Discrete Mathematics

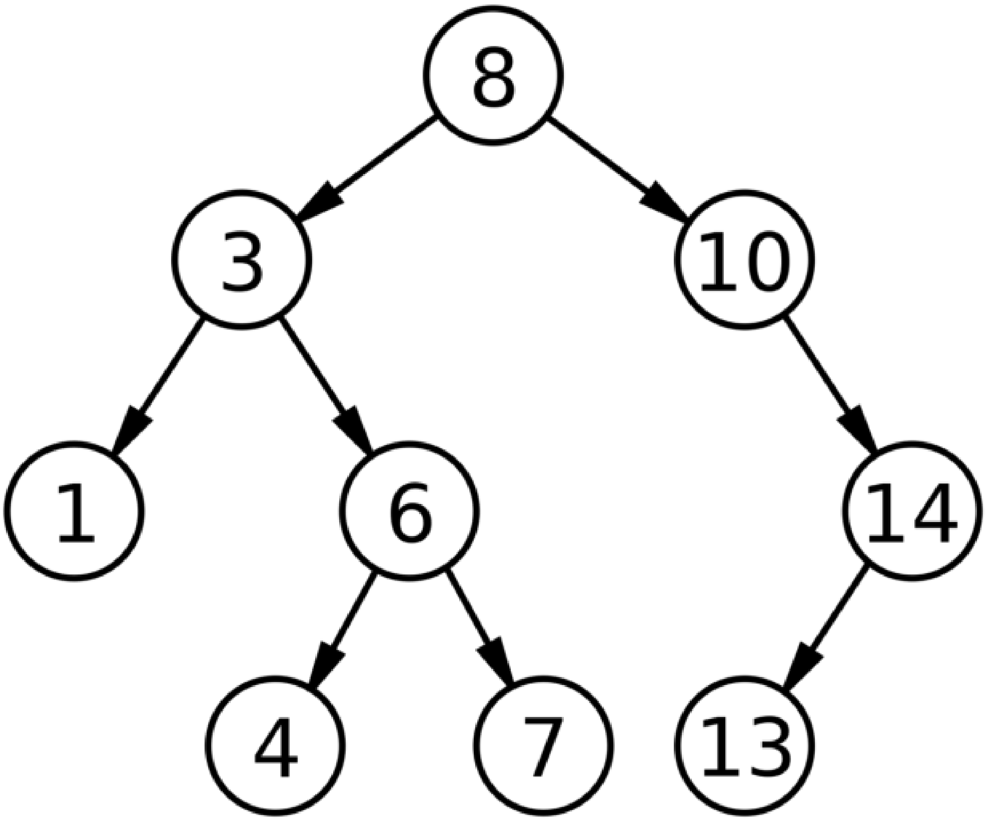
Scribed Notes 14 (13th October)

What is a binary tree?

A binary tree is a special type of tree in which every node or vertex has either no child node or one child node or two child nodes. A binary tree is an important class of a tree data structure in which a node can have at most two children.

Child node in a binary tree on the left is termed as 'left child node' and node in the right is termed as the 'right child node.'

In the figure mentioned below, the root node 8 has two children 3 and 10; then this two child node again acts as a parent node for 1 and 6 for left parent node 3 and 14 for right parent node 10. Similarly, 6 and 14 has a child node.



A binary tree may also be defined as follows:

• A binary tree is either an empty tree

• Or a binary tree consists of a node called the root node, a left subtree and a right subtree, both of which will act as a binary tree once again

Binary Tree: Common Terminologies

- **Root:** Topmost node in a tree.
- **Parent:** Every node (excluding a root) in a tree is connected by a directed edge from exactly one other node. This node is called a parent.
- **Child:** A node directly connected to another node when moving away from the root.
- **Leaf/External node:** Node with no children.
- **Internal node**: Node with at least one children.
- **Depth of a node:** Number of edges from root to the node.
- **Height of a node:** Number of edges from the node to the deepest leaf. Height of the tree is the height of the root.

Types of Binary Tree

Rooted binary tree: It has a root node and every node has at most two children.

Full binary tree: It is a tree in which every node in the tree has either 0 or 2 children.

Perfect binary tree: It is a binary tree in which all interior nodes have two children and all leaves have the same depth or same level.

