GRAPH THEORY & COUNTING ASSIGNMENT

Page No. 01
Date 7 12 2020

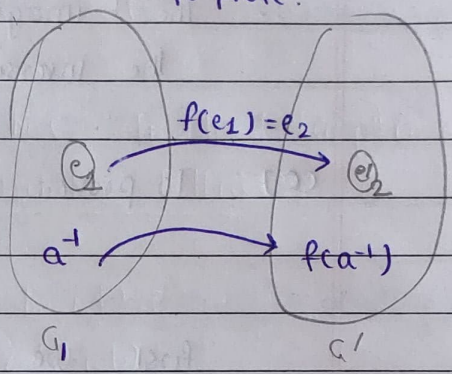
U19CS012 [BHAGVA VINOD RANA]

1) Prove that: If $f: (G_1, *_1) \to (G_2, *_2)$, is homomorphism from G_1 to G_2, the ① $f(e_1) = e_2$

② $f(a^{-1}) = (f(a))^{-1} \quad \forall a \in G$

③ Also, ker f is normal subgroup of G.

1)



(a) let $a \in G$,

By Identity property

$ae_1 = a = e_1a$

Applying function both sides

$\Rightarrow f(ae_1) = f(a) = f(e_1a)$

$\Downarrow$ (∵ homomorphism)

$f(a) * f(e_1) = f(a) = f(e_1) * f(a)$

$f(e_1)$ behaves as identity [$A\, f(e_1) = A = f(e_1)\, A$]

$\Rightarrow f(e_1)$ is identity in G', let identity in $G' = e_2$

$\therefore \underline{[f(e_1) = e_2]}$

Therfore, the image of the identity of G under group morphism f is the identity of G'.

Short = $f(e_1) = f(e_1e_1) = f(e_1)\, f(e_1)$

multiplying both sides $f(e_1)^{-1}$

$[e_2 = f(e_1)]$

(b) let a^{-1} be the inverse of a ($\forall a \in G$), then

$aa^{-1} = e = a^{-1}a$

Applying 'f' $\Rightarrow f(aa^{-1}) = f(e) = f(a^{-1}a)$ $\quad f(aa^{-1}) = f(a)\, f(a^{-1})$

$f(a)\, f(a^{-1}) = f(e) = f(a^{-1})\, f(a)$ [∵ f is homomorphism]

$f(a)\, f(a^{-1}) = \Downarrow e_2 = f(a^{-1})\, f(a)$

[from above (a) $f(e_1) = e_2$

Page No.
Date

$(f(a))(f(a^{-1})) = e = (f(a^{-1}))(f(a))$

inverse

$\therefore \boxed{f(a^{-1}) = [f(a)]^{-1}}$

∴ the f-image of inverse of any element of G under f is the inverse of f-image of a in G'.

(c) To prove: $\frac{\ker(f) = \text{Normal subgroup of } G}{[\ker(f) \triangleleft G]}$

first, we will prove ker(f) is subgroup of G

Let e and e' be the identities of G_1 and G_2 resp.

then

$\ker(f) = K = \{x \in G \mid f(x) = e'\}$ (such that) (subset)

$(\ker(f) \subset G)$

$\therefore f(e) = e'$

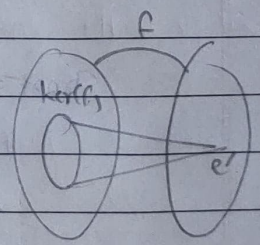
$\Rightarrow e \in K \Rightarrow K \neq \phi$

(Non-empty set)

∴ Let $a, b \in K$;

then $f(a) = e'$

$f(b) = e'$



~~If~~ $ab \in$ Not, we will prove

ab^{-1} → should belong to K [Subgroup defn]

$f(ab^{-1}) = f(a)\,f(b^{-1}) = f(a)\,(f(b))^{-1}$ (We proved earlier)

($\because$ homomorphism)

$a \in K, b \in K$ $= (e')(e')^{-1}$ ($\because$ inverse of (identity) = identity)

$ab^{-1} \in K$ $= (e')(e')$

3. ker(f) $= (e')$

is subgroup

$\Rightarrow [ab^{-1} \in K]$ ∴ ker(f) is subgroup

U19CS012

To prove "Normal" subgroup, we will prove $xax^{-1} \in k$ $\left[\begin{matrix} a \in k \\ x \in G \end{matrix}\right]$

