Discrete Mathematics

Graph Theory -II

Adil M. Khan

Professor of Computer Science

Innopolis University

"In mathematics the art of proposing a question must be held of higher value than solving it!"
- Georg Cantor -

Graph Connectivity

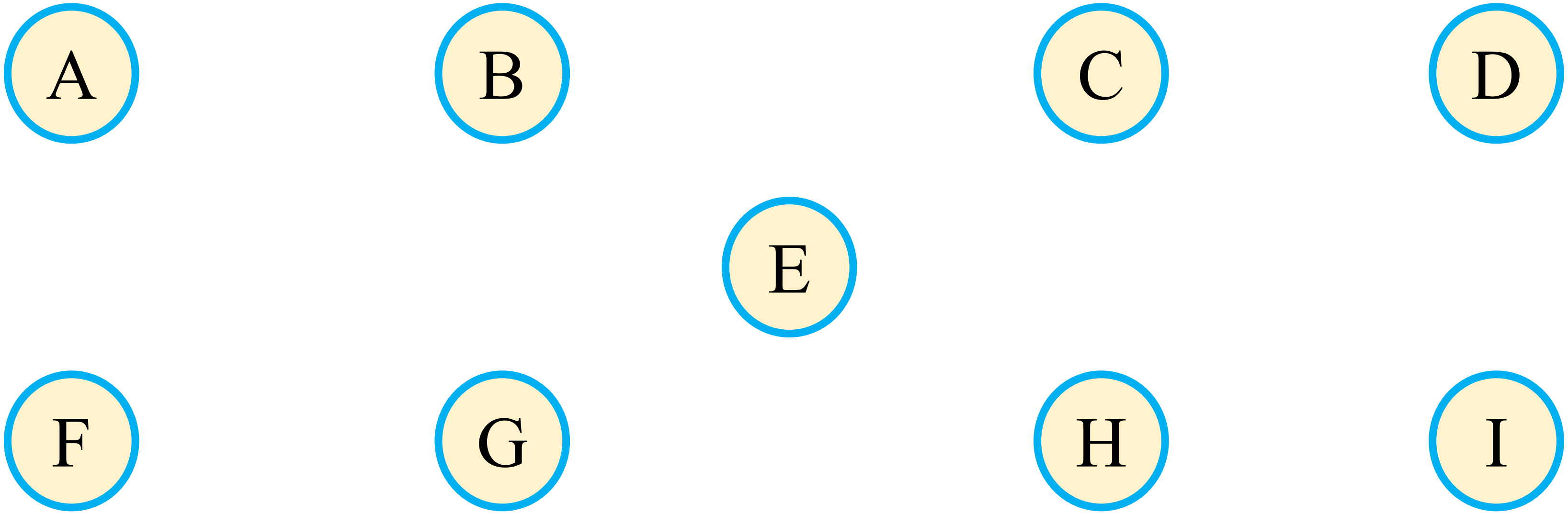
- An undirected graph $G = (V, E)$ is called Connected if for any u , $v \in V$, we have $u \leftrightarrow v$.

- If G is an undirected graph that is not connected, we say that G is disconnected.

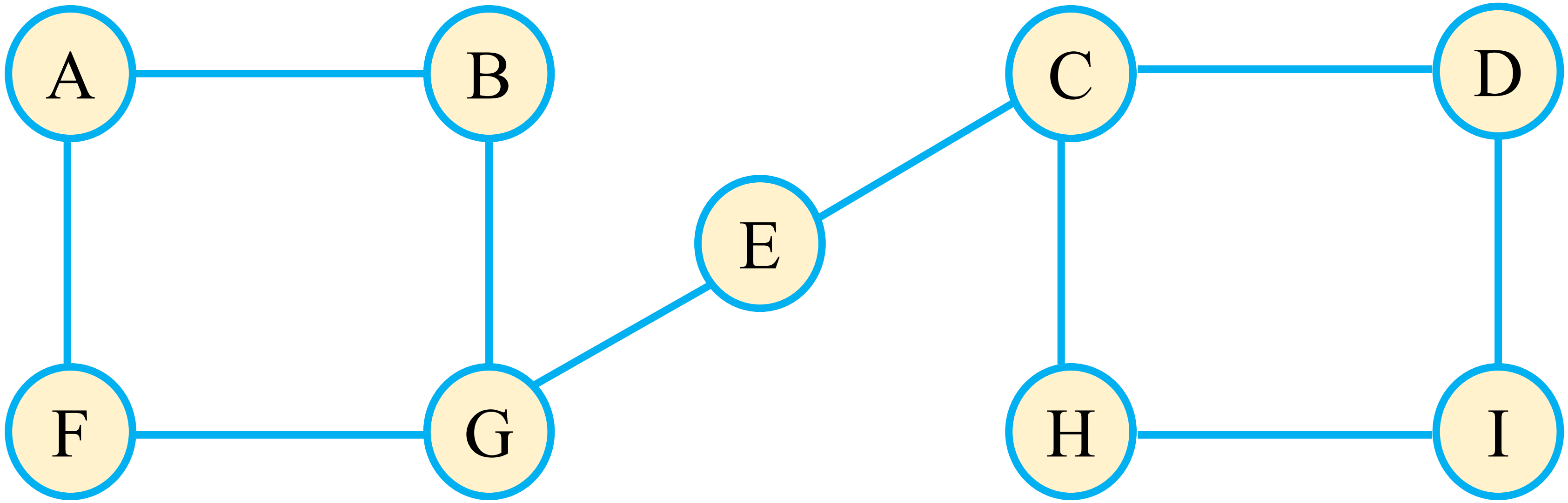
Graph Connectivity

- An important question related to graph connectivity is "how tightly or how fragile" that connectivity is.

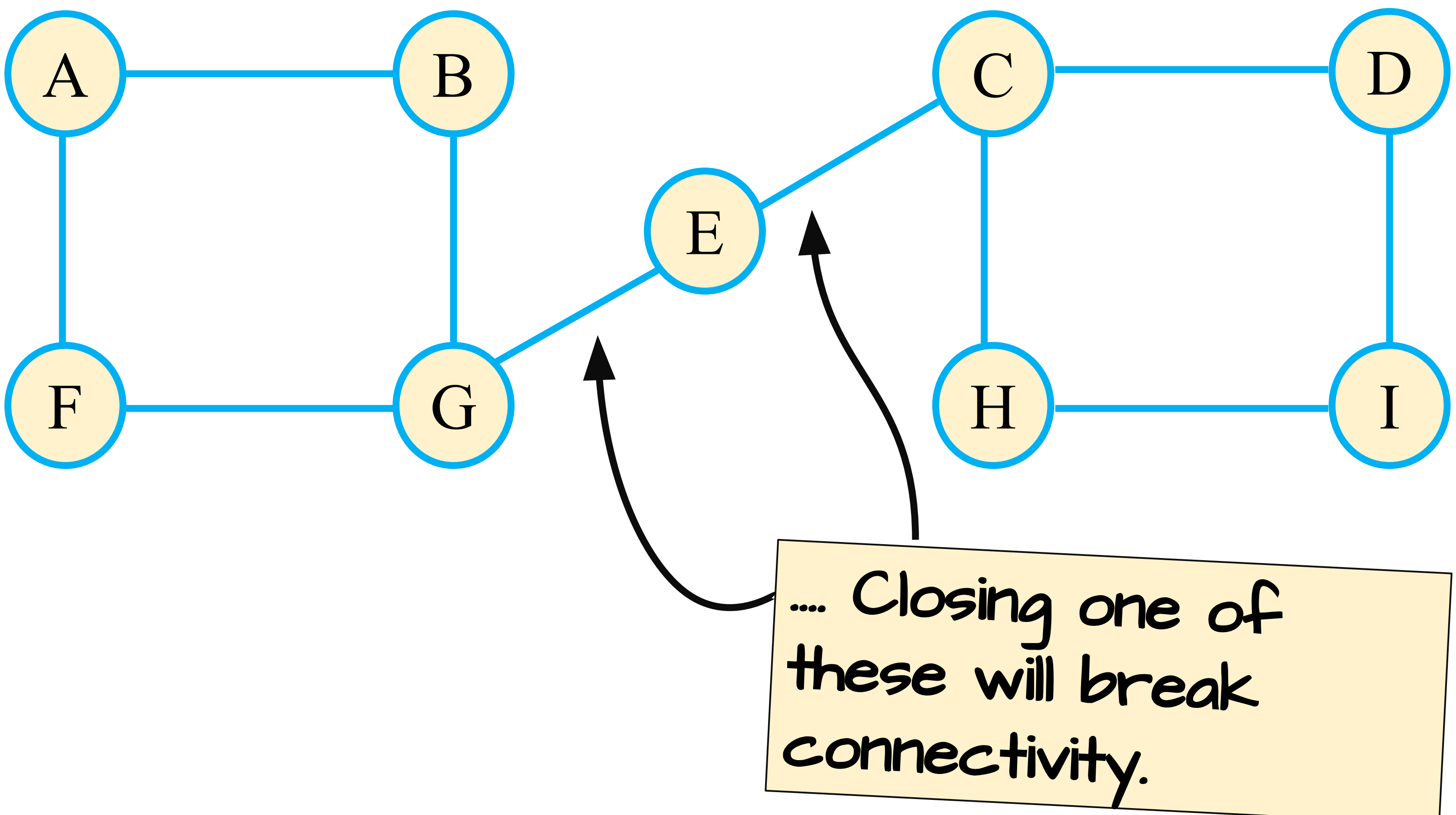
- For example, consider the following scenario



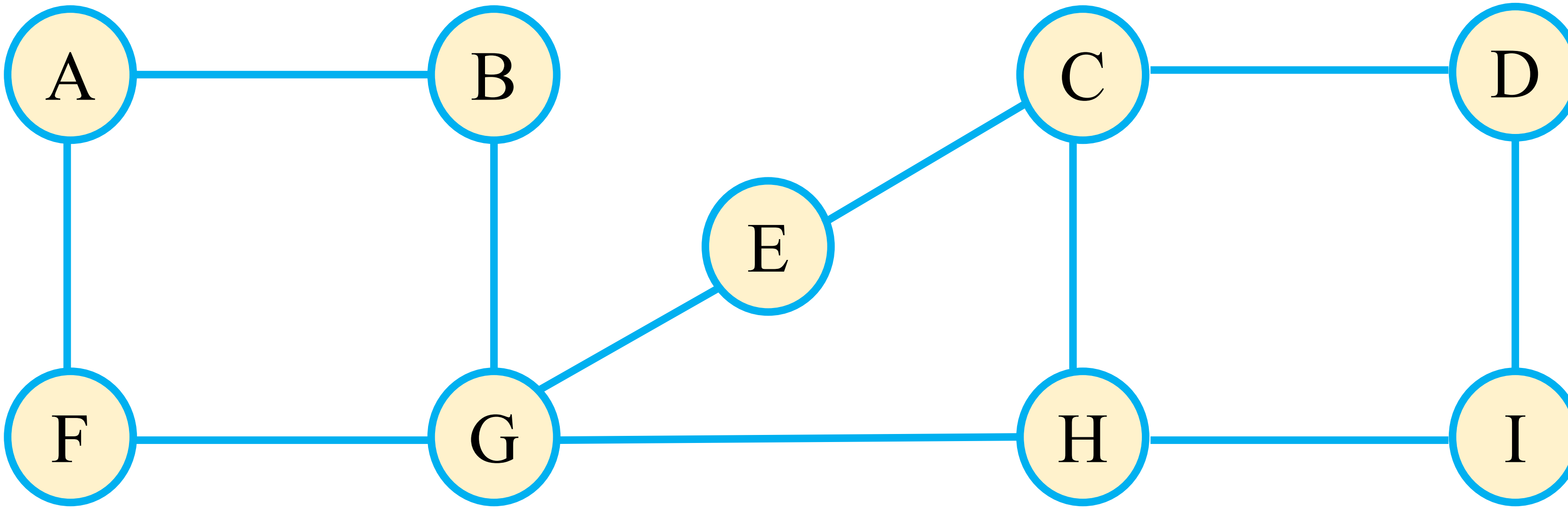
Graph Connectivity



Graph Connectivity



Graph Connectivity



.... More resilient to damage than the last one, but how much resilient?