Complete binary tree: It is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.

Balanced binary tree: A binary tree is height balanced if it satisfies the following constraints:

1. The left and right subtrees' heights differ by at most one, AND

2. The left subtree is balanced, AND

3. The right subtree is balanced.

Degenerate tree: It is a tree is where each parent node has only one child node. It behaves like a linked list.

Application of Binary Tree

Binary trees are used to represent a nonlinear data structure. There are various forms of Binary trees. Binary trees play a vital role in a software application. One of the most important applications of the Binary tree is in the searching algorithm.

A general tree is defined as a nonempty finite set T of elements called nodes such that:

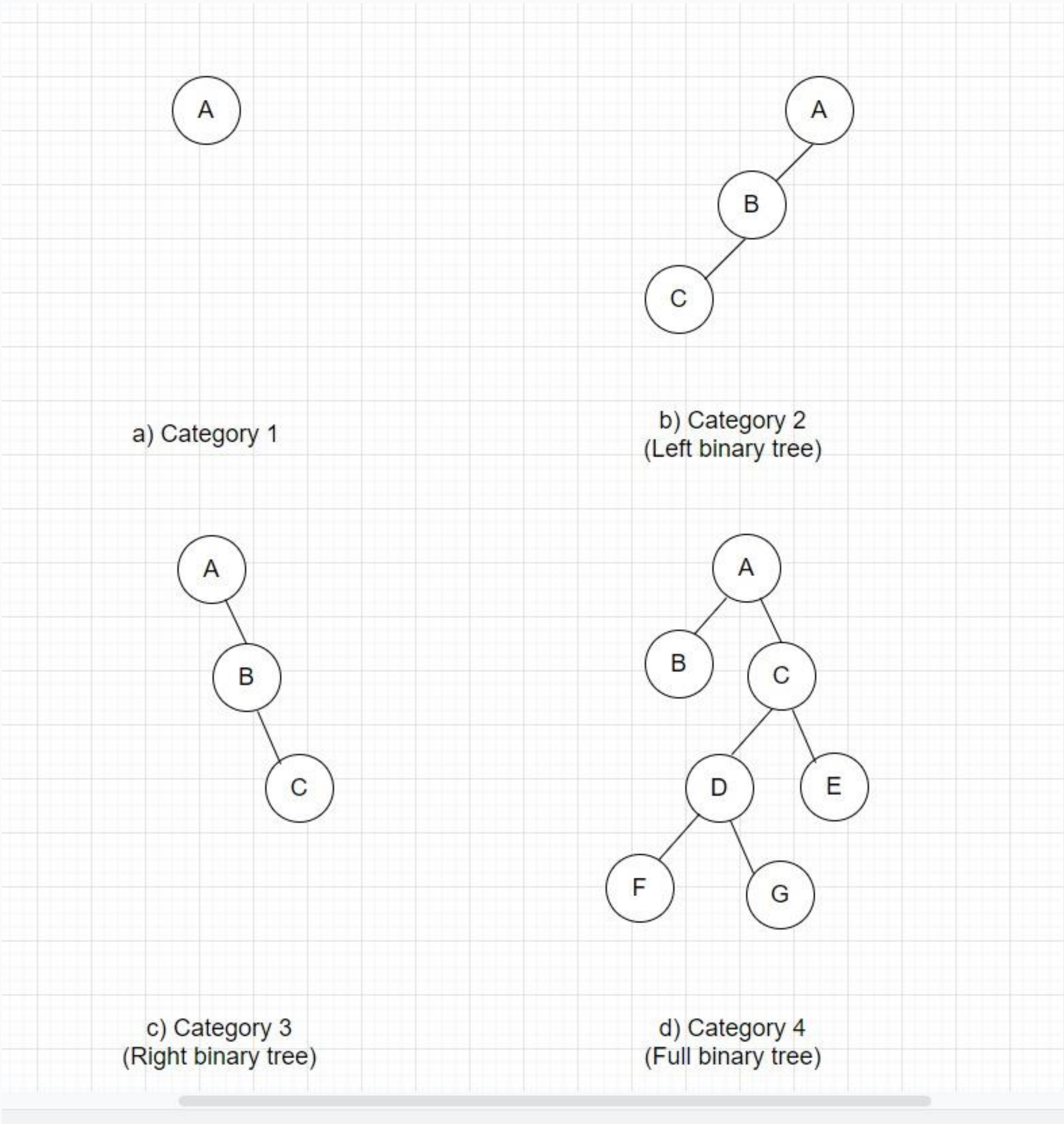
- The tree contains the root element

• The remaining elements of the tree form an ordered collection of zeros and more disjoint trees T1, T2, T3, T4 …. Tn which are called subtrees.

