CS 60047 Autumn 2020

Advanced Graph Theory

Instructor

Bhargab B. Bhattacharya

Lecture #12, #13: 23 Sept. 2020

Indian Institute of Technology Kharagpur

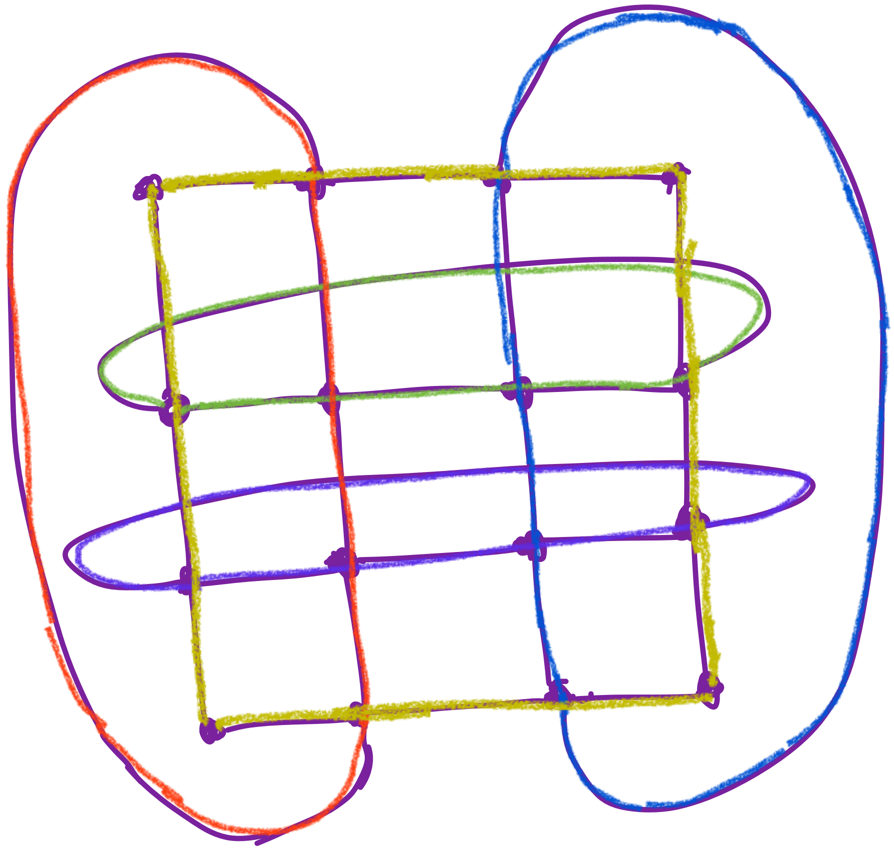
Computer Science and Engineering

Today’s Topics

Traversability

- Finding an Eulerian – Fleury’s Algorithm
- Chinese postman problem
- Hamiltonian graphs

Hierholzer's Algorithm: Find Eulerian Tour



- decomposed into five cycles.

Combine individual cycles to form a single cycle via *splicing*

Another approach: Fleury’s Algorithm

This algorithm finds an *Euler closed trail* or an *Euler open trail* in a connected graph

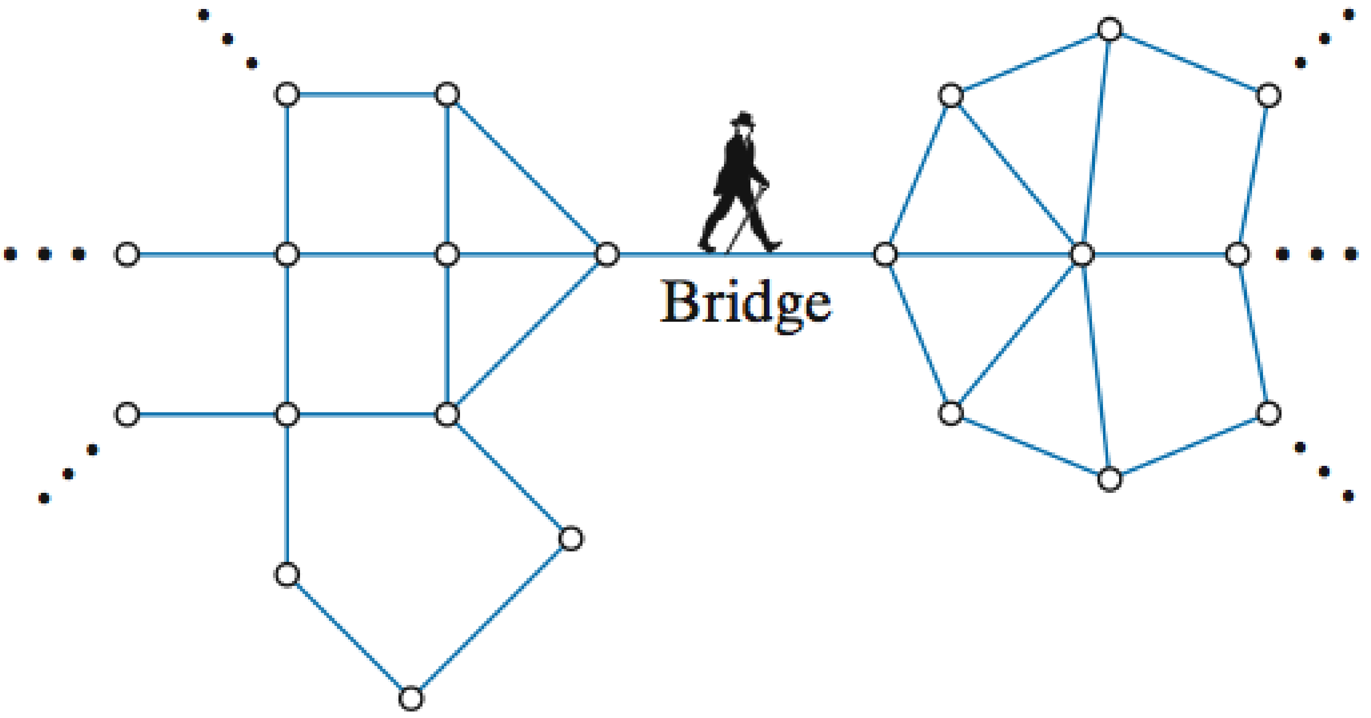
The key idea behind Fleury’s algorithm:

Keep walking but do not burn your bridges behind you

Courtesy: www.ms.uky.edu/~lee

Fleury’s Algorithm

A *bridge* is the only edge connecting two components



Fleury’s algorithm is based on a simple principle:

To find an Eulerian trail, *bridges are the last edges you want to cross*.

Fleury's algorithm for finding Eulerian closed or open trail (with two-odd degree vertices)

- Make sure that the graph is connected and either (i) has no odd vertices (closed trail) or (ii) has just two odd vertices (open trail)

- *Start:* Choose a starting vertex

  In Case-(i) this can be any vertex; in Case-(ii) it must be one of the two odd vertices

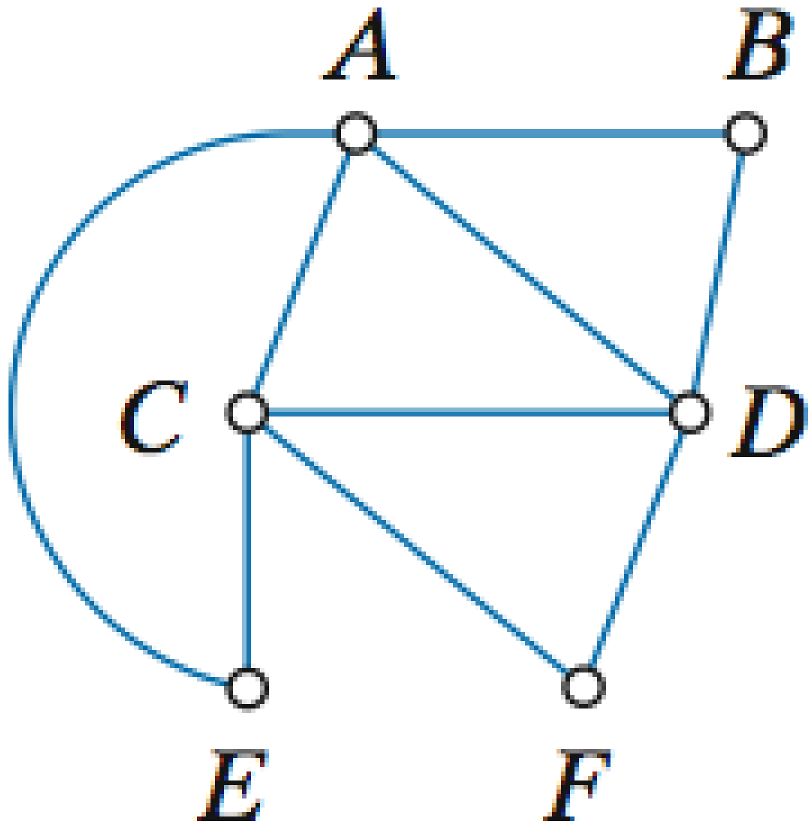
Next,

- **Intermediate steps:** At each step, whenever you have a choice, *do not* choose a bridge of the *yet-to-be-traveled* part of the graph. However, if you have no other choice, take it;

- **End:** When you cannot travel any more, the trail is complete.

In Case-(i) we reach the starting vertex; in Case-(ii) we finish at the other odd-degree vertex

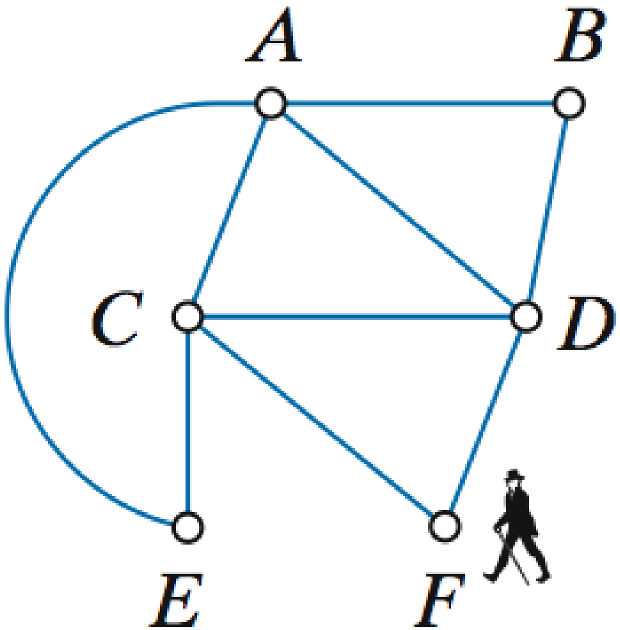
Example: Fleury's Algorithm



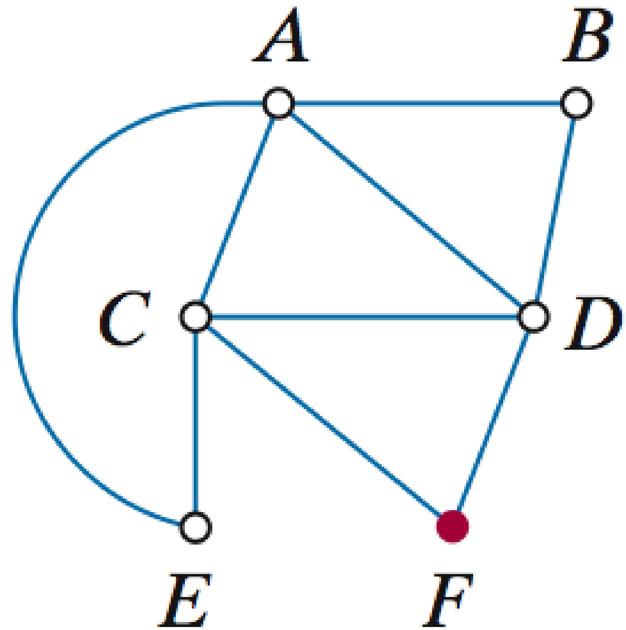
This is an even graph → closed trail exists

Example

Start: Since *G* is even, we can start from any vertex (say *F*). Keep two copies of *G* for convenience.



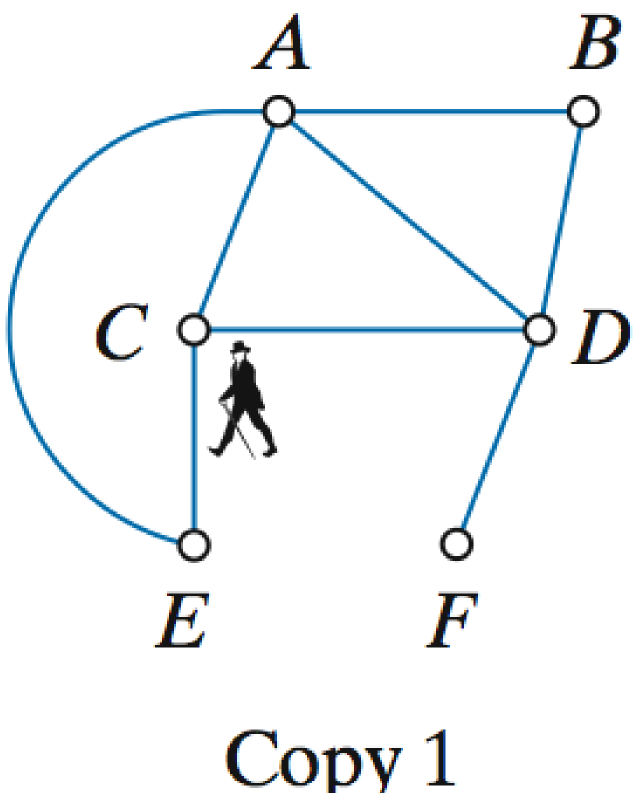
Copy 1

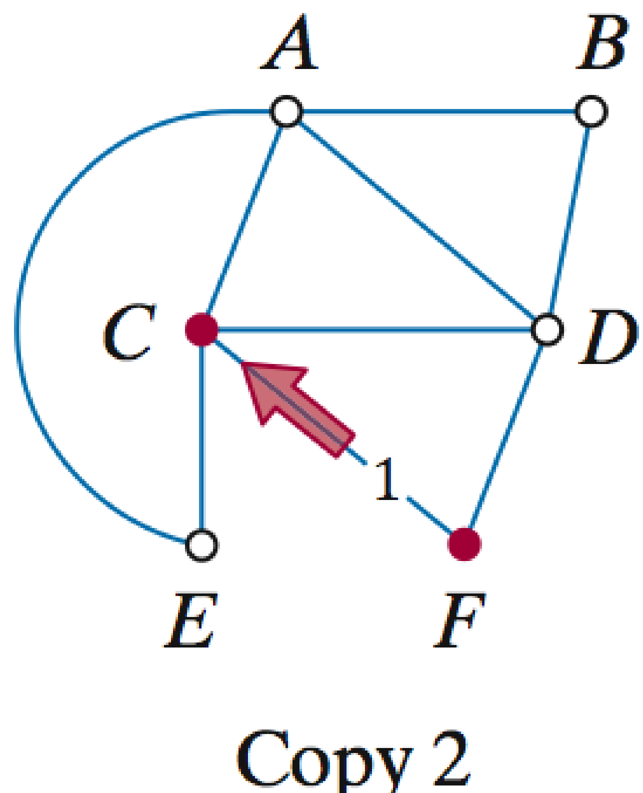


Copy 2

Example

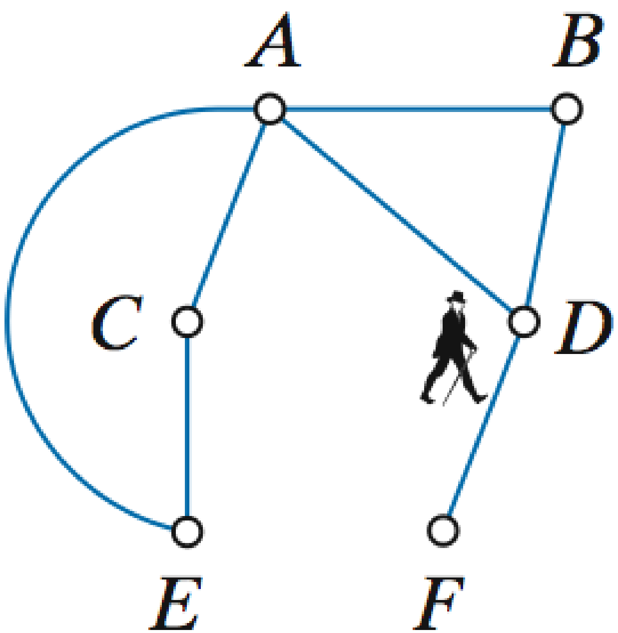
Step 1: Travel from *F* to *C* (*F* to *D* also possible); as we move, delete the corresponding edge (*F*,*C*) in Copy 1.



