Hasse diagram - compact rep. s.t all edges that are reflexive, transitive is not included. (self loops are not drawn)

minimal rep. (DAC) of a partial order (maximal)

B

every DAC — transitive closure

(maximal) partial order

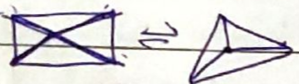
arbitrary DAC → a directed grap (~~& apply reachabil~~ ...

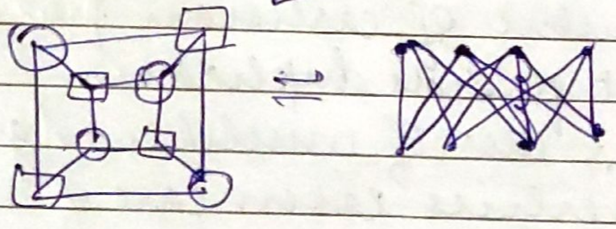
(minimal) hasse diagram

iff transitive closure is ~~not~~ antisymmetric

CQ. Hypercube: hasse diagram of power set partial order

Isomorphism: same structure





→ a family of graphs (Q_0, Q_1, Q_2)

2 copies of Q_0

Q_0 Q_0

2 copies of Q_1

Q_1 Q_1'

OR

→ using bit string

Functions.

6/08/19

- Injective func / one one func has $\forall x, y \in D$
  $(x \neq y) \Rightarrow F(x) \neq F(y)$ $\quad |D| \leq |C|$
- Surjective functions / onto
  $R = C$ $\quad \Rightarrow |D| \geq |C|$
  range, codomain
- Bijective func $\quad \Rightarrow |D| = |C|$
  injective + surjective

* Function composition

$R_f \leq D_g$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 4 & 5 & 6 & 7 & 3 \end{pmatrix} \xrightarrow[\text{to get } 1]{\#\text{op}} 10 \quad (\text{LCM of } (2,5))$$

$G = (V, E)$ → binary relation

simple graph → forbids self loops
→ " multiple edges b/w 2 vertices

A finite graph is a combination of of a finite set of vertices (V) and a binary ~~set~~ irreflexive, symmetric, relations on V called E
(no self loop) (undirected)

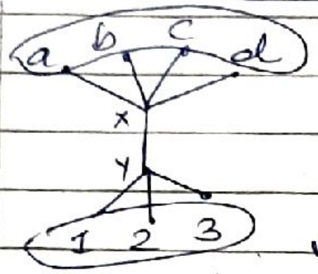
subset of cartesian product
set forbids duplicates
∴ case of multiple edges b/w 2 vertices taken care of

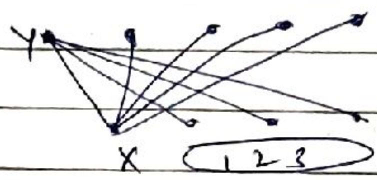
Subgraph : a graph where $V' \subseteq V$ and $E' \subseteq E$
$H(V', E')$

Spanning subgraph
- requires $V' = V$
- ∴ # possibilities $= 2^{|E|}$
- (Need Not be Connected)

Induced subgraph
- edge maximal subgraph for any specified vertex set.
- Only 1 induced SG for a set of vertices.
- $2^{|V|}$

13/08





Isomorphism.

identical structurally

automorphism: a bijective ~~set of a graph~~ fun f from a vertex set of graph to itself such that

$$(u, v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(G)$$

~~Isomorphism is an equivalence~~

Equivalence relation :
~~e.g.~~
- strongly connected components
- isomorphism of graph.

→ reflexive, symmetric, transitive.
(identity) (inverse) (Composition)

- edges & non edges are preserved
- bijective

$$\underbrace{G_1, \; G_2}_{f}, \; \underbrace{\;G_3}_{g}$$

$f \circ g$ → preserve adjacency & non adja cency
→ byective

$(u,v) \in E_1 \Leftrightarrow (F(u), F(v)) \in E_2$

$\Leftrightarrow \begin{pmatrix} G(F(u)), \\ G(F(v)) \end{pmatrix} \in E_3$.

∴ isomorphism from G_1 to G_3.

$(u,v) \in E_1 \Leftrightarrow (G(F(u)), G(F(v))) \in E_3$.