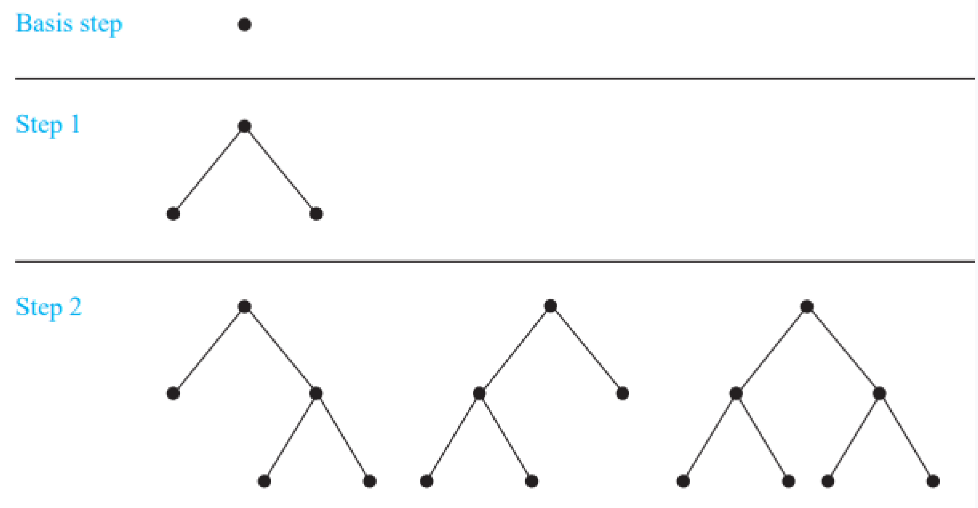
Full Binary Tree

A full Binary tree is a special type of binary tree in which every parent node/internal node has either two or no children. It is also known as **a proper binary tree**. In full binary tree the left subtree or the right subtree can not be empty.

As shown below category 1 and 4 only FBT.



The set of full binary trees can be defined recursively by these steps:

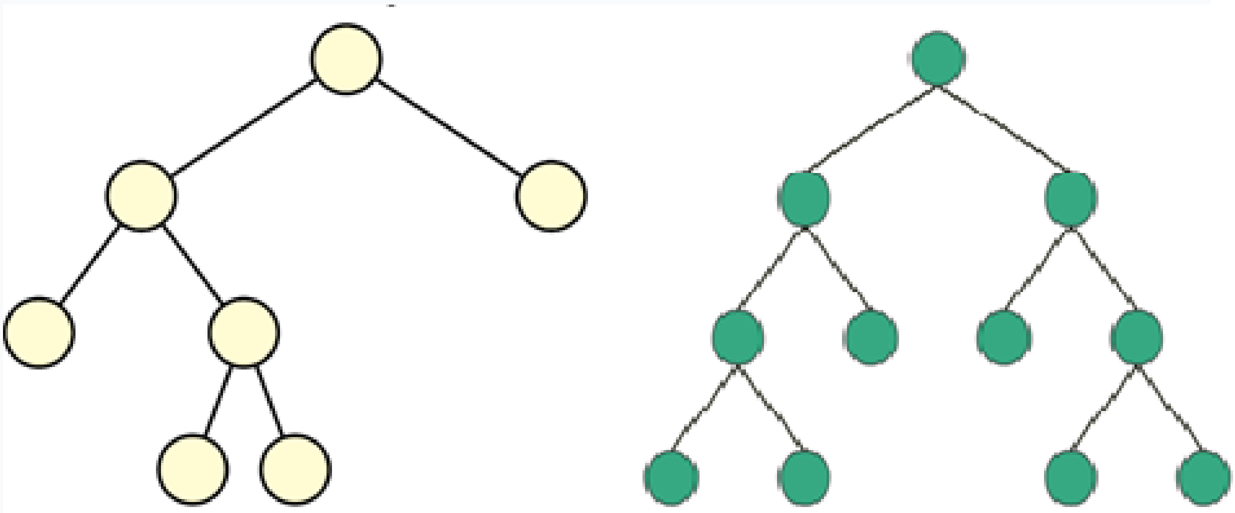


BASIS STEP: There is a full binary tree consisting only of a single vertex r.

