Connectivity

Walk:

Let v and w be two vertices in a graph.
Walk between v & w in the graph is a finite alternate sequence $v = v_0, e_1, v_1, e_2, v_2, \ldots e_n, v_n = w$ of vertices and edges of the graph s.t. an edge e_i in the sequence joins vertex v_{i-1} and vertex v_i. The vertices and edges in a walk need not be distinct.

Two walks $v_0, e_1, v_1 \ldots e_n, v_n$ and $u_0, f_1, u_1, \ldots f_m, u_m$ are equal if $n = m$ $v_i = u_i$ and $e_i = f_i$ for $0 \leq i \leq n$.

Two walks are said to be different if they are not equal

Length of a walk $\rightarrow$ Number of edges in the walk

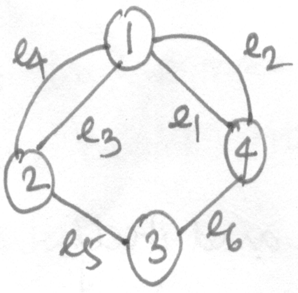
For a simple graph, the sequence defining a walk need not be listed i.e. $v = v_0, e_1, v_1, \ldots e_n, v_n = v_0 - v_1 - v_2 - \cdots v_n$

Trail: A walk in which no edge is repeated

The walk $v = v_0, e_1, v_1, \ldots v_{n-1}, e_n, v_n = w$ in which vertices v_i $(0 < i < n)$ are all distinct is a path between v & w

The $(n-1)$ vertices v_i $(0 < i < n)$ are called intermediate vertices.

Every path is a trail.



The sequence $2, e_4, 1, e_1, 4, e_2, 1, e_1, 4$ is a walk b/w Vertex 2 & vertex 4

" " $2, e_3, 1, e_1, 4, e_2, 1, e_4, 2, e_5, 3$ is a trail b/w 2 & 3

" " $2, e_5, 3, e_6, 4, e_1, 1$ is a path b/w 2 & 1

* Theorem 1: Every walk in a graph b/w v & w contains a path b/w v and w ~~and~~

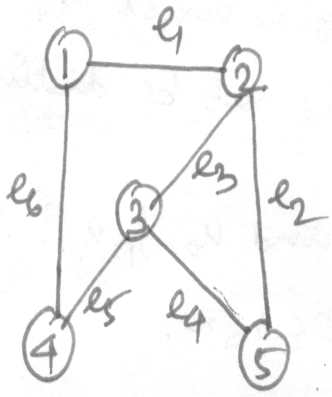
* Theorem 2: If A is an adjacency matrix of a simple graph $G = (V, E)$ where $V = \{1, 2, \ldots n\}$ then, the $(i-j)$th entry in the kth power of A is the no. of different walks of length k b/w vertices i & j

In particular $(i-j)$ diagonal entry in A^2 is the degree i for-

Ex:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad A^2 = \begin{bmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

Graph:



$$A^4 = \begin{bmatrix} 9 & 3 & 11 & 1 & 6 \\ 3 & 15 & 7 & 11 & 8 \\ 11 & 6 & 15 & 3 & 8 \\ 1 & 11 & 3 & 9 & 6 \\ 6 & 8 & 8 & 6 & 8 \end{bmatrix}$$

Degree vertices ~~are~~ of nodes 1, 2, 3, 4, 5 are : $\underbrace{2, 3, 3, 2, 2}_{\text{diagonal entries of } A^2}$

(1,5) entry in A^4 is 6, indicating there a 6 different walks of length 4 between 1 & 5 -

1-4-1-2-5

1-2-1-2-5

1-4-3-2-5

1-2-5-2-5

1-2-3-2-5

1-2-5-3-5

Closed walk $\longrightarrow$ walk b/w a vertex and itself

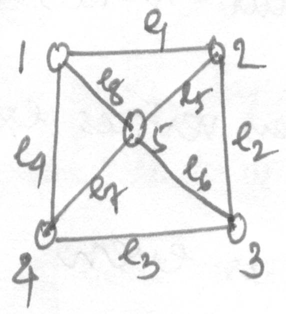
Circuit $\longrightarrow$ A closed walk in which no edge repeats

v, e_1, w, e_2, v is a cycle

v, e_1, w, e_1, v with no repeated intermediate ~~of~~ vertices is not a cycle since it is not a circuit