Graph Connectivity

- How to characterize graph connectivity in circumstances where edges between nodes might suddenly break down?

- What do well-connected, and poorly connected graphs look like?

k-Edge Connected Graph

- An undirected graph G is called ***k-edge-connected*** if G is connected, and there is no set of **k-1** edges that can b removed from G that disconnects it.

k-Edge Connected Graph

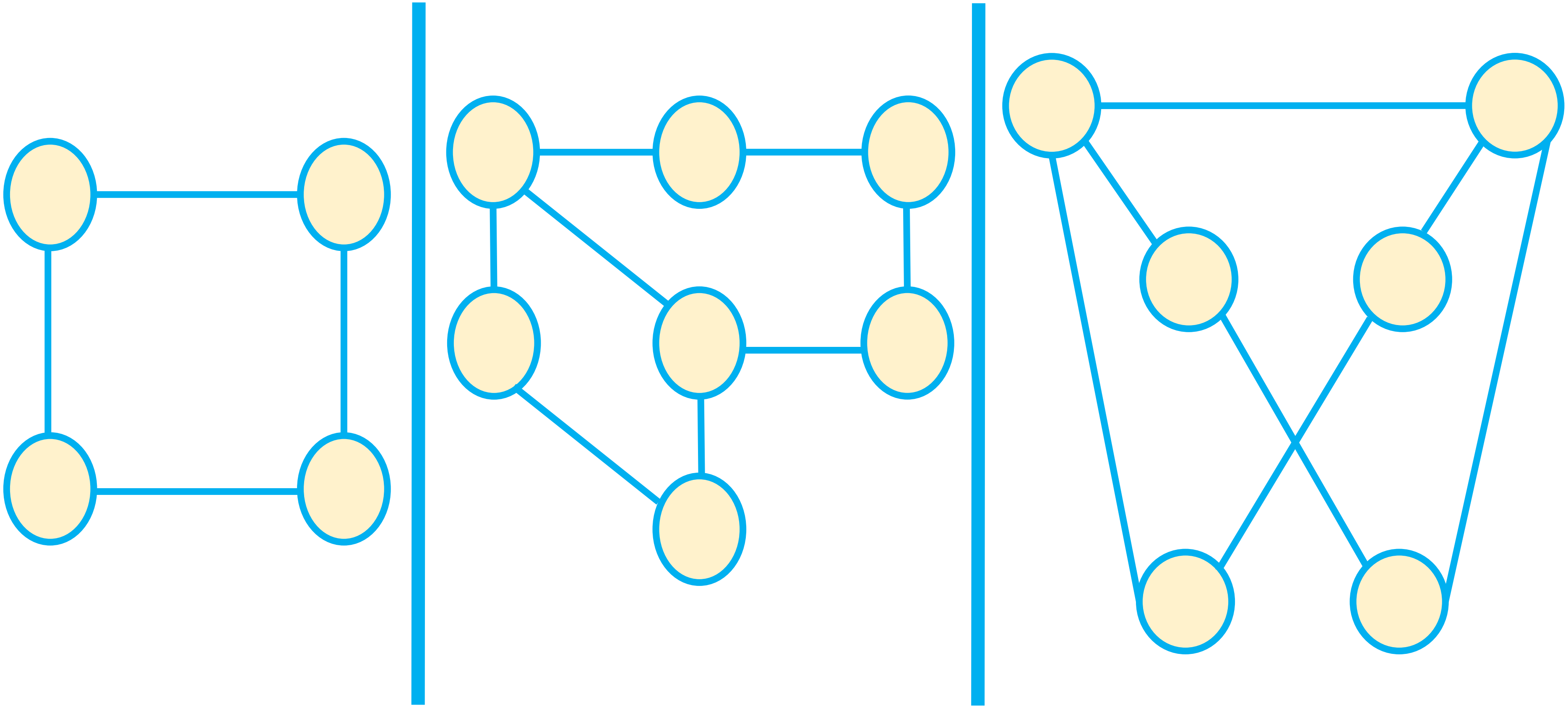
- Here we will specifically talk about ***1-edge-cooonected*** and ***2-edge-connected*** graphs (k =1, and k = 2)

- Though graphs with k >= 3 also exist, ***1 and 2-edge-connected*** graphs are more common

2-Edge Connected Graph

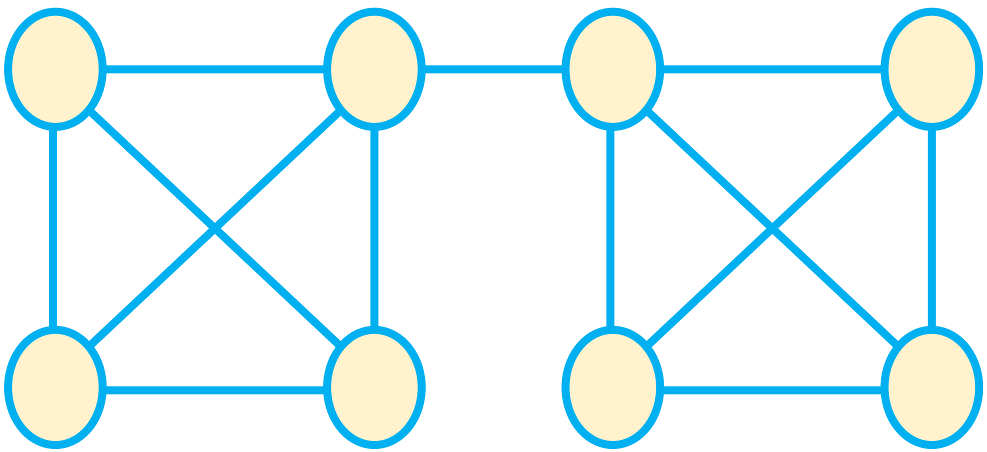
- As per the definition of ***k-edge-connected*** graphs, how many edges can we remove from a ***2-edge-connected*** graph without jeopardizing its connectivity?

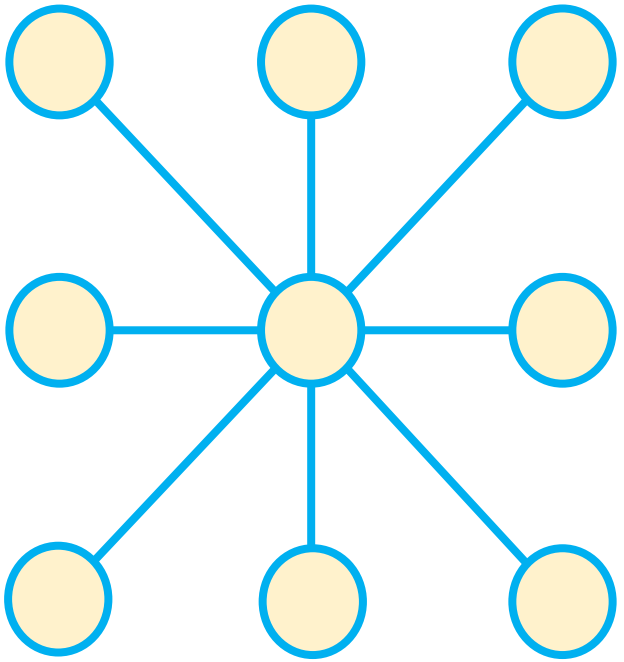
2-Edge Connected Graph



Examples of 2-edge-connected Graphs

2-Edge Connected Graph





Are these graphs 2-edge-connected?

Bridge

- A ***bridge*** in a connected, undirected graph G is an edge in G that, if removed, disconnects G.

Bridge and 2-edge connected graphs

- **Thus,** An undirected graph G is 2-edge connected *iff* it is connected and has no bridges.

Bridge and 2-edge connected graphs

- **Thus,** An undirected graph G is 2-edge connected *iff* it is connected and has no bridges.

- To understand this, let’s take a closer look at 2-edge connected graphs

Bridge and 2-edge connected graphs

- **Theorem:** An undirected graph G is 2-edge connected *iff* it is connected and has no bridges.

- To understand this, let's take a closer look at 2-edge connected graphs

- 2-edge connectivity requires that:

1. There is a path between any pair of nodes in the graph, and
2. After deleting any single edge from the graph, there is still a path between any pair of nodes in the graph

2-Edge Connected Graph

- Let's try to understand 2-edge-connected graphs to see why is it true that deleting any edge from such graphs does not affect its connectivity.

Deleting from 2-edge connected graphs

- Pick a pair of nodes *u* and *v* in a 2-edge connected graph G.

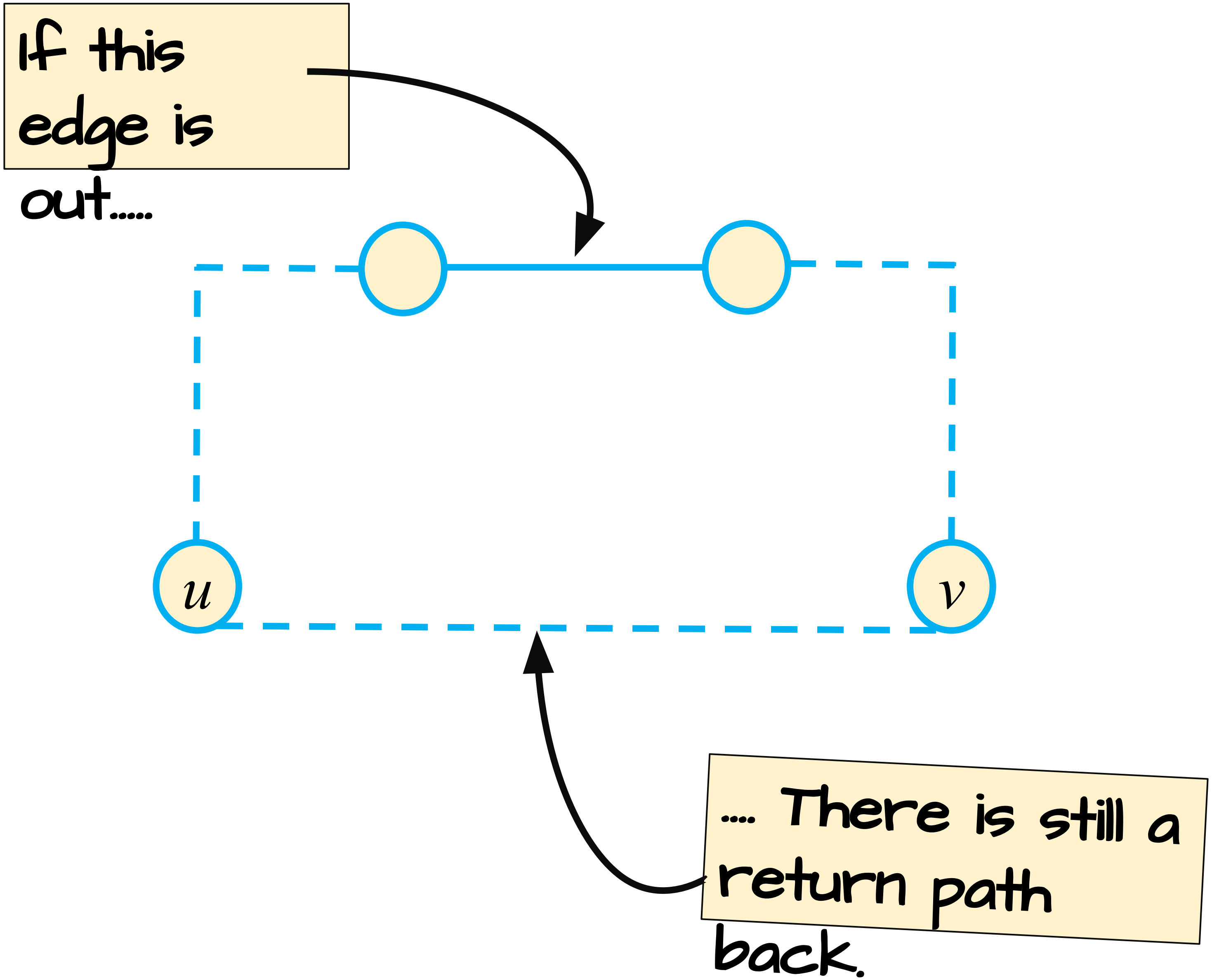
- If we delete an edge from G, *u* and *v* are still connected. Under what circumstances is that possible?