RECURSIVE STEP: If T1 and T2 are disjoint full binary trees, there is a full binary tree, denoted by T1 · T2, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T1 and the right subtree T2.

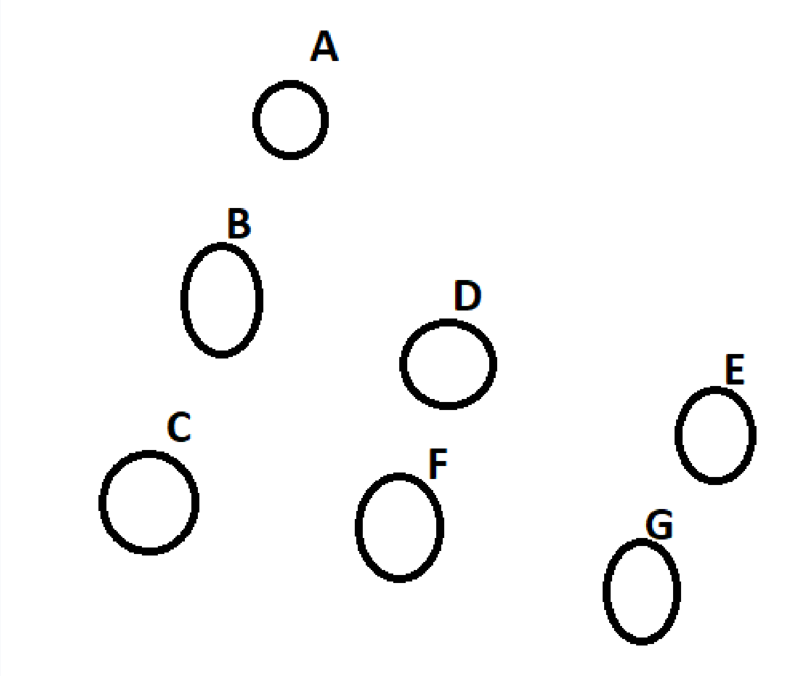
NOTE:- Every full binary tree has even number of links(edges). Every full binary tree having n nodes has exactly n-1 vertices.

Some examples of full binary tree.

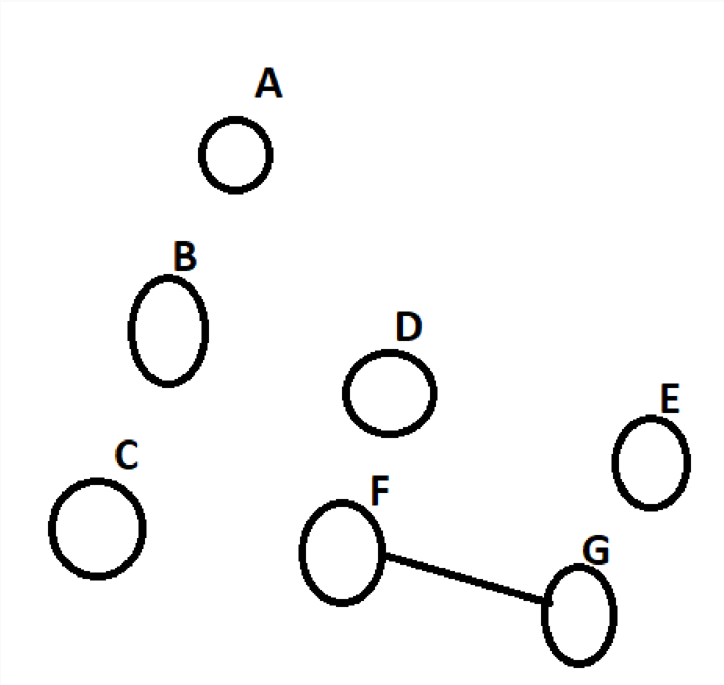


Reachability in terms of traverse edges

Over here there are 7 disjoint so there is no reachability in terms of traverse edges.