$k \lhd G$: let $x \in G$ and $a \in k$

$f(xax^{-1}) = f(x) * f(a) * f(x^{-1})$ [$\because$ f is homomorphism]

$= f(x) * e' * (f(x))^{-1}$ [$f(x^{-1}) = (f(x))^{-1}$]

$= f(x) * (f(x))^{-1}$

$= e'$ [(ele)*(ele)$^{-1}$ = identity]

$\therefore x \in G, a \in k \Rightarrow xax^{-1} \in k$

$\therefore$ $\ker(f) \lhd G$, Hence proved.

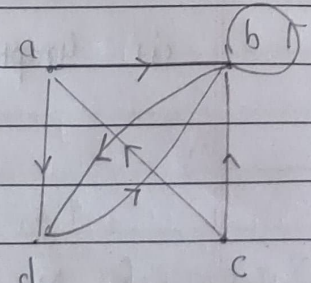
(Normal subgroup)

② Define Digraphs. what is the minimum no. of edges in a strongly connected ~~dig~~ diagraph having n vertices.

② Digraph = Directed Graph or Digraph, consist of a set V of vertices (or nodes) together with a set E of ordered pairs of elements of V called edges (or arcs).

The vertex a is called initial vertex of the edge (a, b) and the vertex b is called terminal vertex of this edge.

[→ = represents direction's]

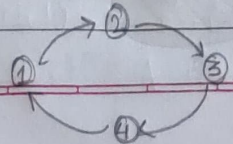


Minm no. of edges = n

∵ you can create a cycle to connect n vertices with "n-1" edges to make strongly-connected comp.

OR

(in-degree / out-degree) >= 1 each vertex

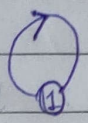


U19CS012

Page No.
Date

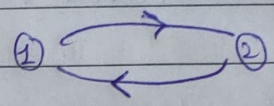
Proof by Mathematical induction

(*) $n = 1$

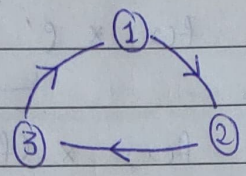


To make di-graph, with one vertex
Strongly connected (path to between all pairs of vertices)

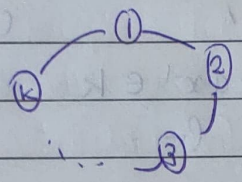
(*) $n = 2$



(*) $n = 3$

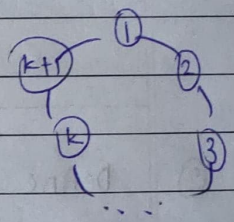


$n = k$



is true

for: $n = k+1$ ∴ Adding one vertex in cycle,



Ans:

[Min^m no. of edges with 'n' vertices = "n" to make digraph Strongly connected]

Such di-graph have cyclic shape. (Strongly connected)
∴ We can reach from any vertex to any other vertex.

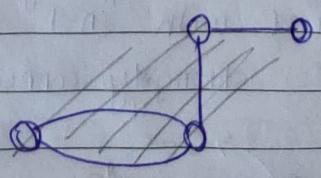
(3) Draw a graph having the given properties or explain why no such graph exists.

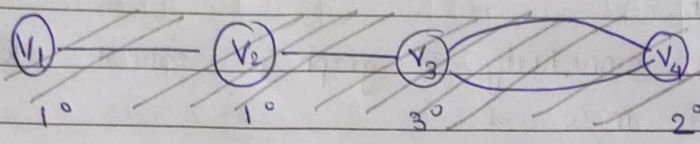
(i) Graph with four vertices of degree 1, 1, 2, 3

$$\sum_i deg(V_i) = 1+1+2+3 = 7 \notin Even$$

↳ odd

No such





(ii) $\sum_{i=1}^{4} \deg(v_i) = 1+1+2+3 = 7 \neq 2*(\text{no. of edges})$ (even)

7 [odd] = ~~(8)~~

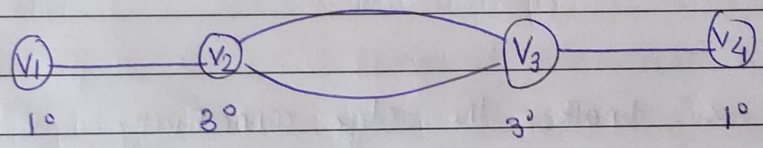
$\therefore$ No such graph exist

(ii) Graph with four vertices of degree 1, 1, 3 & 3

$\sum_{i=1}^{4} \deg(v_i) = 1+1+3+3$

$= 8 = 2*(\text{no. of edges})$ (even)

$\Rightarrow$ [no. of edges $= 8/2 = 4$]



$v_1\ v_2\ v_3\ v_4$

(iii) Simple Graph with four vertices of degree 1, 1, 3 & 3.

