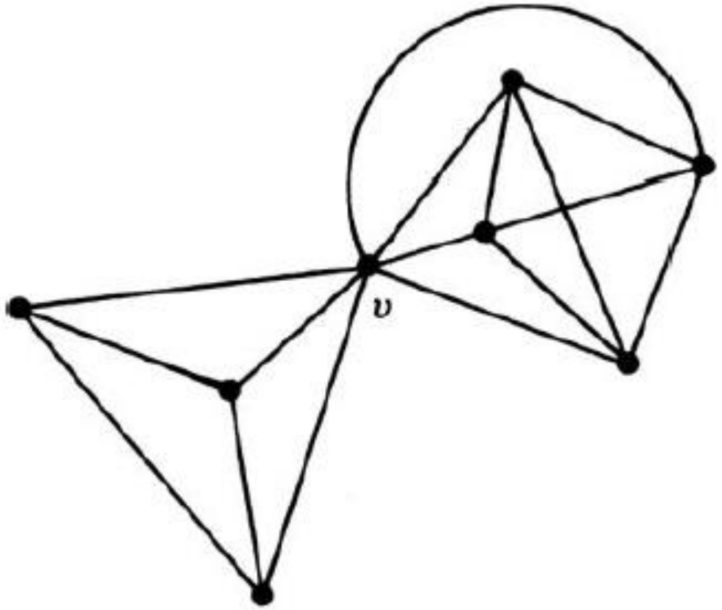
- **Proof of Theorem 7:** Let the fundamental cut-set S determined by a branch b_i be $S = \{b_i, c_1, c_2, \ldots, c_p\}$, and let Γ_1 be the fundamental circuit determined by chord c_1: $\Gamma_1 = \{c_1, b_1, b_2, \ldots, b_q\}$.
- Since the number of edges common to S and Γ_1 must be even, b_i must be in Γ_1.
- The same is true for the fundamental circuits made by chords $c_2, c_3, \ldots, c_p$.
- On the other hand, suppose that b_i occurs in a fundamental circuit Γ_{p+1} made by a chord other than $c_1, c_2, \ldots, c_p$. Since none of the chords $c_1, c_2, \ldots, c_p$ is in Γ_{p+1}, there is only one edge b_i common to a circuit Γ_{p+1} and the cut-set S, which is not possible. Hence, the theorem.

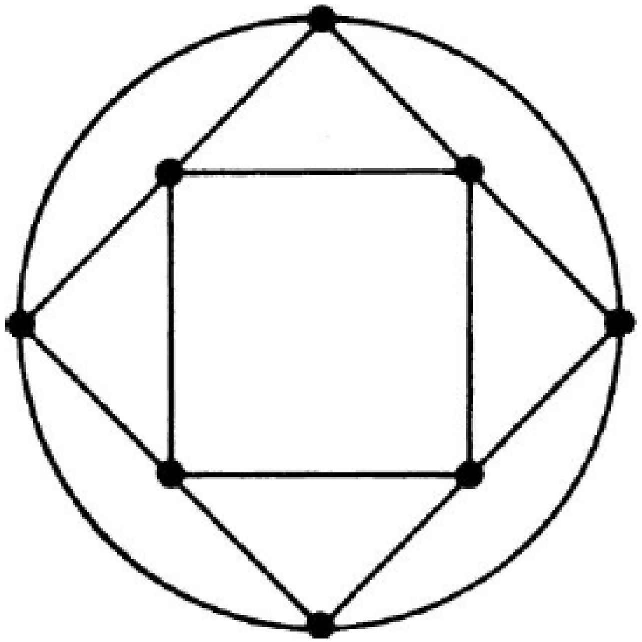


- Consider branch *e* of the spanning tree {*b*, *c*, *e*, *h*, *k*}.
- The fundamental cut-set determined by *e* is {*e*, *d*, *f*}.
- The two fundamental circuits determined by chords *d* and *f* are
  - determined by chord *d:* {*d*, *c*, *e*},
  - determined by chord *f*: {*f*, *e*, *h*, *k*}.
- Branch *e* is contained in both these fundamental circuits, and none of the remaining fundamental circuits contains branch *e*.