2 graphs are isomorphic if their complement are isomorphic

- Regular : every vertex has the same degree.
- 2 regular graphs are disjoint cycles.

as each vertex is similar to itself

- every graph has identity automorphism.
- max # of automorphism $= n!$
(# bijective functions possible)

G $\qquad F_1$. $\qquad F_2$

- Inverse of an automorphism is also an automorphism.

$(u,v) \in E \Leftrightarrow (F(u), F(v)) \in G$

automorphism → preserve structural adjacency
and non adjacency.
-reflexive, symmetric, transitive

$f(x) = y$

$g(f(x)) = z$

- It partitions the vertex set

→ Binary relation b/w vertices based on the existence of an automorphism between any pair is an equivalence relation.

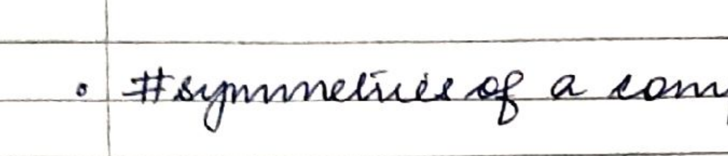
- #symmetries of a complete graph = $n!$

automorphism of a graph =
automorphisms of its complement

complete graph, cycle → 1 equivalence class.

↓

all $n!$ symmetries



any path has 2 automorphisms
equivalence class = $\lceil n/2 \rceil$

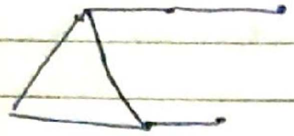
→ cycle

$2n$ = # of automorphisms.

1 = # equivalence classes.

↳ vertex transitive graph (degrees are always equal) $\left[K_n, \overline{K_n}, C_n, \overline{C_n}, Q_n, \overline{Q_n} \right]$

Rigid graph → every vertex is unique (n equivalence classes)



→ rigid graph.

no 2 vertices are identical

- 2 regular graphs cannot be rigid

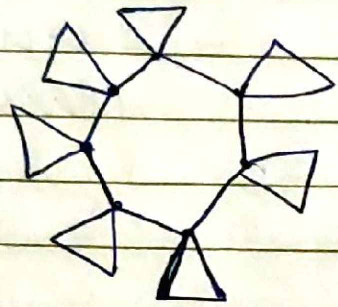
---

19 Aug missed (Vanshika)

20/8

- all Kneser graphs are vertex transitive.

Ex:



2 ~~a~~ equivalence classes.

- 2 sets of isomorphism

P_4, K_3 transitivity

- Not vertex transitive.
Not edge "
- It is triangle transitive.
~~automo~~

- Complete graph → any bijective mapping would work.

* **Decompositions** → identical / similar / arbitrary

- partitioning of edge set of a graph

- any graph can be broken into a complete graphs

$d_G(v)$ → degree of a vertex of a graph G

$$d_G(v) = \sum_{i=1}^{k} d_{G_i}(v)$$

- Impossible to decompose K_6 into triangles (K_3)
  $d(v) = 5$
  ↳ can't be broken into sum of 0's + 2's.

-

- Petersen graph is edge, vertex and P_4, P_3 transitive (find proof)
  → count #cycles in petersen graph.

- decomposing a complete graph into 2 isomorphic graph → self complementry graph.
  ↳ # edges must be even.
  ↳ n or n-1 must be a multiple of 4.
  ie $n(n-1) \% 4 == 0$.

not cycle of < 5 .

· Peterson graph does not have a ~~cycle~~ triangle.

n=5 k=2 .

~~edge~~ ~~but cannot find~~

diameter = 2

- does not have a 4 cycle.

↳ requires a vertex to have 2 common neighbors

but ~~to~~ not possible in peterson graph.

~~Peter~~ · has exactly 5 12 cycles.

6 cycle in peterson ~~cycle~~ graph is induced.

- remove 6 cycles & the cross connections we will be left with 

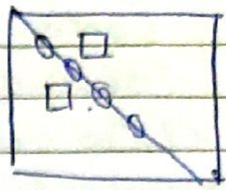
there are 10 such graphs ⇒ there are 10 6 cycles.

