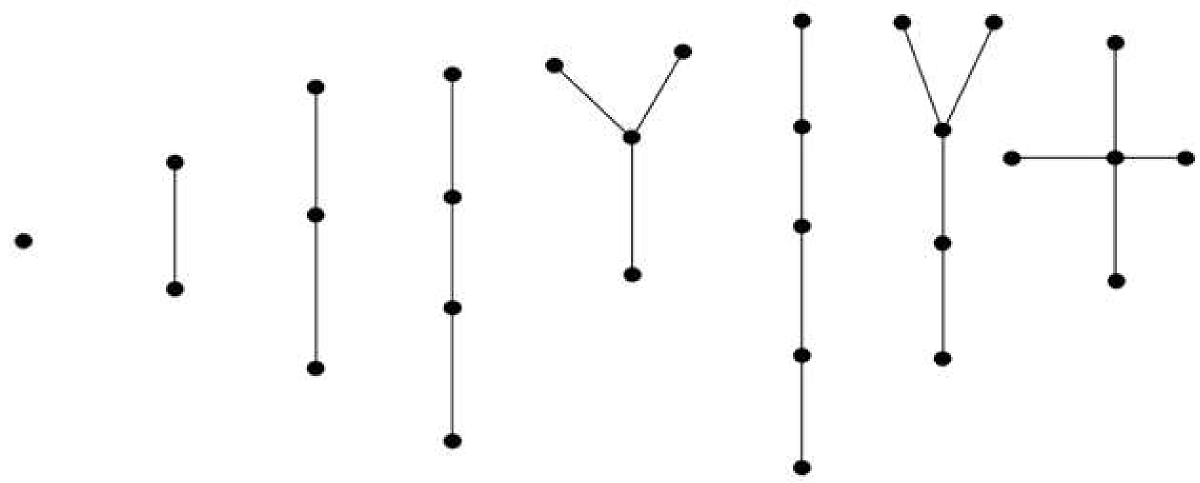
Tree and Forest

1. What is Tree and Forest?

Tree

- In graph theory, a **tree** is an **undirected, connected and acyclic graph**. In other words, a connected graph that does not contain even a single cycle is called a tree.
- A tree represents hierarchical structure in a graphical form.
- The elements of trees are called their nodes and the edges of the tree are called branches.
- A tree with n vertices has (n-1) edges.
- Trees provide many useful applications as simple as a family tree to as complex as trees in data structures of computer science.
- A **leaf** in a tree is a vertex of degree 1 or any vertex having no children is called a leaf.

Example

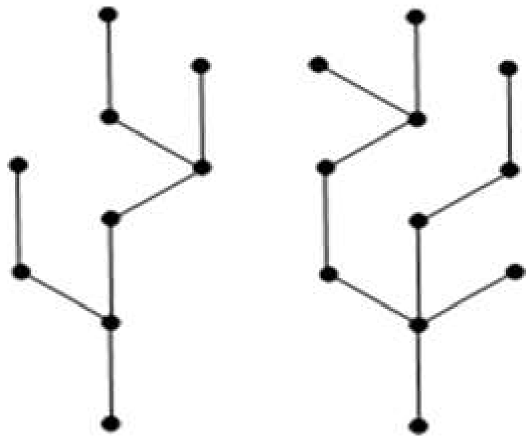


In the above example, all are trees with fewer than 6 vertices.

Forest

In graph theory, a **forest** is **an undirected, disconnected, acyclic graph**. In other words, a disjoint collection of trees is known as forest. Each component of a forest is tree.

Example



The above graph looks like a two sub-graphs but it is a single disconnected graph. There are no cycles in the above graph. Therefore it is a forest.

---

2. Properties of Trees

1. Every tree which has at least two vertices should have at least two leaves.
2. Trees have many characterizations:
   Let T be a graph with n vertices, then the following statements are equivalent:
   - T is a tree.
   - T contains no cycles and has n-1 edges.
   - T is connected and has (n -1) edge.
   - T is connected graph, and every edge is a cut-edge.
   - Any two vertices of graph T are connected by exactly one path.
   - T contains no cycles, and for any new edge e, the graph T+ e has exactly one cycle.
3. Every edge of a tree is cut -edge.
4. Adding one edge to a tree defines exactly one cycle.
5. Every connected graph contains a spanning tree.
6. Every tree has at least two vertices of degree two.