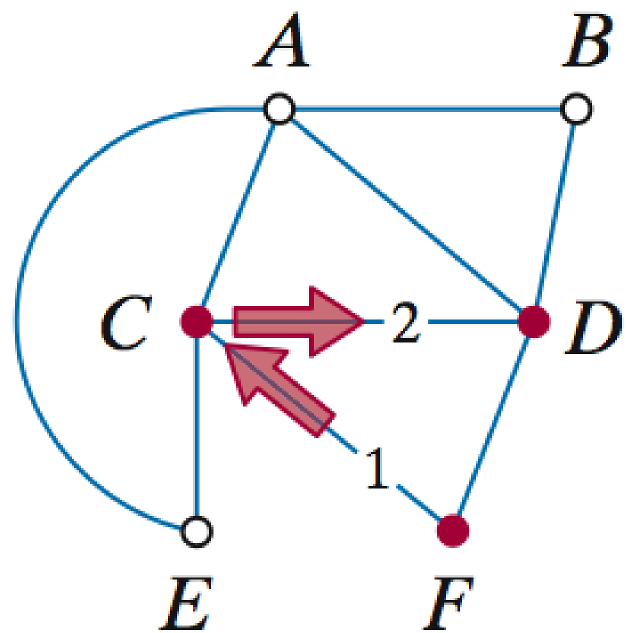


Example

Step 2: Travel from *C* to *D* (to *A* or to *E* also possible); delete the edge (*C*,*D*)



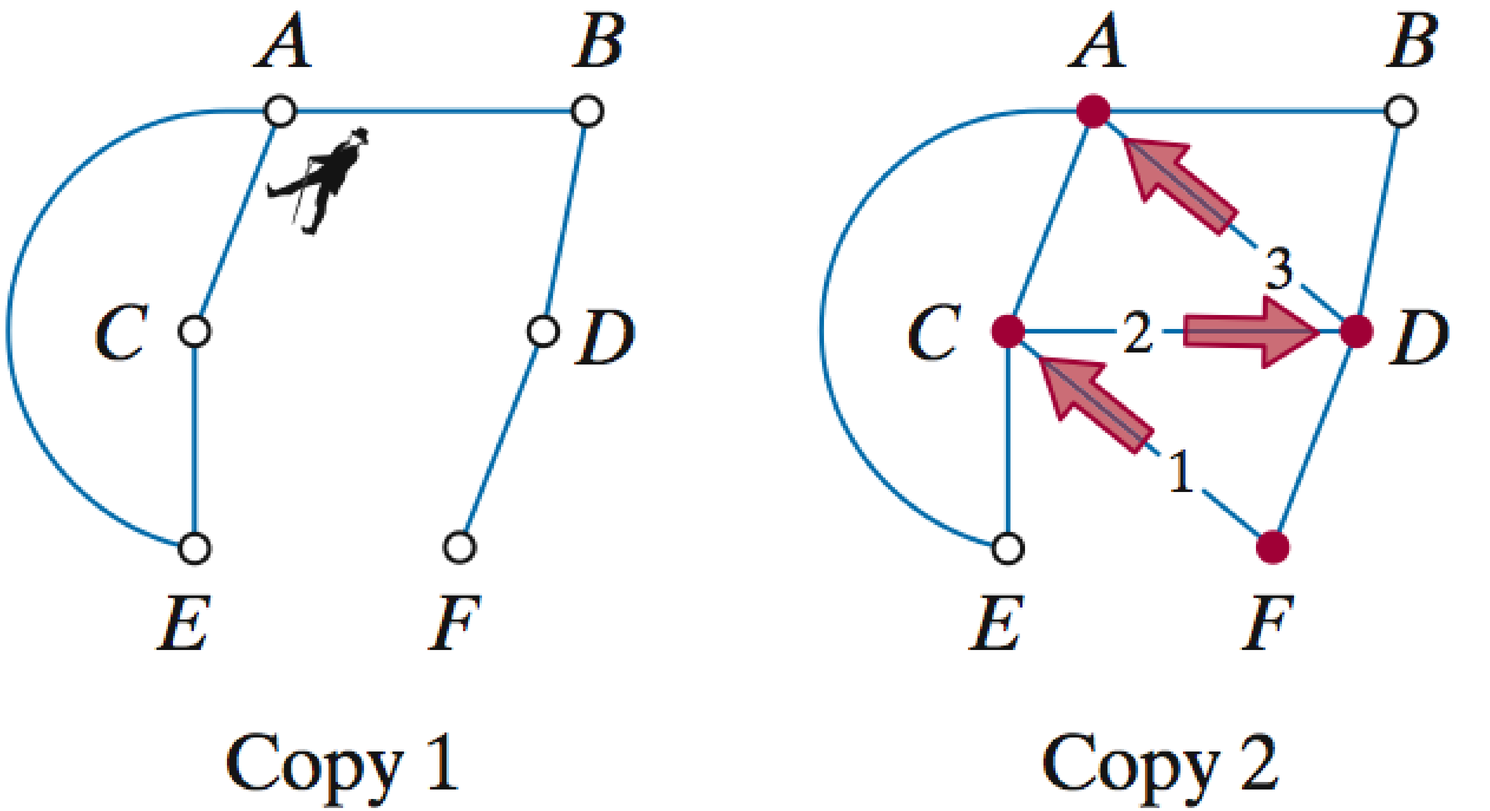
Copy 1



Copy 2

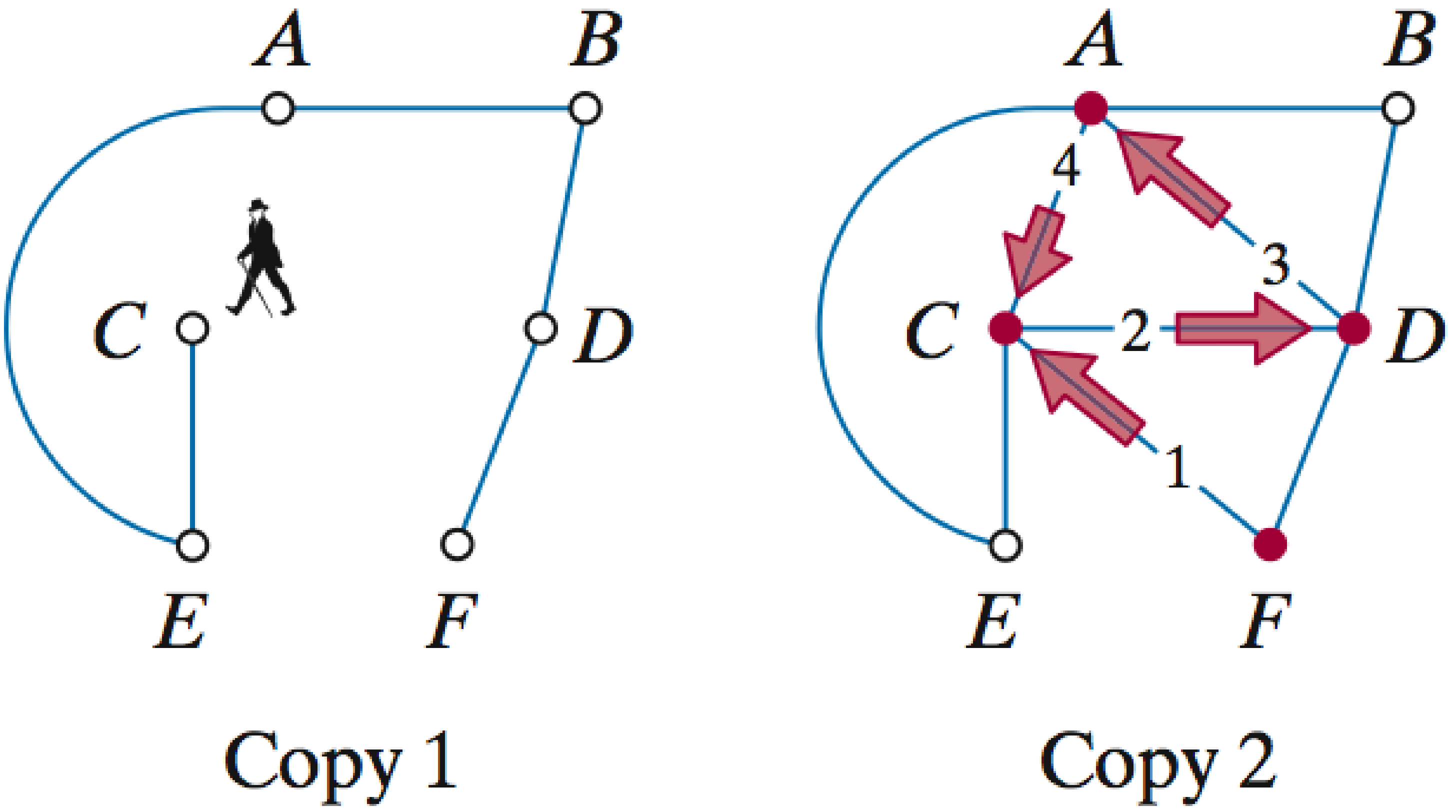
Example

Step 3: Travel from *D* to *A* (to *B* also possible, but not to *F; because DF* is a bridge!); delete (*D*,*A*)



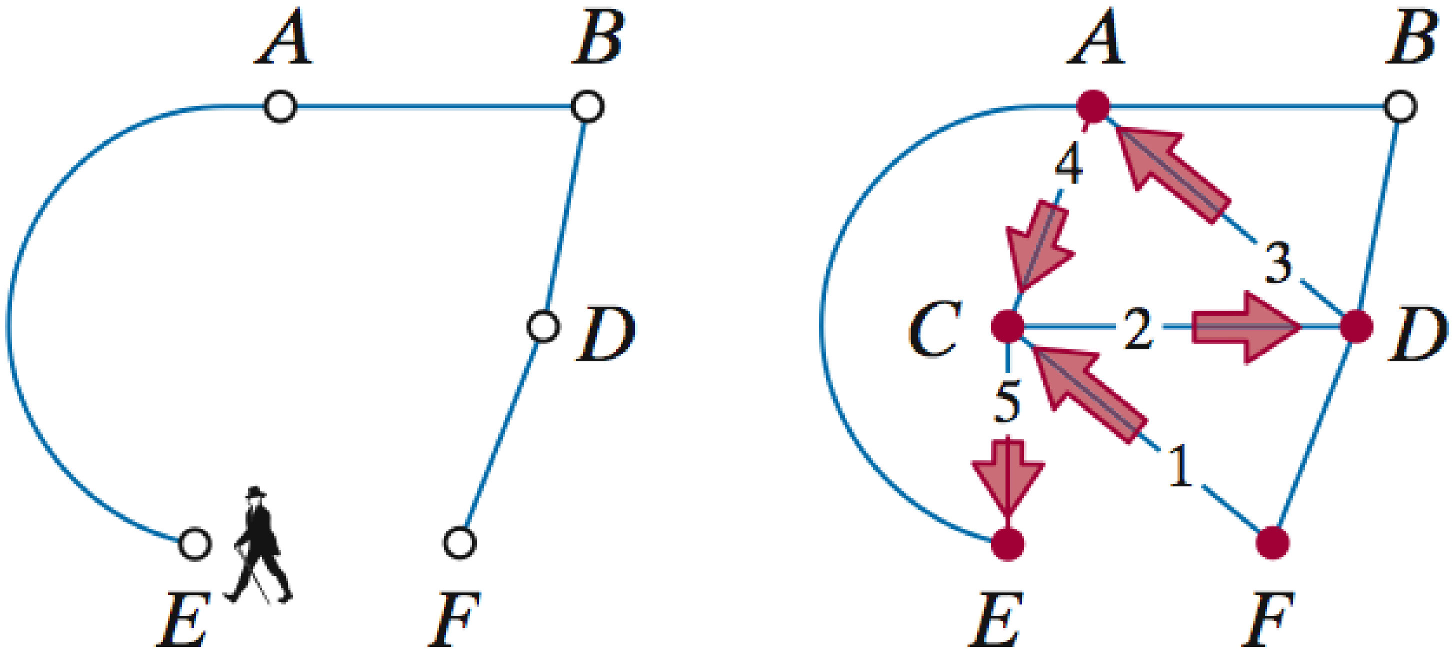
Example

Step 4: Travel from *A* to *C* (to *E* also possible, but not to *B; AB* is a bridge!); delete (*A,C*)



Example

Step 5: Travel from *C* to *E* (it is a bridge, but there is no choice!); delete (*C*,*E*)