22/8 **Representing graphs on computers**

1) → adjacency matrix [n×n matrix] - for simple G → diag = 0
- for undirected G → symmetric

2) → adjacency list

3) → incidence matrix



2) → array of linked list

→ neighbors to the vertex

vertices

3) → matrix of both edges & vertices

$$\text{vertices}\left\{\begin{array}{c|ccc|}
 & e_1 & e_2 \cdots & e_m \\
v_1 & & & \\
\vdots & & & \\
v_n & & &
\end{array}\right.$$

edges

1-0 for undirected graph

1's in a row = degree of vertex

total 1's = total degree

= 2(# edges)

$$\sum_{v \in V} d(v) = 2|E|$$

(first theorem of graph theory)

(handshaking lemma)

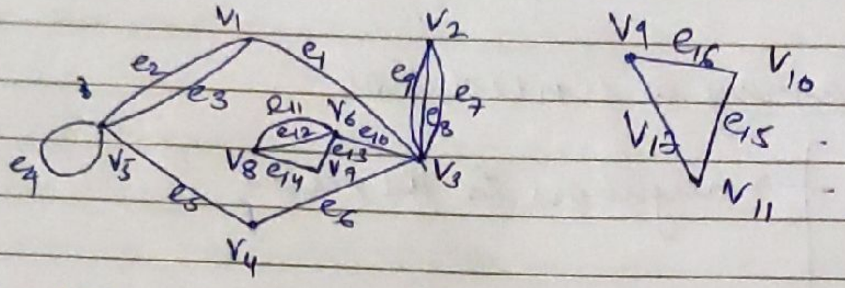
*Walk : An alternating sequence of vertices and edges beginning at a vertex and ending at a vertex such that for each edge in the sequence, the vertices appearing immediatly before + after it are its 2 end points

Length of a walk : number of occurrence of edges in a sequence (including duplicates)

Types :

[odd walk + even walk] } mutually exclusive and exhaustive
[closed walk + open walk]

1st & last vertex of the sequence is the same



· walks cannot cross components

Ex: $v_9 e_{16} v_{10} e_{16} v_9 e_{16} v_{10} e_{15} v_{11}$
length = 4
open walk → end points are distinct

· for simple graphs (no need to write edges) as there are no self loops + parallel edges.

*Trail : A walk w/o repeated edges. (allows reuse of vertices)
Ex: v_3 e_6 v_4 e_4 v_5 e_2 v_1 e_1 v_3.

Types: { [open and closed] / [odd + even] } (cycle is an ex of trail)

*Path : A walk/trail where vertices are not repeated (+ ∴ edges are also not repeated)
- always open

- paths + cycles can be graphs / subgraphs.

{peterson graph, cycles, induced subgraph}

- every shortest path is induced.
- every induced paths need not be shortest.
- shortest cycle must be induced
  vice versa may not be true.

26th Aug missed - something related to adjacency matrix

27th/8 ~~Planes~~, Decomposition, trails.

* EULERIAN TRAILS
  - a trail wherein no edge is left uncovered.
  ⇒ closed trail.
  ~~odd~~ ⇒ spanning (w.r.t edges)

in a trail { degree of end point of a closed trail = even
degree of an internal point = even.

for euler trail:
[Internal point] degree of vertex in graph = degree of vertex in trail

~~d~~ in open trail, end points have odd degree

- the final residual degree of vertices must be even.

- Every odd degree vertex must be in atleast one of the decompositions.

euler ⇒ - Graph must be having every vertex w/i even degree
⇒

- #every vertex w/i ~~odd~~ odd degree must be even (first theorem of graph theory)

→ ~~componts~~ Only 1 non trivial compononent

* Necessary + sufficient conditions (G_c)

- every vertex has even degree
- Only 1 non trivial component
  concurrent w/i
  atleast 1 edge

∘ either you can't complete euler trail or can be completed by starting from any vertex.

algorithm
→ greedy
→ string of vertices, edges motlies
→ string of residue edges

paste into this

→ polynomial time

⇒ can we decompose the graph into a single open trial?

[every open trail has 2 vertices of odd deg]

∴ to decompose a graph w/i 2K vertices having odd deg needs $\frac{2K}{2} = K$ decompositions.

OR K rounds
OR K trails (atlease)