Separable Graph

- A connected graph is said to be separable if its vertex connectivity is one.
- All other connected graphs are called non-separable.
- An equivalent definition is that a connected graph G is said to be separable if there exists a subgraph g in G such that g'(the complement of g in G) and g have only one vertex in common.





- Two graphs with 8 vertices and 16 edges.
- First graph has Vertex connectivity, $K(G) = 1$ and edge connectivity, $K'(G) = 3$
- Second one has edge connectivity as well as the vertex connectivity of four
- Thus, the network of Second is better connected than that of First.

- **THEOREM 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.

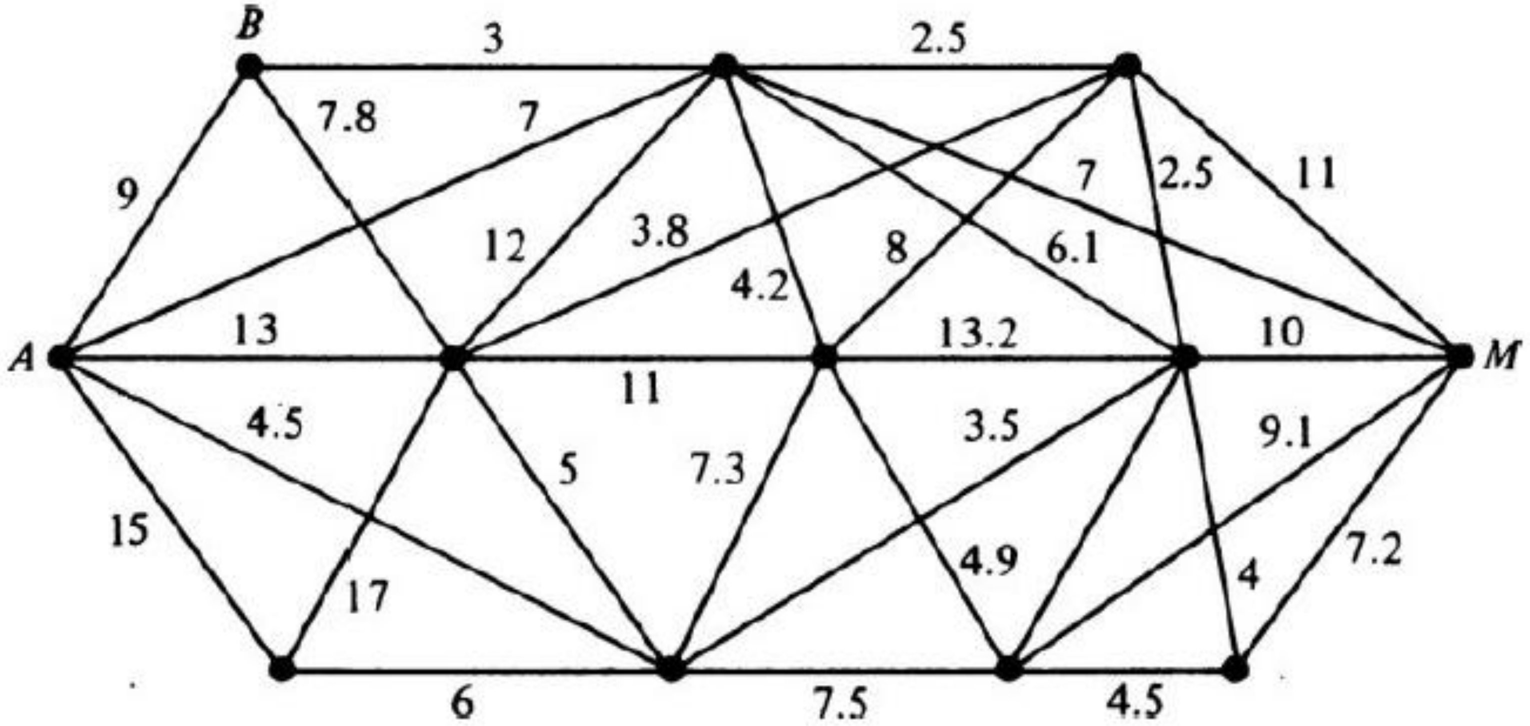
- **THEOREM 9:**
- The vertex connectivity of any graph G can never exceed the edge connectivity of G.
- ***Proof:***
- Let α denote the edge connectivity of G. Therefore, there exists a cut-set S in G with α edges.
- Let S partition the vertices of G into subsets $V1$ and $V2$.
- By removing at most α vertices from $V1$ (or $V2$) 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.