* The subgraph C of a simple graph G is a cycle in G if and only C is a cyclic graph

Simple graph G, any cycle consisting of K-vertices is a K-cycle
; K is odd ⇒ odd cycle and if K is even, ⇒ even cycles



Closed walk → 1, e_1, 2, e_5, 5, e_6, 3, e_3, 4, e_7, 5, e_8, 1 → eg circuit

1–2–3–4–1 is an even cycle

* If there is an odd cycle in a simple ~~graph~~ graph G, with 3 or more vertices then G is not bipartite
or a simple graph ~~is~~ with three or more vertices is Bipartite if it has no odd cycles.

Connected Graphs

A pair of vertices in a graph is a connected pair if there is a path b/w them.
G is a connected graph if every pair of vertices in G is a connected pair.

A connected subgraph H of graph G is a component of G if H = H' whenever H' is a connected subgraph (of G) that contains H or a component of a graph is a maximal connected subgraph.

A graph is connected if ~~the~~ and only if the no. of components is one.

* **Theorem**:

A graph G is disconnected if and only if its vertex V can be partitioned into two ~~or~~ non-empty ~~sets~~ disjoint subsets V_1 and V_2 such that there exists no edge in G where one vertex is in V_1 and other vertex is in V_2

Proof: Suppose such a partitioning exists. Consider two arbitary vertices a and b s.t. $a \in V_1$ and $b \in V_2$.
No path can exist b/w a & b ~~els~~ otherwise, there would be atleast one edge whose one end vertex would be in V_1 and other in V_2. Hence if a partition exists, G is not connected

Theorem:

If a graph (connected or disconnected) has exactly two vertices of odd degree, there must be a path joining these two vertices.

Proof: Let G be a graph with all even vertices except v_1 and v_2 which are odd.

$\because$ No. of vertices of odd degree is even

$\therefore$ For every component of a disconnected graph, no graph can have odd no. of ~~disconnected graph~~ odd vertices

Therefore, in graph G, v_1 and v_2 belong to same component and thus must have a path between them.

Theorem:

A simple graph with n-vertices and k-components have atmost $(n-k)(n-k+1)/2$ edges

Proof:

Let the no. of vertices in each of k-components of a graph G be $n_1, n_2 \ldots n_k$

Thus, $n_1 + n_2 + \cdots + n_k = n$

$n_i \geq 1$

$$\sum_{i}^{k} n_i^2 \leq n^2 - (k-1)(2n-k)$$

Now, max. no. of edges in the i-th component of G is $^{n_i}C_2 = \frac{1}{2} n_i (n_i - 1)$

$\therefore$ max. no. of edges in G:

$$\frac{1}{2} \sum_{i=1}^{k} (n_i - 1) n_i = \frac{1}{2} \sum_{i=1}^{k} n_i^2 - \frac{n}{2} \quad \left(\because \sum_{i=1}^{k} n_i = n\right)$$

$$\leq \left[n^2 - (k-1)(2n-k)\right] - \frac{n}{2}$$

$$= \frac{1}{2} \cdot (n-k)(n-k+1)$$

…x Deletion and Edge Deletion

…ven, a graph G and a vertex $v \in V(G)$:

$G-v$ denotes graph by removing v and all edges incident with v from G

If S is a set of vertices, then $G-S$ denotes the graph obtained by removing each vertex of S and all associated incident edges.

If e is an edge of G, then $G-e$ is the graph obtained by removing only the edge e (its end vertices stay).

If T is the set of edges then $G-T$ is the graph obtained by deleting each edge of T from G.