Deleting from 2-edge connected graphs

- Pick a pair of nodes *u* and *v* in a 2-edge connected graph G.
- If we delete an edge from G, *u* and *v* are still connected. Under what circumstances is that possible?
- Two things must have happened:

1. The deleted edge was not on the path between *u* and *v*
2. If it was, there must be an alternative route between *u* and *v*

The Key Insight



Connectivity and Cycle

- **Theorem:** Let $G = (V, E)$ be any graph containing a simple cycle C. Let $u, v \in V$ be nodes in G. if $u \leftrightarrow v$, then after deleting any single edge in C from graph G, it is still the case $u \leftrightarrow v$.

Edge Connected Graphs and Cycles

- **Theorem:** Let G be an undirected graph. If G is connected and every edge of G belongs to at least one simple cycle, then G is 2-edge-connected.

Edge Connected Graphs and Cycles

- **Theorem:** Let G be an undirected graph. If G is connected and every edge of G belongs to at least one simple cycle, then G is 2-edge-connected.

- **Proof:** Consider any undirected connected graph $G=(V, E)$, where each edge lies on at least on simple cycle
  - We need to show that G is 2-edge connected.

  - To this, we need to show that G satisfies the two properties:

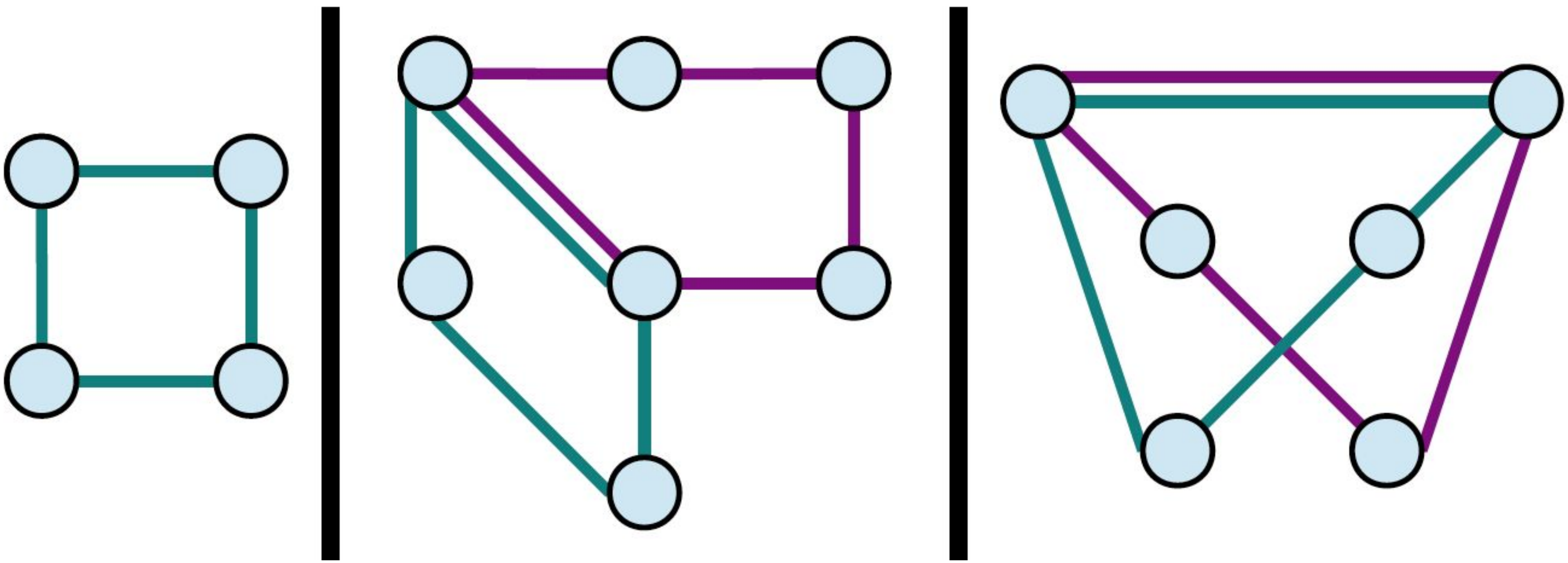
1. G is connected. This is true from our assumption
2. Removing a single edge leaves G connected

Edge Connected Graphs and Cycles

- **Theorem:** Let G be an undirected graph. If G is connected and every edge of G belongs to at least one simple cycle, then G is 2-edge-connected.
- **Proof:** Consider any undirected connected graph $G = (V, E)$, where each edge lies on at least on simple cycle
  - We need to show that G is 2-edge connected.
  - To this, we need to show that G satisfies the two properties:
  1. G is connected. This is true from our assumption
  2. Removing a single edge leaves G connected

- Consider any edge e.
- By assumption e lies on some simple cycle.
- Thus removing e leaves every pair of nodes in the cycle stay connected.
- Since our choice of e was arbitrary, this means that removing any e from G does not disconnect it.
- Thus G is 2-edge connected.

Now Look at These!



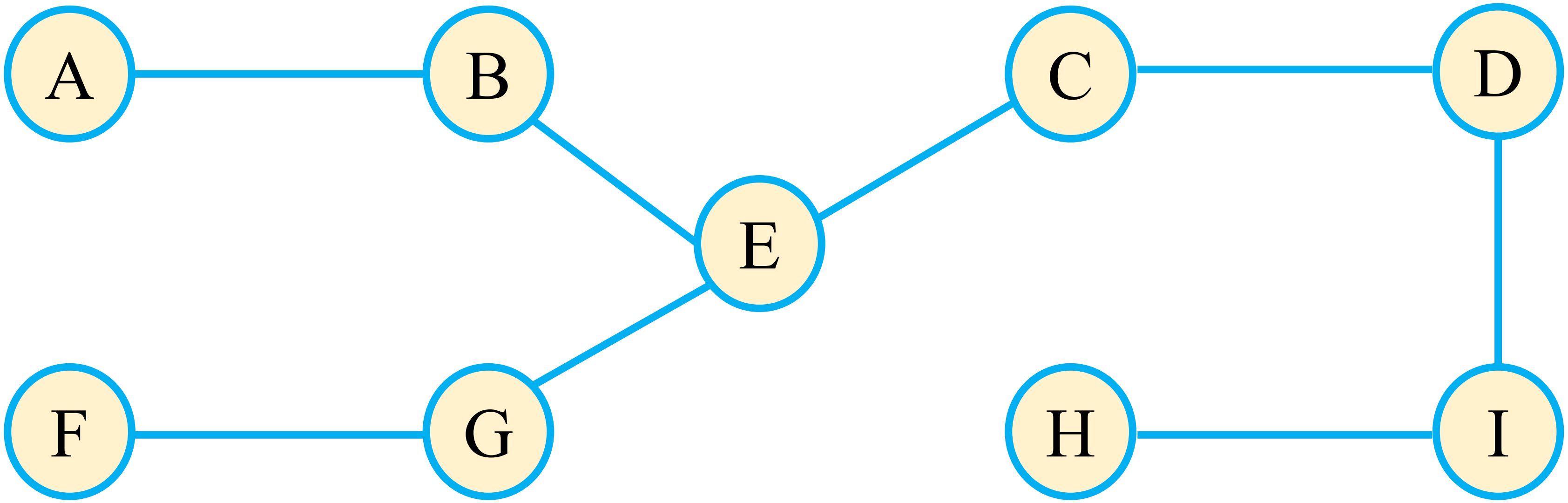
Edge Connected Graphs and Cycles

- In Other Words!

- **Theorem:** If G is 2-edge-connected, then every edge in G lies on a simple cycle.

Connected Graphs with No Redundancy

- **Example**

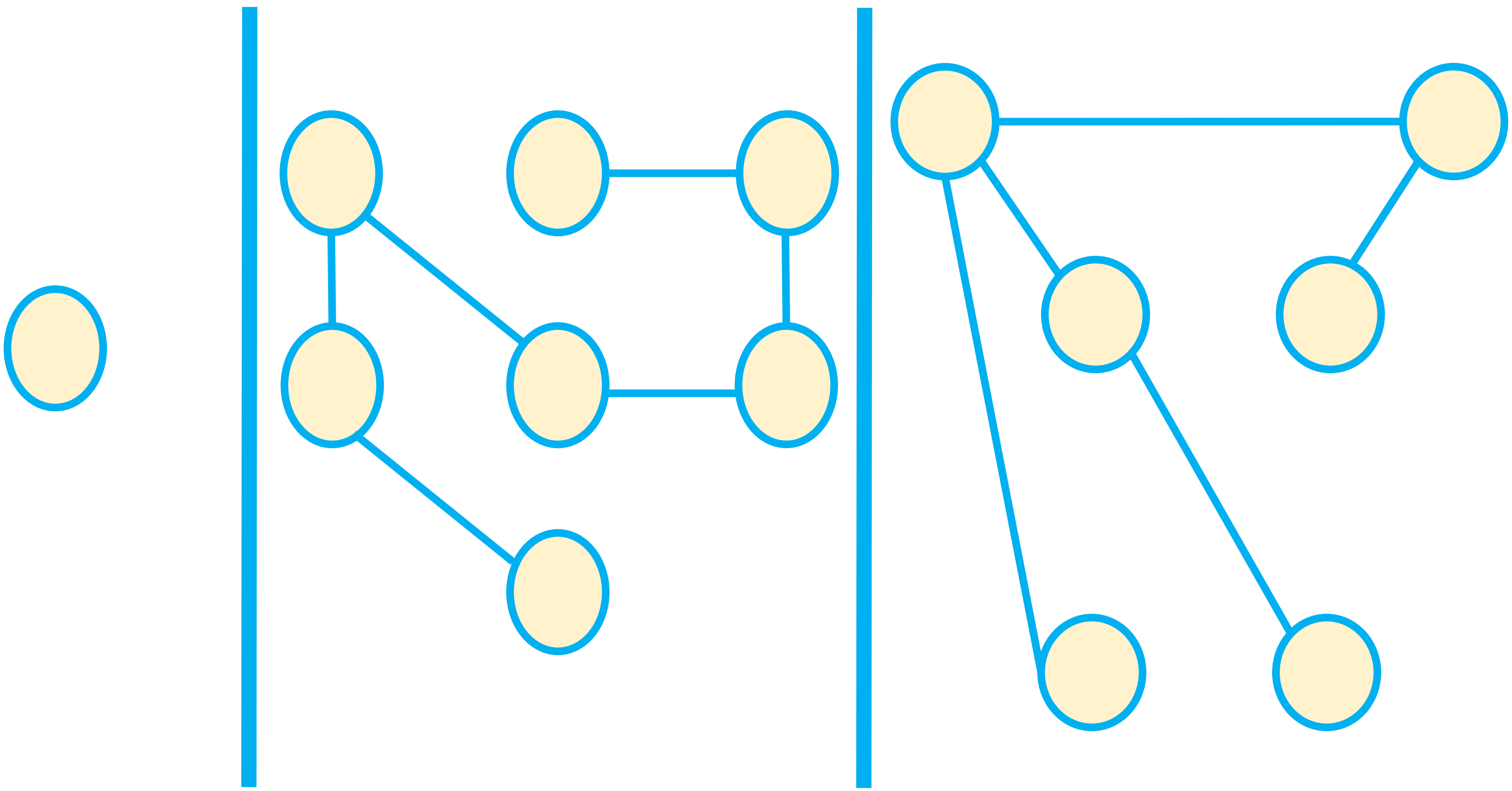


Minimally Connected Graphs

- An undirected graph G is called minimally connected if G is connected, but the removal of any edge from G leaves G disconnected.