$\because$ Simple Graph $\rightarrow$ x no self loops

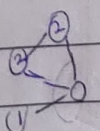
$\rightarrow$ x no parallel edges

vertex (v_3) ~~with degree 3~~ (degree 3) $\Rightarrow$ will need incoming edge from v_1, v_2, v_4

vertex (v_4) $\Rightarrow$ will need incoming edge from v_1, v_2, v_4

$\Rightarrow$ v_1 & v_2 have 2 degrees, but this is false.
(connected to v_3 & v_4)

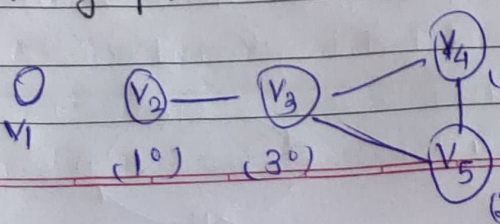
$\therefore$ No such Graph exist



(iv) Simple graph with 5 vertices (0), 1, 2, 2, 3

(0) trivial

$\sum(\deg(v_i)) = 8$ ✓



4.) Prove That

(A) " Edge connectivity of graph G cannot exceed the smallest degree in G &

(B) vertex connectivity of graph G cannot exceed the edge connectivity of graph G "

(A) Let vertex v_i be the vertex with the smallest degree in G.

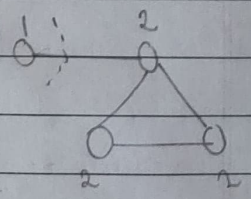
Let $d(v_i)$ be the degree of v_i.

⇒ Vertex v_i can be seperated from G by removing $d(v_i)$ edges incident on v_i

IMP

Hence the theorem states

Edge connectivity cannot exceed smallest degree in G



(B) Let α denote the edge connectivity of G.

∴ Therfore, there exist a cut set S in G with α edges

Let S partition the vertices of G into subset

V_1 V_2

By removing at most α vertices from V_1 (or V_2) on which the edges in S are incident,

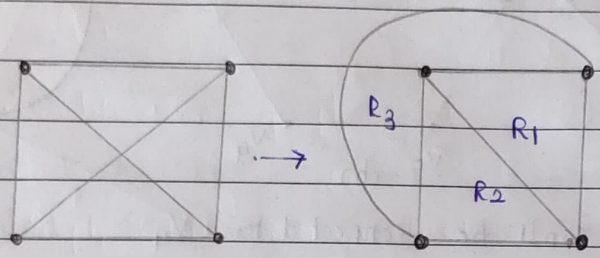
We can affect the removal of S (together with all other edges incident on these vertices) from G. Hence the theorem is proved.

Kuratowski's two Graph's

Page No. 07
Date

(5) Define Planar Graph & Regions. Are K_5 & $K_{3,3}$ planar graphs? Why?

A Graph is called Planar if it can be drawn in the Plane without any edges crossing.



The Graph K_4 (with crossing)

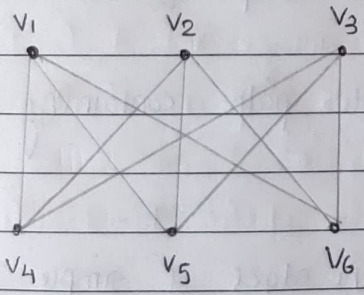
K_4 with no crossing

A planar representation splits the plane into "regions"

[R_1, R_2, R_3 & R_4]

(R_4) = Outer Region

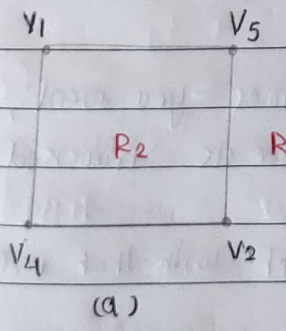
$K_{3,3}$ Graph (Bipartite)