Copy 1 Copy 2

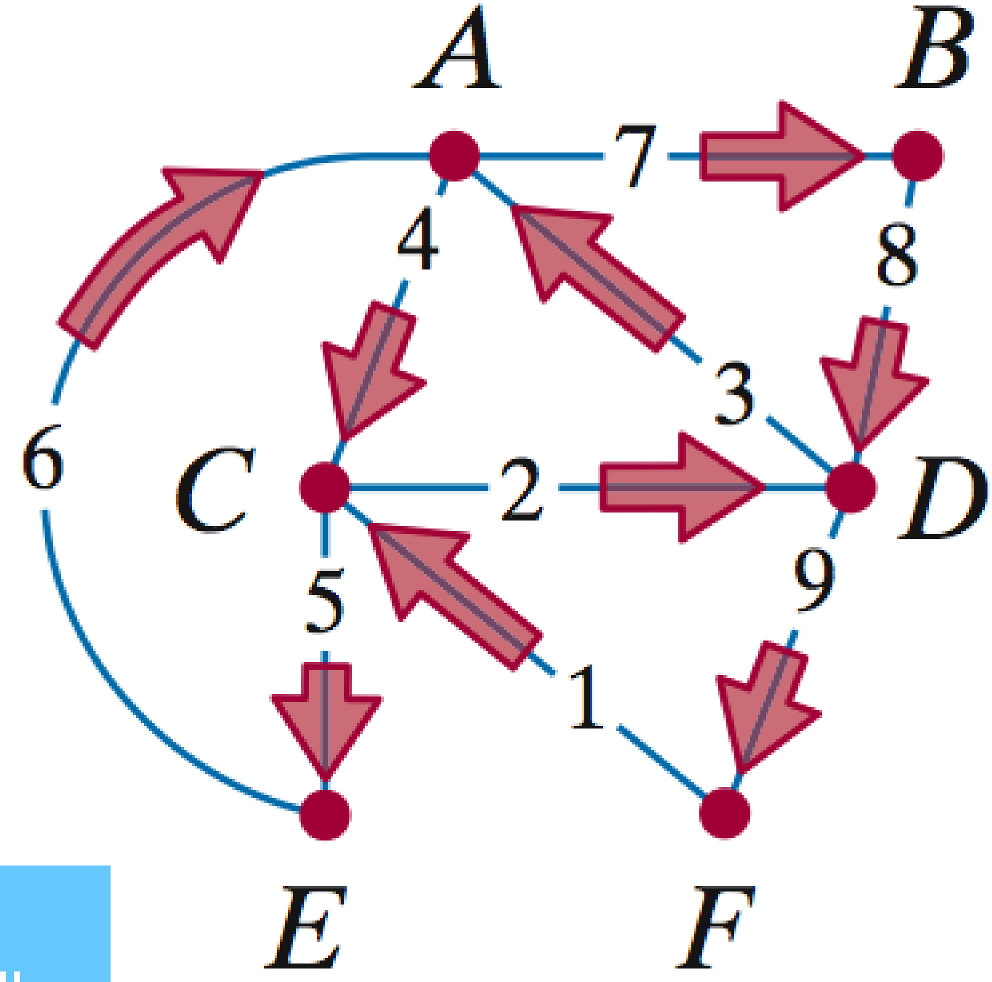
Example

time complexity: $O(|E|^2)$; later improved to $O(|E|\log^3|E|\ \log\log|E|)$

Steps 6, 7, 8, and 9:

Only one way to go at each step;

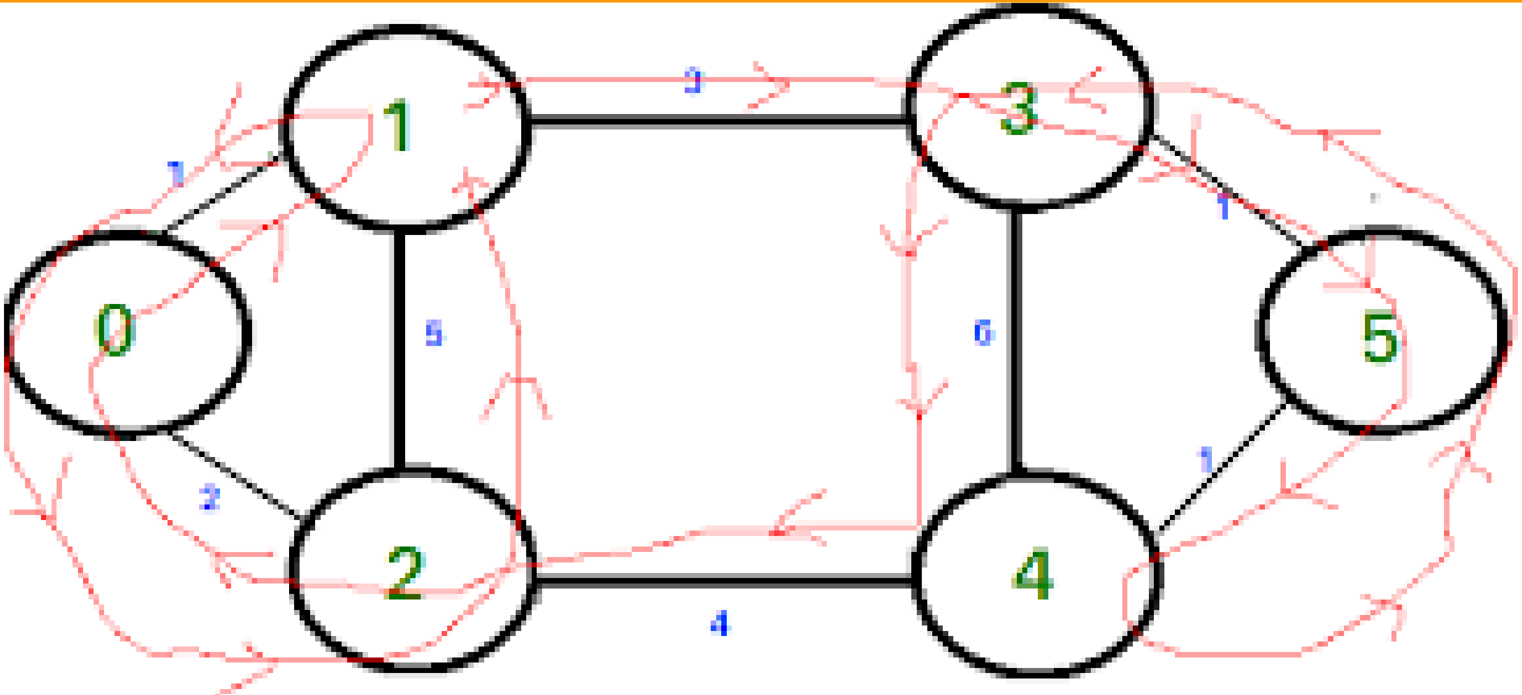
Found successfully the Eulerian closed trail



Homework:
Study Fleury's algorithm formally, with complexity analysis

Route Inspection Problem: A Variant of Eulerian

Chinese Postman Problem (CPP): edge repetition allowed; delivery of letters by a postman to houses

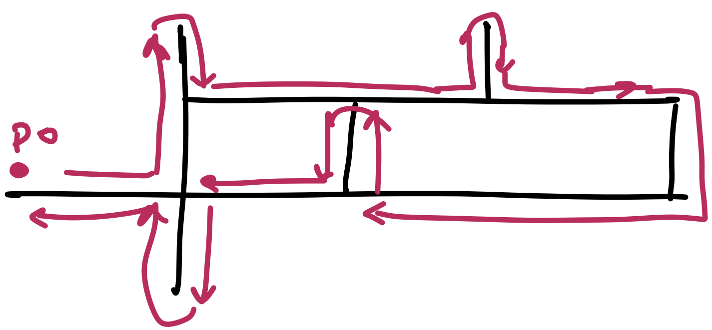


Chinese Postman Tour: Total route cost
= 3 + 1 +1 +1 +1 + 6 + 4 + 2 + 1 + 1 + 2 +5 = 28

Chinese Postman Problem (CPP) 1960

DW: 2.3.9, 2.3.10

Given a post-office and a road network, the postman starts from post-office, travels through each road segment (edges) at least once to deliver letters and return to the post office, minimizing travel cost.



Formulation: Network → a graph
PO → a vertex
road segments → edges
non-negative edge weights → distance, time

Applications

① postman's delivery of letters; ② garbage collection; ③ snow removal; ④ police patrolling

CPP vs TSP?

Observation

If G is even, then $CPP \equiv ET$ (Eulerian closed trail)

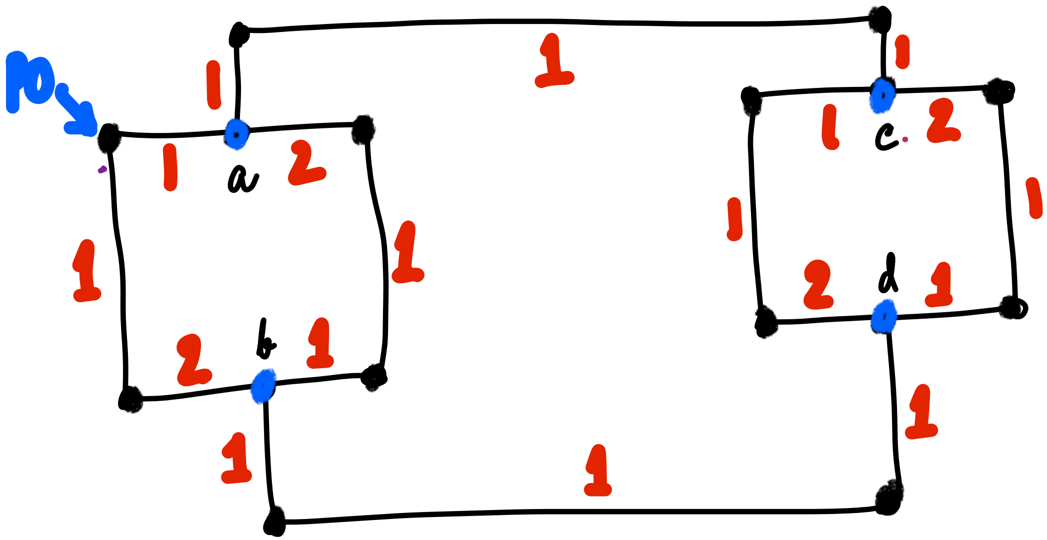
otherwise, we must repeat traversing some edges to create a closed trail.

$$\text{Total cost} = \underbrace{\text{Closed trail cost}}_{\text{constant}} + \underbrace{\text{repeat-cost}}_{\text{minimize}}$$

objective

- Repeat or duplicate edge travel so that
  (a) all degrees become even
  (b) repeat cost is minimized

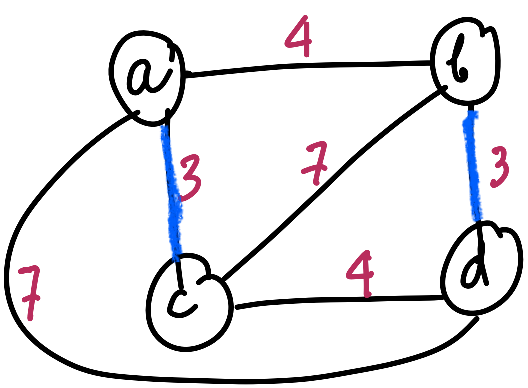
Example: Chinese Postman Problem (CPP)



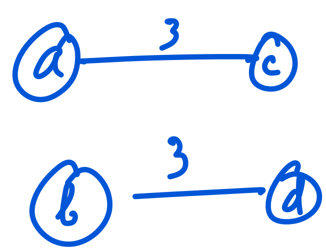
HW K: Trace the postman's travel route

Perfect matching exists (handshaking lemma)

a, b, c, d → odd-degree vertices



Need least-cost perfect matching

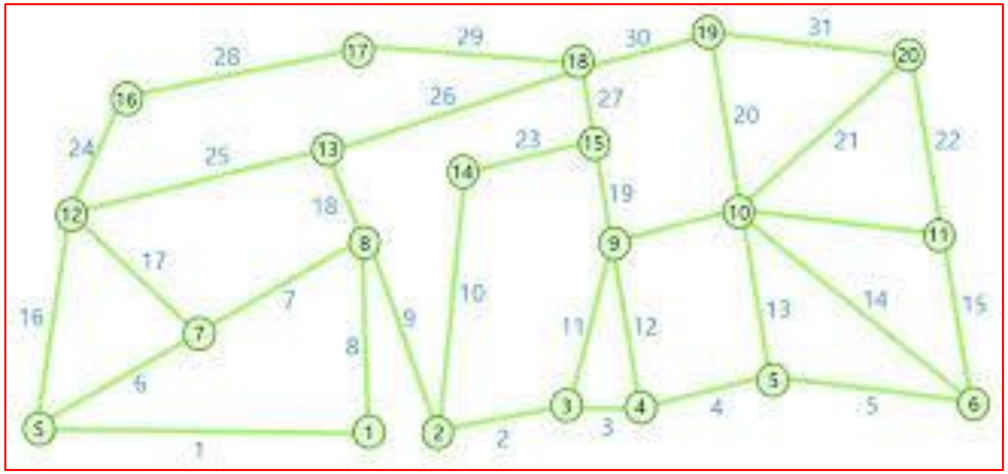


Additional cost = 6

Total optimized travel cost = 22+6=28

Traveling Salesperson Problem (TSP): Finding a Hamiltonian

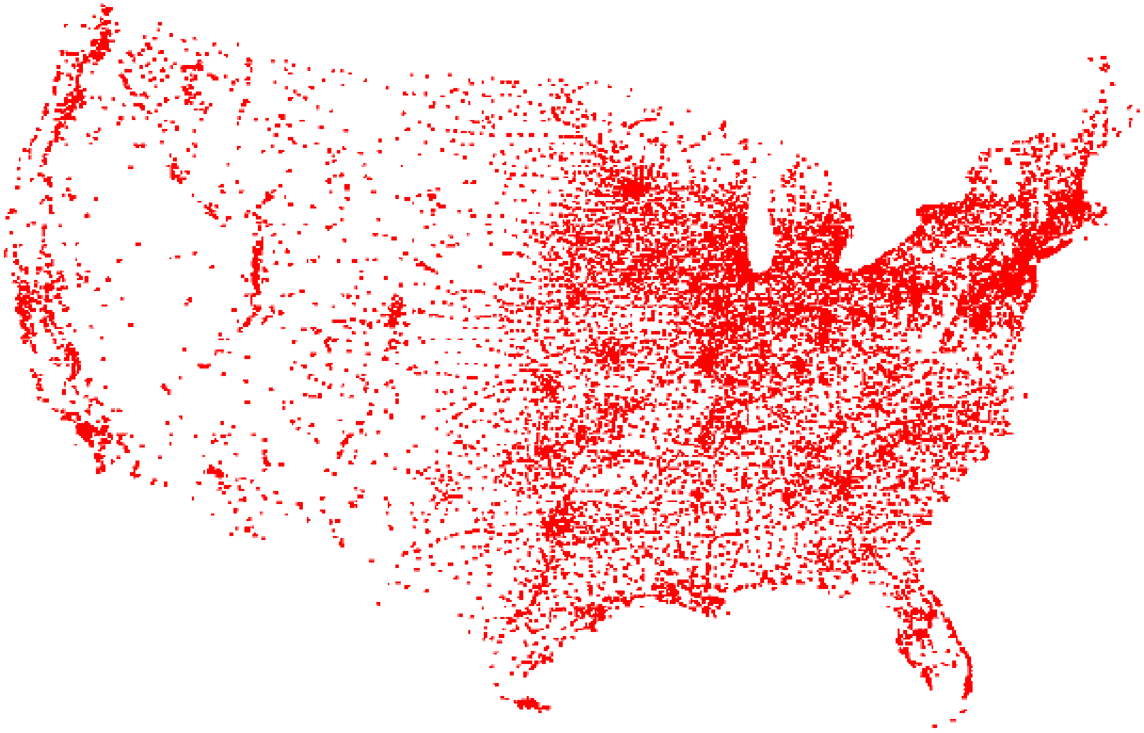
Find a closed tour of minimum length (cost) visiting certain number of places



TSP → Numerous applications:
Transportation: scheduling deliveries; picking up students in a school-bus; collection of mails from post-boxes (why *different* from CPP?);
Engineering: Scheduling of a machine to drill holes in a metal sheet; molecular biology