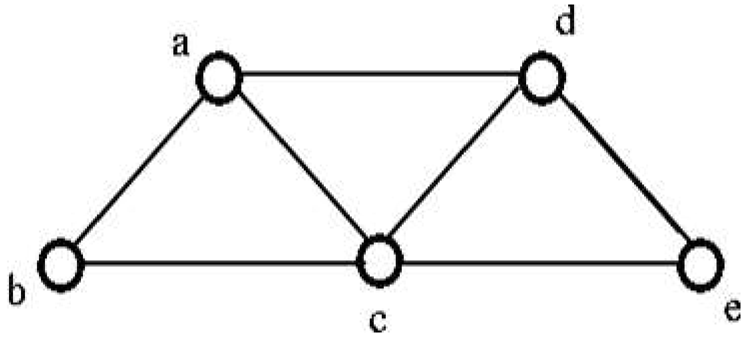
---

3. Spanning Tree

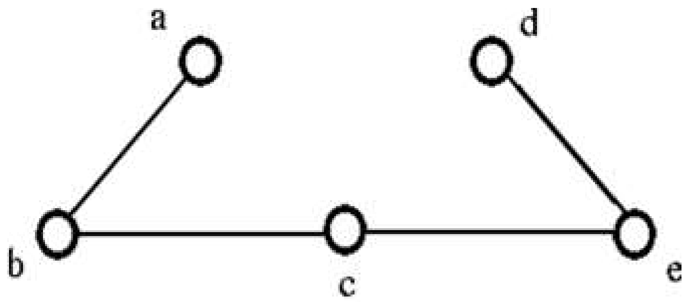
A **spanning tree** in a connected graph G is a sub-graph H of G that includes all the vertices of G and is also a tree.

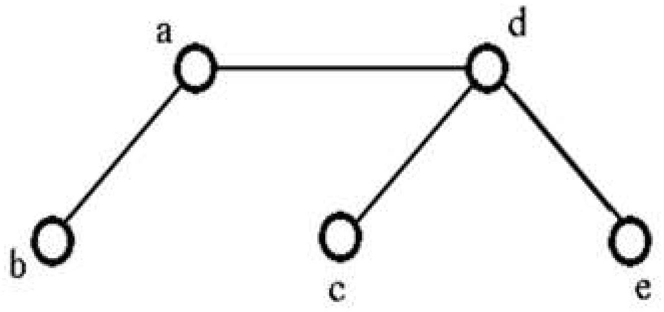
Example

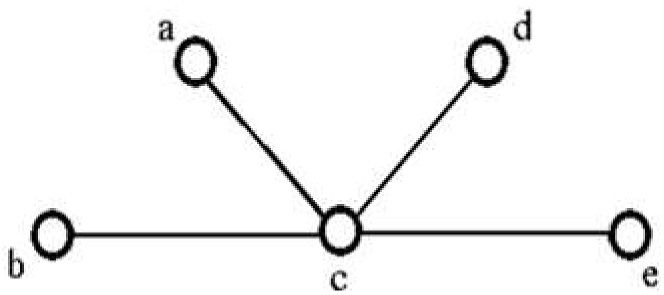
Consider the following graph G:



From the above graph G we can implement following three spanning trees H:







Methods to find the spanning Tree

We can find the spanning tree systematically by using either of two methods:

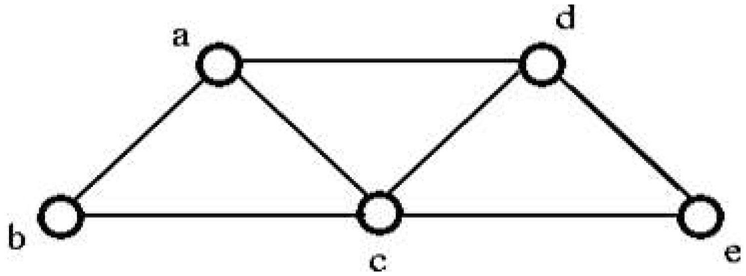
1. Cutting- down Method
2. Building- up Method

1. Cutting- down method

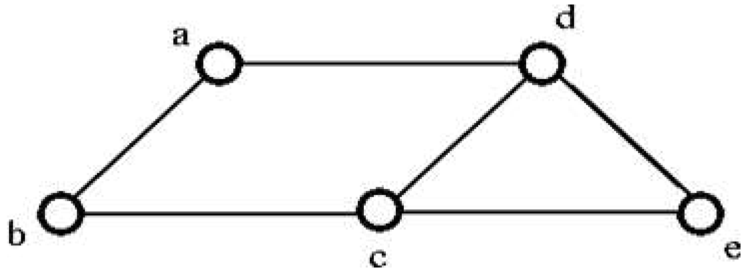
- Start choosing any cycle in Graph G.
- Remove one of cycle's edges.
- Repeat this process until there are no cycles left.