Minimally Connected Graphs

- **Example:**



- **Note:** Not a single on of these contains a cycle.

Acyclic Graphs

- A graph is called acyclic if it contains no simple cycles.

Minimal Connectivity and Acyclic Graphs

- **Theorem:** If an undirected graph G is minimally-connected, then it is **connected** **and** **acyclic**.
- **Proof: By Contradiction:** Let's assume that G is minimally-connected but it is **not connected** **or** **it is not acyclic**,
- But every minimally-connected graph is connected, thus it cannot be the case that G is not connected. **(1)**
- So G must be acyclic.
- This means that G contains a simple cycle. **(2)**

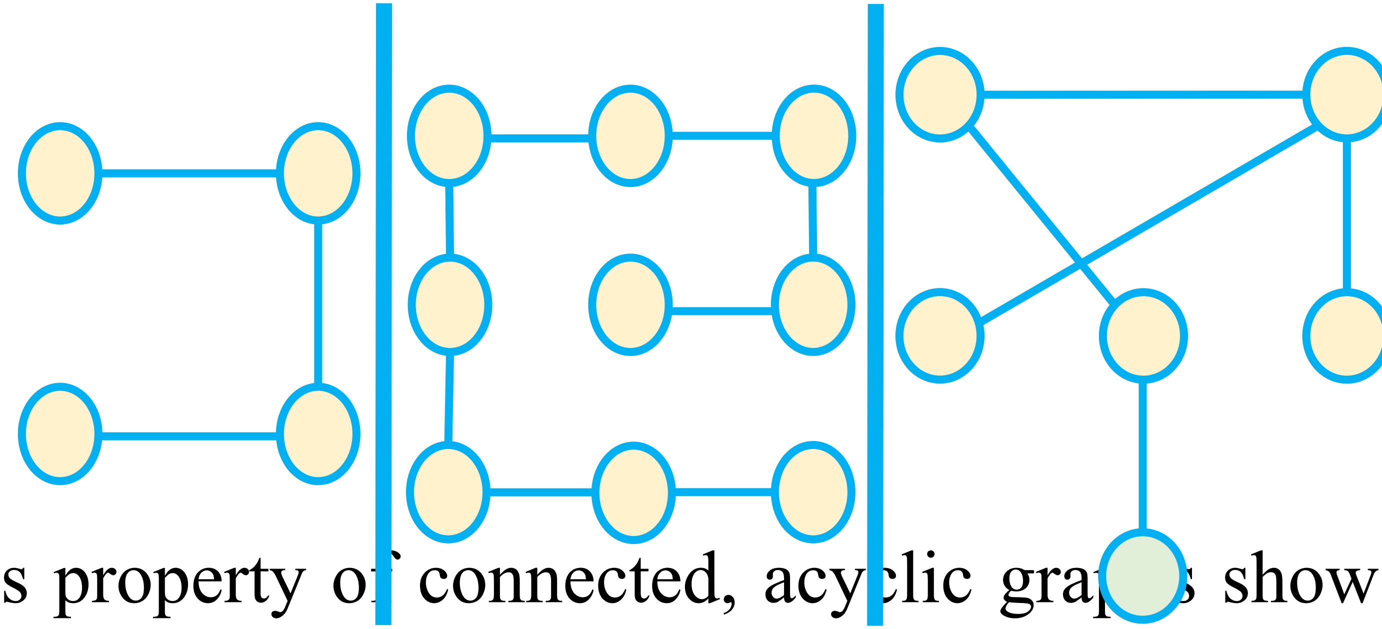
Minimal Connectivity and Acyclic Graphs

- **Theorem:** If an undirected graph *G* is minimally-connected, then it is connected and acyclic.
- **Proof: By Contradiction:** Let's assume that *G* is minimally-connected but it is not connected or it is not acyclic,
- But every minimally-connected graph is connected, thus it cannot be the case that *G* is not connected. **(1)**
- So *G* must be acyclic.
- This means that *G* contains a simple cycle. **(2)**

- From **(1) and (2),** it can be deduced that removing an edge e from *G*, leaves *G* connected. **(3)**
- **(3)** contradicts our assumption that *G* is minimally-connected
- Since we have reached a contradiction, so our assumption must be wrong. Thus if *G* is minimally connected, then it must be connected and acyclic.

Further Properties of such Graphs

- What happens if we try adding edges into such graphs?



- This property of connected, acyclic graphs shows that this graphs are in a sense, as large as they can get while still being acyclic. Adding any missing edge into the graph is guaranteed to give us a cycle.

Maximally Acyclic Graph

- An undirected graph G is called maximally acyclic if it is acyclic, but the addition of any edge introduces a simple cycle.

- **Theorem**: If an undirected graph G is connected and acyclic, then it is maximally acyclic.

- **Proof**: Consider $G = (V, E)$, an undirected, connected, acyclic graph.

- Consider nodes u and v, such that $(u,v) \notin E$. We will prove that adding the edge (u,v) will produce a simple cycle.

Maximally Acyclic Graph

- An undirected graph G is called maximally acyclic if it is acyclic, but the addition of any edge introduces a simple cycle.
- **Theorem:** If an undirected graph G is connected and acyclic, then it is maximally acyclic.
- **Proof:** Consider $G = (V, E)$, an undirected, connected, acyclic graph.
- Consider nodes u and v, such that $(u,v) \notin E$. We will prove that adding the edge (u,v) will produce a simple cycle.
- Since G is connected, there must be simple path $(u, x_1, x_2, \ldots, x_n, v)$, where none of the intermediate nodes are equal to either u or v.
- Now, consider adding the edge (u, v). The above path will extend to $(u, x_1, x_2, \ldots, x_n, v, u)$, which a simple cycle.
- Since our choice of edge was arbitrary, hence if G is acyclic, it proves that G is maximally acyclic.

What have we done so far?

- G is minimally connected → G is connected and acyclic → G is maximally acyclic
- As it happens to be, the reverse is also true (you are encouraged to prove them yourselves)
- That is,
- G is maximally acyclic → G is connected and acyclic → G is minimally connected

Overall

- **Theorem:** Let G be an undirected graph. The following are all equivalent:
  1. G is minimally connected.
  2. G is connected and acyclic.
  3. G is maximally acyclic

- These graphs are incredibly important in computer science, and are called trees.
- A tree is an undirected graph that is minimally connected. Equivalently, a tree is an undirected graph that is connected and acyclic, or an undirected graph that is maximally acyclic.

Trees

- Trees are the basis of many important data structures, such as binary search trees.

- They also play an important role in certain types of algorithmic analysis, and many problems involving trees are known to be computationally simple.

Trees

- We just learned three properties of a Tree (minimal connectivity, connectivity & acyclicity, and maximal acyclicity)

- There are many other properties, for example,

  Let $G = (V, E)$ be an undirected graph. Then any pair of nodes $u, v \in V$ have exactly one simple path between them iff G is a tree.

- You will cover them in detail in the next semester.

- You are encouraged to do the further reading on Trees by yourself.

More on Walks, Trails, Paths and Circuits

Definitions

• **Definition**

Let G be a graph, and let v and w be vertices in G.

A **walk from *v* to *w*** is a finite alternating sequence of adjacent vertices and edges of G. Thus a walk has the form

$$v_0 e_1 v_1 e_2 \cdots v_{n-1} e_n v_n,$$

where the v's represent vertices, the e's represent edges, $v_0 = v$, $v_n = w$, and for all $i = 1, 2, \ldots n$, v_{i-1} and v_i are the endpoints of e_i. The **trivial walk from *v* to *v*** consists of the single vertex v.

A **trail from *v* to *w*** is a walk from v to w that does not contain a repeated edge.

A **path from *v* to *w*** is a trail that does not contain a repeated vertex.

A **closed walk** is a walk that starts and ends at the same vertex.

A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge.

A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

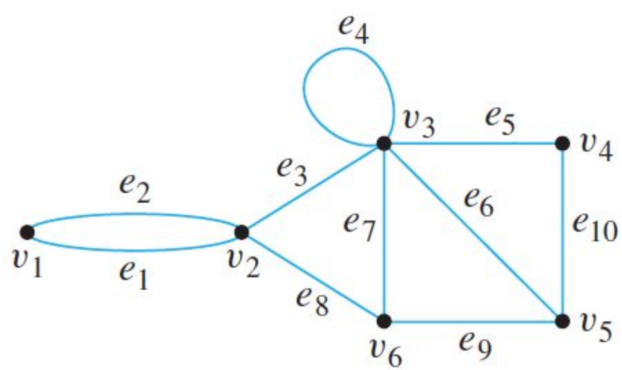
Summary

	Repeated Edge?	Repeated Vertex?	Starts and Ends at Same Point?	Must Contain at Least One Edge?
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple circuit	no	first and last only	yes	yes

Example

In the graph below, determine which of the following walks are trails, paths, circuits, or simple circuits.

a. $v_1 e_1 v_2 e_3 v_3 e_4 v_3 e_5 v_4$ b. $e_1 e_3 e_5 e_5 e_6$ c. $v_2 v_3 v_4 v_5 v_3 v_6 v_2$
d. $v_2 v_3 v_4 v_5 v_6 v_2$ e. $v_1 e_1 v_2 e_1 v_1$ f. v_1



Euler Trails and Circuits

- How do we create efficient routes for the delivery of goods and services (such as mail delivery, pizza delivery, garbage collection) along the streets of a city?
- These types of problems are known as routing problems.
- In terms of graph theory, they are known as Euler Circuit Problems.

Two Basic Questions

- Is an actual route possible?
- If the answer is yes, then which one is the optimal route?
- Measure of “being optimal” can be cost, distance, time and so on.

Euler Circuit Problems

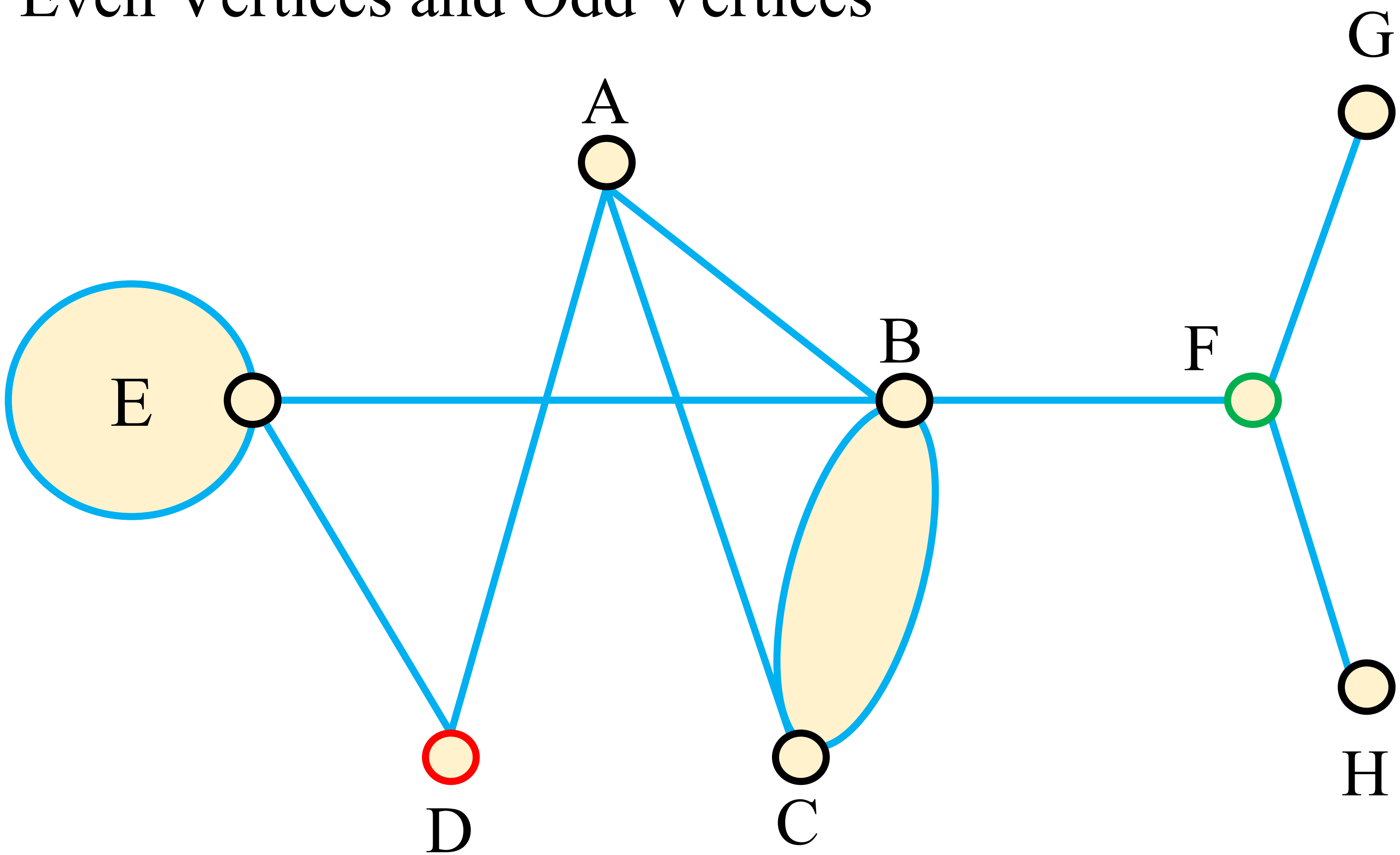
- The common thread in all such problems is "the exhaustion requirement"

- That means that the route must pass through … everywhere.