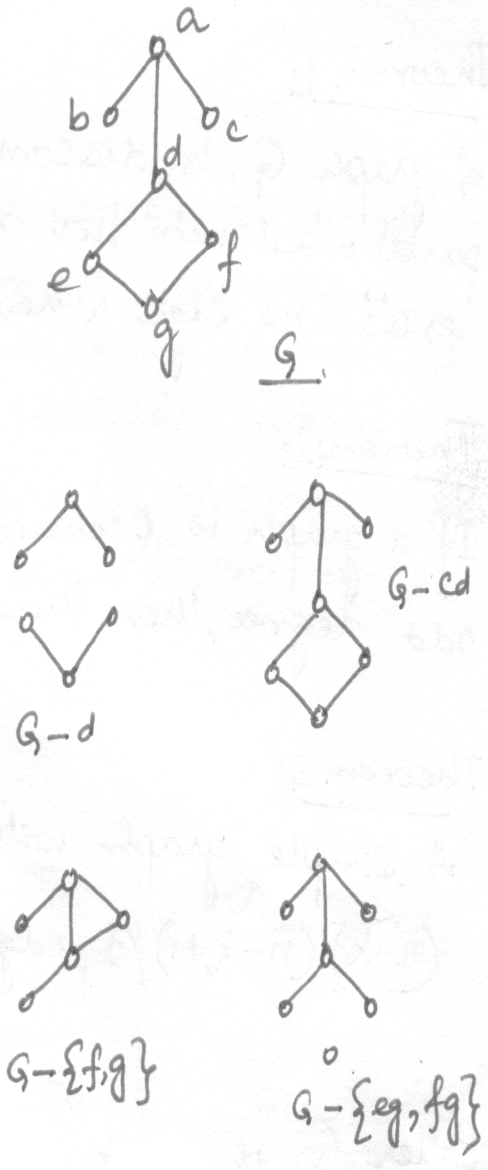
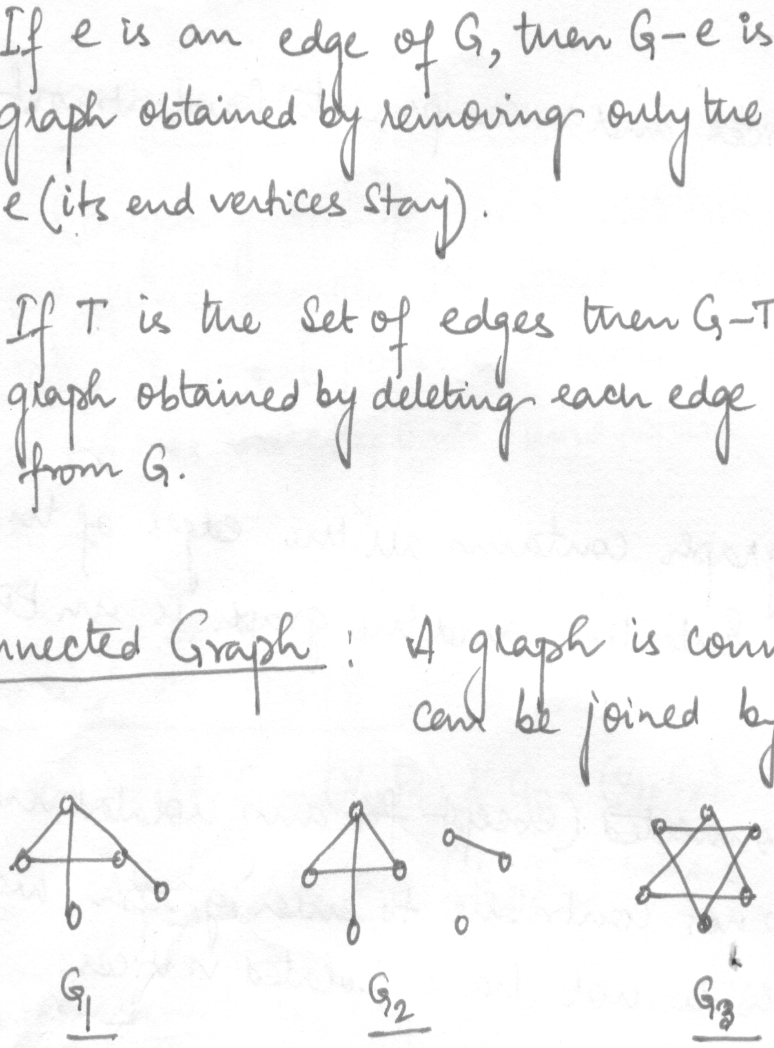


Fig 1

Connected Graph: A graph is connected if every pair of vertices can be joined by a path



G_1 is Connected

G_2 & G_3 are disconnected

Each maximal connected piece of a graph is called Connected Component.

