QA: Discrete Mathematics

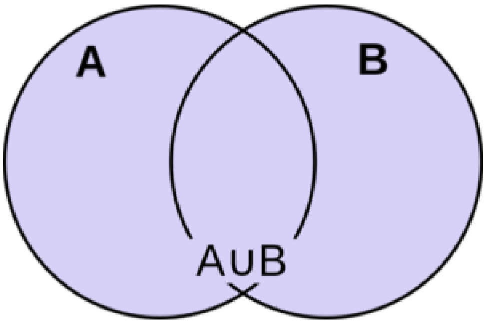
Created	@January 16, 2023 4:57 PM
Last Edited Time	@January 17, 2023 7:00 PM
By	**BORHAN<ASH2101008M>**
Email	

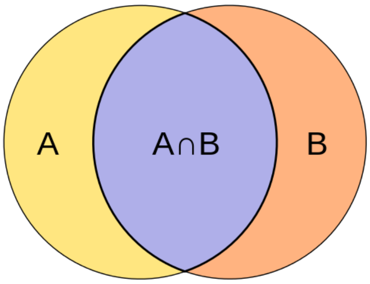
Sets

Union vs Intersection

UNION	INTERSECTION
The union of two sets is defined as the set of all elements that belong to either A or B, or possibly both.	The intersection of two sets is defined as the set of all elements that belong to both A and B at the same time.
The union of two sets is represented by ∪.	The intersection of two sets is represented by ∩.
It corresponds to the logical OR.	It corresponds to the logical AND.
It discards the duplicate values from the sets.	It is an associative operation which lists the common values from the sets.
Example: If A = {1, 2, 3} and B = {2, 3, 4}, then A∪B = {1, 2, 3, 4}.	Example: If A = {1, 2, 3} and B = {2, 3, 4}, then A∩B = {2, 3}.

DifferenceBetween.net





Disjoint Set

(https://byjus.com/maths/what-is-disjoint-set/#:~:text=What is a Disjoint Set,3 as a common element.)

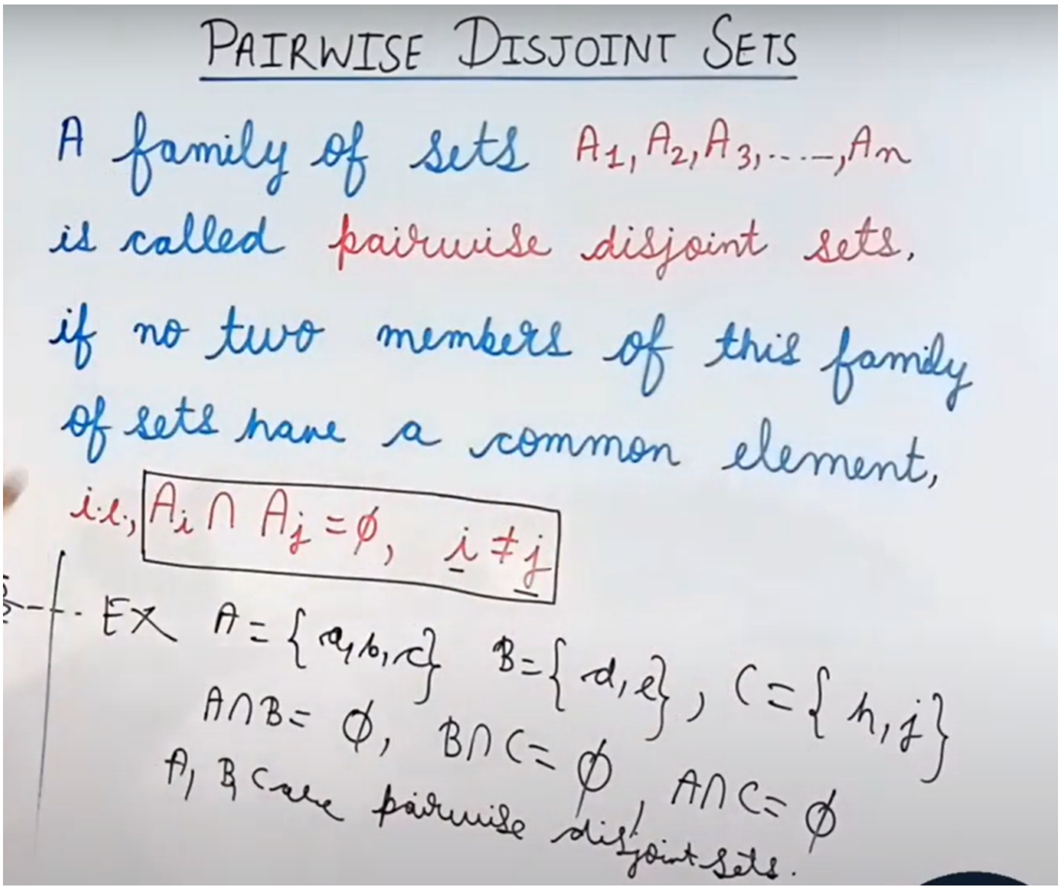
Definition:

A pair of sets which does not have any common element are called **disjoint sets.** For example, set A={2,3} and set B={4,5} are disjoint sets. But set C={3,4,5} and {3,6,7} are not disjoint as both the sets C and D are having 3 as a common element.

Another definition:

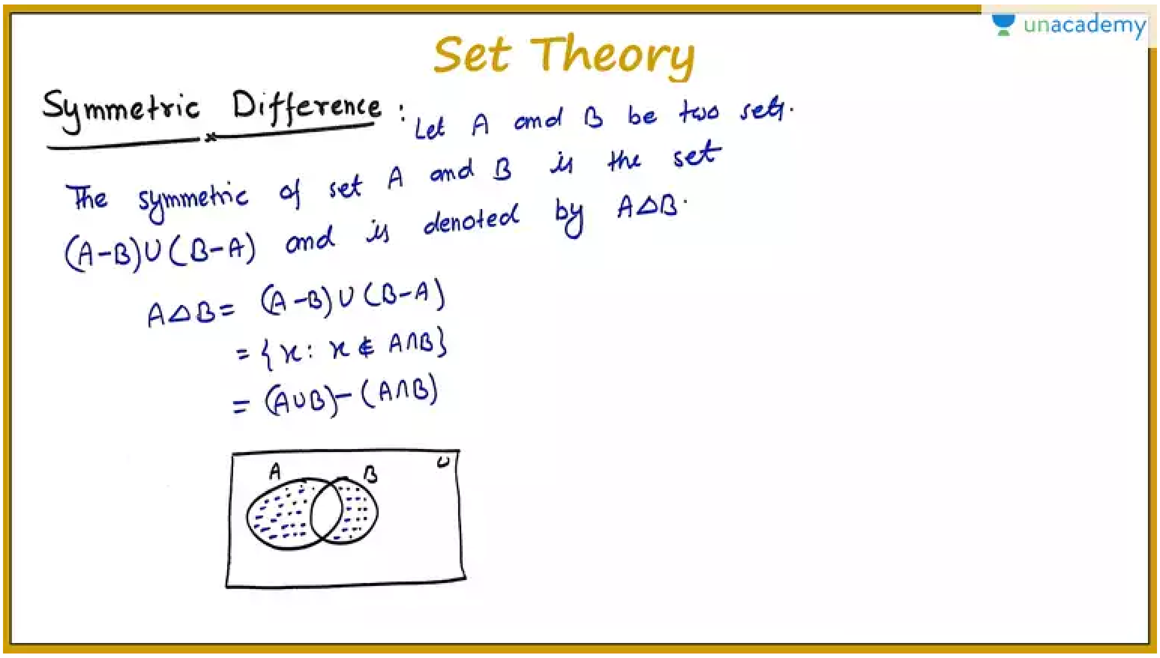
When the intersection of two sets is a null or empty set, then they are called disjoint sets. Hence, if A and B are two disjoint sets, then;

$A \cap B = \phi$



Symmetric Difference :

- Also denoted by **(A⊕B)**



Bit String:

$U = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

Bit representation of U : 1111111111

$A = 2, 3, 10$

Bit representation of A : 0110000001

$B = 3, 6$

AUB in bit representation : 01110110000

From SSROY bhai:

Bit string:

Example:

$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$A = \{1, 3, 5, 7, 9\}$

$B = \{2, 4, 6, 8, 10\}$

$C = \{1, 2, 3, 4, 5\}$

Bit string of $A = ?$, $B = ?$, $C = ?$

[Set এ যে সংখ্যাগুলো থাকবে সে স্থানে 1 বসাতে হবে]

Solution:

Bit string of $A = 1010101010$

Bit string of $B = 0101010101$

Bit string of $C = 1111100000$

Example:

Bit string for the sets $\{1, 2, 3, 4, 5\}$ and $\{1, 3, 5, 7, 9\}$.