Example

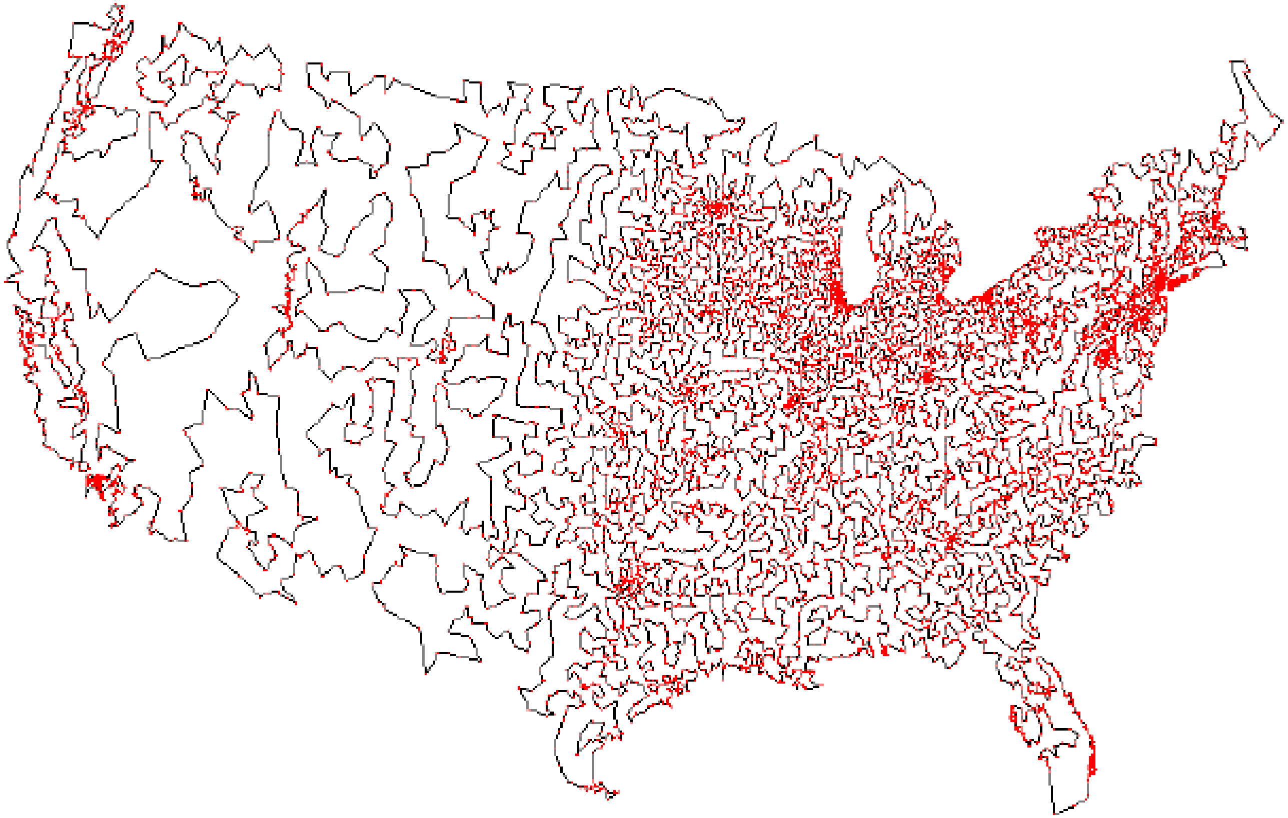
13,509 cities in the US



13508!= 1.476e+49936

Optimal tour?

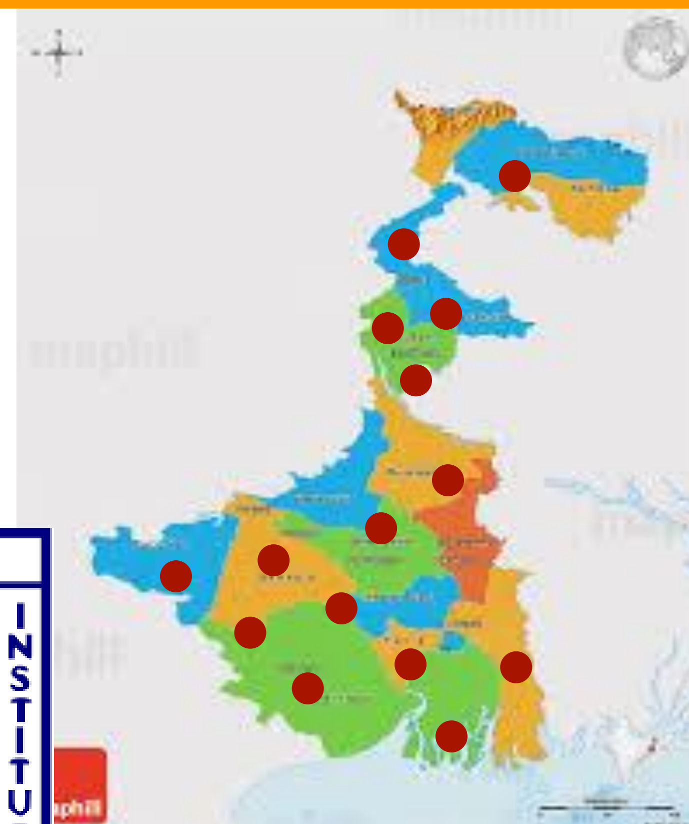
13509 cities in the USA

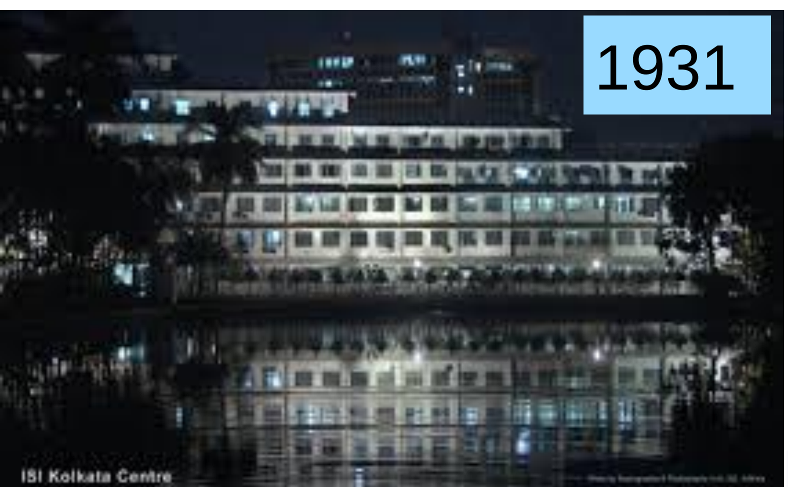


(Applegate, Bixby, Chvatal and Cook, 1998)

A TSP application (1940): Optimal transportation of farming equipment to multiple locations for conducting soil test in Bengal; *P. C. Mahalanobis*, Indian Statistical Institute, Calcutta

TSP research in India?





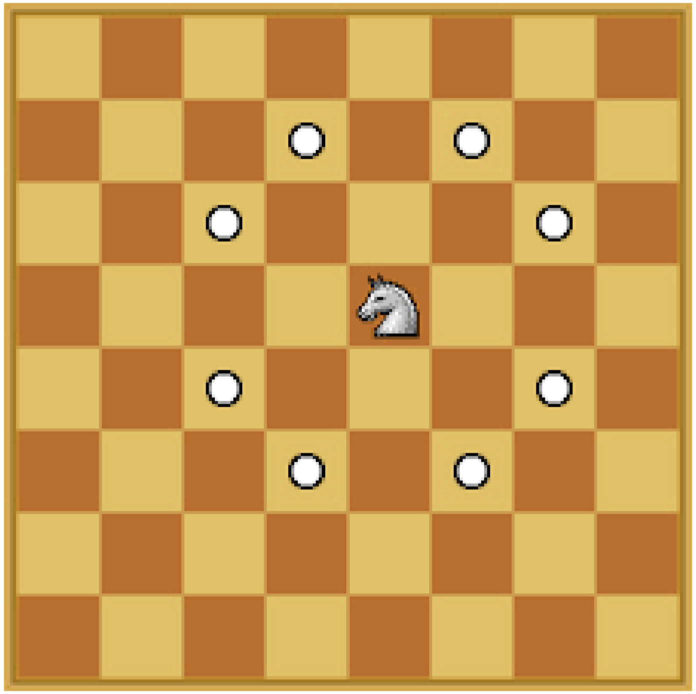
1931

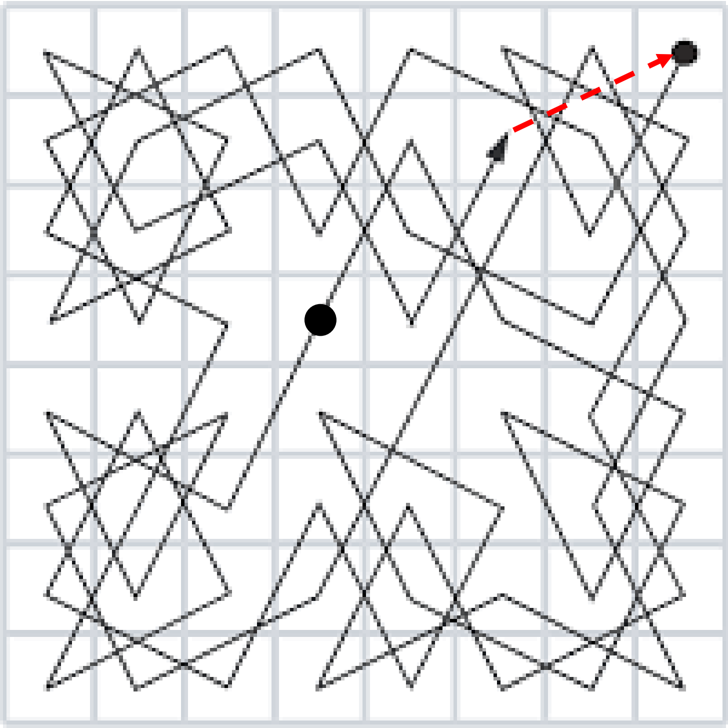
STATISTICAL INDIAN INSTITUTE

भिन्नेष्वैक्यस्य दर्शनम्

UNITY IN DIVERSITY

Can all the cells of an (8×8) chessboard be reached with knight's move without repeating visits to cells and returning to the start position?



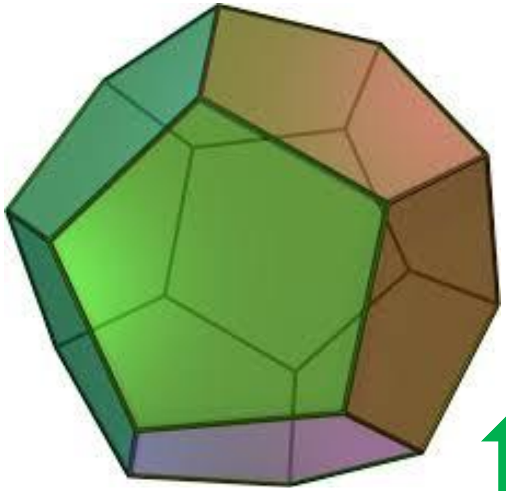


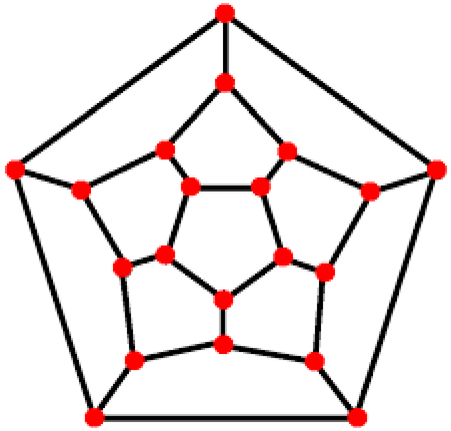
Ali C. Mani and al-Adli ar-Rumi

Circa 840

Platonic Solids and Hamiltonian

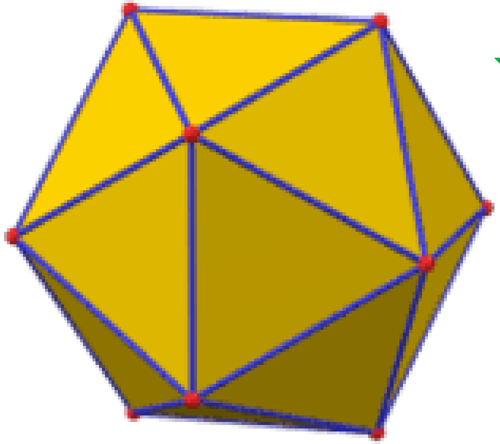
dodecahedron: 20 vertices, 12 pentagons, 30 edges