- In other words, Euler circuit problems are about finding "exhaustive routes" --- that pass through every single street (or bridges, or lanes, or highways) within a predefined area (be it a town, or a city).

Euler Trail & Circuit

- Even Vertices and Odd Vertices



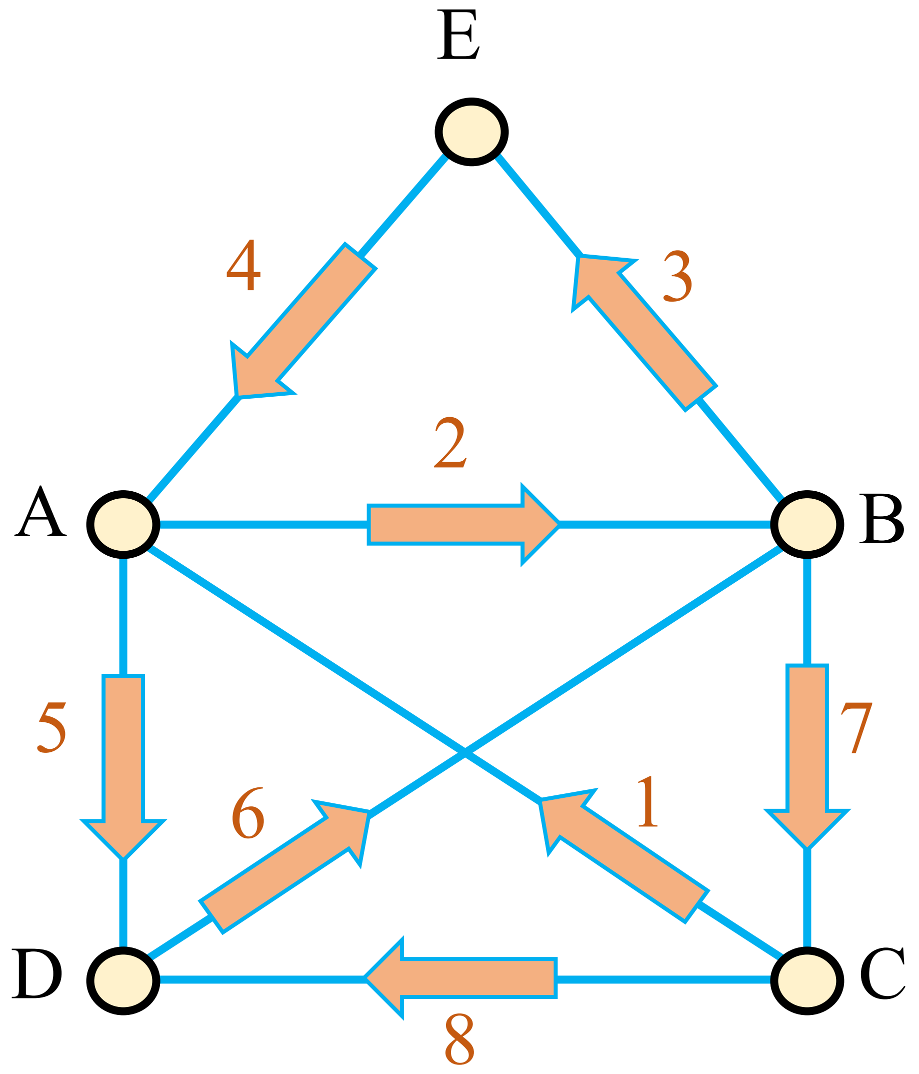
Euler Trail & Circuit

- **Euler Trail (*u* , *v*):**
  - It is a trail in a connected graph G that travels from *u* to *v*, passes through all the edges of G exactly once and all the vertices of G at least once.

- **Euler Circuit:**
  - A circuit in a connected graph G that uses every vertex of G at least once, and every edge of G exactly one.

- **Note:** A connected graph cannot have both Euler trail and Euler circuit. It can have one or the other or neither.

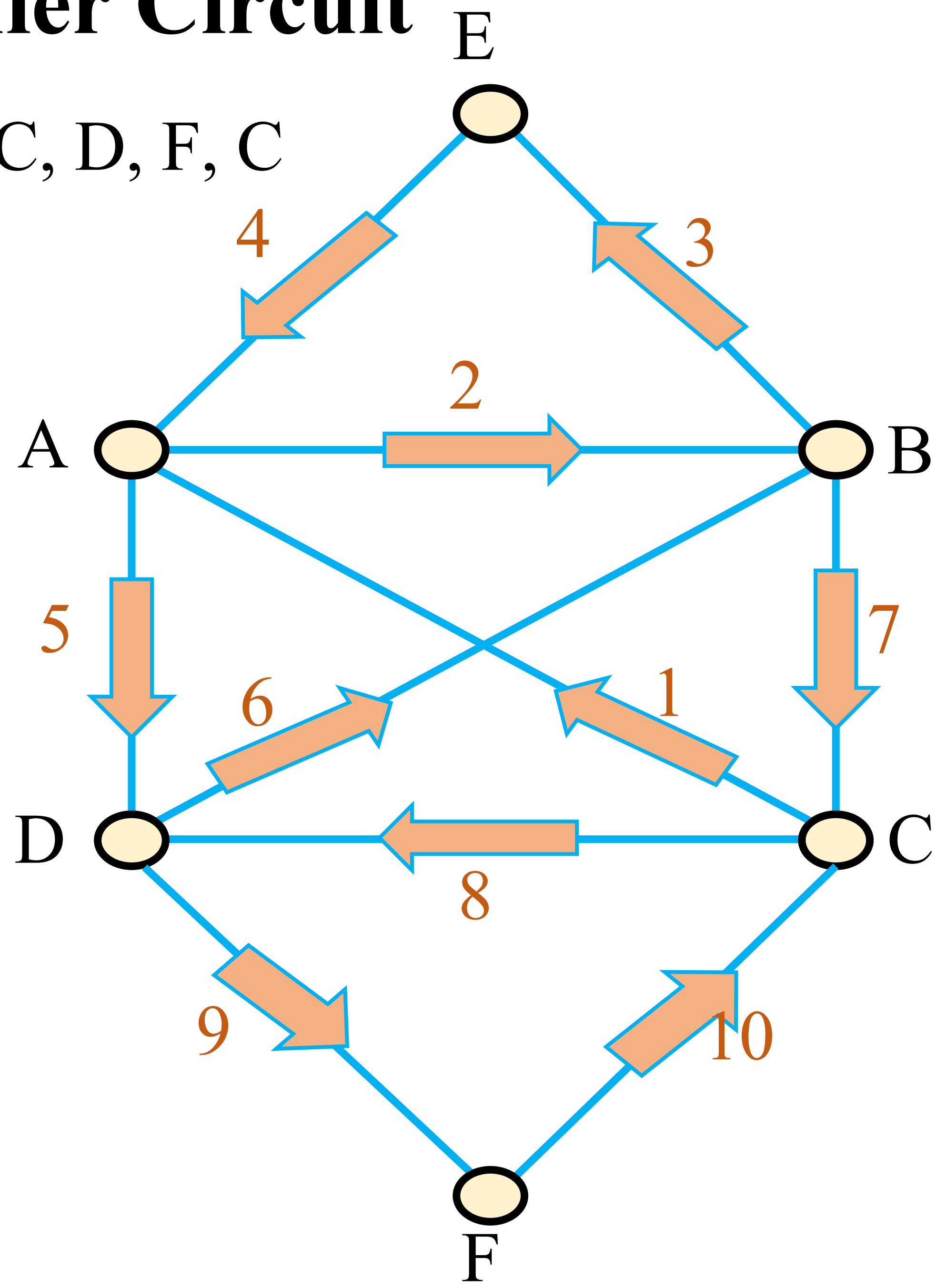
Euler Trail



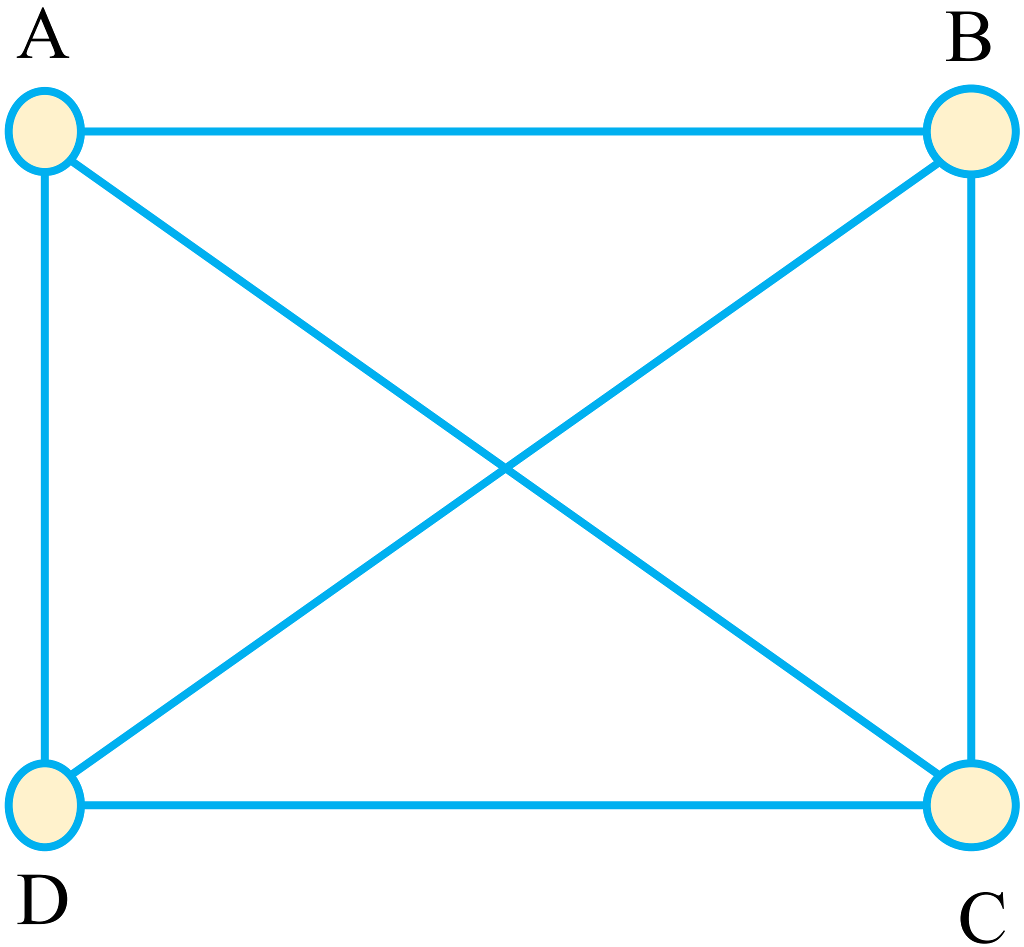
- C, A, B, E, A, D, B, C, D

Euler Circuit

- C, A, B, E, A, D, B, C, D, F, C



A graph with Neither



Euler’s Theorem

- In this section we are going to develop the basic theory that will allow us to determine if a graph has a Euler circuit, a Euler trail, or neither.