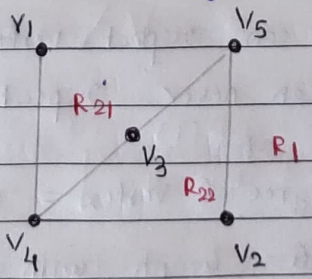
In any planar representation, of $K_{3,3}$ the vertices V_1 & V_2 must be connected to both V_4 and V_5. These (4) edges form a closed curve and split plane into two regions R_1 & R_2. figure - (a)

The vertex V_3 is in either R_1 or R_2.

V_3 → inside → seperate into two sub-region [R_{21}, R_{22}] (b)



(a)



∴ There is no way to place vertex V_6 without forcing a crossing.

if → R_1 edge V_3 to V_6 cannot be drawn

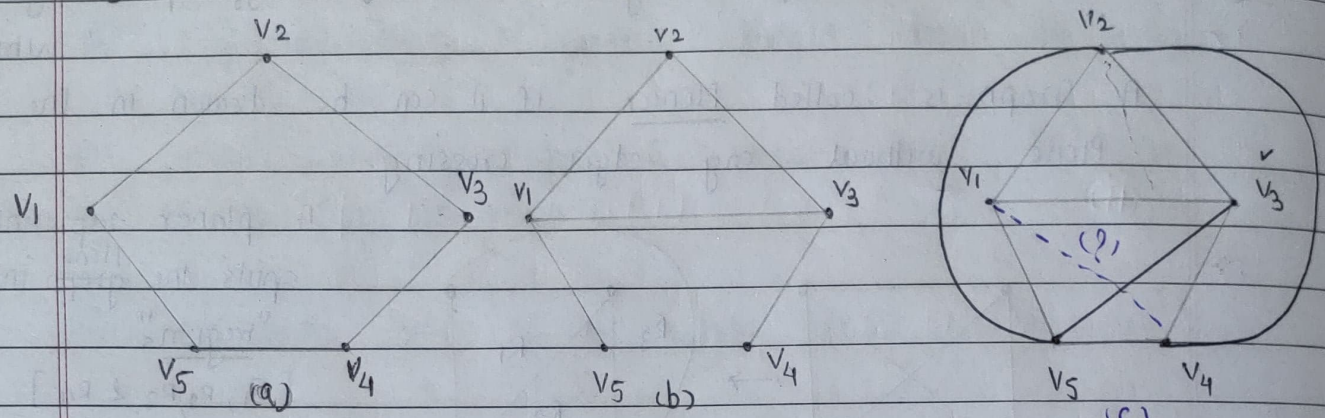
V_6 → R_{21} " V_2 to V_6 " "

→ R_{22} " V_1 to V_6 " "

∴ $K_{3,3}$ is Not Planar similar Approach K_5 is also Non-planar

U19CS012

K_5



V_1 can't be connected to V_4

⑥ Define Euler Graph & S.T a graph G is Eule graph iff every vertex in G has even degree.
What happens if there are exactly two odd vertices.

⑤

An Euler Circuit in a graph G is a simple circuit containing every edge of G.
An Euler path in G is a simple path containing every edge of G.

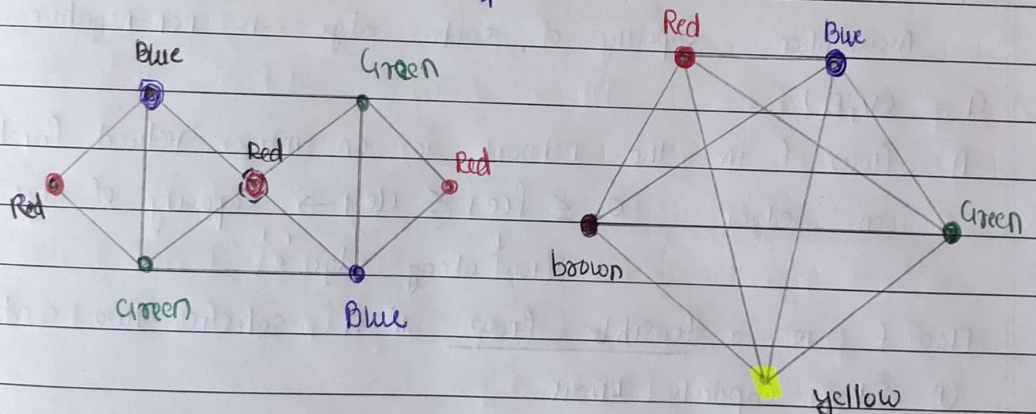
Eulerian Circuit = traversal of all the edges of simple graph once and only once, starting at one vertex and ending at same vertex.
You can repeat vertices as many times you want, but you can never repeat an edge once its traversed.