Example

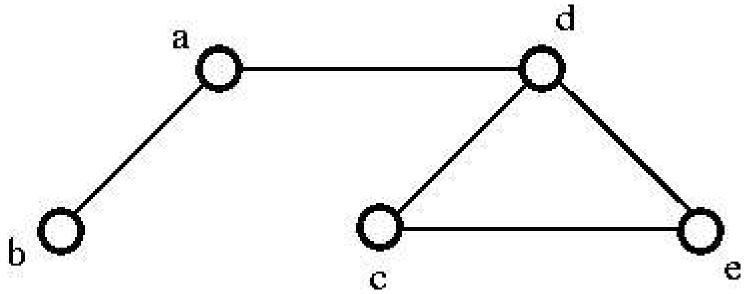
- Consider a graph G,



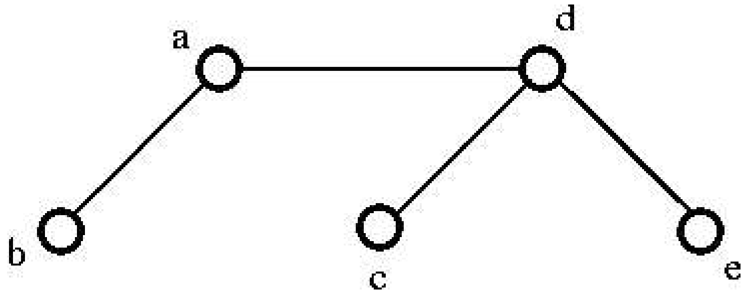
- If we remove the edge ac which destroy the cycle a-d-c-a in the above graph then we get the following graph:



- Remove the edge cb, which destroy the cycle a-d-c-b-a from the above graph then we get the following graph:



- If we remove the edge ec, which destroy the cycle d-e-c-d from the above graph then we get the following spanning tree:

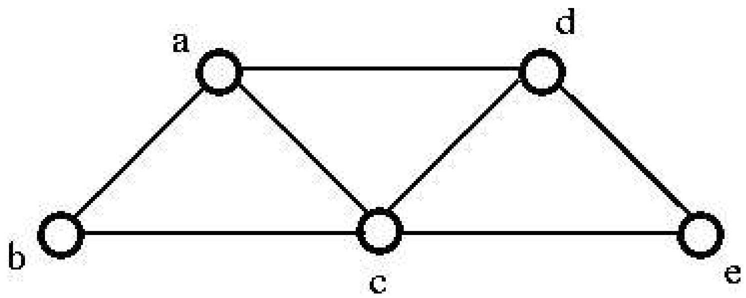


2. Building - up method

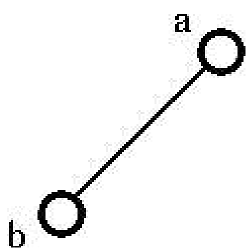
- Select edges of graph G one at a time. In such a way that there are no cycles are created.
- Repeat this process until all the vertices are included.

Example

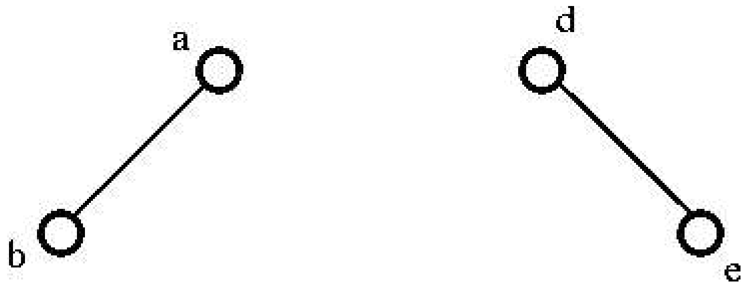
Consider the following graph G,



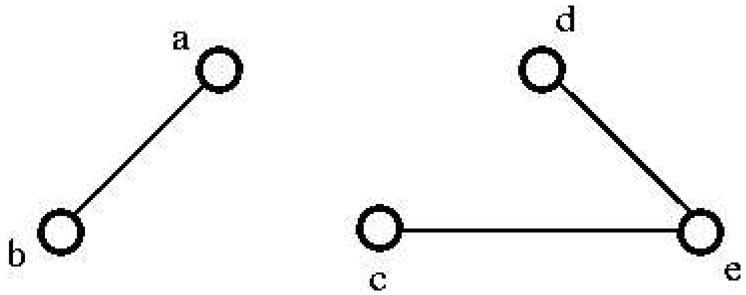
- Choose the edge **ab**,



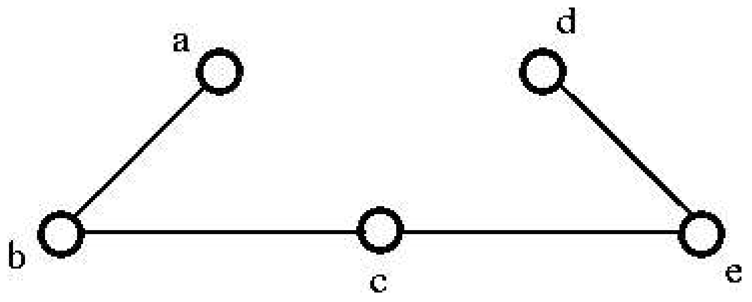
- Choose the edge **de**,



- After that , choose the edge **ec**,



- Next, choose the edge **cb**, then finally we get the following spanning tree:



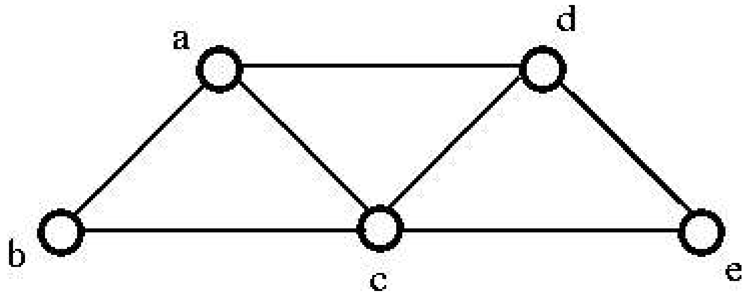
Circuit Rank

The number of edges we need to delete from G in order to get a spanning tree.

Spanning tree G = m- (n-1), which is called the **circuit rank** of G.

1. Where, m = No. of edges.
2. n = No. of vertices.

Example



In the above graph, edges m = 7 and vertices n = 5

Then the circuit rank is,

1. G = m - (n - 1)
2. = 7 - (5 - 1)
3. = 3

Graph Theory - Connectivity

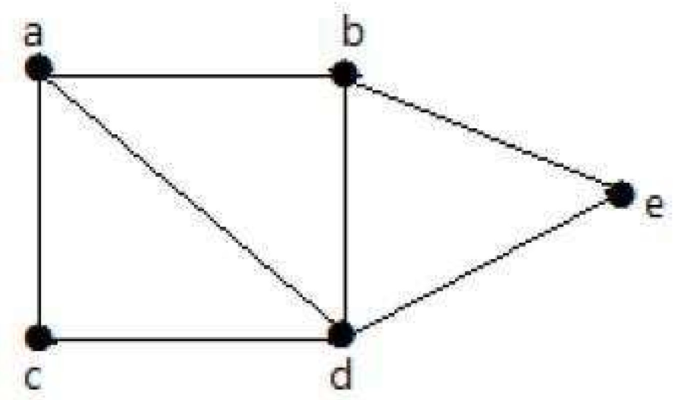
Whether it is possible to traverse a graph from one vertex to another is determined by how a graph is connected. Connectivity is a basic concept in Graph Theory. Connectivity defines whether a graph is connected or disconnected. It has subtopics based on edge and vertex, known as edge connectivity and vertex connectivity. Let us discuss them in detail.

Connectivity

A graph is said to be **connected if there is a path between every pair of vertex**. From every vertex to any other vertex, there should be some path to traverse. That is called the connectivity of a graph. A graph with multiple disconnected vertices and edges is said to be disconnected.

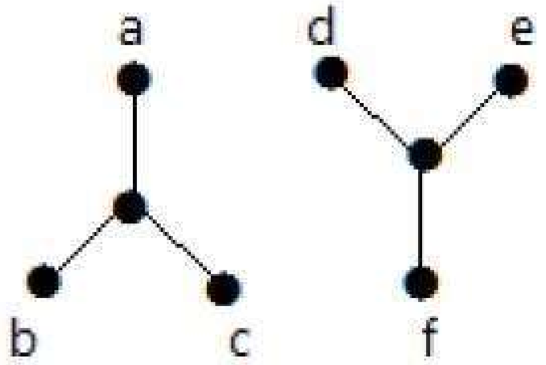
Example 1

In the following graph, it is possible to travel from one vertex to any other vertex. For example, one can traverse from vertex 'a' to vertex 'e' using the path 'a-b-e'.



Example 2

In the following example, traversing from vertex 'a' to vertex 'f' is not possible because there is no path between them directly or indirectly. Hence it is a disconnected graph.