Euler's Circuit Theorem

- If a graph G is connected and every vertex is even, then it has a Euler circuit. If G has any odd vertices, then it does not have a Euler circuit.

Summary Euler’s Circuit Theorem

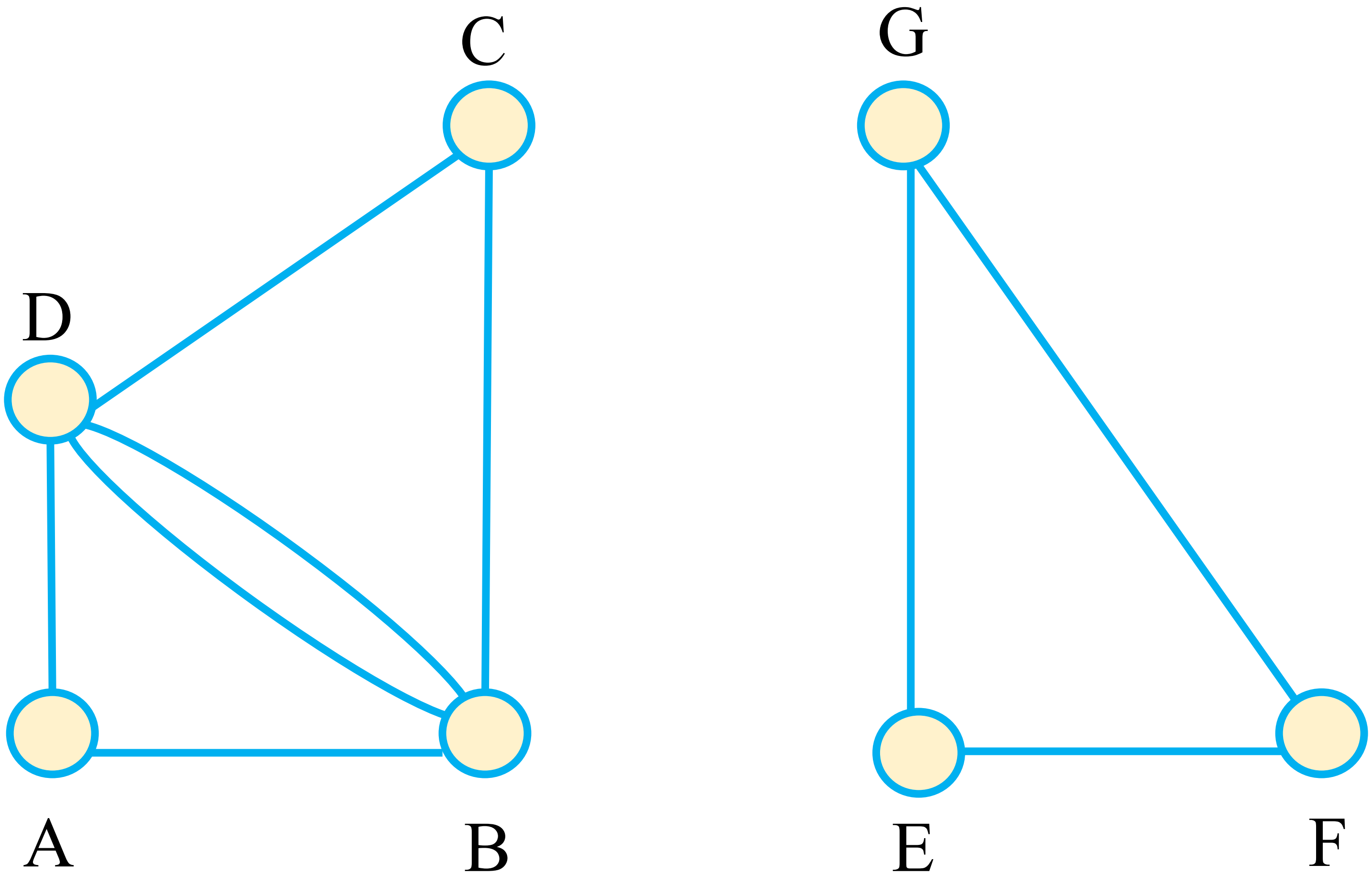
- Think of an arbitrary vertex *v* in a connected graph *G*

- A Euler circuit passing through *v* can take us there multiple times, but each time we arrive at *v* via an unvisited edge and leave via another unvisited edge. Thus a single visit to *v* consumes 2 edges.

- We can keep doing this as long as *v* is an even vertex. If it was odd, at some point we are going to come into it and not be able to get out!

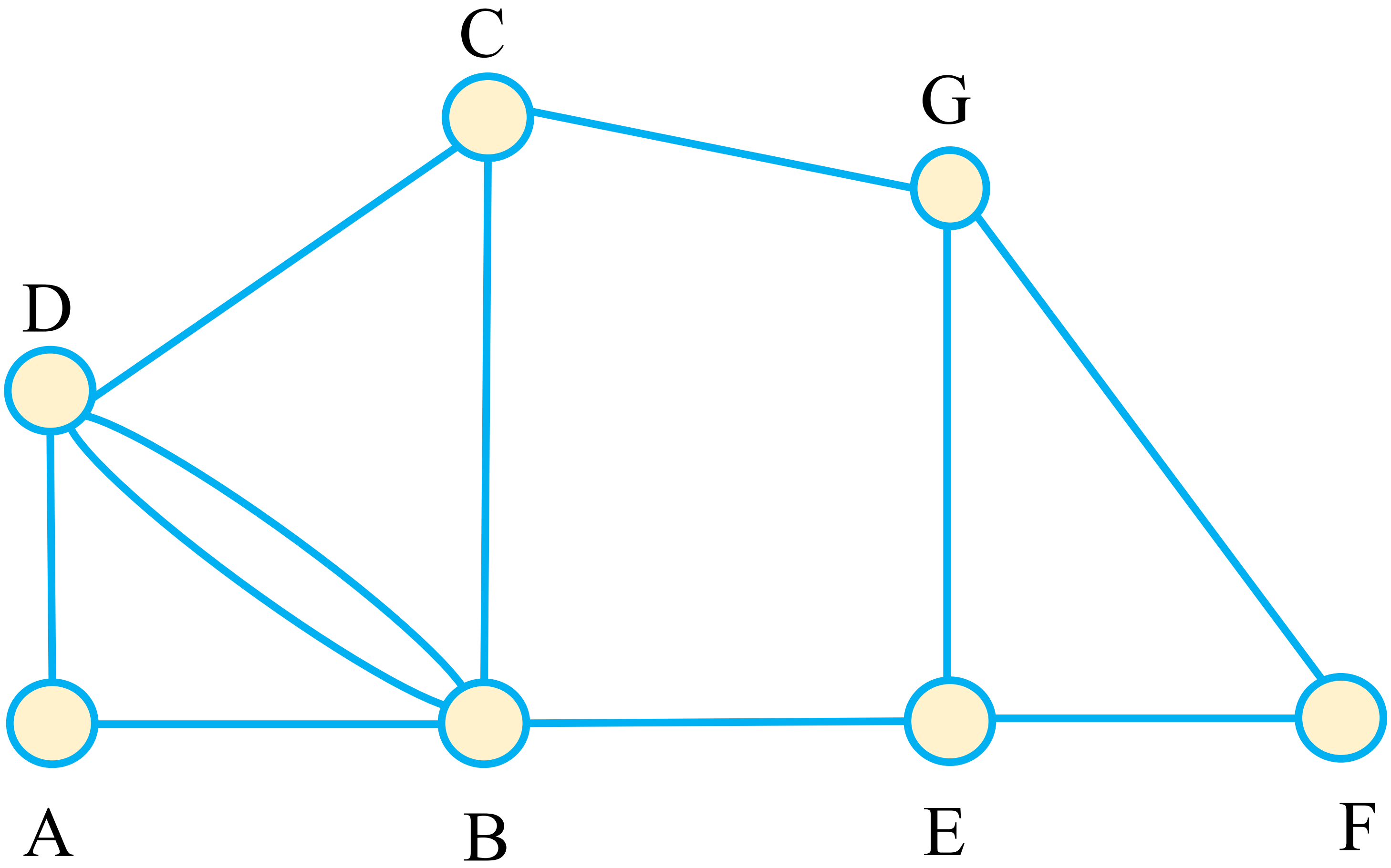
Apply Euler's Circuit Theorem

- Does this graph contain a Euler circuit.



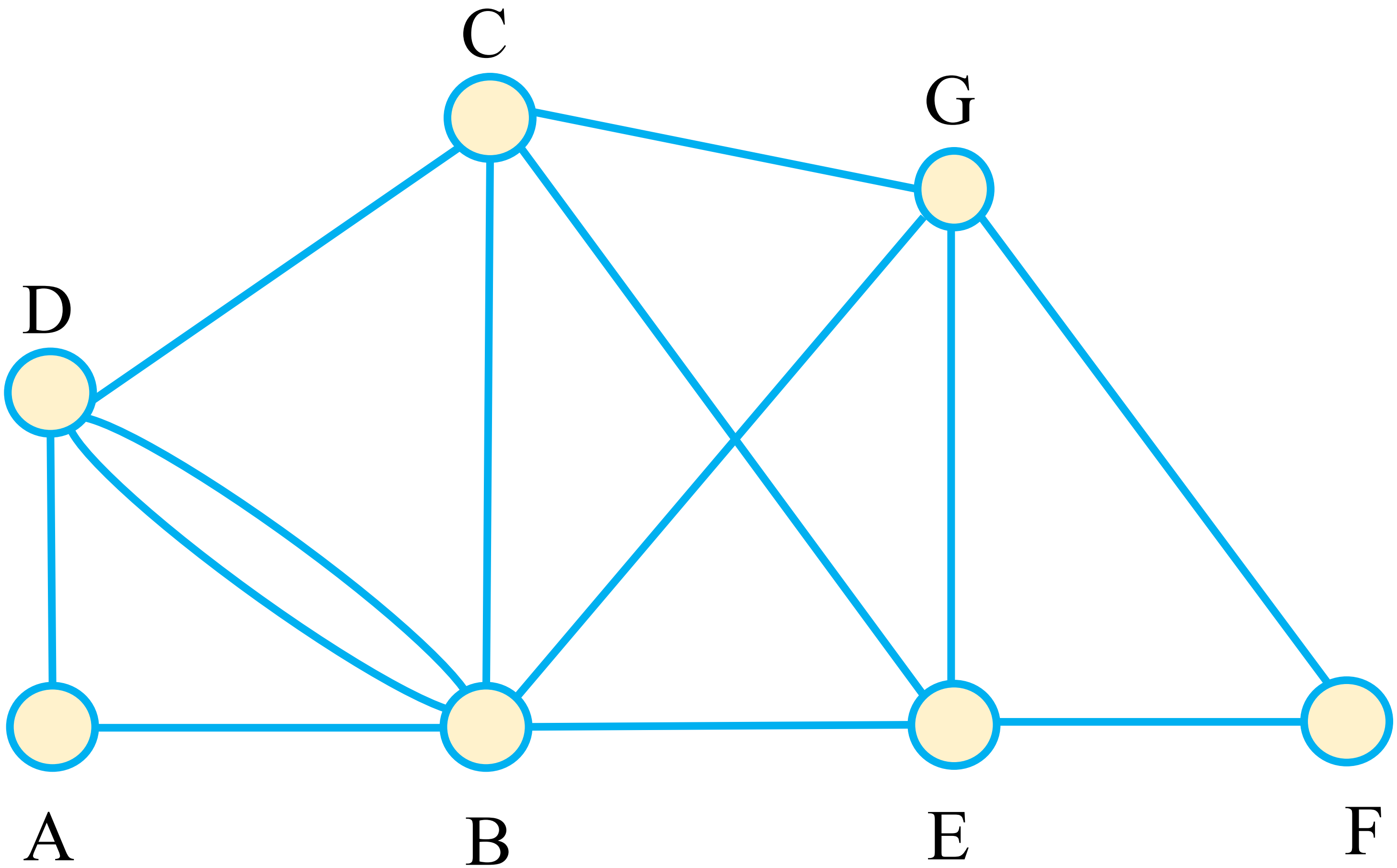
Apply Euler's Circuit Theorem

- Does this graph contain a Euler circuit.