Cutvertex: If deletion of a vertex v from G causes the no. of components to increase then v is cut vertex

Ex: Vertex d is cut vertex / vertex c is not a cut vertex

Bridge: An edge e in G is said to be a bridge if the graph $G-e$ has more components than G.

ex: ab is the bridge in G

Vertex Cutset: A proper subset S of vertices is called Vertex cut set (cut set) if the graph $G-S$ is disconnected

Complete graph: Every vertex is adjacent to every other vertex

* Complete graph do not have cutsets

<u>Theorem 1:</u>

A graph G. is disconnected if and only if its vertex set V can partitioned into two non-empty, disjoint subsets V_1 & V_2 s.t. there exists no edge where one vertex is in V_1 and other in V_2

<u>Theorem 2:</u>

If a graph is (connected or disconnected) has exactly two vertices of odd degree, then there must be a path joining these two vertices

<u>Theorem 3:</u>

A simple graph with n-vertices and k-components have atmost $(n-k)(n-k+1)/2$ edges

<u>Euler Graph</u>

If some closed walk in a graph contains all the edges of the graph, then the walk is an Euler line and the graph is an Euler graph.

→ Euler graph is always connected (except for any isolated vertices)
∵ isolated vertices do not contribute to euler graph, we assume Euler graph do not have isolated vertices.

<u>Theorem 4:</u>

A given connected graph G is an Euler graph if and only if all vertices of G are of even degree.

<u>Proof:</u>

Suppose G is an Euler graph ⇒ G contain an Euler line (walk) while tracing the walk, every time the walk meets a vertex 'v', it goes through two new edges incident on v — with one it entered v and with the other exited v. It is also true for the terminal vertices through one it entered and through the ~~last~~ other it exited.

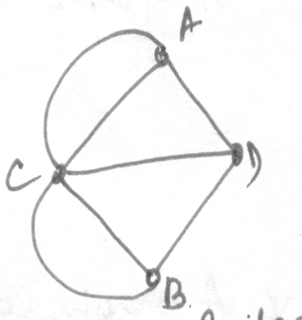
Thus, if G is an Euler graph, degree of each vertex is even.

...ficiency:

...ume all vertices are of even degree.
...nstruct a walk starting at an arbitary vertex v and going through the edges of G s.t. no edge is traced more than once. We continue tracing as far as possible. Since every vertex is of even degree, we can exit from every vertex we enter. Tracing cannot stop at any vertex but v. Since v is also of even degree we shall eventually reach v when tracing comes to an end.

~~Koni~~ Königsberg Bridge Problem

In The Königsberg Bridge graph not all its vertices are of even degree. Hence it is not an Euler graph. ~~Thus,~~ Thus, it is not possible to walk over all the seven bridges exactly once and return to the starting point.



Königsber Bridge Graph

Operations on graph

1. **Union**: If $G_1 = (V_1, E_1)$ & $G_2 = (V_2, E_2)$
   $G_3 = G_1 \cup G_2 = (V_3, E_3)$ where $V_3 = V_1 \cup V_2$ and $E_3 = E_1 \cup E_2$

2. **Intersection**: $G_4 = G_1 \cap G_2$ where $V_4 = V_1 \cap V_2$ and $E_4 = E_1 \cap E_2$
   edges belong to ~~either~~ both G_1 & G_2