Cut Vertex

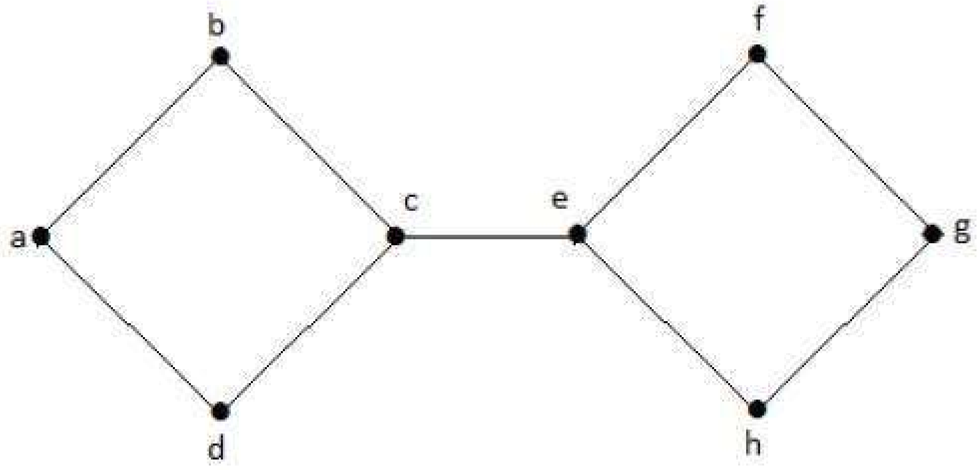
Let ‘G’ be a connected graph. A vertex V ∈ G is called a cut vertex of ‘G’, if ‘G-V’ (Delete ‘V’ from ‘G’) results in a disconnected graph. Removing a cut vertex from a graph breaks it in to two or more graphs.

Note − Removing a cut vertex may render a graph disconnected.

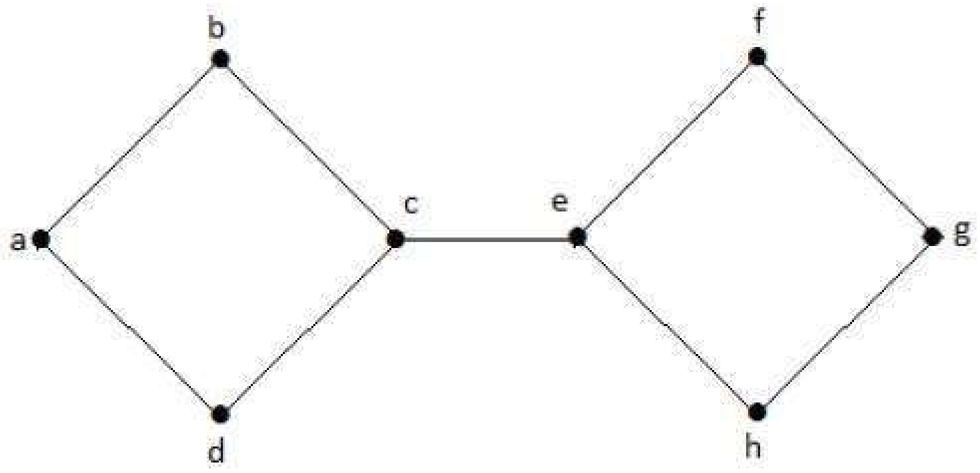
A connected graph ‘G’ may have at most (n–2) cut vertices.

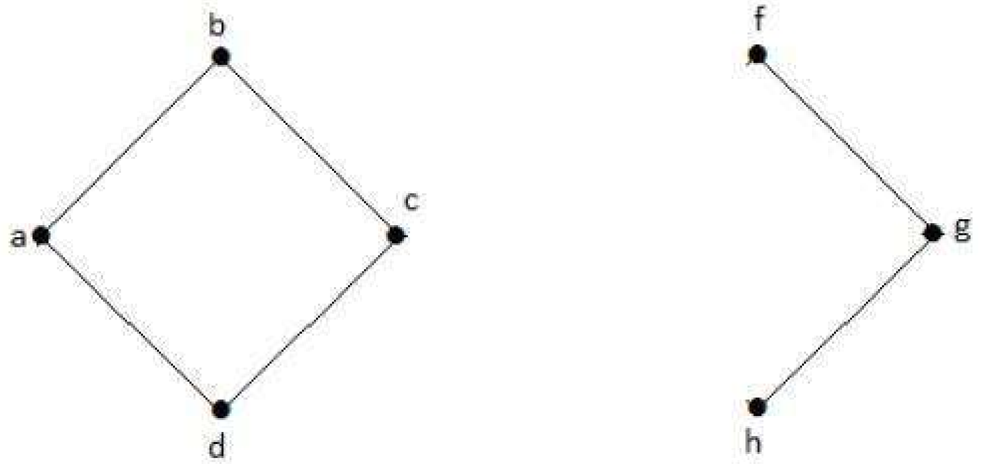
Example

In the following graph, vertices ‘e’ and ‘c’ are the cut vertices.



By removing ‘e’ or ‘c’, the graph will become a disconnected graph.