Find out union and intersection of these sets of $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Solution:

Let, $A = \{1, 2, 3, 4, 5\}$

$B = \{1, 3, 5, 7, 9\}$

$\vee \rightarrow$ or

$\wedge \rightarrow$ and

Bit string of $A = 1111100000$

Bit string of $B = 1010101010$

Functions

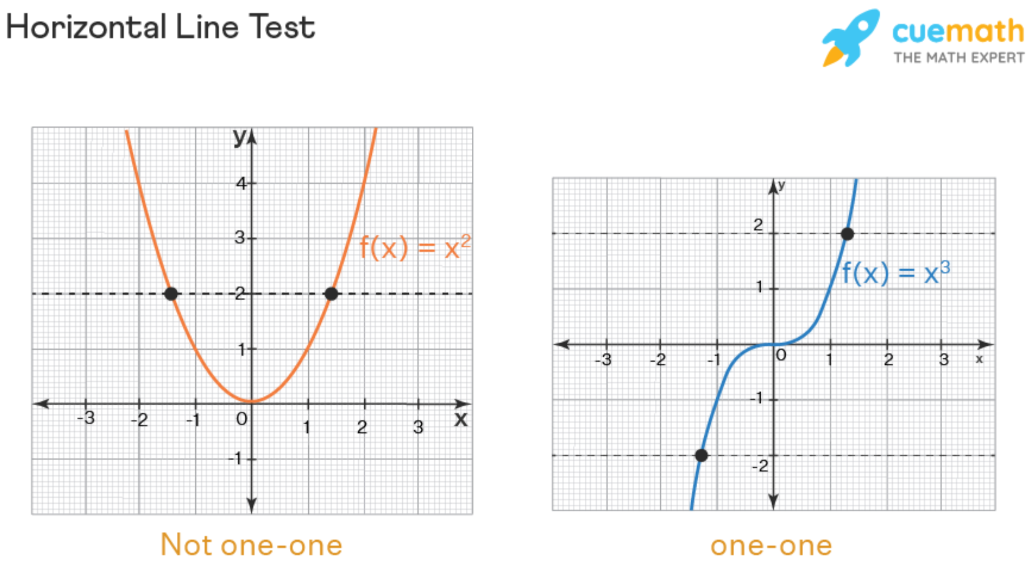
(https://www.geeksforgeeks.org/functions-properties-and-types-injective-surjective-bijective/)

One to one function(Injective):

A function is called one to one if for all elements a and b in A, if f(a) = f(b),then it must be the case that a = b.

- A function is one to one if it is either strictly increasing or strictly decreasing

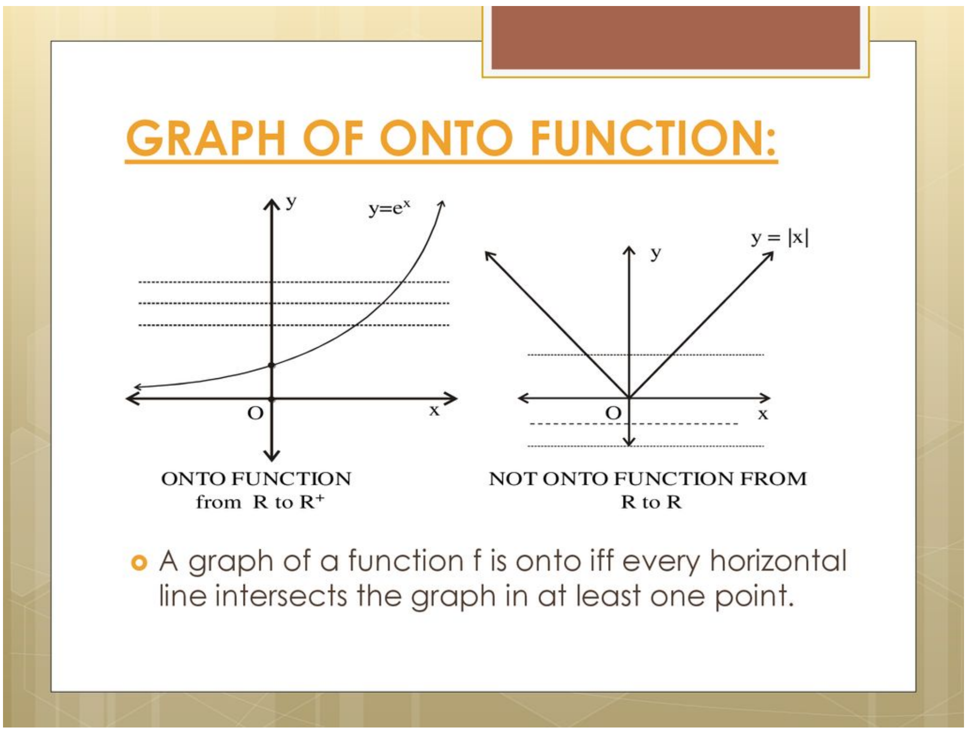
- one to one function never assigns the same value to two different domain elements.



Onto Function (surjective):

If every element b in B has a corresponding element a in A such that f(a) = b. It is not required that a is unique;

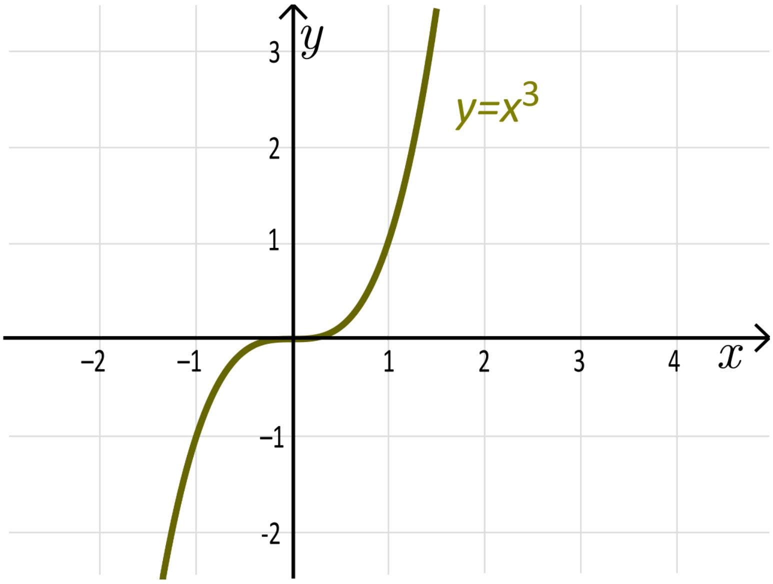
- For onto function, range and co-domain are equal



One to one correspondence function(Bijective/Invertible):

A function is Bijective function if it is both one to one and onto function.

- If a function f is not bijective, inverse function of f cannot be defined.



How to prove?

Step 1 of 1

Given function is $f(x) = 2x + 1$
$f(x_1) = 2x_1 + 1$
and, $f(x_2) = 2x_2 + 1$
Let $f(x_1)= f(x_2)$
$2x_1 + 1 = 2x_2 + 1$
$2x_1 = 2x_2$
$x_1 = x_2$
If $f(x_1) = f(x_2)$
$\Rightarrow x_1 = x_2$
The function is one-to-one.

Step 2 of 2

Now we need to proceed with the second check :
Let $y = 2x + 1$
$x = \frac{y-1}{2}$
$\Rightarrow \frac{y-1}{2}$ = x .. . . equation (i)
From equation (i) we can say each value of $\frac{y-1}{2}$ gives a unique value of ' x '. Therefore the function is onto.
As this function passes both the checks so that it's bijective from R to R.

Final answer

The function $f(x)$ = 2x + 1 is a bijective function from R to R .

Find Inverse Function: (Not Important)

Explanation

Given function is $f(x) = x^2$.
We have to calculate the value of $f^{-1}\{(1)\}$.

- First, calculate the inverse function. The process of to find f Let $y = f(x)$. Then calculate the value of x in term of y.
- Plug $y = 1$ in the inverse function.

Step 1 of 2

Given function is $f(x) = x^2$.
Let $f(x) = y$.
$\therefore y = x^2$
$x^2 = y$
$y = \pm\sqrt{x}$.
Therefore the inverse function of the function is $f^{-1}(y) = \pm\sqrt{y}$.