degree of vertex = no. of edges incident with that vertex
Let G = Graph with Eulerian circuit
IMP "Every time we arrive at a vertex during our traversal of G, we enter via one edge and exit via another."
∴ Thus, there must be even number of edges at every vertex
∴ Every vertex of G has even degree.

A connected multigraph has an Euler path
X Not Euler circuit
if and only if it has exactly two vertices of odd degree
You will start at one vertex, but will not ~~retu~~ return
to some vertex.

⑦ What is K-chromatic Graph? Explain with Example.
Also find chromatic number of Complete graph K_n.

⑦ The chromatic number of a graph = Least number of colors to color a particular graph. if chromatic number = k, then graph is called k-chromatic Graph.



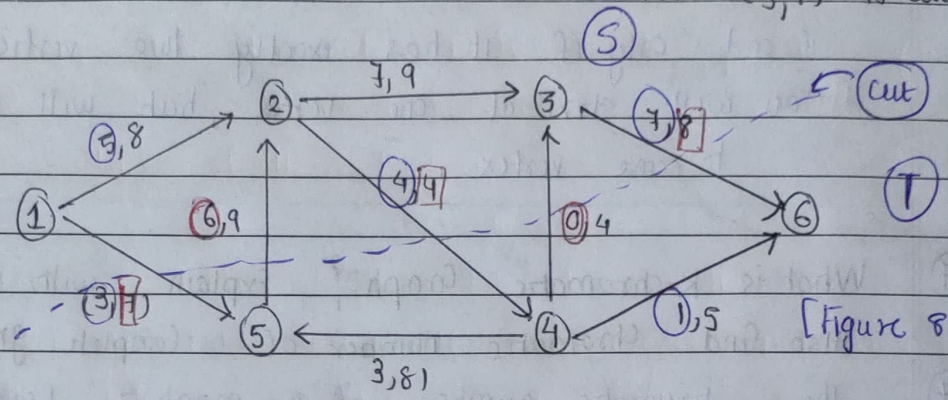
Coloring in K_5 (complete graph)

A coloring of K_n can be constructed using n colors by (fully connected)
assigning a different color to each vertex.
[few colors? No]
No two vertices can be assigned same color, because
"Every two vertices in K_n are adjacent"

$\therefore \quad \chi(K_n) = n$ ($K_5 \rightarrow$ 5 colors
→ Not planar
→ x contradict 4 color theorem)

(chi) Chromatic No.

8.) What is feasible flow? Find cut in the capacited flow given below & hence verify $f(G) = f(S,T) - f(T,S)$ for (S,T) is cut



[Figure 8(a)]

8.) Vertices $\{1, 2, \ldots n\}$ (V) — source: 1 (indegree 0) — (intermediate vertex): $(2 \ldots n-1)$ — sink: n (outdegree 0)

Assumption: Capacity of each edge is non-negative.

$G = (V, E)$

A flow f in the network is an integer-valued function defined on edges $0 \le f(e) \le c(e)$ → capacity of edge 'e'
↓ flow along edge 'e'

Flow f is a <u>feasible flow</u> if it satisfies two condition

① Edge Capacity Limit

$\forall e \in E \quad 0 \le f(e) \le c(e)$

② conservation of Flow

$\forall v \in V \setminus \{s,t\} \quad \sum_{e \text{ leaving } v} f(e) = \sum_{e \text{ entering } v} f(e)$

[OUTFLOW = INFLOW] at vertex (v)

[source (s) & sink (t) do not obey this]

Theorem: For an arbitrary cut of the network G the value of flow is given by

$f(G) = f(S, T) - f(T, S)$

source

(S,T) = cut

In Figure 8(a)
(←)

Inflow into the sink = 7+1 = 8
Outflow from the source = 5+3 = 8
which is current flow value = (8)
} (conservation) ✓ of Flow

If S = {1,2,3} and T = {4,5,6}

the flow value = f(S,T) - f(T,S)

⇓

(→) from ② to ① — from ① to S (←)

= (3+4+7) - (6+0)

= 8

⑤ ① (→) □

Flow value is less than or equal to the capacity = 7+4+8 = 19 of this (cut)

Hence, f(G) = f(S,T) - f(T,S) is verified.