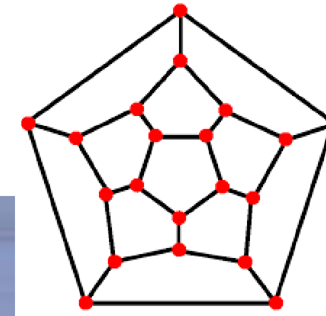
planar dual of
each other

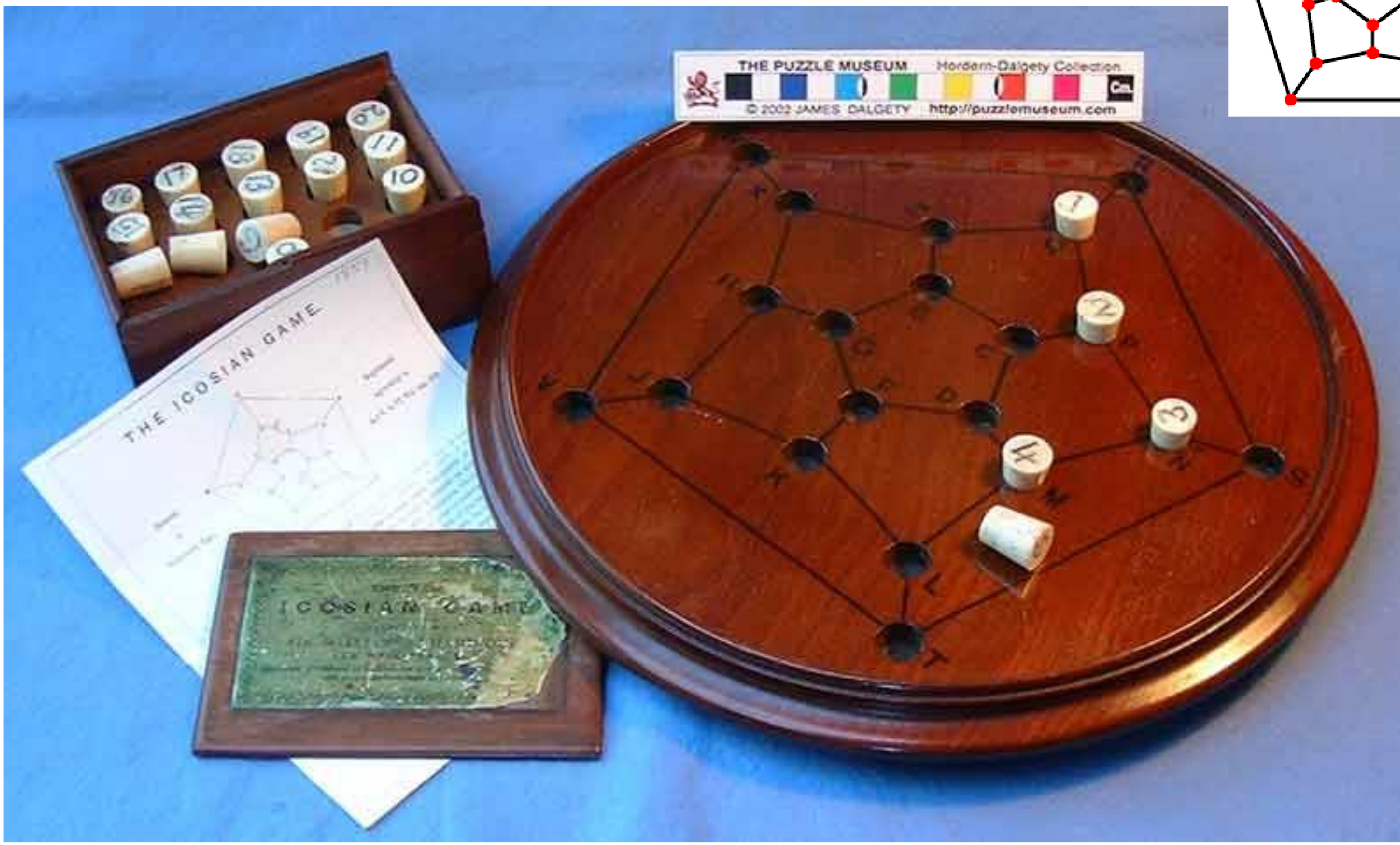
planar drawing of dodecahedron



icosahedron:
20 faces, 30 edges and 12 vertices

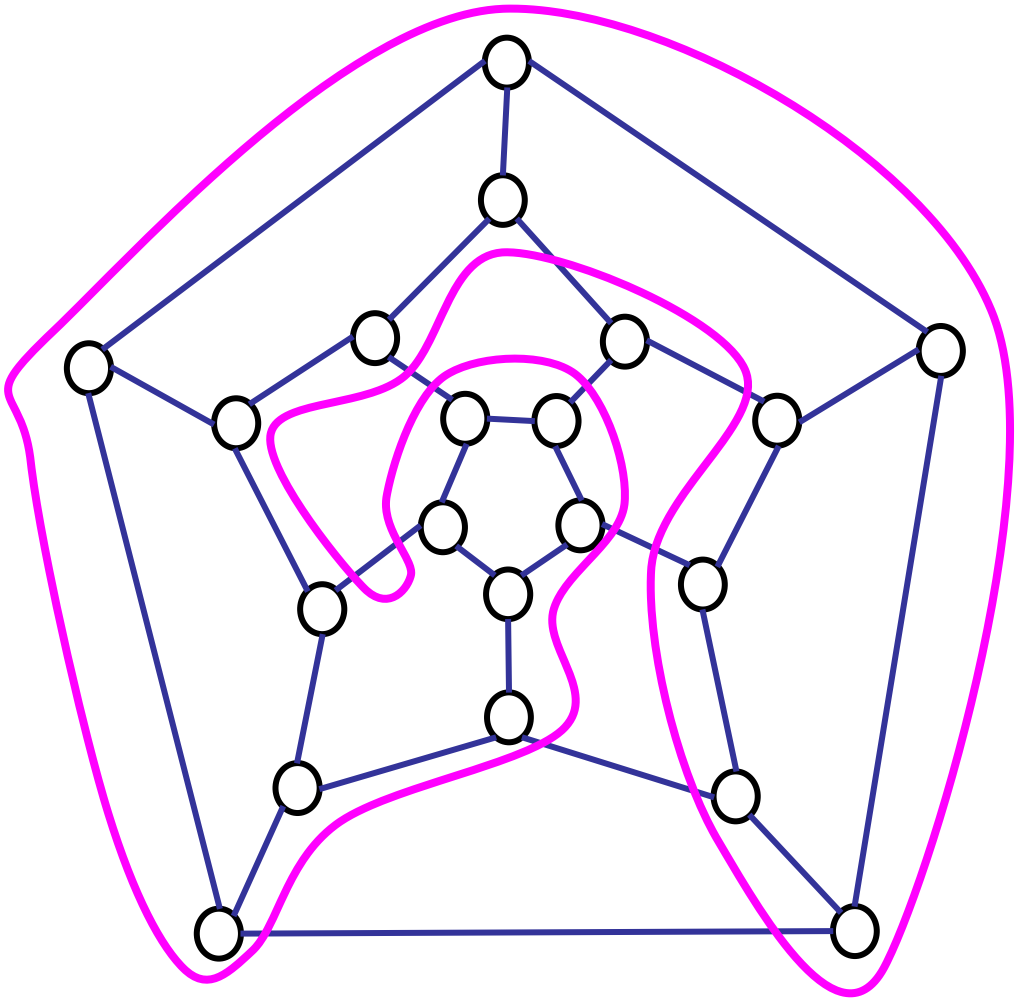
The Icosian Game





William R. Hamilton (1805-1865), a mathematician, physicist and astronomer of Ireland, invented the puzzle in 1857

Visit every vertex of a dodecahedron exactly once and finish where you had started

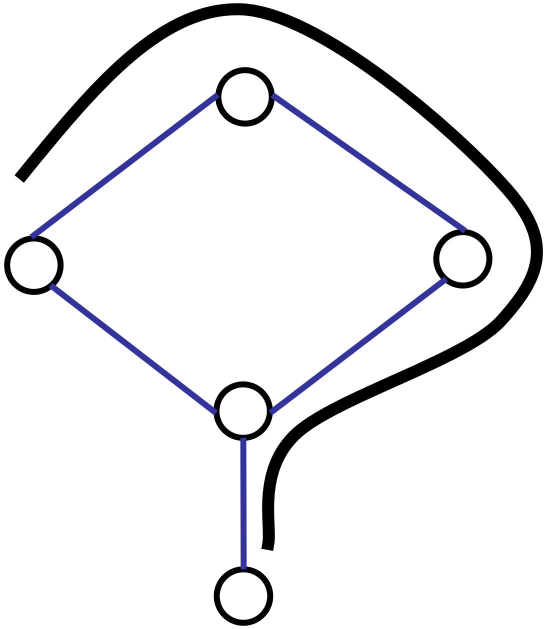


Hamiltonian Cycle $\equiv$ TSP (when edge-cost is attached)

Hamiltonian Graph

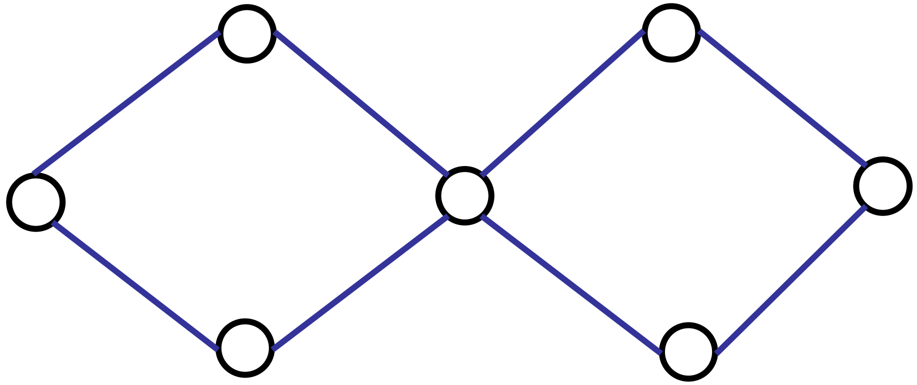
- **Hamiltonian path** is a path that visits each vertex exactly once
- A **Hamiltonian cycle** (also called *Hamiltonian circuit*) is a cycle that visits each vertex exactly once (except for the starting vertex, which is visited once at the start and once again at the end)
- A graph that contains a Hamiltonian cycle is called a **Hamiltonian graph**. Any Hamiltonian cycle can be converted to a Hamiltonian path by removing one of its edges, but a Hamiltonian path can be extended to Hamiltonian cycle only if its endpoints are adjacent

Hamiltonian Graph



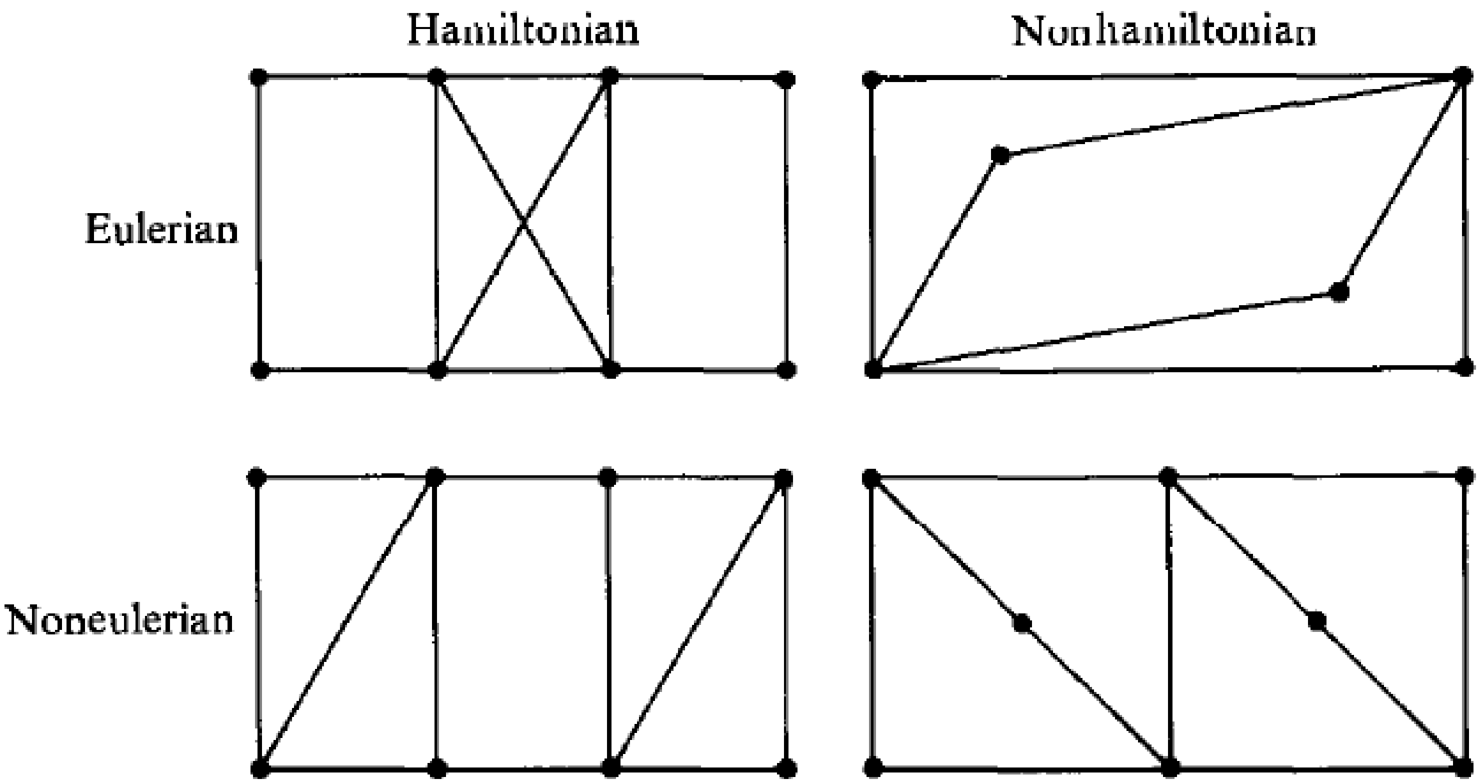
This one has a Hamiltonian path, but not a Hamiltonian tour (cycle)

Hamiltonian Graph

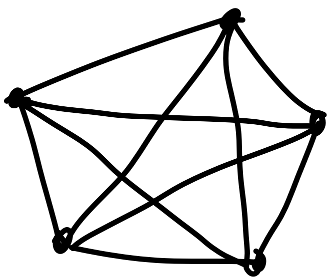


This one has an Euler tour but no Hamiltonian cycle

Example (Harary)

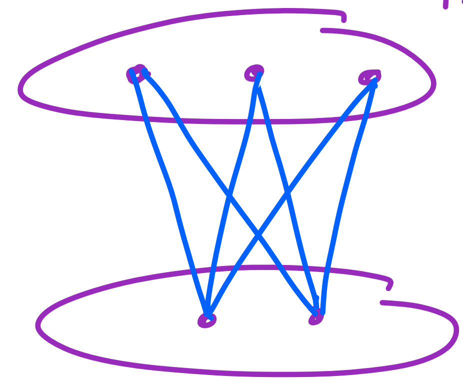


Hamiltonian (HC)

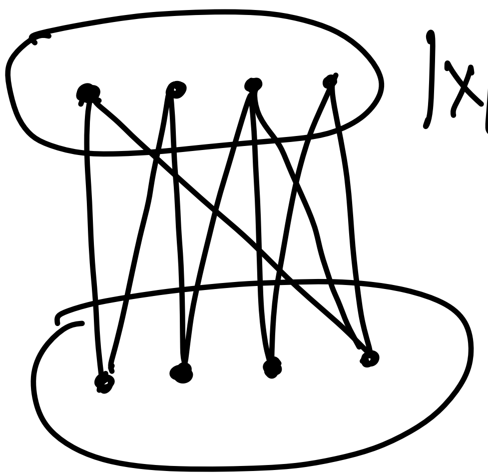


K_5

$\Rightarrow$ K_n is Hamiltonian



cannot be HC



$|X| = m$

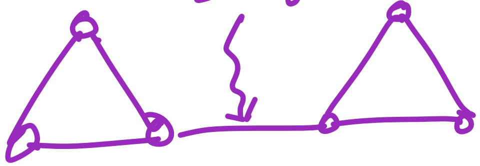
$|Y| = n$

A bipartite graph $K_{m,n}$ will be Hamiltonian only if $m = n$

necessary Condition

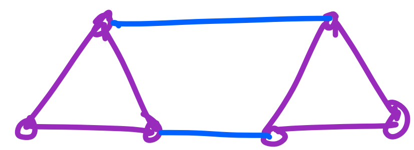
Hamiltonian

bridge

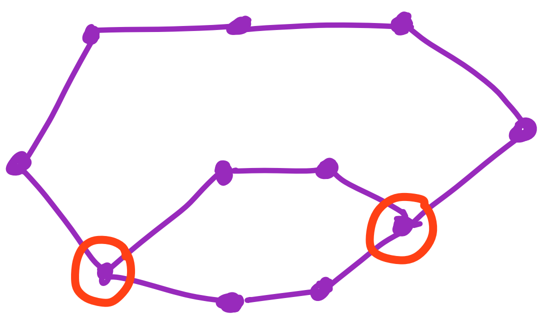


cannot be HC

⇒ Every Hamiltonian graph must be 2-connected



⇒ every pair of vertices must lie on a cycle.



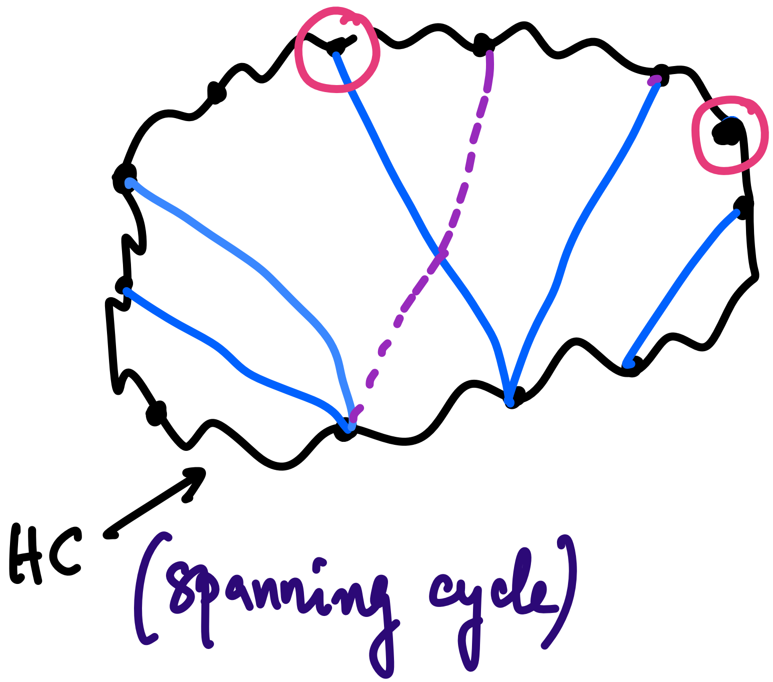
← 2-connected

HC ✗

Removal to two vertices ⇒ three components ⇒ HC ✗

Not possible.