Step 2 of 2

Plug $y = 1$
$f^{-1}(1) = \pm\sqrt{1}$
$f^{-1}(1) = \pm 1$

Final answer

- Hence $\mathrm{f}^{-1}(1) = \{1, -1\}$

Ceiling and Flooring Function

Ceiling Function: The ceiling function of *x*, denoted by $\lceil x \rceil$ or ceil (*x*), is defined to be the least integer that is greater than or equal to *x*.

Flooring Function: Let *x* be a real number. The floor function of *x*, denoted by $\lfloor x \rfloor$ or floor (*x*), is defined to be the greatest integer that is less than or equal to *x*.

Ceil : (positive number) → integer part + 1

(negative number) → integer part

Floor : (positive number) → integer part

(negative number) → integer part - 1

Rounding Functions						
	-3.5	-2.9	-1.1	1.1	2.9	3.5
FLOOR	-4	-3	-2	1	2	3
CEIL	-3	-2	-1	2	3	4
ROUND	-4	-3	-1	1	3	4
INT	-3	-2	-1	1	2	3
INT + SIGN	-4	-3	-2	2	3	4

Relation:

Let A & B two sets and some elements of A be in certain correspondence with some elements of B then that correspondence defines a relation between elements of those two sets.

Properties of Relation: (***)

(https://www.youtube.com/watch?v=ItfxKIvq1LI&ab_channel=mathematicaATD)

1) Reflexive Relation: If A relation to R, if **for all value** x belongs to A and **(x,x)** belongs to R

- Irreflexive → not reflexive

2) Symmetric : (a, b) belongs to R and **(b,a)** belongs to R

3) Transitive : (x,y) belongs to R and **(y,z)** belongs to R and **(x,z)** belongs to R

From 15th Batch, Anwar Maruf bhai:

Reflexive : Relation is reflexive, If $(a,a) \in R$ for every $a \in A$.

Transitive: Relation is transitive, If $(a,b) \in R$ & $(b,c) \in R$, then $(a,c) \in R$.

Here,

$R_1 = \{(1,1), (1,2), (2,1), (2,2), (3,4), (4,1)\}$

$R_2 = \{(1,1), (2,2), (2,1)\}$

$R_3 = \{(1,1), (1,2), (1,4), (2,1), (2,2), (3,3), (4,1), (4,1), (4,4)\}$

$R_4 = \{(2,1), (3,1), (3,2), (4,1), (4,2), (4,3)\}$

$R_5 = \{(1,1), (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), (3,3), (3,4), (4,4)\}$

R_3 and R_5 are relations of reflexive.

R_2 and R_5 are relations of transitive.

!**Important:**

Note : → x divides y = y/x

→ x is divided/divisible by y = x/y

Composition of Relation

$* \; S \circ R \rightarrow$ zeta aage oita sheshe
zeita sheshe oita aage (aage gele, baghe khay)
have to check one by one like below

R = {(2, 2), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5), (5, 3)}
S = {(2, 3), (2, 5), (3, 4), (3, 5), (4, 2), (4, 3), (4, 5), (5, 2), (5, 5)}.

$S \circ R =$ (2, 3) (2, 4) (2, 2) (3, 3)

What is the composite of the relations R and S, where R is the relation from $\{1, 2, 3\}$ to $\{1, 2, 3, 4\}$ with $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$ and S is the relation from $\{1, 2, 3, 4\}$ to $\{0, 1, 2\}$ with $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$?

Solution: $S \circ R$ is constructed using all ordered pairs in R and ordered pairs in S, where the second element of the ordered pair in R agrees with the first element of the ordered pair in S. For example, the ordered pairs $(2, 3)$ in R and $(3, 1)$ in S produce the ordered pair $(2, 1)$ in $S \circ R$. Computing all the ordered pairs in the composite, we find

$$S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}.$$

$\overline{A \cup B} = \overline{A} \cap \overline{B}$
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\rightarrow$ De Morgan law.

A	B	$A \cup B$	$\overline{A \cup B}$	$A \cap B$	$\overline{A \cap B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cap \overline{B}$	$\overline{A} \cup \overline{B}$
0	0	0	1	0	1	1	1	1	1
0	1	1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	1	0	1
1	1	1	0	1	0	0	0	0	0

$\overline{A \cup B} = \overline{A} \cap \overline{B}$

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

Recurrence Relation

A recurrence relation is a formula for the next term in a sequence as a function of its previous terms.

- If $< a_n >$ is a sequence, an equation connecting a_n with a finite number of previous terms as $a_{n-1}, a_{n-2}...a_{n-k}$ of the sequence is known as **Recurrence Relation.**

Example:

Fibonacci : 1 1 2 3 5 ..

where $a_n = a_{n-1} + a_{n-2}$, it’s called **Recurrence Relation/Difference Equation.**

Problem Type: Is a solution of … the recurrence ?

From 15th Batch, Anwar Maruf bhai:

c) Is the sequence $\{a_n\}$ a solution of the recurrence relation for $a_n=8a_{n-1}-16a_{n-2}$ if i). $a_n=2^n$ and ii). $a_n=n^2 4^n$? 2

(ii)

$a_n = n^2 4^n$

$\therefore a_0 = 0^2 \times 4^0 = 0$

$a_1 = 1^2 \times 4^1 = 4$

$a_2 = 2^2 \times 4^2 = 4 \times 16 = 64$

$\therefore a_2 = 8 \times a_1 - 16 a_0$

$= 8 \times 4 - 16 \times 0$

$= 32$

So, this is not a recurrence relation

Inclusion-Exclusion Principle (PIE):

(Concept : https://www.youtube.com/watch?v=xFhogXqV_1M&ab_channel=OnnoRokomPathshala

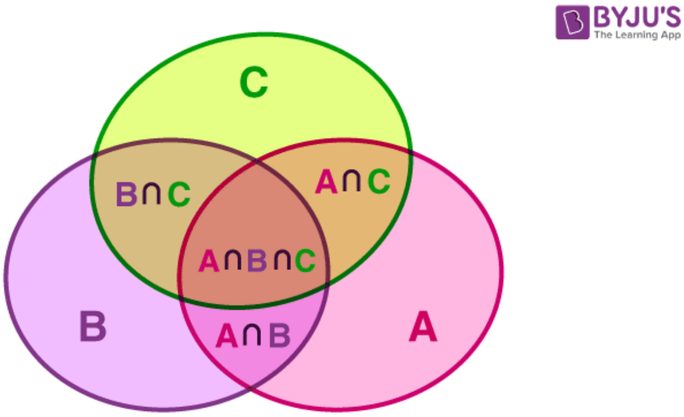
Math:https://www.youtube.com/watch?v=szUTQRJU76Q&ab_channel=MathsStatsUNSW)

Principle of Inclusion and Exclusion is an approach which derives the method of finding the number of elements in the union **(at least one time)** of two finite sets

Example:

$n(A \cup B) = n(A) + n(B) - n(A \cap B)$

$n(A \cup B \cup C) = n(A) + n(B) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap B \cap C)$



Math Problem:

- *and* means intersection
- *or* means union

Among a group of students, 49 study Physics, 37 study English and 21 study Biology. If 9 of these students study Maths Physics and English, 5 study English and Biology, 4 study Physics and Biology and 3 study Physics, English and Biology, find the number of students in the group.

1) 91

2) 92

3) 86

4) none of these

Solution:

Let P represent the number of students who study Physics, E represent the number of students who study English and B represent the number of students who study Biology.

Number of students in the group = $n(P \cup E \cup B)$

Given $n(P) = 49$, $n(E) = 37$, $n(B) = 21$

$n(P \cap E) = 9$

$n(E \cap B) = 5$

$n(P \cap B) = 4$

$n(P \cap E \cap B) = 3$

$n(P \cup E \cup B) = n(P) + n(E) + n(B) - n(P \cap E) - n(E \cap B) - n(P \cap B) + n(P \cap E \cap B)$

$= 49 + 37 + 21 - 9 - 5 - 4 + 3$

$= 92$

Option 2 is the answer.