9) Explain Hoare's Logic for program verification in Brief.

Hoare's Logic : is a formal system with a set of logical rules for reasoning rigorously about the correctness of computer program.

Central feature of Hoare's Logic = Hoare's Triplet
It describes how the execution of a piece of code changes the state of computation.
It is in the form ← command

{P} C {Q}

↖ Assertion ↗

P → is the Precondition and Q is Post condition

Page No.
Date

Continued
9.) Explain Hoare's Logic for Programme Verification in Brief.

9.) continued
Using Hoare's Logic, only partial correctness can be proven, while termination needs to be proven seperately

Intuitive reading of Hoare triple is:

Whenever P holds of the state before the execution of C the Q will hold afterwards or C does not terminate. In the latter case, there is no "after", & so Q can be any statement at all. Indeed, one can chose Q to be false to express that C does not terminate.

10.) Explain in Brief : Linear recurrence relation, inclusion and exclusion for counting.

10.) A recurrence relation is an equation that recursively defines a sequence where the next term is a function of previous terms (Expressing F_n as some characteristics of F_i with $i < n$)

Eg: Fibonacci seriel $F_n = F_{n-1} + F_{n-2}$ $(a_1 = a_2 = 1)$
Tower of Hanoi $F_n = 2F_{n-1} + 1$

⊛ Linear Recurrence Relation: A linear recurrence relation of degree k / order k is a recurrence relation which is in the format

$$x_n = A_1 x_{n-1} + A_2 x_{n-1} + \dots A_k x_{n-k} \quad (A_n = \text{constant}, A_k \neq 0)$$

on sequence of number as First degree polynomial

(*) Principle of inclusion - Exclusion (PIE)

It is a counting technique that computes the number of elements that satisfy atleast one of several properties while guaranteeing that elements satisfying more than one property are not counted twice.

Idea behind PIE is summing the number of elements that satisfy atleast one of two categories and subtracting the overlap prevents double counting.

Eg: $n(A \cup B) = \underbrace{n(A) + n(B)}_{\text{inclusion}} - \underbrace{n(A \cap B)}_{\text{Exclusion}}$

$n(A \cup B \cup C) = \underbrace{n(A) + n(B) + n(C)}_{\text{inclusion}} - \underbrace{(n(A \cap B) + n(B \cap C) + n(A \cap C))}_{\text{Exclusion}} + \underbrace{n(A \cap B \cap C)}_{\text{inclusion}}$

11) Explain in Brief : first counting principle, second counting principle & circular permutation.

11) ① First counting principle [Sum Rule]

Suppose some event E can occur in m ways } event F can occur in n ways }

Suppose both event cannot occur simultaneously.

(E or F) can occur in "m+n" ways.

Generalizing

Event	ways	
E_1	n_1	& (no two events can occur at same time)
E_2	n_2	
E_3	n_3	
:	:	

one event can occur in $(n_1 + n_2 + n_3 + \dots)$ ways.

② Second Counting Principle [Product Rule Principle]

Suppose Event E $\xrightarrow{\text{occur}}$ m ways

← independent to Event 'E'

Event F $\xrightarrow{\text{occur}}$ n ways

Combination E and F $\xrightarrow{\text{can occur}}$ m×n ways.

Generalizing

Event	ways
E_1	n_1
E_2	n_2
E_3	n_3
⋮	⋮

[All E_1, E_2, E_3 independent of each other]

all events can occur in order $n_1 * n_2 * n_3 * \ldots$ ways.

③ Circular Permutation

Circular permutation = total number of ways in which n distinct object can be arrange around fix circle.

Two types

Clockwise & Anticlockwise orders are [same]

$$P_n = (n-1)!/2$$

(n = no. of objects)

Clock wise & Anticlockwise orders are [different]

$$P_n = (n-1)!$$

Circular Permutation [n = no. of obj.]