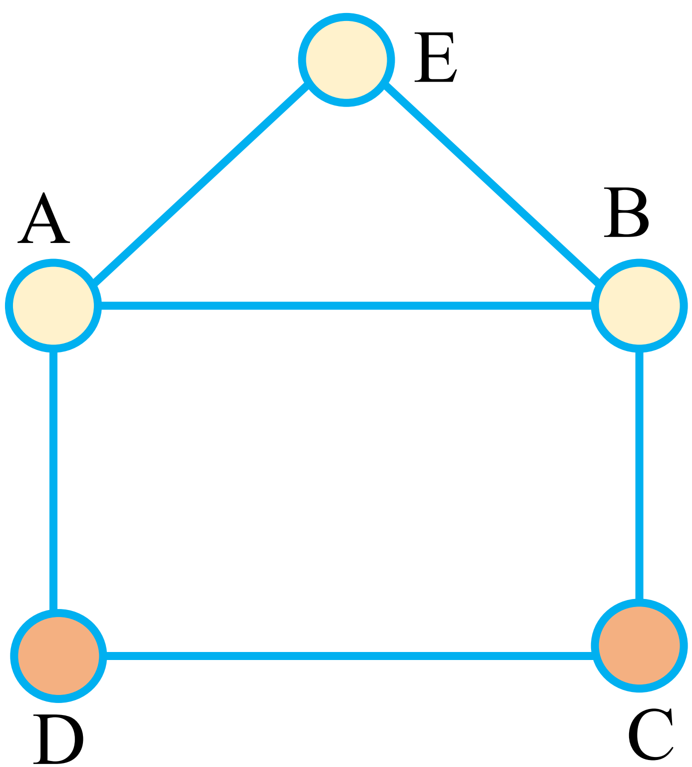
Apply Euler's Circuit Theorem

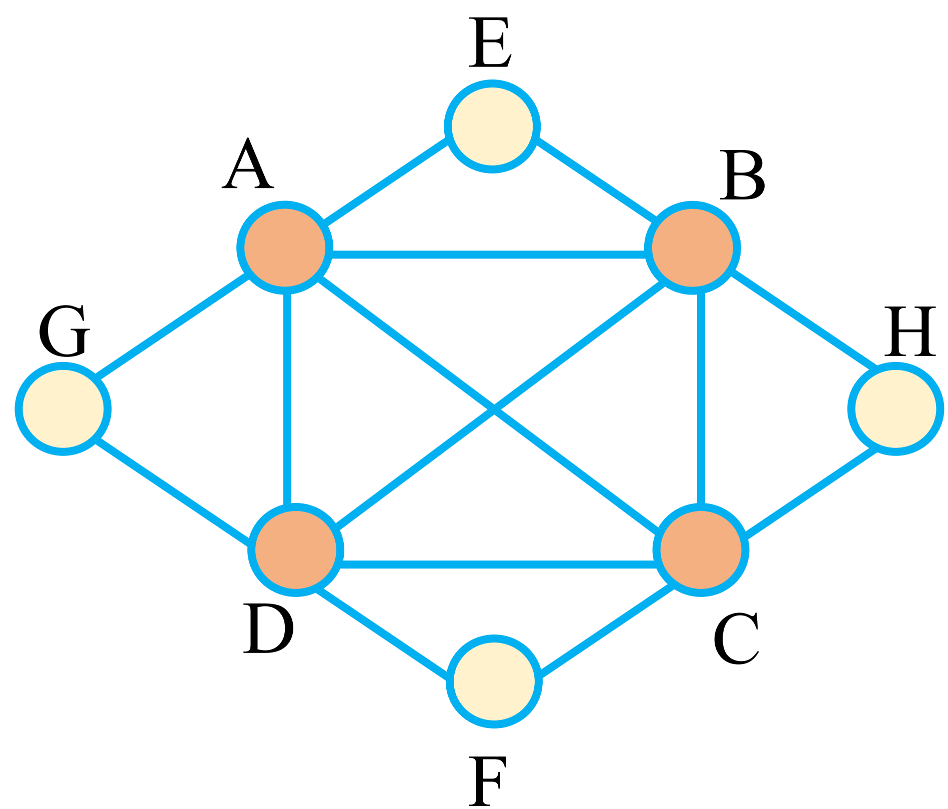
- Does this graph contain a Euler circuit.



Apply Euler's Trail Theorem

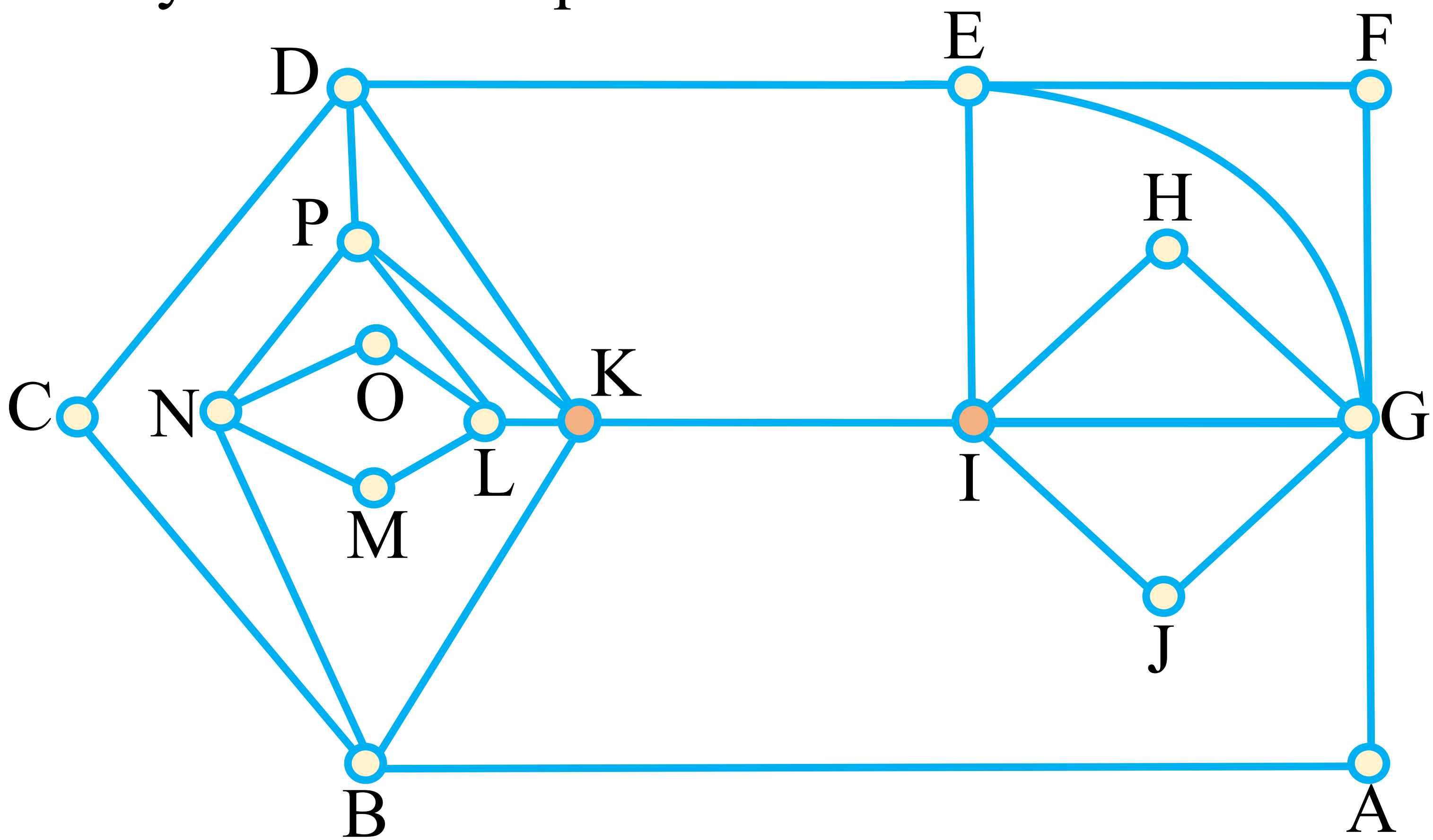
- If a graph G is connected, and has exactly two odd vertices, then G has a Euler trail. Any such trail must start at one of the odd vertices and end at the other one. If G has more than two odd vertices, it does not have a Euler trail.





Euler's Theorem

- The full power of Euler's theorems is best understood when graphs we are dealing with bigger graphs.
- **For example,** here is an average size graph, but it is already difficult to spot a Euler circuit or a trail



Euler's Third Theorem

- Circuit theorem: deals with graph having zero odd vertices
- Trail Theorem: deals with graph having exactly two or more odd vertices
- Missing piece: a graph with just one odd vertex

➢ Euler's third theorem rules out "missing piece"

Euler's Sum of Degree Theorem

- The sum of the degrees of all the vertices of a graph equals twice the number of edges; therefore, it is an even number.

- A graph always has an even number of odd vertices.

Intuition Behind the Theorem

- Consider $G = (V, E)$. Consider nodes u and v, such that $(u,v) \in E$.
- (u,v) contributes once to the degree of vertex u and once to the degree of vertex v.
- Thus, (u,v) makes a total contribution of 2 to the sum of the degrees of G.
- Thus, the sum of the degrees of all the vertices of G is twice the number of edges.

Intuition Behind the Theorem

- We just showed that the total sum is an even number. Therefore, it is impossible to have just one odd vertex, or three odd vertices, or five odd vertices, and so on.

- In other words, the odd vertices of a graph always come in twos.

Summary of Euler's Three Theorems

Number of Odd Vertices	Conclusion
0	G has Euler Circuit
2	G has Euler Trail
4, 6, 8, …	G has Neither
1, 3, 5, …	Better go back and double check! This is impossible!

Hamiltonian Paths and Circuits

- Euler trails and Euler circuits are all about finding trails and circuits that include every edge of the graph exactly once and every vertex of the graph at least once.

- Hamiltonian paths and Hamiltonian circuits (cycle) are all about finding paths and circuits that include every vertex of the graph exactly once.

Hamiltonian Paths and Circuits

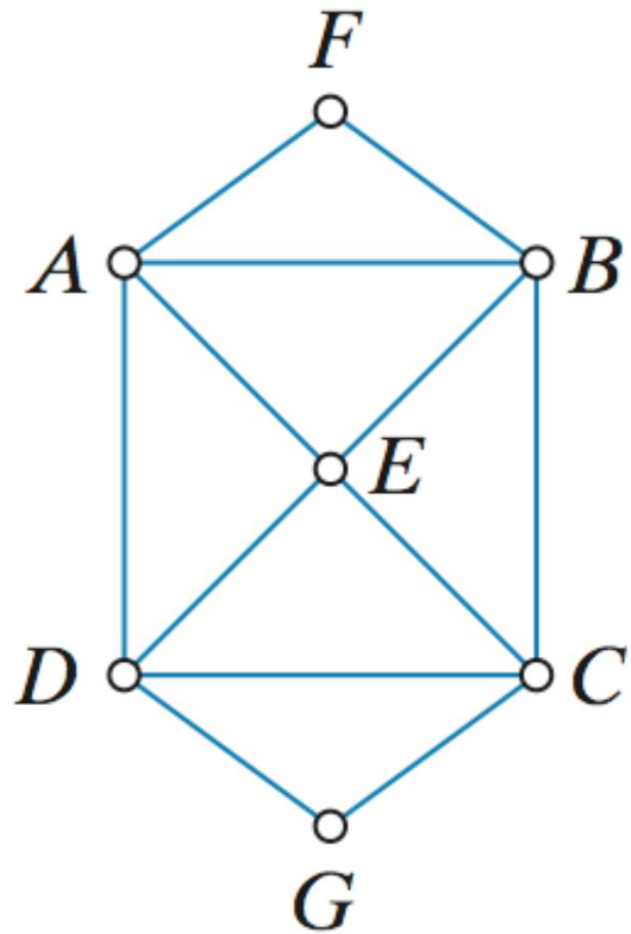
- If a graph has a Hamilton circuit, then it automatically has a Hamilton path – (the Hamilton circuit can always be truncated into a Hamilton path by dropping the last vertex of the circuit.)