Why?



Removal of two vertices must leave at most two components, provided ∃ HC.

Theorem (DW: 7.2.3)

If $G(V,E)$ has a HC, then for every non-empty set $S \subseteq V$, the graph $G-S$ has at most $|S|$ components.

Proof:

$\omega(G)$: # components in G

Let C be a spanning cycle.

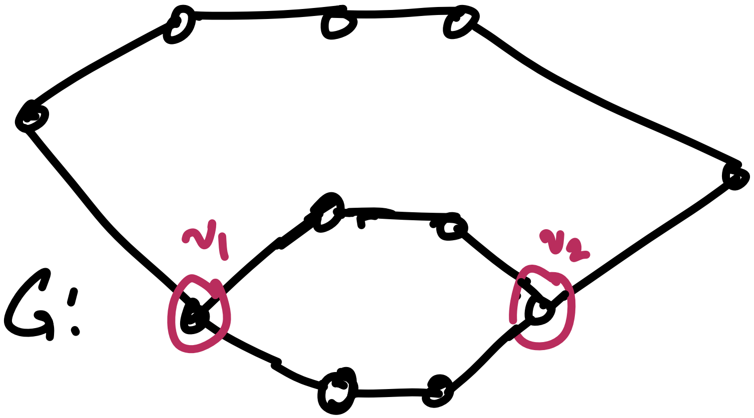
$$\omega(C-S) \le |S|$$

$C-S$ is also a spanning subgraph of $G-S$.

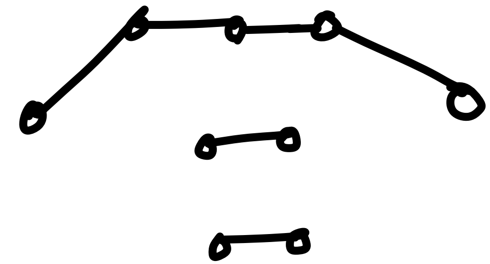
$$\omega(G-S) \le \omega(C-S) \le |S|$$

$\boxtimes$

G:



$\Rightarrow$ G-S:



Three components

$S = \{v_1, v_2\}$

$|S| = 2$

$\Downarrow$

G is not Hamiltonian.

Necessary condition, <u>not</u> sufficient

CS 60047 Autumn 2020

Advanced Graph Theory

Instructor

Bhargab B. Bhattacharya

Lecture #14, #15: 25 Sept. 2020

Indian Institute of Technology Kharagpur
Computer Science and Engineering

Today’s Topics

Traversability

- Hamiltonian graphs
- Sufficient conditions
- Line graphs and traversability
- Problem-solving tutorial

Sufficient Conditions for HC

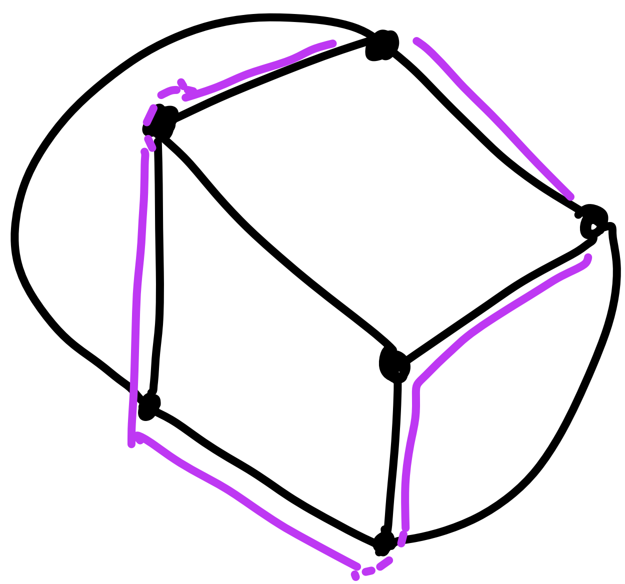
Theorem (Dirac) : DW: 7.2.8

Given a simple graph $G(V,E)$, $|V| = n \geq 3$.

If $\delta(G) \geq \frac{n}{2}$, then G is Hamiltonian.

minimum degree

Example



$n = 6$

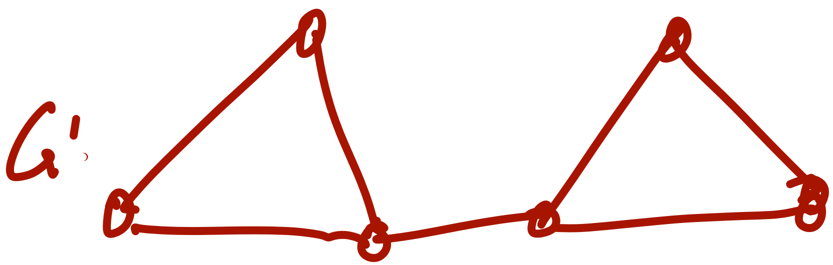
$\delta(G) = 3 \geq \frac{n}{2}$ ✓

$n = 2$

$\delta(G) = 1 \leq \frac{n}{2}$ ✓

No HC

No smaller minimum degree is sufficient



$n = 6, \quad n > 3$

$\delta(G) = 2 < \frac{n}{2}$;

G does not admit HC

Proof (Dirac's Theorem): $\delta(G) \geq \frac{n}{2} \Rightarrow G$ admits HC.

By contradiction and maximality

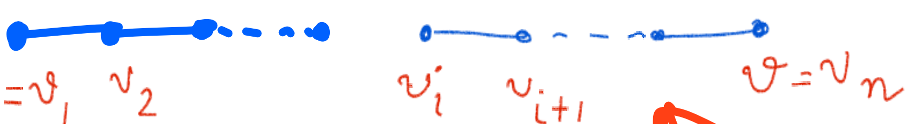
Let G be a maximal non-H simple graph with $n \geq 3$ and $\delta \geq \frac{n}{2}$

Let u, v be two non-adjacent vertices in G.

$\Rightarrow$ each HC in $(G+uv)$ must include (u,v).

Consider the spanning path P: $u(v_1), v_2, \cdots, v(v_n)$

P:

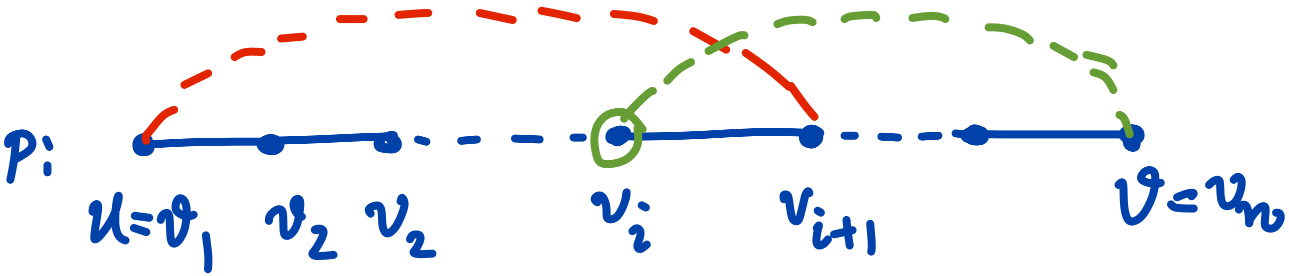


maximal path

To prove: $\delta(G) \geq \frac{n}{2} \Rightarrow G$ admits a HC

$\Rightarrow \delta(G) < \frac{n}{2}$

$\Rightarrow$ Contradiction $\boxtimes$



Define $S = \{ v_i \mid (u, v_{i+1}) \in E \}$; $T = \{ v_i \mid (v_i, v) \in E \}$

Note that $v_n \notin S \cup T$

$\Rightarrow |S \cup T| < n$

Also, $S \cap T = \phi$, otherwise $G \Rightarrow HC$

$$\therefore d(u) + d(v) = |S| + |T| = \underbrace{|S \cup T| + |S \cap T|}_{< n}$$

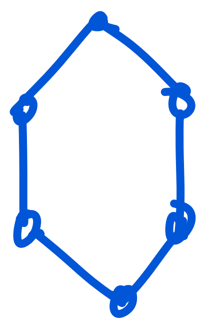
Hamiltonian

$$\boxed{\delta(G) \geq \frac{n}{2} \Rightarrow G \text{ is } H.}$$

Gabriel Dirac's results provides a sufficient condition, but not necessary.

Example:

C_6:

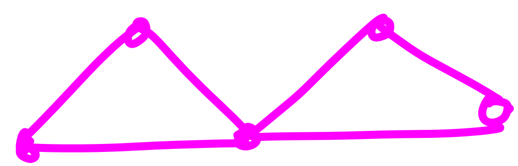


$n=6$

$\delta(G) = 2 < \frac{n}{2} = 3$

But, C_6 is Hamiltonian

G:



$n=5$; $\delta \geq \frac{n-1}{2} = 2$

The sufficient condition is tight:
G is not Hamiltonian
(However, a Hamiltonian path exists).

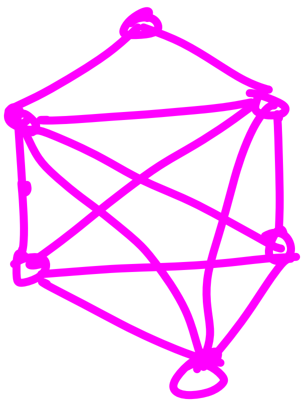
In 1960, Oystein Ore improved Dirac's result.

Theorem (Ore): Let $G(V,E)$ be a simple graph with $|V| = n \geq 3$. If for each pair of non-adjacent vertices u, v in G, $d(u) + d(v) \geq n$, then G is Hamiltonian.

Variant (DW: 7.2.9): If for a pair (u,v) of non-adjacent vertices $d(u) + d(v) \geq n$, then G is Hamiltonian if and only if $G + uv$ is Hamiltonian.

Example

Dirac $\Rightarrow$ $\underset{(2)}{\delta(G)} < \underset{(3)}{\frac{n}{2}}$ $\Rightarrow$ inconclusive



G : $n = 6$

$\delta(G) = 2$

Ore $\Rightarrow$ For every pair of non-adjacent vertices u, v

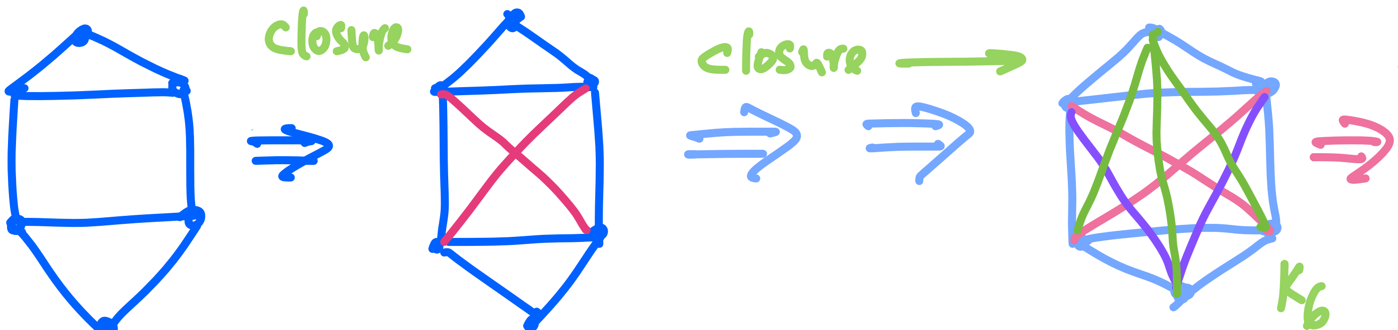
$d(u) + d(v) \geq 6 = n$

$\Rightarrow$ G is Hamiltonian.

Ore $\Rightarrow$ Hamiltonian Closure

Given a graph G $\Rightarrow$ put an edge between a non-adjacent pair $u, v \in V$, whenever $d(u) + d(v) \geq n$.

$\Longrightarrow$ Hamiltonian closure of G



closure is unique

$n = 6$

$\delta(G) = 2$

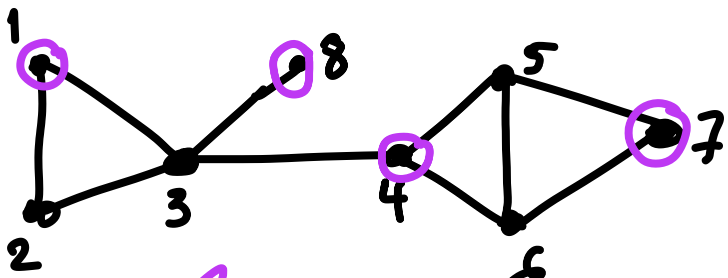
Necessary & sufficient

Bondy & Chvátal: A simple n-vertex graph is H iff its closure is H.

Connectivity, independence, and HC

independence number $\alpha(G)$
↳ size of max. ind. set.