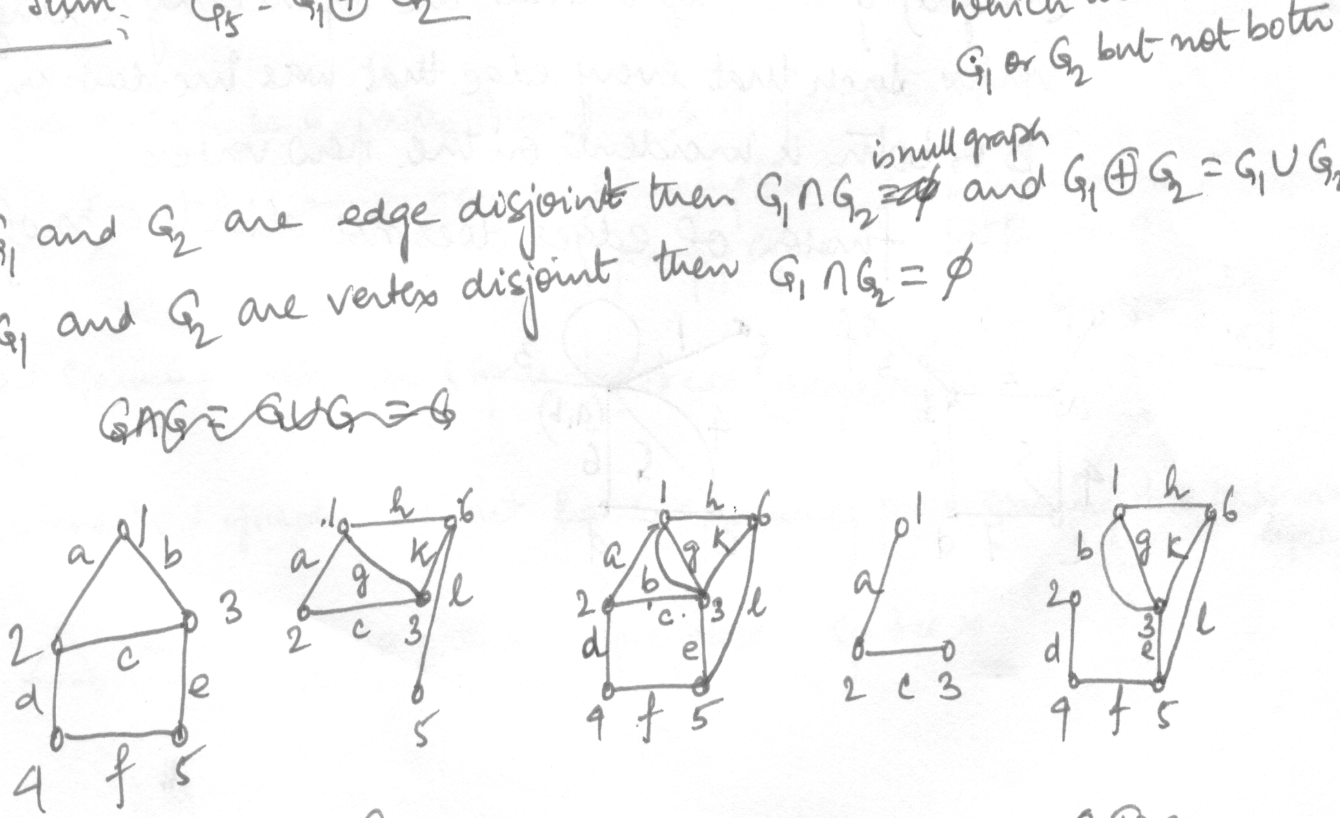
3. **Rising Sum**: $G_5 = G_1 \oplus G_2$ where $V_5 = V_1 \cup V_2$ and E_3 contain edges which are either in G_1 or G_2 but not both

If G_1 and G_2 are edge disjoint then $G_1 \cap G_2 = \phi$ is null graph and $G_1 \oplus G_2 = G_1 \cup G_2$

If G_1 and G_2 are vertex disjoint then $G_1 \cap G_2 = \phi$

~~$G \cap G = G \cup G = G$~~

Ex:



G_1 G_2 $G_1 \cup G_2$ $G_1 \cap G_2$ $G_1 \oplus G_2$

$G \cup G = G \cap G = G$

$G \oplus G$ = null graph

If g is a subgraph of G, then $G \oplus g$ is the subgraph that remain after all edges of g have been removed from G.

$\therefore$ $G \oplus g = G - g$ and Known as complement of G.

<u>Decomposition:</u>

A graph G is decomposed into two subgraphs g_1 and g_2 if

$g_1 \cup g_2 = G$.

$g_1 \cap g_2$ = null graph

* A graph containing m edges $\{e_1, e_2, \ldots e_m\}$ can be decompose in $2^{m-1} - 1$ different ways into pairs of subgraphs g_1, g_2

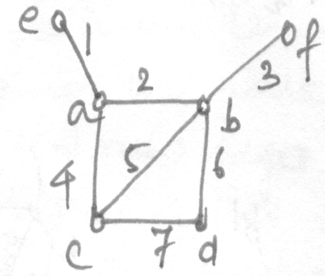
* G can be decomposed into more than two subgraphs — that are pairwise edge disjoint and collectively include every edge in G

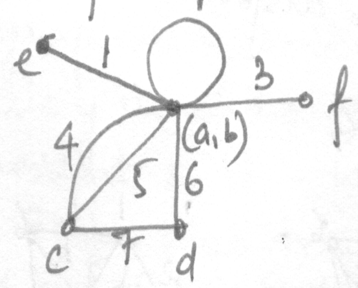
<u>Deletion</u>: discussed already

<u>Fusion</u>: A pair of vertices a, b in a graph is said to fused (merged) if the two vertices are replaced by a single new vertex such that every edge that was incident on a or b or both is incident on the new vertex.