Without ‘g’, there is no path between vertex ‘c’ and vertex ‘h’ and many other. Hence it is a disconnected graph with cut vertex as ‘e’. Similarly, ‘c’ is also a cut vertex for the above graph.

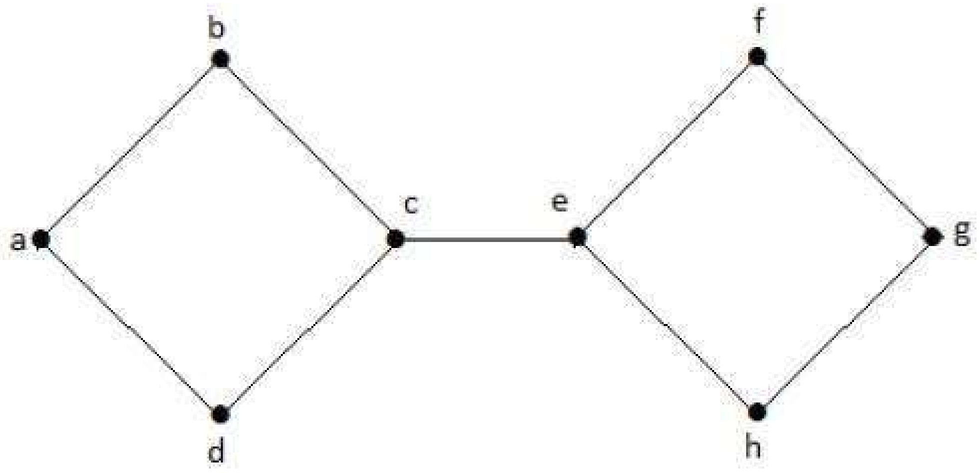
Cut Edge (Bridge)

Let ‘G’ be a connected graph. An edge ‘e’ ∈ G is called a cut edge if ‘G-e’ results in a disconnected graph.

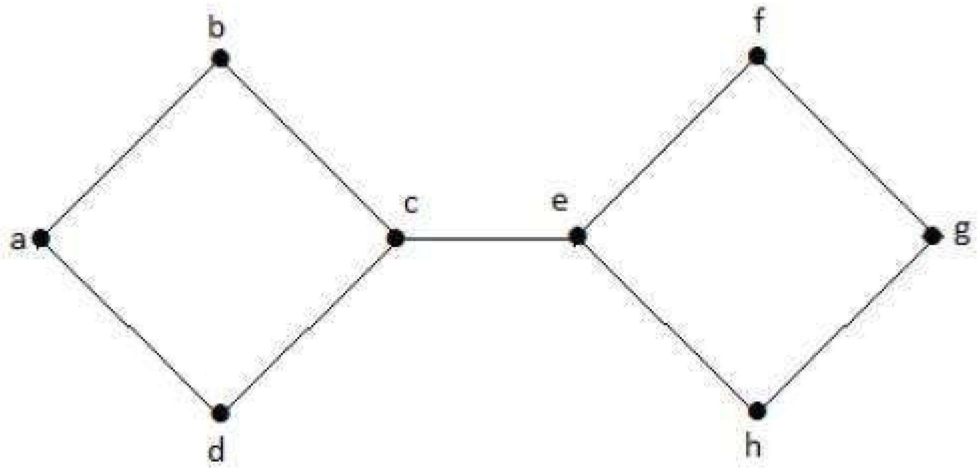
If removing an edge in a graph results in to two or more graphs, then that edge is called a Cut Edge.

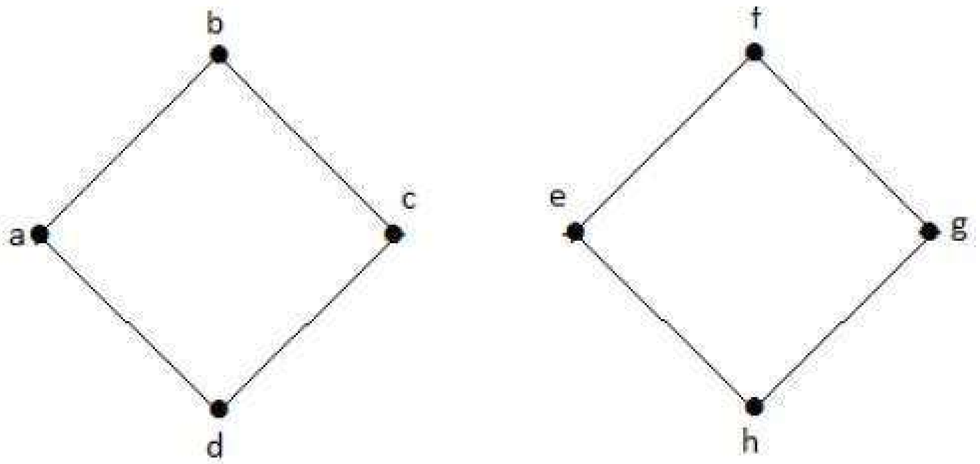
Example

In the following graph, the cut edge is [(c, e)].