- **For example,** the Hamilton circuit A, F, B, C, G, D, E, A can be truncated into the Hamilton path A, F, B, C, G, D, E.

- Contrast this with the mutually exclusive relationship between Euler circuits and paths: If a graph has a Euler circuit it cannot have a Euler path and vice versa.

Hamiltonian vs. Euler (1)

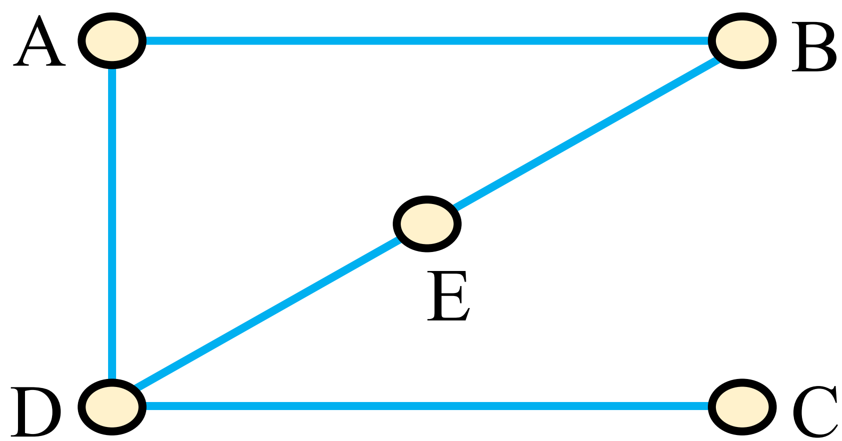
- The following graph
- Has Euler circuits,
  since all the vertices are even
- Has Hamiltonian Circuits



- For example:
  - A, F, B, C, G, D, E, A

Hamiltonian vs. Euler (2)

- The following graph
- Has no Euler circuits, but it does have Euler trail (C, D, E, B, A, D)
- Has no Hamiltonian circuits, but it does have Hamiltonian paths (A, B, E, D, C)



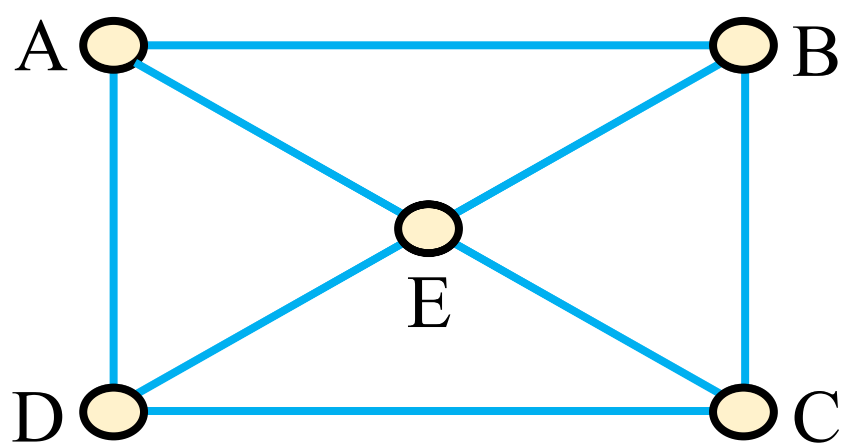
- Thus, a graph can have a Hamiltonian path but no Hamiltonian circuit!

Hamiltonian vs. Euler (3)

- The following graph

- Has neither Euler circuits nor trails (four odd vertices)

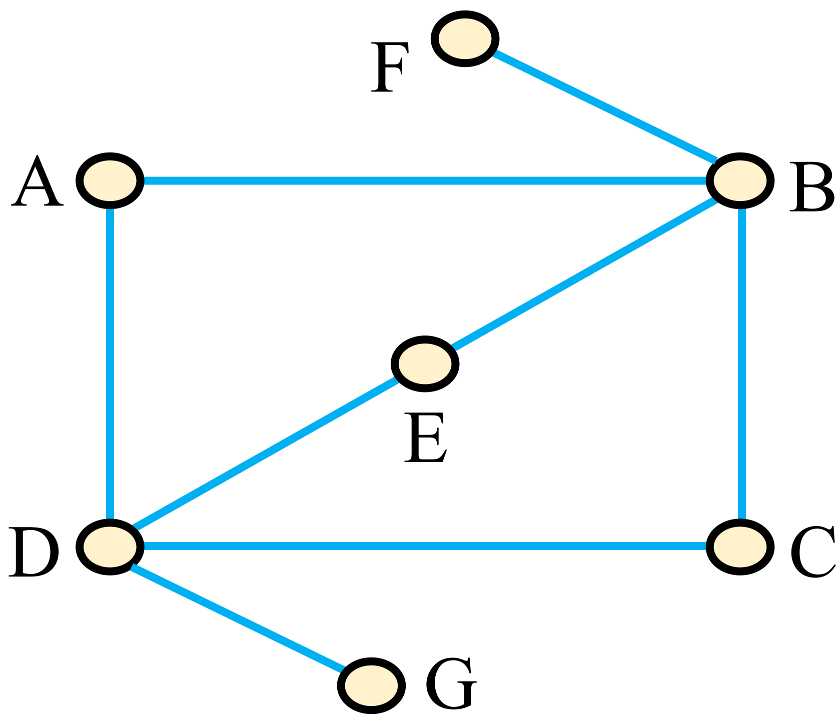
- Has Hamiltonian circuits (A, B, C, D, E, A)



- **From (2)** it follows that the graph has Hamiltonian paths (A, B, C, D, E)

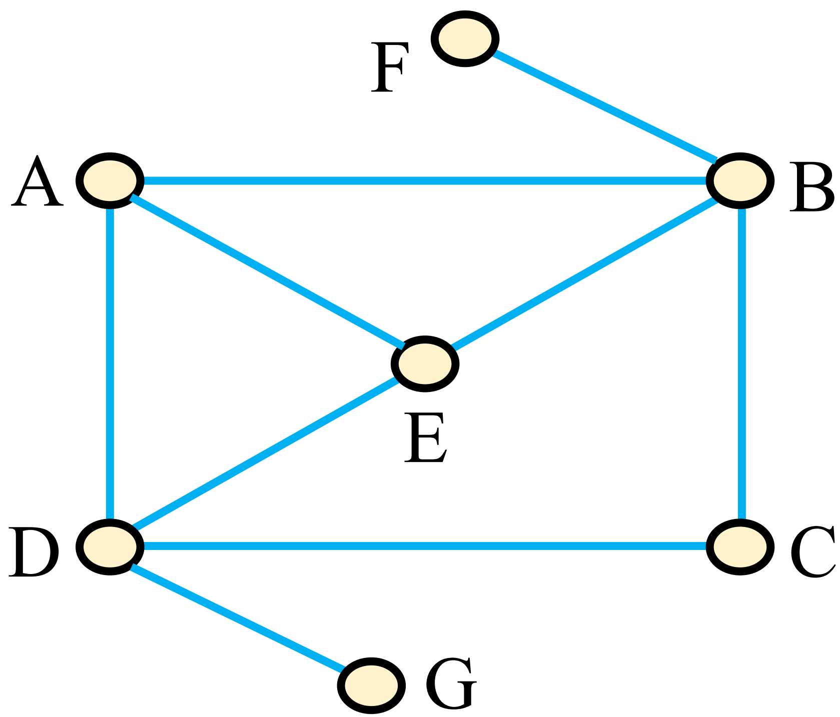
Hamiltonian vs. Euler (4)

- The following graph
- Has no Euler circuits but has Euler trail (F and G are the two odd vertices)
- Has neither Hamiltonian circuits nor paths



Hamiltonian vs. Euler (5)

- The following graph
- Has neither Euler circuits nor trails (more than 2 odd vertices)
- Has neither Hamiltonian circuits nor paths



Summary of Examples (1-5)

- The existence of a Euler circuit or a trail in a graph tells us nothing about the existence of a Hamiltonian circuit of a path in that graph

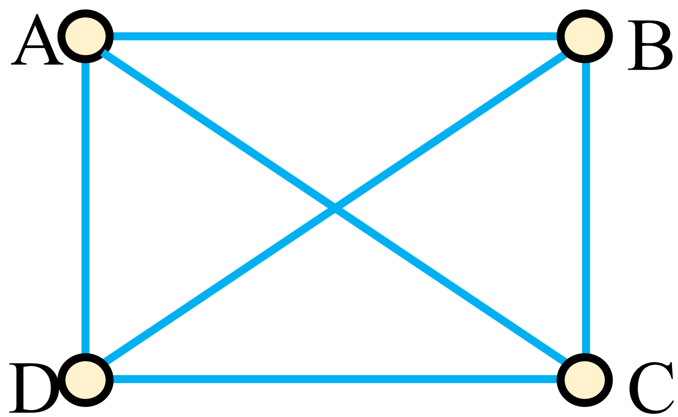
Lack of Theorems

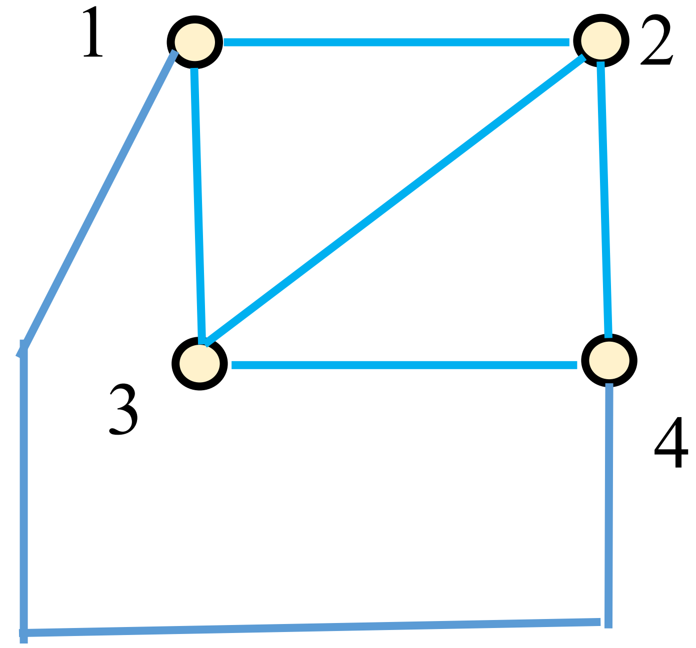
- Unlike Euler circuit and trail, there exist no “Hamiltonian circuit and path theorems” for determining if a graph has a Hamiltonian circuit, a Hamilton path, or neither.

- In today's tutorials, TAs will present a method for checking the presence of Hamiltonian circuit or a path in a graph.

Graph Isomorphism

- Look at the following two graphs.





- They are really the same graph.
- We say that these graphs are isomorphic

Graph Isomorphism

- Suppose G1 = (V1, E1) and G2 = (V2, E2). Then, G1 and G2 are isomosphic if.

  - There exists a bijection $f: V1 \rightarrow V2$

  - If for all *u* and *v* in V1
    if *u* and *v* are adjacent in G1

  - Then
    f(v1) and *f(v2)* are adjacent in G2

Though I won't include it in exam, graph Isomosphism is an important topic, so you are encouraged to read Sec 10.4 from Epp's book.