Connectivity $\kappa(G)$ of a graph (V,E)
↳ minimum-size vertex set $S \subset V$, s.t.
$G-S$ is disconnected or has only one vertex



$\Rightarrow \kappa(G) = 1 \Rightarrow \{3\}$

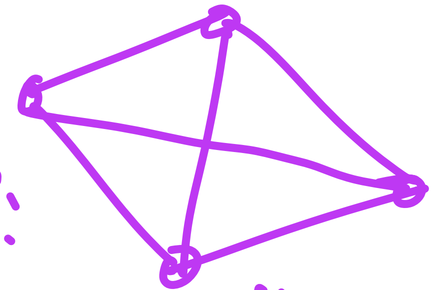
$\alpha(G) = 4 = \{1,4,7,8\}$

$\beta(G) = 4 = \{2,3,5,6\}$

size of minimum vertex cover

$$\boxed{\alpha(G) + \beta(G) = n}$$

G:

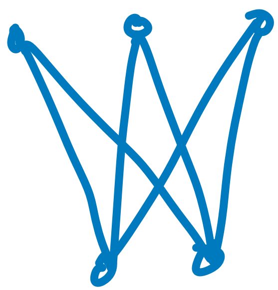


K_4

$\kappa(G) = 3$; $\alpha(G) = 1$, $\beta(G) = 3$

m

G:



n

$K_{m,n}$

$\kappa(G) = \min(m,n)$; $\alpha(G) = \max(m,n)$

$\beta(G) = \min(m,n)$

Theorem (Chvátal and Erdős, 1972): DW: 7.2.19

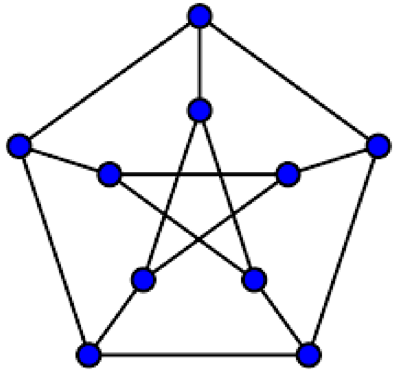
Let $G(V,E)$ be a graph, s.t. $|V| \geq 3$.
If $\kappa(G) \geq \alpha(G)$ then G admits a HC.

connectivity

independence number

Proof: Reading assignment $\boxtimes$

Problem 1



Show that PG is not Hamiltonian,
but $\overline{PG}$ is Hamiltonian

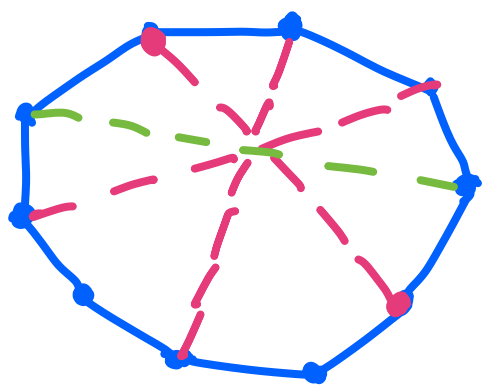
__Proof:__ Suppose PG is Hamiltonian
$\Rightarrow \exists$ a spanning cycle C_{10}

$e(PG) = \frac{10 \times 3}{2} = 15$

$e(C_{10}) = 10$

$\exists$ 5 chords

girth(PG) = 5



$\Rightarrow$ 4 cycle $\Rightarrow$ contradiction. $\boxtimes$

$\overline{PG}$ is Hamiltonian

Proof: $\forall i$, $d(v_i) = 3$ in PG

Therefore, $\forall i$ $d(v_i) = 6$ in $\overline{PG}$

$|V| = n = 10$ in PG and $\overline{PG}$

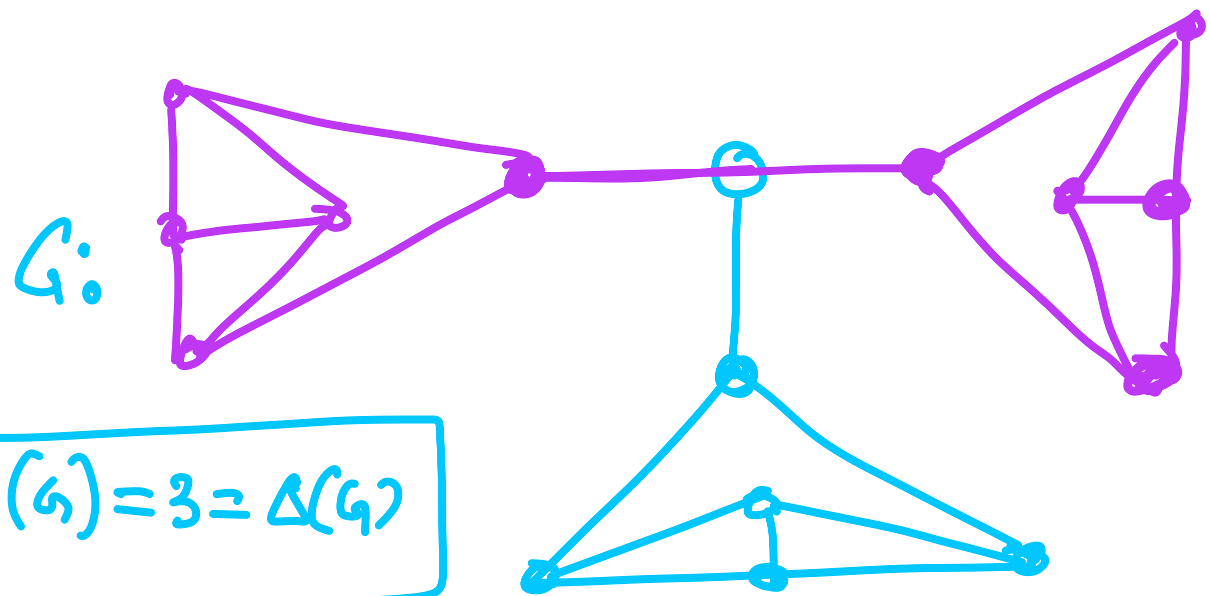
$$\delta(\overline{PG}) = 6 \geq \frac{n}{2} = 5$$

By Dirac's Theorem, $\overline{PG}$ is Hamiltonian

$\boxtimes$

Problem 2

Construct a 3-regular graph G which does not have a Hamiltonian cycle or a Hamiltonian path.



← No HC but HP ✓

$\boxed{\delta(G) = 3 = \Delta(G)}$

⇐ No HC No HP

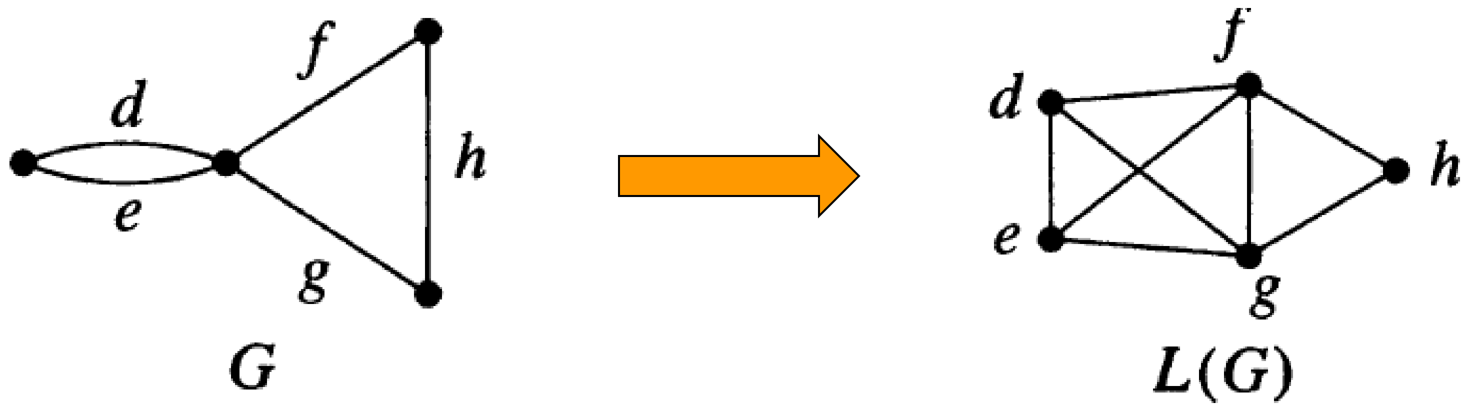
Summary: Hamiltonian Graph

- **Dirac's Theorem:** if G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$ then G has a Hamilton circuit.

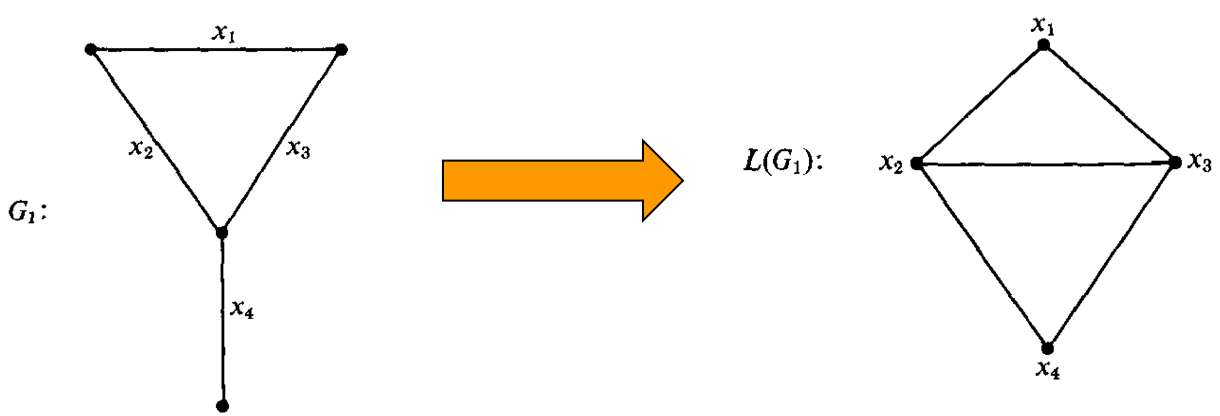
- **Ore's Theorem:** if G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

Open Question: What is the necessary and sufficient condition for a graph to become Hamiltonian??

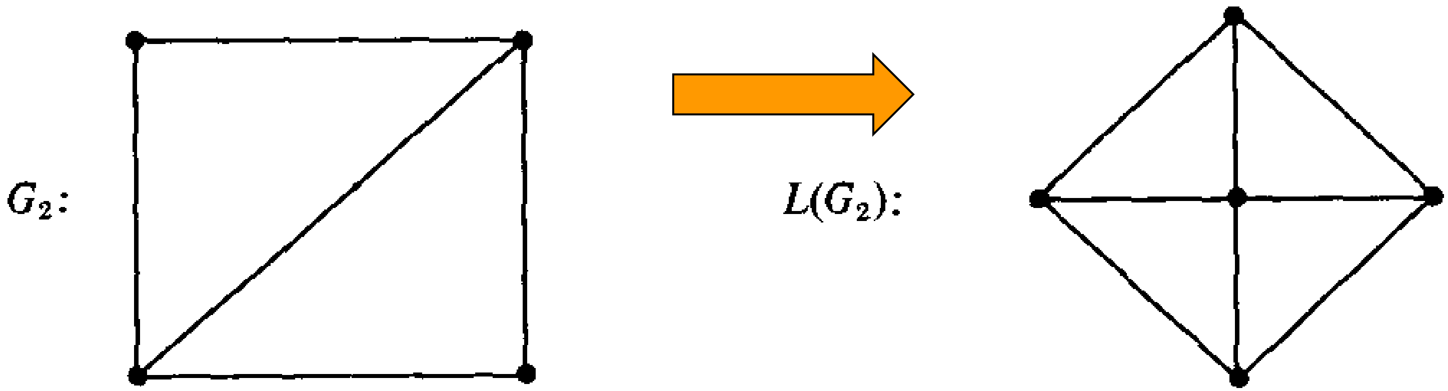
Line graphs and traversability (Textbook DW: 7.1.1)

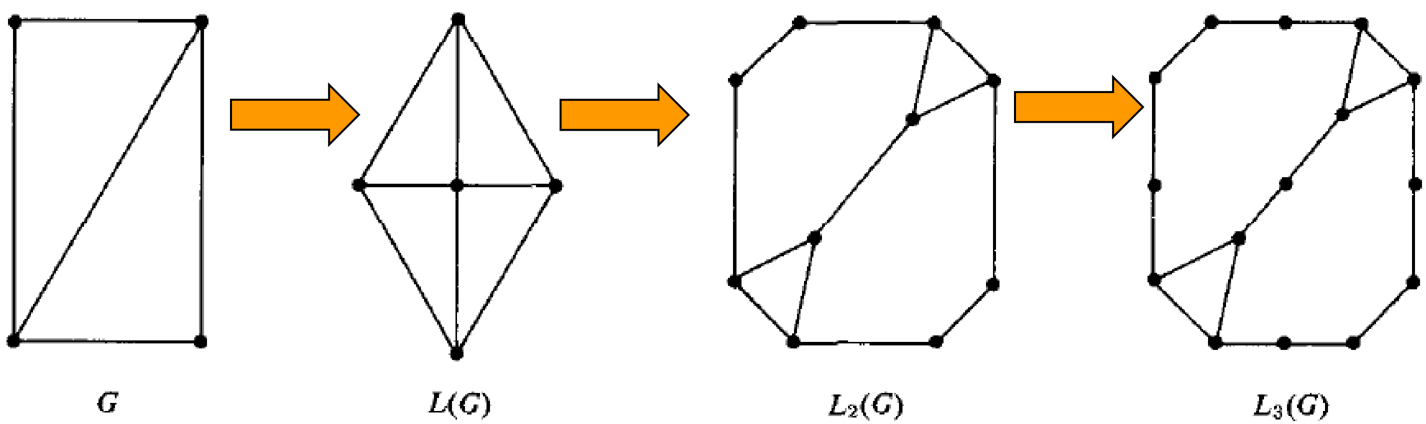


Edge in G → a vertex in line graph $L(G)$
$(u, v) \in E(L(G))$ if in G, edge u and edge v share a common vertex



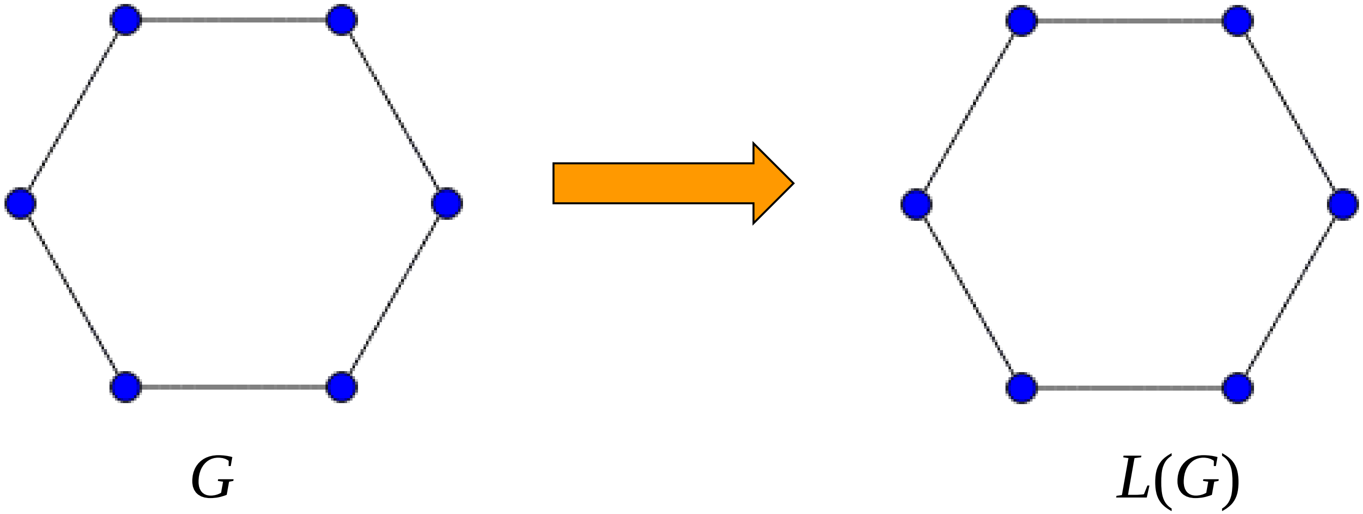
Line graphs





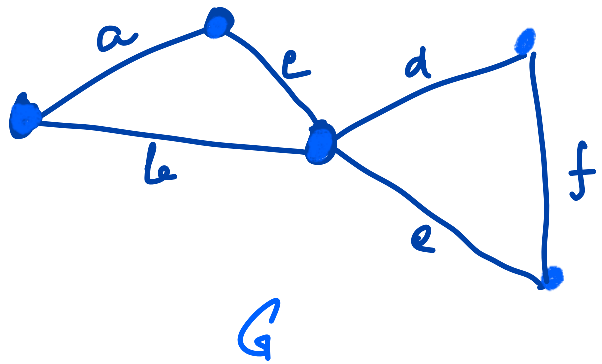
sequence of line graphs

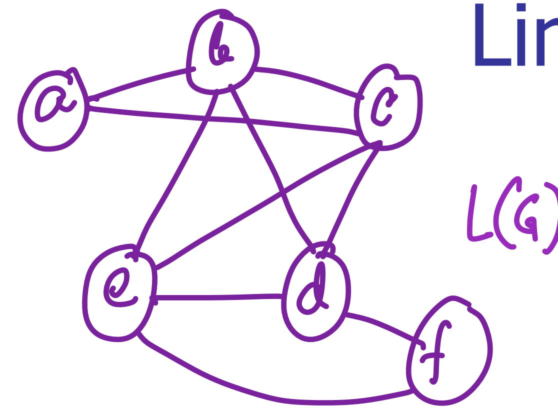
Line graphs

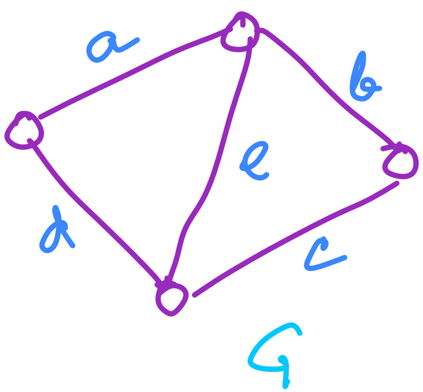


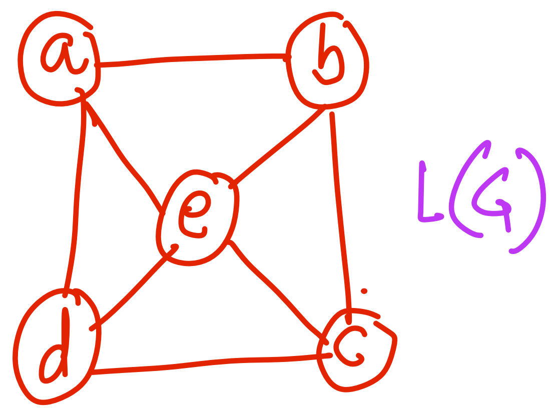
A connected graph G is isomorphic to $L(G)$ if and only if G is a cycle! (*Ref.* Harary)