The fusion of edges does not alter the no. of edges

<u>Ex.</u>



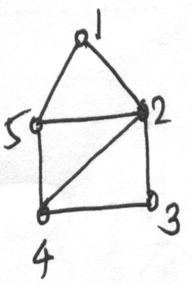


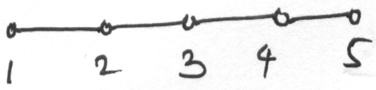
Spanning Cycles/Trees/Path

Spanning Path → Given a graph G, a spanning Path is a subgraph H s.t $V(H)=V(G)$ and H is a path

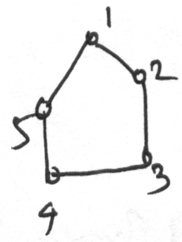
Spanning Cycle → A cycle C s.t. ~~V(G)~~ $V(C)=V(G)$

" Tree → A tree T s.t. $V(T)=V(G)$



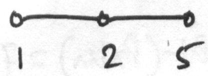


↳ Subgraph & a path

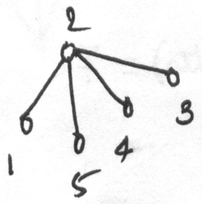


C_5

Spanning cycle

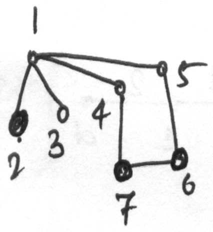


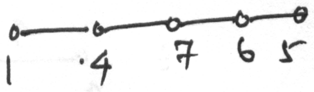
× Spanning path



Tree (connected & acyclic) and contains all $V(G)$

Spanning tree



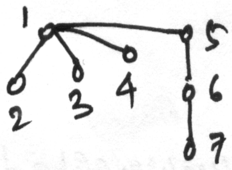


path ✓

Spanning ×

No Spanning path exists.

No spanning cycle exist



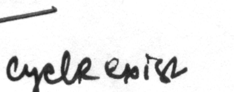
Spanning tree ✓

Conclusion: For a graph, spanning path and spanning cycles may not exist, but spanning tree exists

* If the Graph is a path, then it has a spanning Path

ex. P_4  Spanning Path

Spanning cycle



Spanning tree

all spanning paths are spanning trees (acyclic)

* Disconnected graphs do not have spanning path, spanning tree or spanning cycle

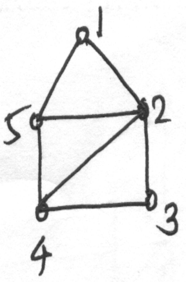
sp. path × sp. cycle × sp. tree ×

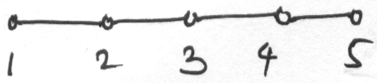
Spanning Cycles/Trees/Path

Spanning Path → Given a graph G, a spanning Path is a Subgraph H s.t. V(H) = V(G) and H is a path

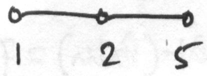
Spanning Cycle → A cycle C s.t. ~~V(G)~~ V(C) = V(G)

" Tree → A tree T s.t. V(T) = V(G)

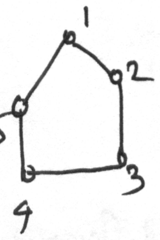




↳ Subgraph & a path

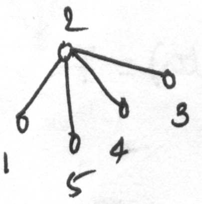


× Spanning path



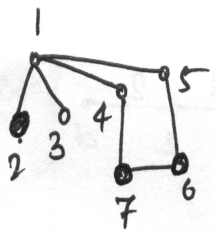
C_5

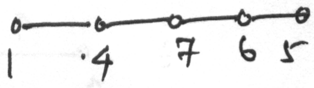
Spanning cycle



Tree (connected & acyclic) and contains all V(G)

Spanning tree

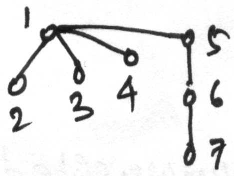




path ✓
Spanning ×

NO Spanning path exists.