If we connect two vertices f and g as shown in figure below there is decrease by two vertices and generate one edge (FG) . so total component arise -2+1=1 .so now 5 component as A,B,C,D,E and (FG) as combined component. So , every time one edge is formed it reduce two vertices.



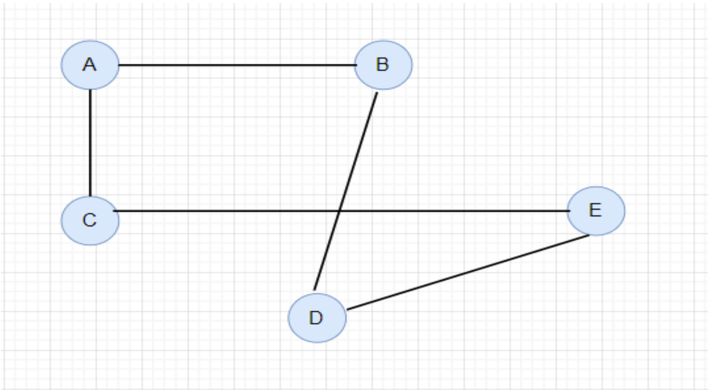
Every tree having N nodes has exactly N-1 edges.

First Theorem of Graph Theory

Adjacency Matrix

The **adjacency matrix**, also called the **connection matrix**, is a matrix containing rows and columns which is used to represent a simple labelled graph, with 0 or 1 in the position of (V_i , V_j) according to the condition whether V_i and V_j are adjacent or not. It is a compact way to represent the finite graph containing n vertices of a **m x m** matrix M. Sometimes adjacency matrix is also called as **vertex matrix**.

Some Example of Adjacency matrix for an undirected graph:-



	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	0
C	1	0	0	0	1
D	0	1	0	0	1
E	0	0	1	1	0

Degree of vertex in a graph:- The degree of a vertex in an undirected graph is **the number of edges associated with it**.

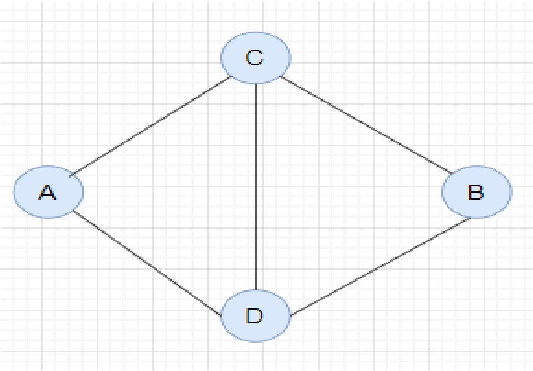
Deg(A)=2 (B,C)

Deg(B)=2 (A,D)

Deg(C)=2 (A,E)

Deg(D)=2 (B,E)

Deg(E)=2 (C,D)



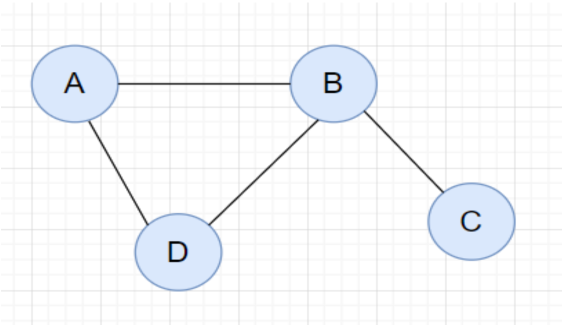
	A	B	C	D
A	0	0	1	1
B	0	0	1	1
C	1	1	0	1
D	1	1	1	0

Deg(A)=2 (C,D)

Deg(B)=2 (C,D)

Deg(C)=3 (A,B,D)

Deg(D)=3 (A,B,C)



	A	B	C	D
A	0	1	0	1
B	1	0	1	1
C	0	1	0	0
D	1	1	0	0

Deg(A)=2 (B,D)

Deg(B)=3 (A,C,D)