By removing the edge (c, e) from the graph, it becomes a disconnected graph.





In the above graph, removing the edge (c, e) breaks the graph into two which is nothing but a disconnected graph. Hence, the edge (c, e) is a cut edge of the graph.

Note − Let ‘G’ be a connected graph with ‘n’ vertices, then

- a cut edge e ∈ G if and only if the edge ‘e’ is not a part of any cycle in G.

- the maximum number of cut edges possible is ‘n-1’.
- whenever cut edges exist, cut vertices also exist because at least one vertex of a cut edge is a cut vertex.
- if a cut vertex exists, then a cut edge may or may not exist.

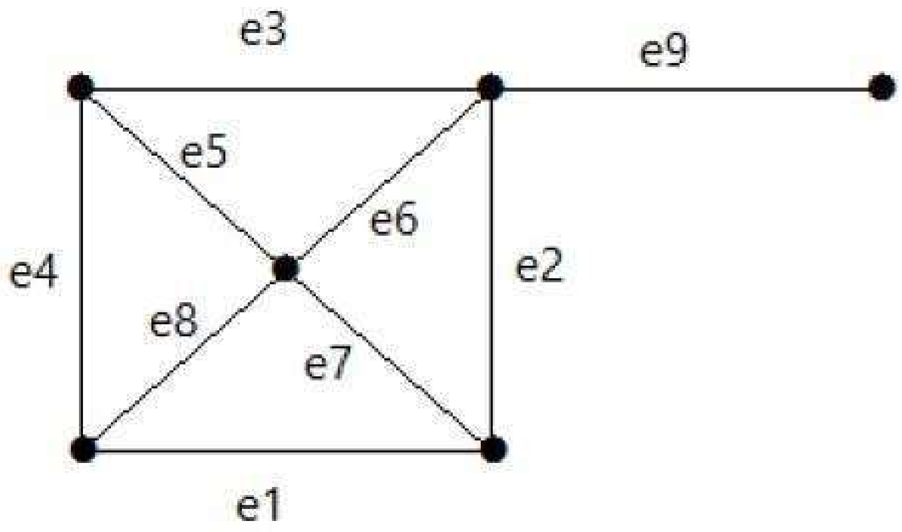
Cut Set of a Graph

Let ‘G’= (V, E) be a connected graph. A subset E’ of E is called a cut set of G if deletion of all the edges of E’ from G makes G disconnect.

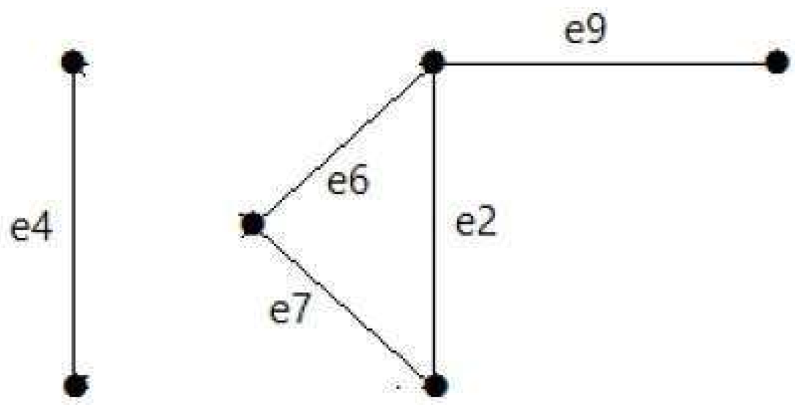
If deleting a certain number of edges from a graph makes it disconnected, then those deleted edges are called the cut set of the graph.

Example

Take a look at the following graph. Its cut set is E1 = {e1, e3, e5, e8}.



After removing the cut set E1 from the graph, it would appear as follows −



Similarly, there are other cut sets that can disconnect the graph −

- E3 = {e9} – Smallest cut set of the graph.
- E4 = {e3, e4, e5}

Edge Connectivity

Let ‘G’ be a connected graph. The minimum number of edges whose removal makes ‘G’ disconnected is called edge connectivity of G.

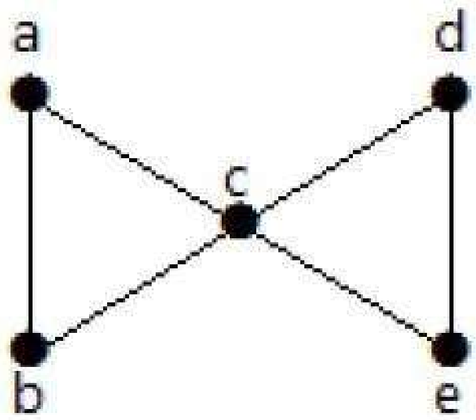
Notation − λ(G)

In other words, the **number of edges in a smallest cut set of G** is called the edge connectivity of G.