Line graphs







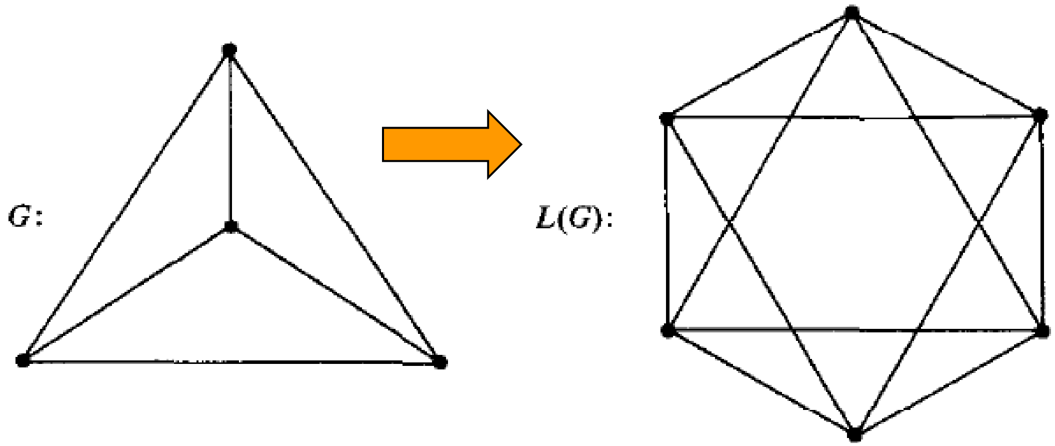


If a simple graph *G* is Eulerian, then *L*(*G*) is both Eulerian and Hamiltonian (a → b → e → f → d → c → a);

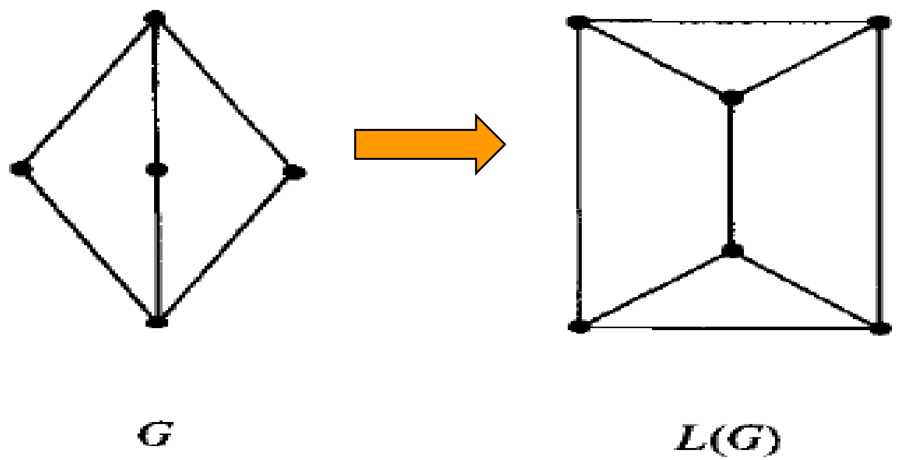
If a simple graph *G* is Hamiltonian, then *L*(*G*) is also Hamiltonian.

Proof: Homework

However, the converse is not always true

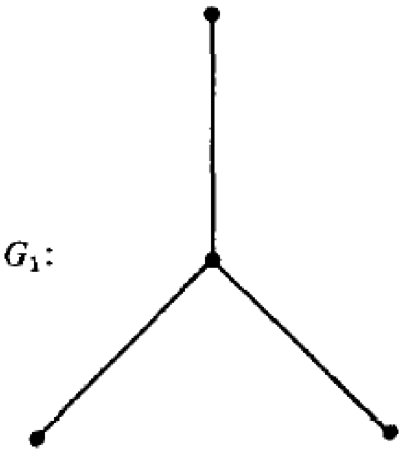


L(*G*) is Eulerian and Hamiltonian, but *G* is not Eulerian



L(*G*) is Hamiltonian, but *G* is not

Homework



Show that the graph shown above cannot be a line graph

Homework 1: Graphical Sequences

which of the following sequences are graphic?

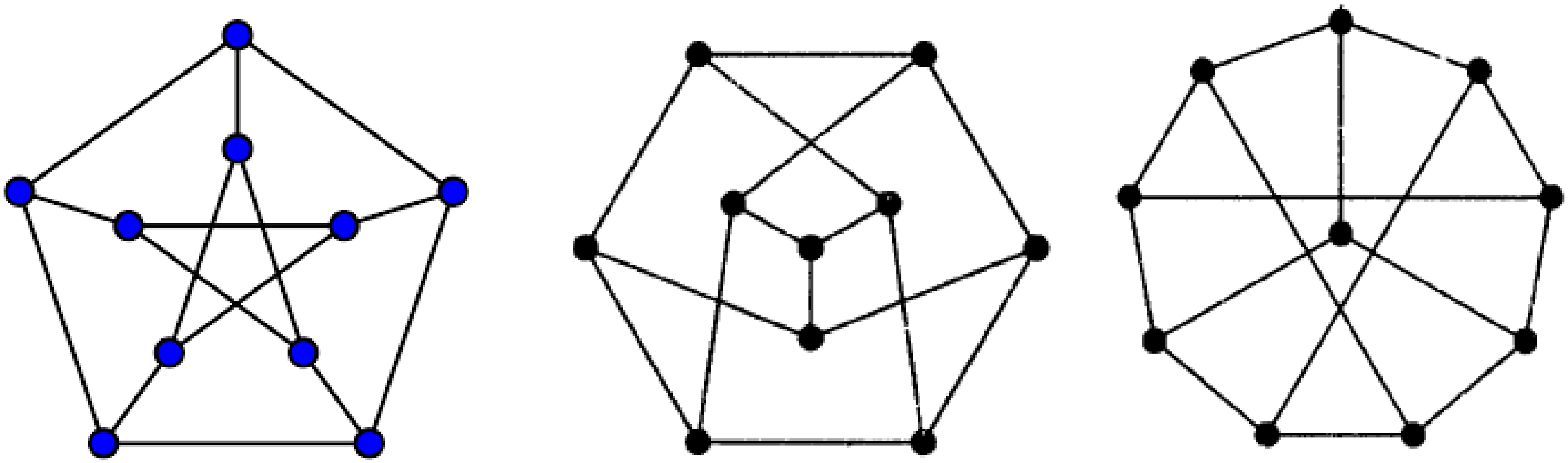
1. $\{5, 5, 4, 3, 2, 2, 2, 1\}$

2. $\{5, 5, 4, 4, 2, 2, 1, 1\}$

3. $\{5, 5, 3, 2, 2, 1, 1\}$

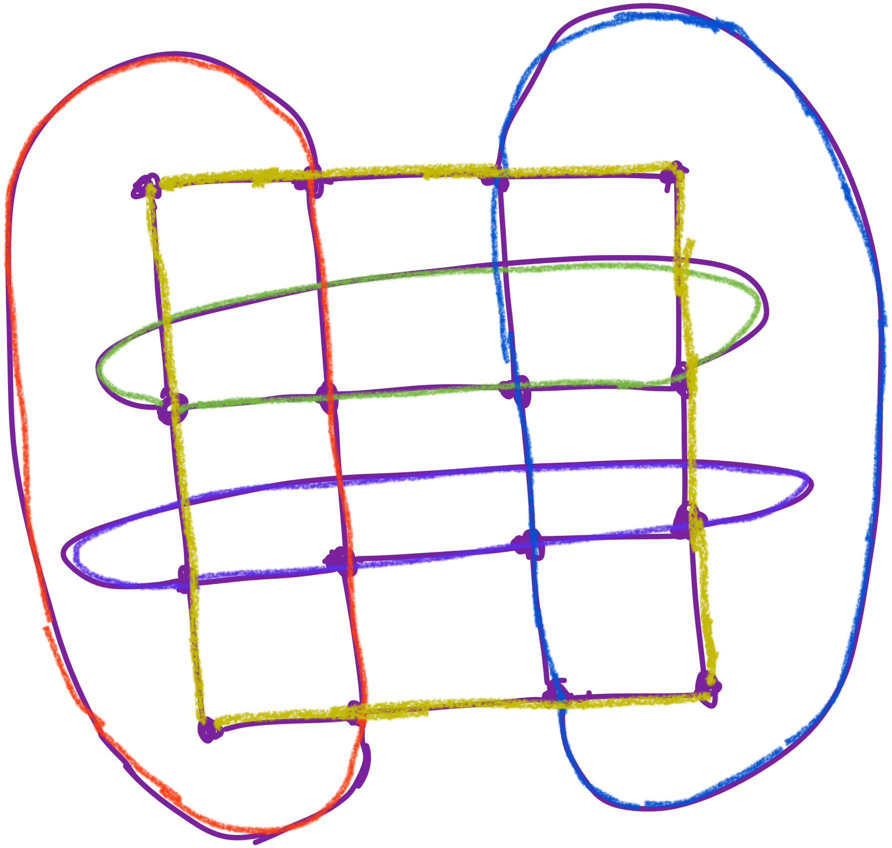
4. $\{5, 5, 5, 4, 2, 1, 1, 1\}$

Homework 2: Petersen Graph



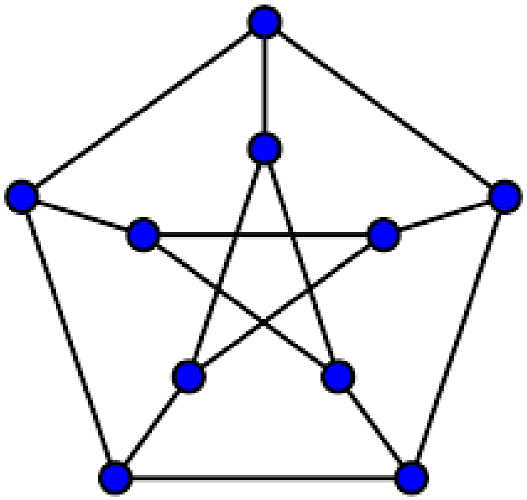
Does PG have C_7, C_8, C_9, C_{10} ?

Homework 3: Hierholzer's Algorithm



Label the vertices of this graph as 1, 2, …, 16. Then construct the Eulerian by combining the five cycles as shown, using Hierholzer's algorithm.

Homework 4: Open Trail



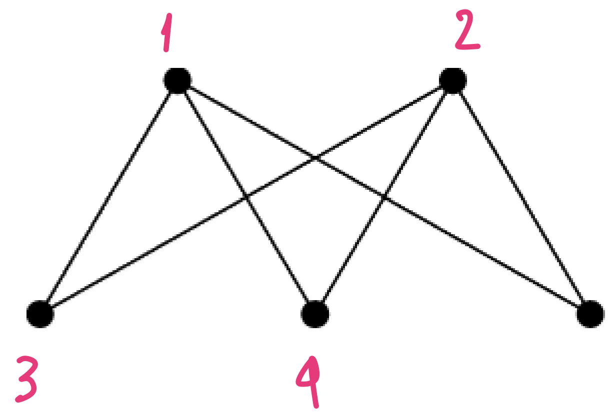
Label the vertices of PG with 1, 2, …, 10.

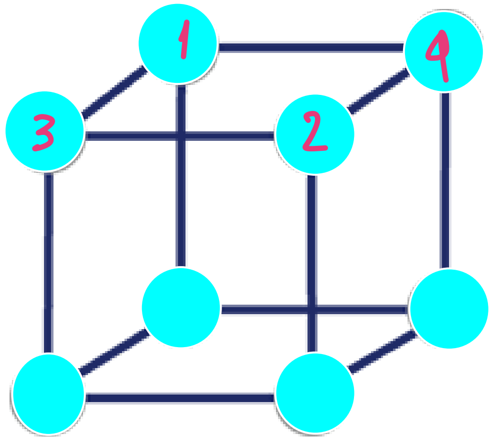
Add minimum number of extra edges in PG such that it admits an open trail covering all edges;

Show the trail.

Homework 5: Hypercube

Show that the above graph cannot be a subgraph of a Hypercube Q_k, for any $k \geq 3$.





Note that *dist*(1, 2) = 2, via both 3 and 4; hence, 1, 2, 3, 4 would lie in a 2-cube of Q_k. Two such vertices (e.g., 1, 2) can have at most two neighbors (such as 3, 4) in the same 2-cube. Since Q_k can be recursively composed, 1 and 2 cannot have any other common neighbor in higher order coordinates, because their successors are mirrored. Hence, the proof.

Announcement

1. Will collate the homework set and allocate them to some students randomly in advance, for presenting the solution in class, one each, on 25 September 2020; this exercise will be rewarded with grade points.

2. Will Schedule an online MCQ test on Saturday, 26 Sept. 2020, 12:00 noon – 1:30 PM covering the material discussed till 18 Sept. 2020. Open-book, open-notes test; Credit: 20%.

Twenty Problems from the textbook (D. West) for practicing

Problem #

1.1.8, 1.1.11, 1.1.12, 1.1.19, 1.1.22, 1.1.25, 1.1.38,

1.2.2, 1.2.9, 1.2.10, 1.2.20, 1.2.27, 1.2.34,

1.3.2, 1.3.4, 1.3.7, 1.3.9, 1.3.40, 2.1.12, 2.1.16