- If G is a simple graph, then $K(G) \leq K'(G) \leq \delta(G)$
- Every cut-set in a non-separable graph with more than two vertices contains at least two edges.
- **THEOREM 10:** The maximum vertex connectivity one can achieve with a graph *G* of *n* vertices and *e* edges ($e \geq n - 1$) is the integral part of the number $2e/n$; that is, $\lfloor 2e/n \rfloor$.
- ***Proof:***
  - Every edge in *G* contributes two degrees. The total ($2e$ degrees) is divided among *n* vertices. Therefore, there must be at least one vertex in *G* whose degree is equal to or less than the number $2e/n$. The vertex connectivity of *G* cannot exceed this number, in light of Theorems 8 and 9.

- **THEOREM 11:** A connected graph *G* is *k*-connected if and only if every pair of vertices in *G* is joined by *k* or more paths that do not intersect, and at least one pair of vertices is joined by exactly *k* non-intersecting paths.
- **THEOREM 12:** The edge connectivity of a graph *G* is *k*: if and only if every pair of vertices in *G* is joined by *k* or more edge-disjoint paths and at least one pair of vertices is joined by exactly *k* edge-disjoint paths.
- A graph G is nonseparable if and only if any pair of vertices in G can be placed in a circuit.

Network Flows

- In a network of telephone lines, highways, railroads, pipelines of oil (or gas or water), and so on, it is important to know the maximum rate of flow that is possible from one station to another in the network.
- This type of network is represented by a weighted connected graph in which the vertices are the stations and the edges are lines through which the given commodity (oil, gas, water, number of messages, number of cars, etc.) flows.
- The weight, a real positive number, associated with each edge represents the capacity of the line, that is, the maximum amount of flow possible per unit of time.



- The graph represents a flow network consisting of 12 stations and 31 lines. The capacity of each of these lines is also indicated in the figure.

- It is assumed that at each intermediate vertex the total rate of commodity entering is equal to the rate leaving. In other words, there is no accumulation or generation of the commodity at any vertex along the way.
- The flow through a vertex is limited only by the capacities of the edges incident on it. In other words, the vertex itself can handle as much flow as allowed through the edges.
- The lines are lossless.
- **THEOREM 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 subgraph G − S (the subgraph 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.

Fundamental Circuits – Algorithm

- Each edge is tested to see if it forms a circuit with the tree constructed so far; but instead of taking the edges themselves in an arbitrary order, we select a vertex z and examine this vertex by looking at every edge incident on z.
- Vertex z, is the vertex added most recently to the partially formed tree.
- Let the vertices of the given connected graph G = (V, E) be labeled 1, 2, . . . , n, and the graph be given by its adjacency matrix X.
- Let T be the current set of vertices in the partially formed tree, and let W be the set of vertices that are yet to be examined (i.e., those vertices, in T as well as not in T, which have one or more unexamined edges incident on them).
- Initially, T = ∅ and W = V, the entire set of vertices.
- We start the algorithm by setting T = 1, the first vertex, and W = V. Vertex 1 will be regarded as the root of the tree to be formed

1. If $T \cap W = \emptyset$, then the algorithm is terminated.
2. If $T \cap W \neq \emptyset$, choose a vertex z in $T \cap W$.
3. Examine z by considering every edge incident on z. If there is no such edge left, remove z from W, and go to step 1.
4. If there is such an edge (z, p), test if vertex p is in T.
5. If $p \in T$, find the fundamental circuit consisting of edge (z, p) together with the unique path from z to p in the tree (formed so far). Delete edge (z, p) from the graph, and go to step 3.
6. If $p \notin T$, add edge (z, p) to the tree and vertex p to set T. Delete edge (z, p) from the graph, and go to step 3.