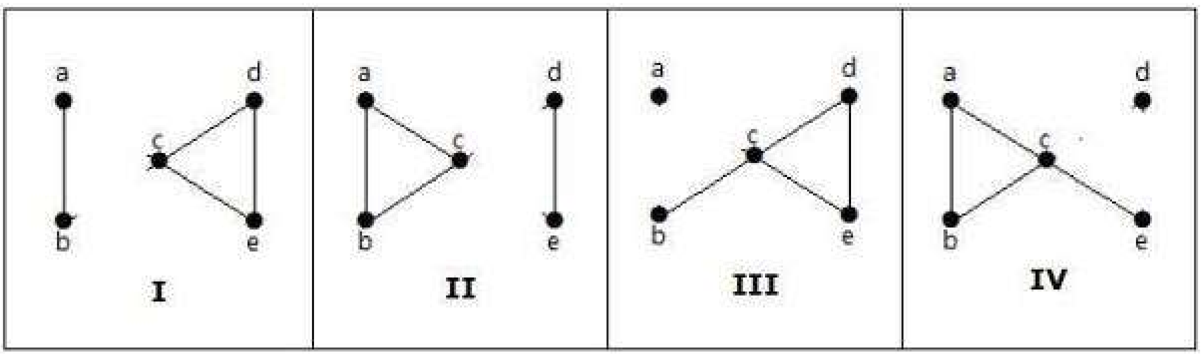
If ‘G’ has a cut edge, then λ(G) is 1. (edge connectivity of G.)

Example

Take a look at the following graph. By removing two minimum edges, the connected graph becomes disconnected. Hence, its edge connectivity (λ(G)) is 2.



Here are the four ways to disconnect the graph by removing two edges −



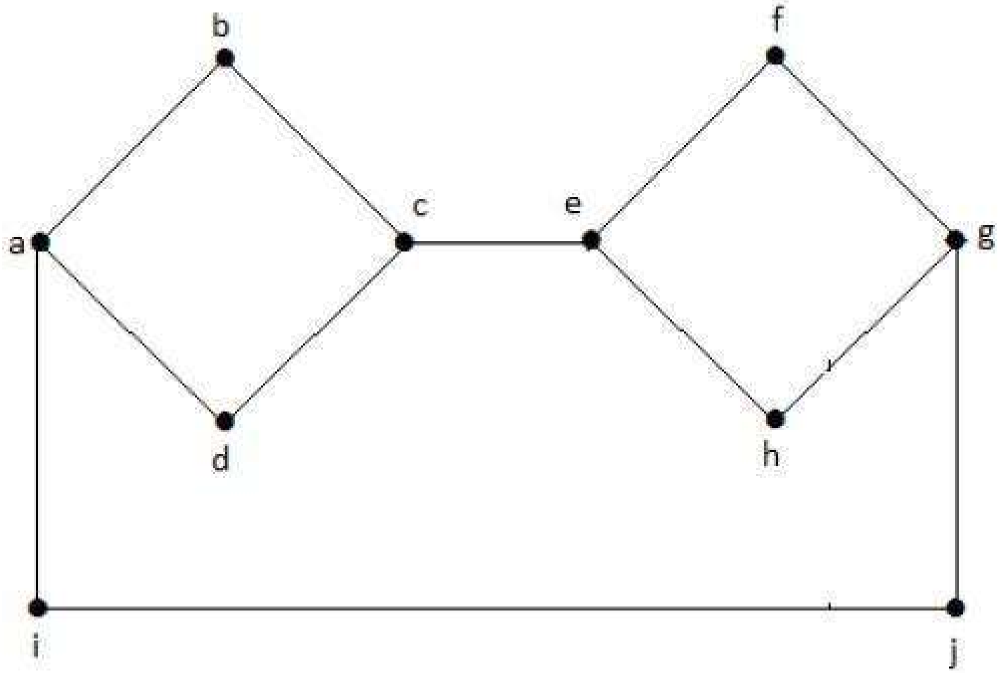
Vertex Connectivity

Let 'G' be a connected graph. The minimum number of vertices whose removal makes 'G' either disconnected or reduces 'G' in to a trivial graph is called its vertex connectivity.

Notation − K(G)

Example

In the above graph, removing the vertices 'e' and 'i' makes the graph disconnected.



If G has a cut vertex, then K(G) = 1.