NO spanning cycle exist

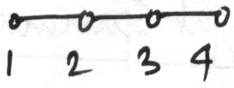


Spanning tree ✓

Conclusion: For a graph, spanning path and spanning cycles may not exist, but spanning tree exists

* If the Graph is a path, then it has a spanning Path

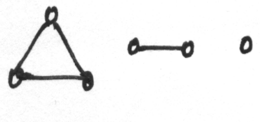
ex. P_4  Spanning Path ✓
Spanning cycle ×



Spanning tree

all spanning paths are spanning trees (acyclic)

* Disconnected graphs do not have spanning path, spanning tree or spanning cycle



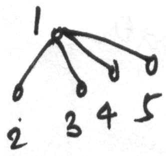
sp. path × sp. cycle × sp. tree ×

Obs: ① If G is connected, there exists a spanning tree

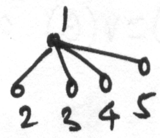
② If G is connected, spanning path, spanning cycle may or mayn...

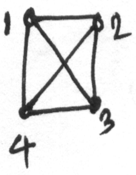
③ If G is connected, we may find atleast one spanning tree

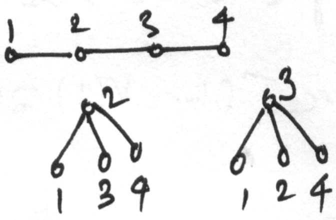
ex.



Sp. tree

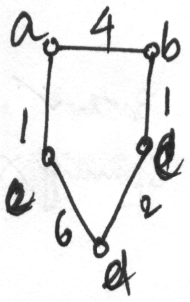


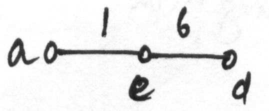




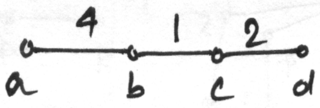
sp. tree for $K_4 = \underline{16}$

Weighted Graphs

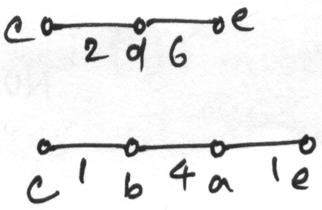




Weight (Path) = 7



weight (Path) = 7

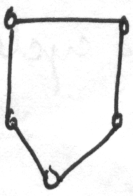


weight( ) = 8
weight( ) = 6

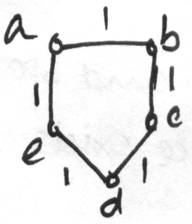
} Prefer min-wt. path

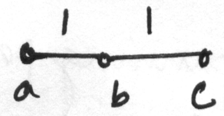
Shortest path

undirected-unweighted

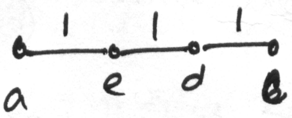


≡



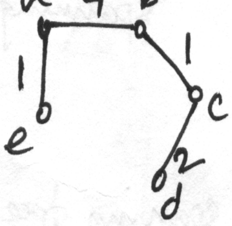


wt( ) = 2 → Shortest path



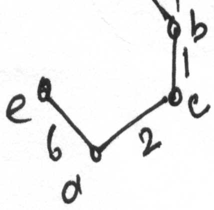
wt. = 3 ~~→~~

Min-weight Spanning tree & Min wt. spanning cycle



Spanning (connected & acyclic)

wt = 8 → Minimum



Spanning (connected & acyclic)

wt = 13

Hamiltonian | Eulerian

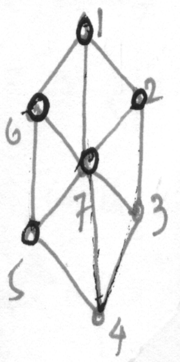
Hamilton Graph:

↳ Hamiltonian Path (A Path containing all vertices – Spanning Path)

each vertices appears exactly once

↳ " Cycle (A cycle " " " – Spanning cycle)

A Gh is hamiltonian if the Gr. has Hamiltonian Path, i.e. ∃ Sp. Path
if the Gr " " cycle, i.e. ∃ Sp. cycle

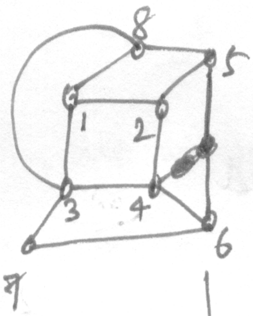


W_7 o—o—o—o—o—o—o
1 2 3 4 5 6 7 (Hamiltonian Path)

Does it have hamiltonian cycle – No

has H path but no H-cycle ✓

Ex.