Permutation & Combinations:

Difference between Permutation & Combination:

Permutation	Combination
Permutations are used when order/sequence of arrangement is needed.	Combinations are used to find the number of possible groups which can be formed.
Permutations are used for things of different kind.	Combinations are used for things of similar kind.
Permutation of two things from three given things a, b, c is ab, ba, bc, cb, ac, ca	Combination of two things from three given things a, b, c is ab, bc, ca
For the different possible arrangement of 'r' things taken from 'n' things is ${}^nP_r = \frac{n!}{(n-r)!}$	For different possible selection of 'r' things taken from 'n' things is ${}^nC_r = \frac{n!}{r!(n-r)!}$

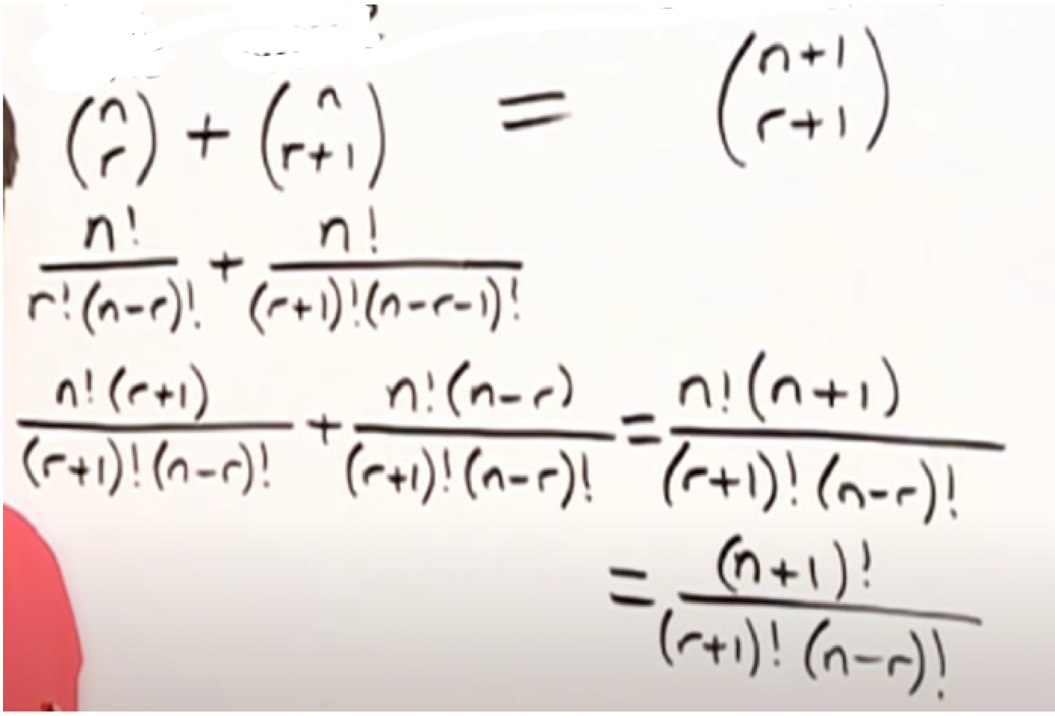
Binomial Theorem:

The Binomial Theorem is the method of expanding an expression that has been raised to any finite power.

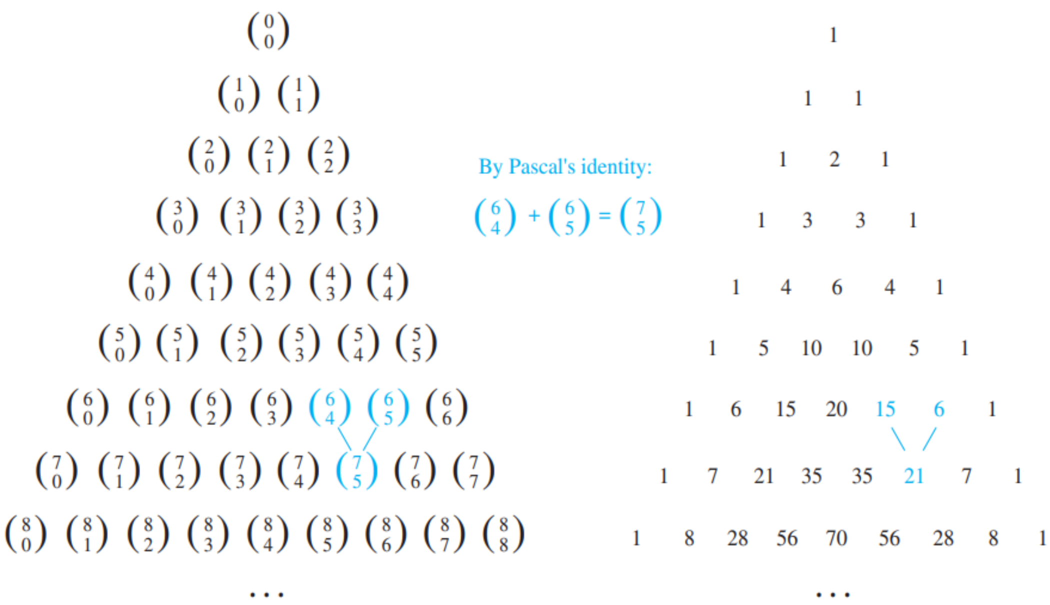
$$nC_r = \frac{n!}{(n-r)!r!}$$

Pascal Identity:

Proof:



Proof by Pascal Identity*:**



(a) (b)

The keywords like-**selection, choose, pick, and combination**-indicates that it is a combination question.

The Keywords like-**arrangement, ordered, unique- indicates that it is a permutation question.**

Product Rule : Subtask to solve a problem/and

Sum Rule : when these are different way of solving the same problems

Permutation of a word

$n \rightarrow$ numbers of letters

$$\left.\begin{matrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{matrix}\right\} \rightarrow \text{number of repetition of letters}$$

$$\frac{n!}{r_1! \cdots r_n!}$$

$n \rightarrow$ male

$m \rightarrow$ female

choosing any only one of them ? $\rightarrow {}^{(n+m)}C_1$

creating a team with 'a' male and 'b' female? $\rightarrow {}^{n}C_a \times {}^{m}C_b$

Alternative method :

$$^{n}C_{r} = \frac{n!}{r!(n-r)!}$$

$$^{n}C_{n-r} = \frac{n!}{(n-r)!(n-n+r)!} = \frac{n!}{r!(n-r)!}$$

$$\therefore\ ^{n}C_{r} =^{n} C_{n-r}.$$

To count the number of bit strings with 2 consecutive one bits (bad strings), I would let

$$\begin{aligned} S_1 &= 11xx & 4 \\ S_2 &= x11x & 4 \\ S_3 &= xx11 & 4 \\ N_1 &= & 12 \end{aligned}$$

Then

$$\begin{aligned} S_1 \cap S_2 &= 111x & 2 \\ S_1 \cap S_3 &= 1111 & 1 \\ S_2 \cap S_3 &= x111 & 2 \\ N_2 &= & 5 \end{aligned}$$

and

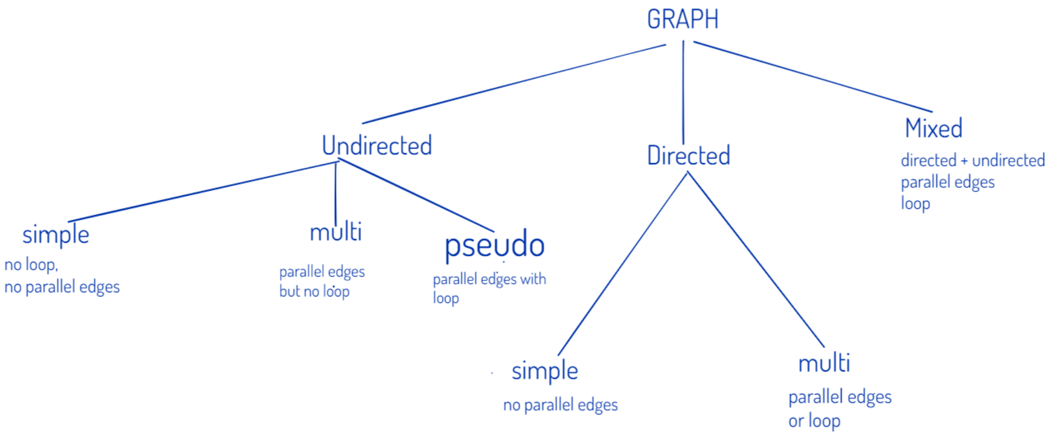
$$\begin{aligned} S_1 \cap S_2 \cap S_3 &= 1111 & 1 \\ N_3 &= & 1 \end{aligned}$$

The count of bad strings is $N_1 - N_2 + N_3 = 8$. → $4! - 12 + 5 - 1 = 8$
The count of good strings is $16 - 8 = 8$.