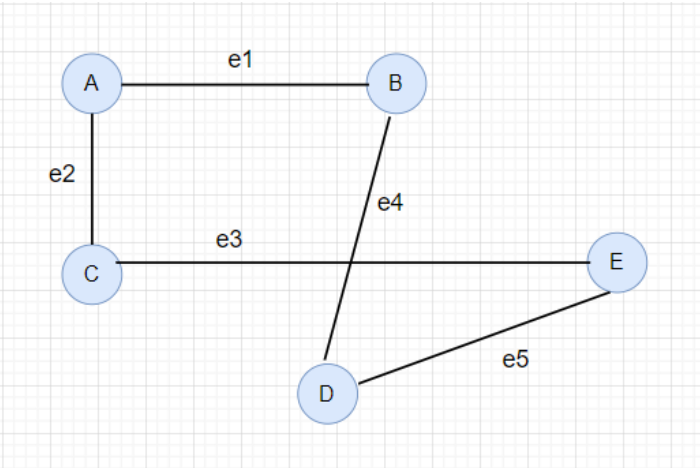
Deg(C)=1 (B)

Deg(D)=2 (A,B)

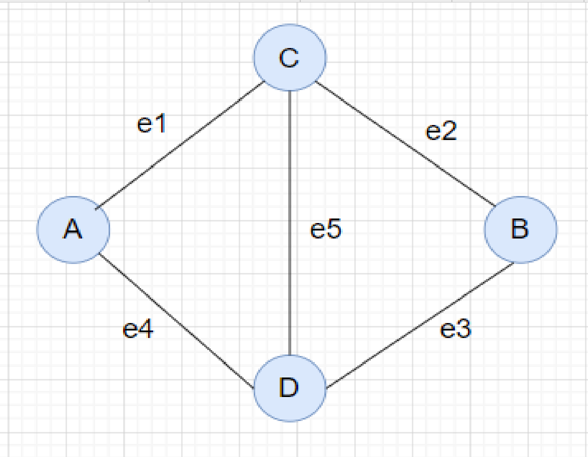
INCIDENCE MATRIX

Let G = (V, E) be a graph where V = {1, 2, . . . , n} and E = {e1, e2, . . . , em}. The incidence matrix of G is an n × m matrix B = (bik), where each row corresponds to a vertex and each column corresponds to an edge such that if ek is an edge between i and j, then all elements of column k are 0 except bik = bjk = 1

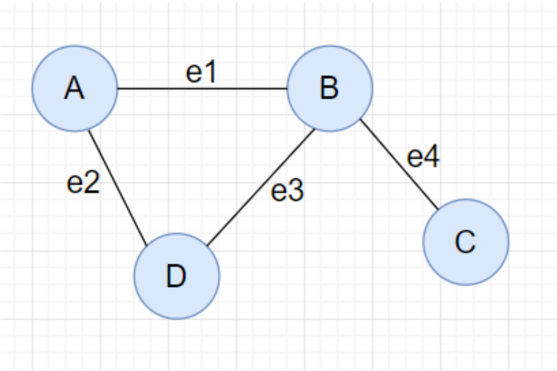
Some Example of Incidence matrix for an undirected graph



	e1	e2	e3	e4	e5
A	1	1	0	0	0
B	1	0	0	1	0
C	0	1	1	0	0
D	0	0	0	1	1
E	0	0	1	0	1



	e1	e2	e3	e4	e5
A	1	0	0	1	0
B	0	1	1	0	0
C	1	1	0	0	1
D	0	0	1	1	1



	e1	e2	e3	e4
A	1	1	0	0
B	1	0	1	1
C	0	0	0	1
D	0	1	1	0

Double Counting Matrix

Another theorem that is commonly proven with a double counting argument states that every undirected graph contains an even number of vertices of odd degree. That is, the number of vertices that have an odd number of incident edges must be even. In more colloquial terms, in a party of people some of whom shake hands, an even number of people must have shaken an odd number of other people's hands; for this reason, the result is known as the handshaking lemma.

To prove this by double counting, let ***d(v)*** be the degree of vertex ***v***. The number of vertex-edge incidences in the graph may be counted in two different ways: by summing the degrees of the vertices, or by counting two incidences for every edge. Therefore

$$\sum_v d(v) = 2e$$

where ***e*** is the number of edges. The sum of the degrees of the vertices is therefore an even number, which could not happen if an odd number of the vertices had odd degree.