2 5 6 3 4 7 1 2

$\underbrace{2\ 5\ 6\ 3\ 4\ 7\ 1}_{\text{H. Path}}$ 2

$\underbrace{2\ 5\ 6\ 3\ 4\ 7\ 1\ 2}_{\text{H-cycle}}$

ex.

If G has Hcycle ⇒ G has H-path
If G has Hpath ⇏ G has Hcyce

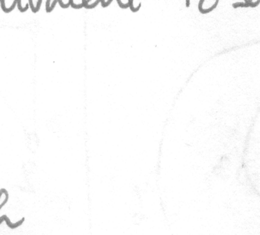
Applications of Hamiltonian ~~Pa~~ Graph:

Road Network / Campus Network

Say: there is a monitor node that needs to visit every node in the N/w collect information and come back.

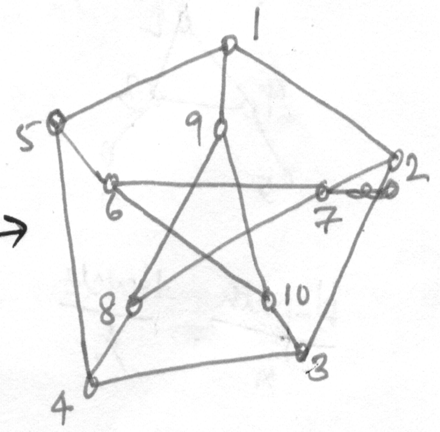
It is equivalent to saying whether there is a hamiltonian ~~Path~~ cycle

Petersson - Graph



H-path ✓ →

H-cycle ×



o—o—o—o—o
1 2 3 10

× Path satisfying property
↓
vertex repeatation

|| 3-Regular $(\underbrace{3,3,\ldots 3}_{10})$

Connected

Induced C_5

P_n Hpath ✓

Hcycle ✗

K_n - Hpath ✓

Hcycle ✓

Regular graph depends on graph structure. Ex: Peterson graph which is 3-regular has H-path but ~~not~~ no H-cycle

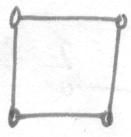
C_n – 2-regular (each vertex is of degree 2)

Hpath, Hcycle

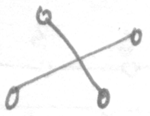
W_n – Hpath ✓

Hcycle ✗

G



G^c



| Disconnected

H-path ✓

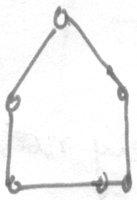
Hcycle ✓

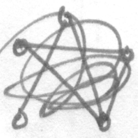
Hpath ✗

Hcycle ✗

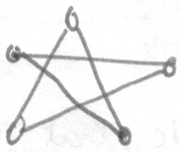
If G is Hamiltonian (Hpath/Hcycle) ⇒ G is connected

G





G^c

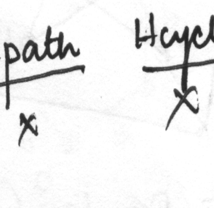


| Self complementary graph

Hcycle ✓

H-cycle ✓

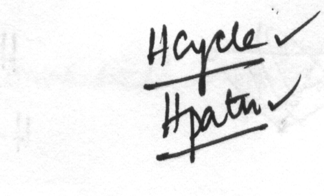
G



H-path ✗

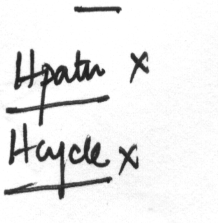
Hcycle ✗

G^c



Hcycle ✓

Hpath ✓



G

Hpath ✗

Hcycle ✗

G^c

Hpath ✗

Hcycle ✗