Graph

- a mathematical structure of two set V (Vertices or Nodes) and E (edges) where V and E both are non-empty

Classifications in a nutshell :



Adjacency Matrix:

Undirected Graph:

- Edges : 1
- No edges : 0
- Self-loop: 1

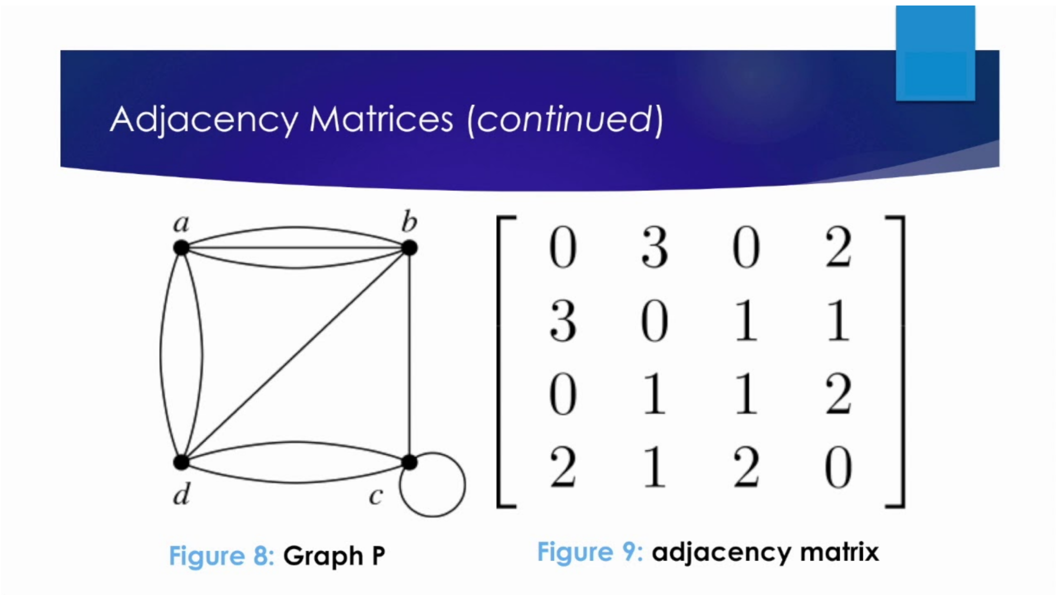


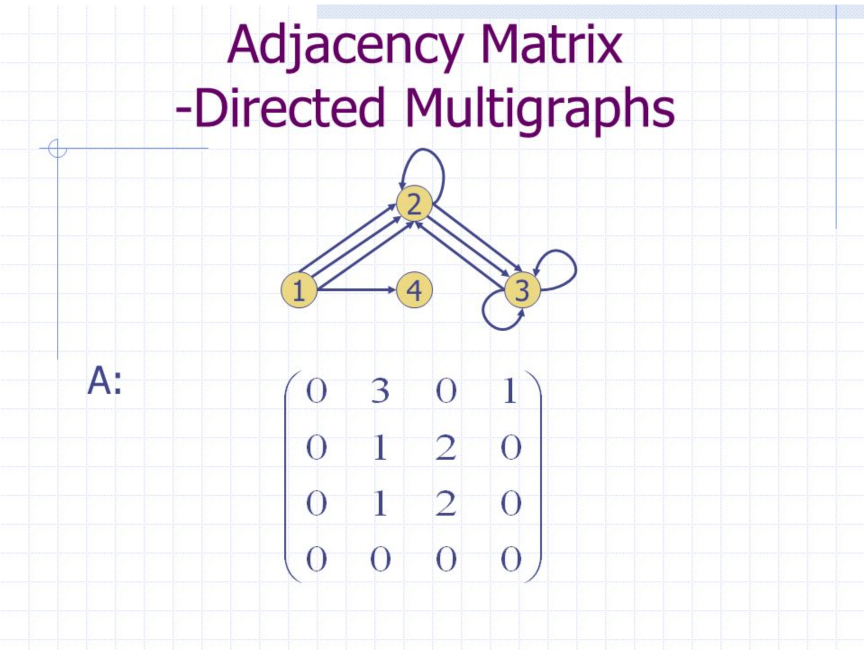
Figure 8: Graph P

Figure 9: adjacency matrix

Directed Graph:

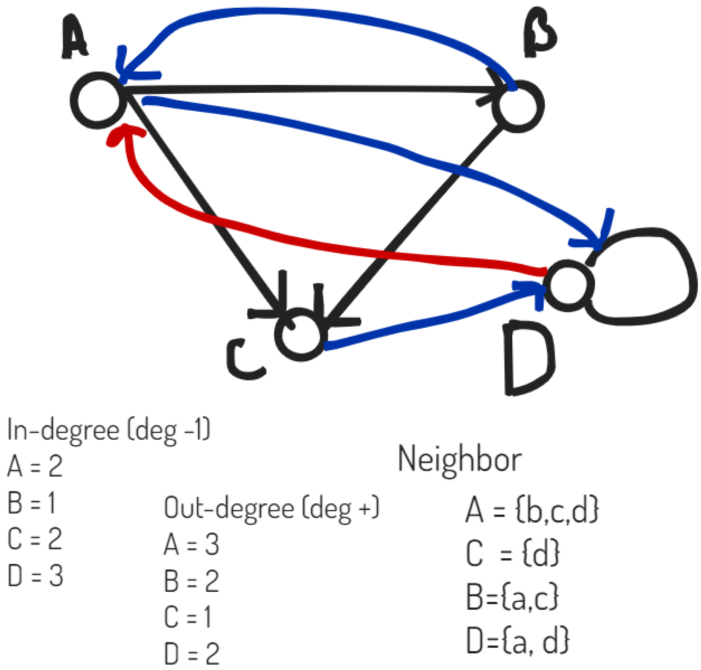
- In-degree : edges entering to the nodes
- Out-degree: edges going out of the node
- Neighbors : connected by out-degree nodes

**** Calculate the out-going vertex**



$$\begin{pmatrix} 0 & 3 & 0 & 1 \\ 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Edges : 1
- No edges : 0
- Self-loop: 1

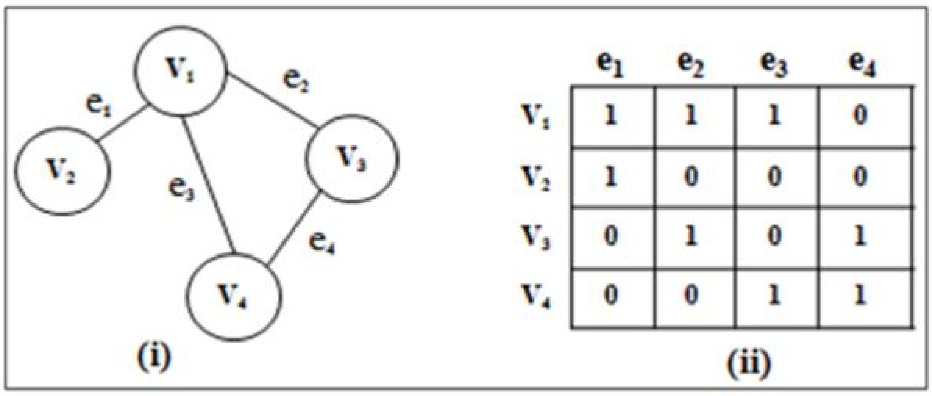


A self-loop adds 2 to the degree of a vertex, or 1 to both indegree and outdegree in case of directed graphs

Incident Matrix:

Undirected :

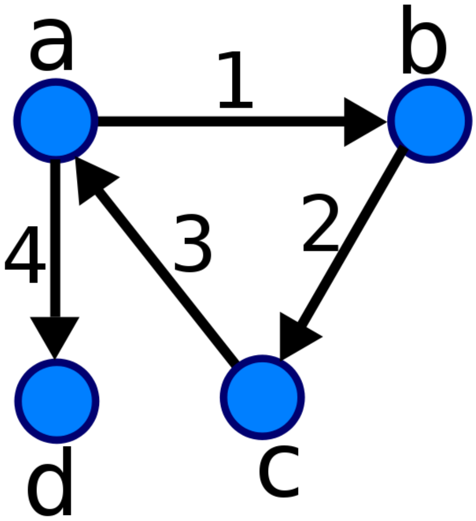
- If edge is connected to the node 1
- else 0



	e_1	e_2	e_3	e_4
V_1	1	1	1	0
V_2	1	0	0	0
V_3	0	1	0	1
V_4	0	0	1	1

Directed Graph:

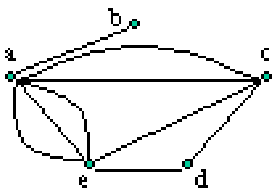
- If out-degree +1
- if in-degree -1

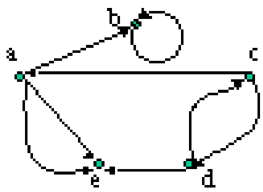


	1	2	3	4
a	1	0	-1	1
b	-1	1	0	0
c	0	-1	1	0
d	0	0	0	-1

Math:

7.3 Adjacency matrices for multigraphs





Adjacency matrix

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 3 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

m_{ij}= The number of edges $\{v_i, v_j\}$

Adjacency matrix

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

m_{ij}= The number of edges (v_i, v_j)

From 15th batch, Anwar Maruf Bhai:

A directed graph (minimum 6 vertices, 2 edges between vertices and 3 loop):

Fig: Graph G

In-degrees in G	Out-degrees in G
$deg^-(A) = 2$	$deg^+(A) = 2$
$deg^-(B) = 2$	$deg^+(B) = 3$
$deg^-(C) = \cancel{0}\ 4$	$deg^+(C) = 1$
$deg^-(D) = 2$	$deg^+(D) = 2$
$deg^-(E) = 3$	$deg^+(E) = \cancel{2}\ \cancel{0}\ 4$
$deg^-(F) = 1$	$deg^+(F) = 2$
Total = 14	Total = 14

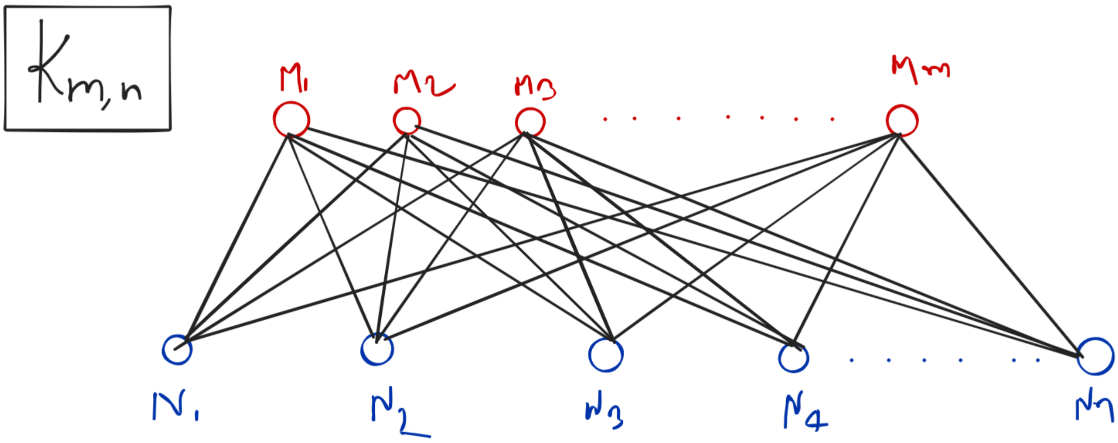
Bipartite Graph/Coloring Graph:

Theory : A **Bipartite Graph** is a graph whose vertices can be divided into two independent sets, U and V such that every edge (u, v) either connects a vertex from U to V or a vertex from V to U. In other words, for every edge (u, v), either u belongs to U and v to V, or u belongs to V and v to U.

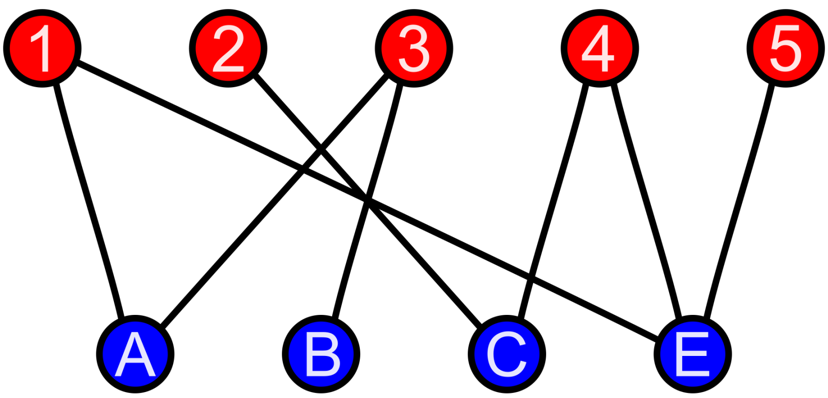
Two neighbor nodes shouldn't be the same color.

- adjacent nodes should be different colors

- (2n+1) nodes cycle is not bipartite, n≥1

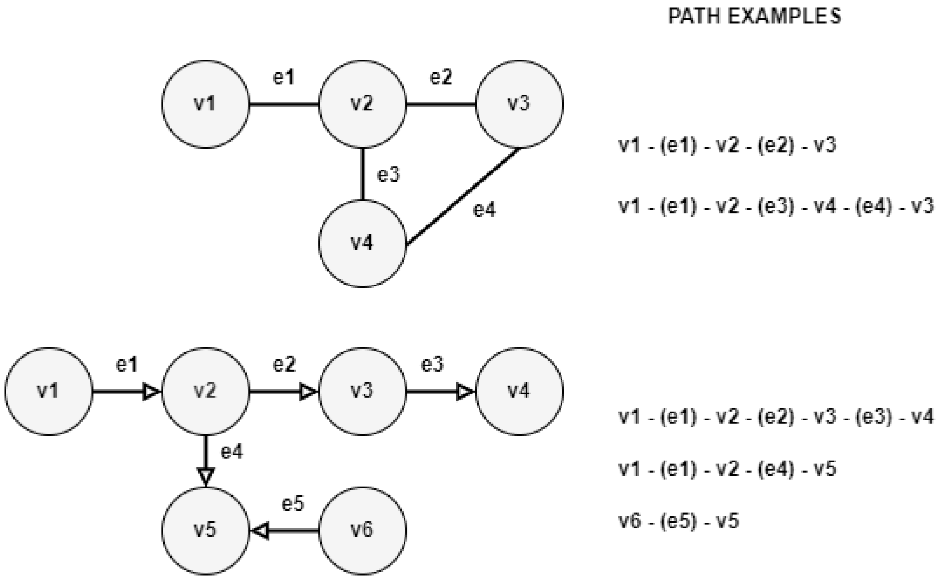


$Example : K_{5,4}$



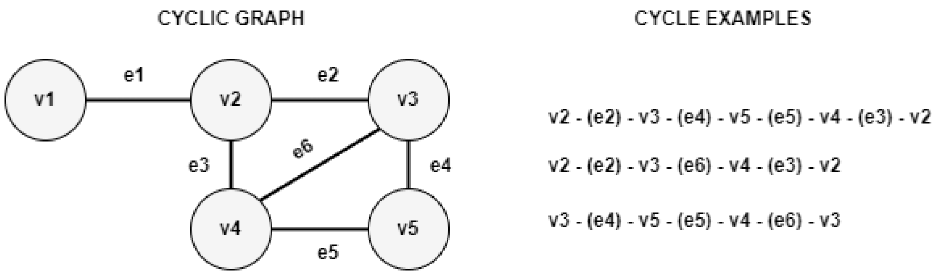
Path:

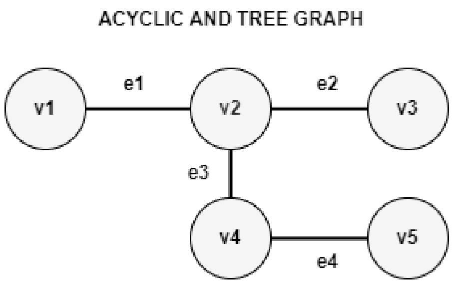
- traverse a graph no repetition of a vertex nor an edge
- an open walk (if starting vertex and ending vertex are different than it's called open walk)



Circuit

- A circuit is a sequence of adjacent nodes starting and ending at the same node
- a close walk (starting and ending node is same)
- not repeated edges





Differences

	Can have repeated nodes?	Can have repeated edges?	Open or Closed?
Walk	Yes	Yes	Both
Trail	Yes	No	Open
Path	No	No	Open
Circuit	Yes	No	Closed
Cycle	No	No	Closed

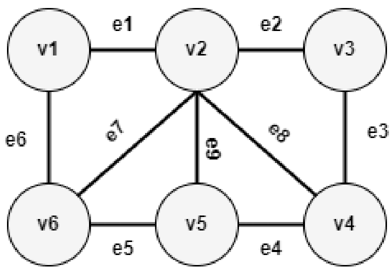
Eulerian Circuit:

- A circuit if it traverses each edge in the graph one and only once
- Edges can't be repeated
- Edge < Vertex

Hamiltonian Circuit:

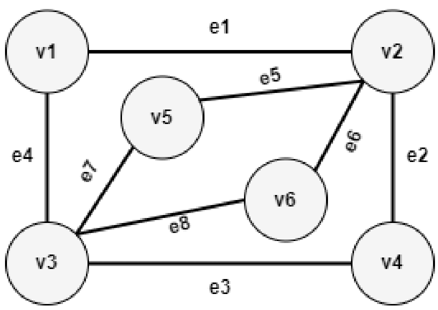
- A circuit passing through **all vertex** of a graph
- Vertex can't be repeated
- Edge > Vertex

CIRCUIT EXAMPLES



v1 - (e1) - v2 - (e2) - v3 - (e3) - v4 - (e4) - v5 - (e5) - v6 - (e6) - v1

Hamiltonian Circuit



v1 - (e1) - v2 - (e5) - v5 - (e7) - v3 - (e8) - v6 - (e6) - v2 - ...
... (e2) - v4 - (e3) - v3 - (e4) - v1

Eulerian Circuit

Isomorphism:

A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity. Such graphs are called isomorphic graphs.

Two Graphs G1 and G2 will Isomorph iff,

- Vertices number are same
- Edges numbers are same
- Equal number of vertices with equal degrees
- Number of cycle must be same

Math:

From SSROY bhai:

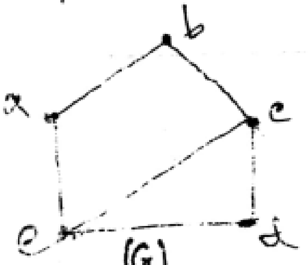
[II] Isomorphism:

Number of verotex and number of edge should be same. Then its called isomorphic.

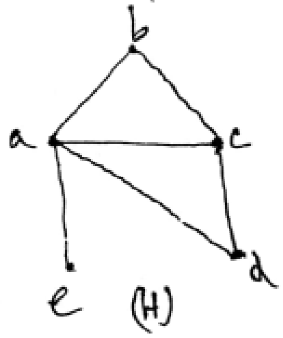
i) Number of vertices should be same.
ii) Number of edges should be same.
iii) Vertices of degree should be same.
iv) No of circuit/cycle should be same.

Example:

[path - at least two edges needed]



(G)



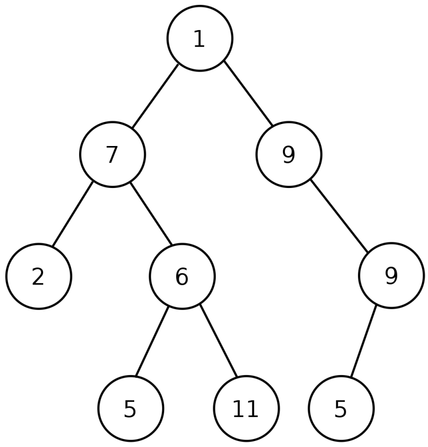
(H)

(G)	(H)
1) No of vertices - 5	1) No of vertices - 5
2) No of edges - 5	2) No of edges - ~~5~~6
3) No of degree two - (a, b, d)	3) No of degree two - (b, d).
4) No of circuit - 3	4) No of circuit - 3

∴ So these two graphs (G) and (H) are not isomorphic.

Tree:

Definition : A **connected undirected acyclic graph** is called a tree. In other words, a connected undirected graph with no cycles is called a tree.



(https://www.geeksforgeeks.org/difference-between-graph-and-tree/)

The basis of Comparison	**Graph**	**Tree**
Definition	Graph is a non-linear data structure.	Tree is a non-linear data structure.
Structure	It is a collection of vertices/nodes and edges.	It is a collection of nodes and edges.
Edges	Each node can have any number of edges.	If there is n nodes then there would be (n-1) number of edges
Types of Edges	They can be directed or undirected	They are always **undirected**
Root node	There is no unique node called root in graph.	There is a unique node called root(parent) node in trees.
Loop Formation	A cycle can be formed.	There will not be any cycle.
Traversal	For graph traversal, we use Breadth-First Search (BFS), and Depth-First Search (DFS).	We traverse a tree using in-order, pre-order, or post-order traversal methods.
Applications	For finding shortest path in networking graph is used.	For game trees, decision trees, the tree is used.

Binary Tree: Binary Tree is defined as a tree data structure where each node has **at most 2 children.**

From Anwar Maruf bhai:

Internal vertices: The nodes/vertices which are having child is called internal vertices.
Ex: A, B, C, D, E and G.

Parent: The immidiate predicisor of any node is known as parent node/parent.
Ex: G is parent of L&M.

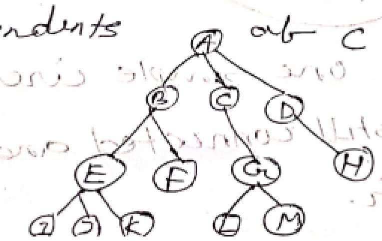
Child: The immidiate successor of any node is known as child node/child.
Ex: L and M are children of G.

Leaf node: The node which is having no child. (External node/Leaf) Ex: I, J, K, L, M & H.

Siblings: All the children of same parent is called siblings. Ex: B, C and D are siblings.

Ancestors: Any predecessor on the path from root to that node. Ex: The ancestors of L is G, C and A.

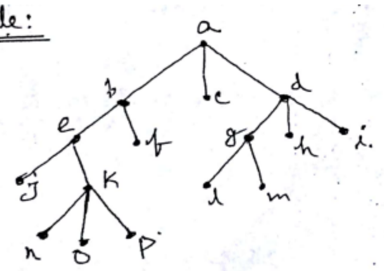
Decendents: Any successors node on the path from that node to leaf node.
Ex: The decendents of C is G, L and M.



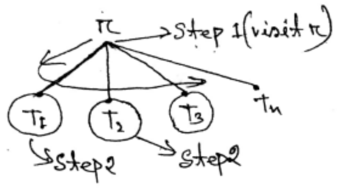
Traversing :

From SSRoy bhai:

Example:

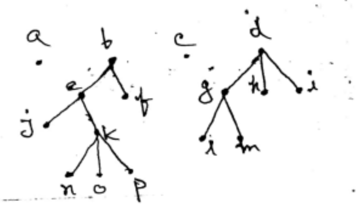


Pre-order:

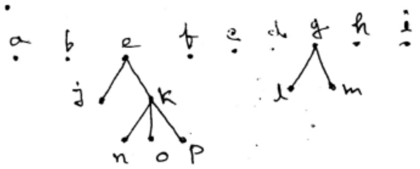


(Visit root)
(Visit subtrees left to right)

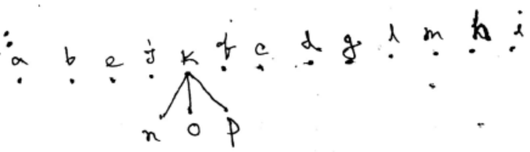
Step 1:



Step 2:



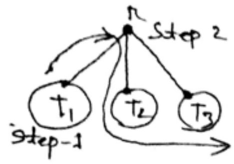
Step 3:



Step 4:

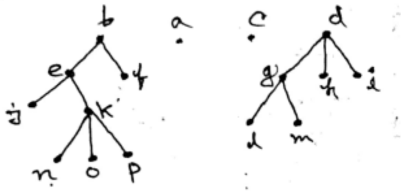
a b e j k n o p f c d g l m h i

In-order:

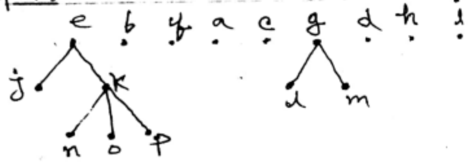


Visit left most subtree, visit root, visit subtrees from left to right.

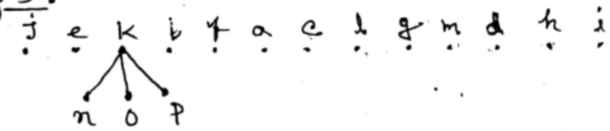
Step 1:



Step 2:

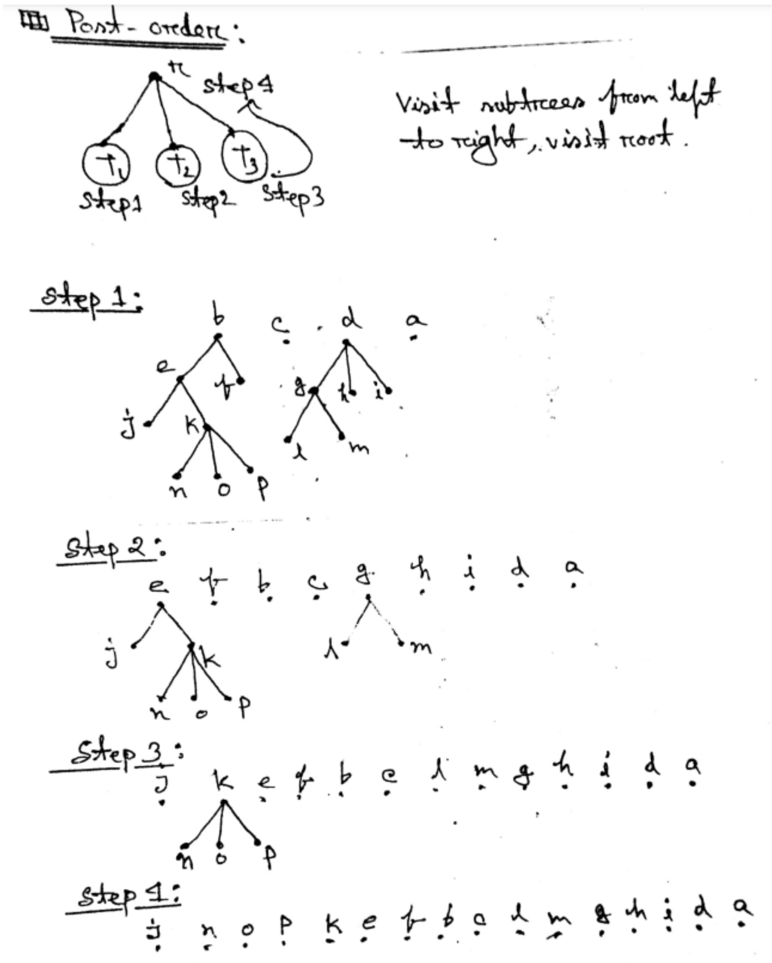


Step 3:



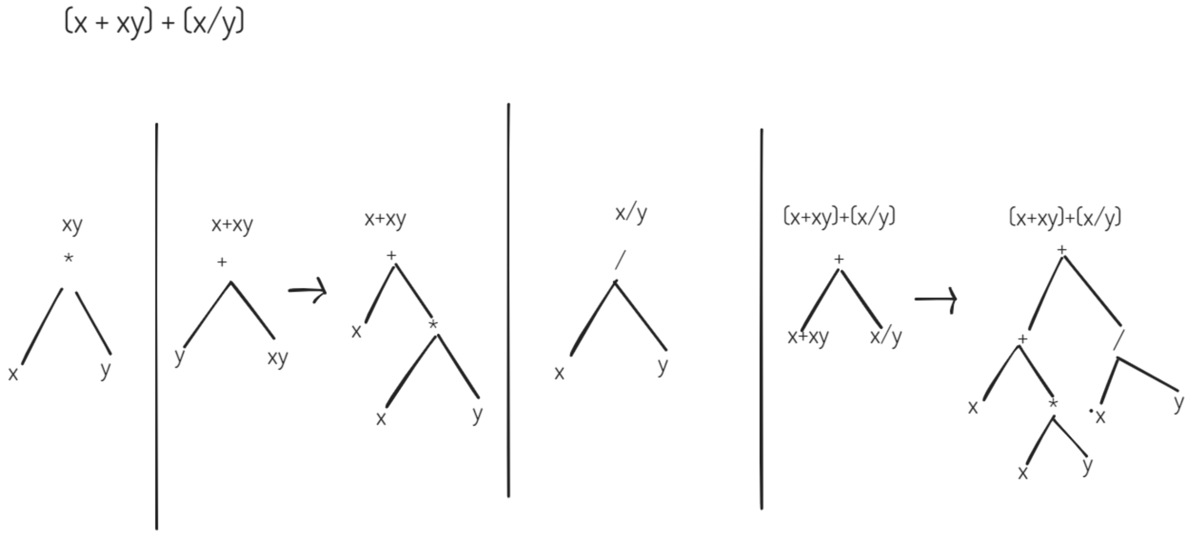
Step 4:

j e n k o p b f a c l g m d h i



Converting an expression to an expression tree:

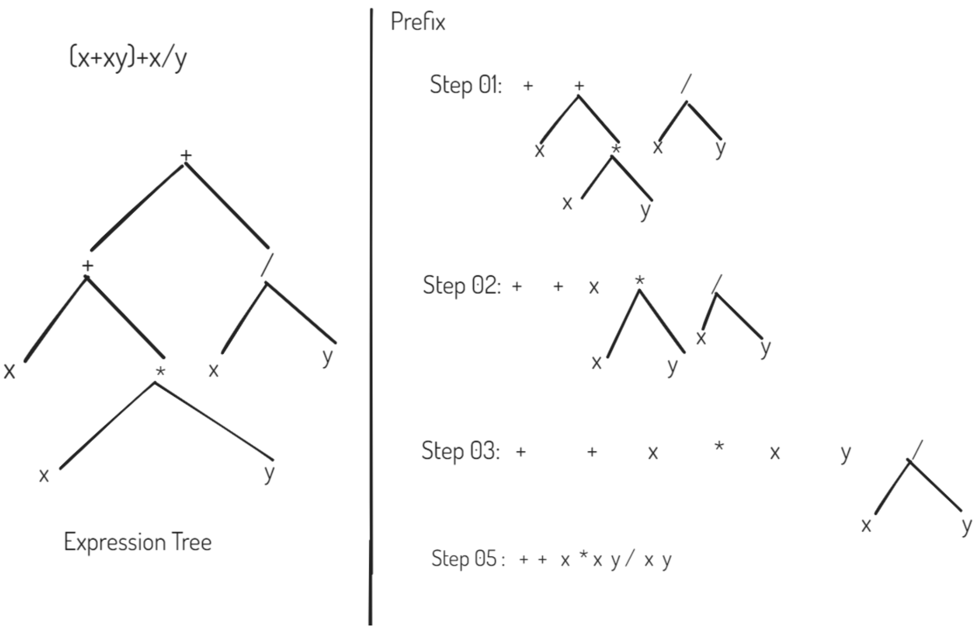
- **It's an infix expression.**



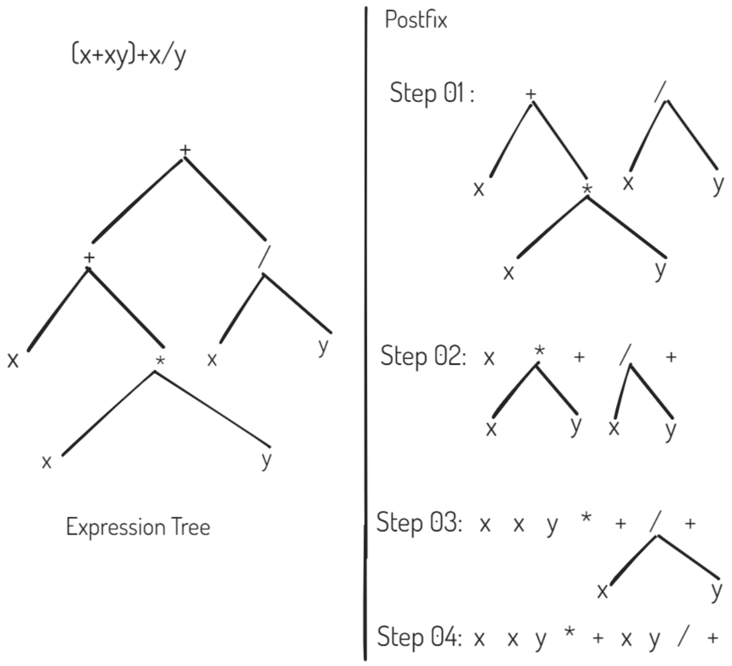
Any expression to postfix or prefix expression:

1. Create Expression Tree
2. Traverse the tree according to question (postfix, prefix)

Expression to Prefix Traversing:



Expression to Postfix Traversing:


(x+xy)+x/y
Expression Tree
Postfix
Step 01 :
Step 02: x * + / +
Step 03: x x y * + / +
Step 04: x x y * + x y / +