Discrete Mathematics

Graph Theory

Adil M. Khan

Professor of Computer Science

Innopolis University

"If I were again beginning my studies, I would follow the advice of **Plato** and start with mathematics!"
- Galileo Galilei -

Abstraction

- A key idea in Software Engineering

- Instead of writing multiple codes to create list of numbers, list of names, list of car components etc., build one single code to represent a “list of objects” then use that list to store data of all different types

Graph

- A powerful abstraction that allows to reason about the relations between "objects" and how individual "connections" between objects can give rise to larger structures

- **For example:**

  - Molecules in chemistry
  - Social networks like Facebook
  - The Internet
  - Highway systems
  - Our brain

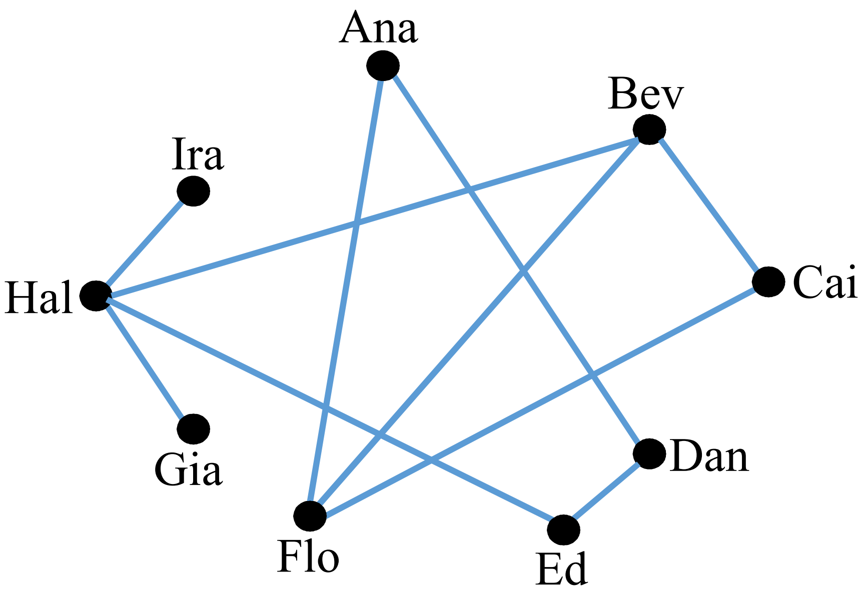
Graph

- **Example:**
  - ➢ Imagine an organization that wants to set up teams of three to work on some projects.
  - ➢ In order to maximize the efficiency, it is decided to put people on each team who had previous experience working together successfully.
  - ➢ The director asked the members to provide names of their past partners.

Graph

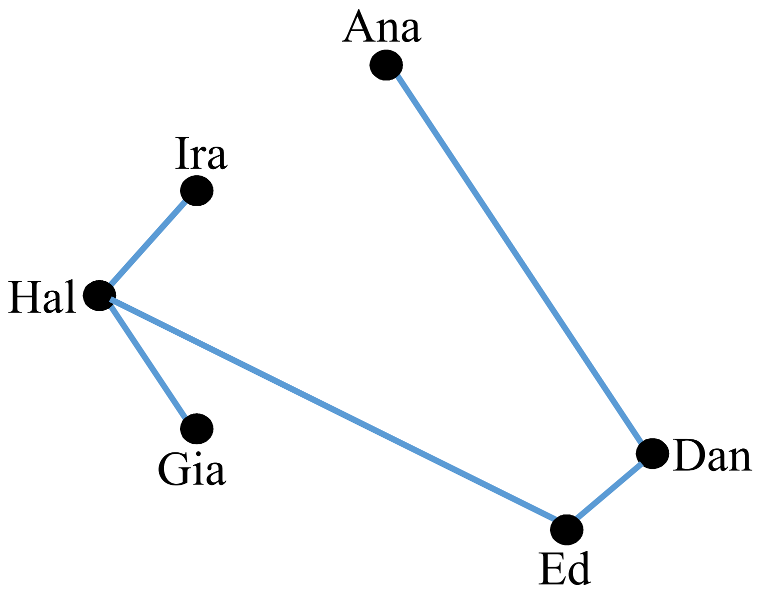
- **Example --- Cont.:**

Name	Past Partners
Ana	Dan, Flo
Bev	Cai, Flo, Hal
Cai	Bev, Flo
Dan	Ana, Ed
Ed	Dan, Hal
Flo	Cai, Bev, Ana
Gia	Hal
Hal	Gia, Ed, Bev, Ira
Ira	Hal



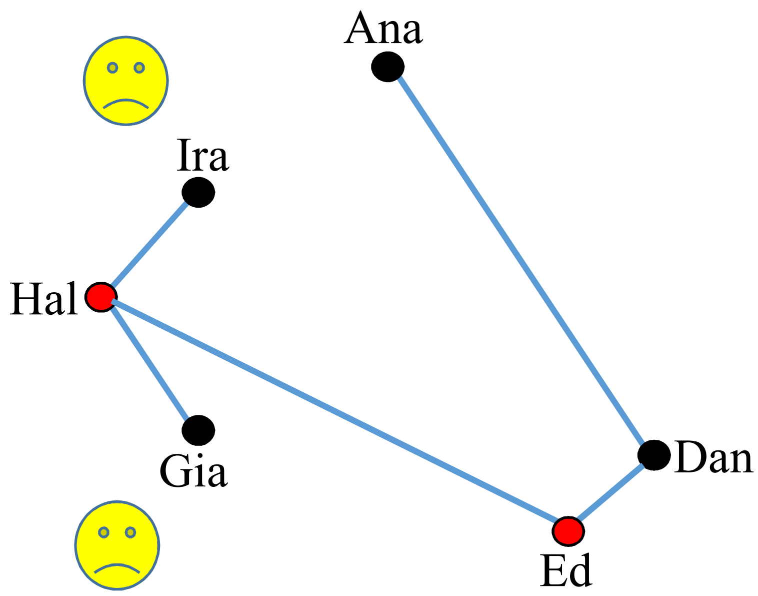
Graph

- **Example --- Cont.:**
  - From the diagram, it is easy to see that Bev, Cai, and Flo are a group of three past partners, and so they should form one of these teams.
  - The following figure shows the result when these three names are removed from the diagram



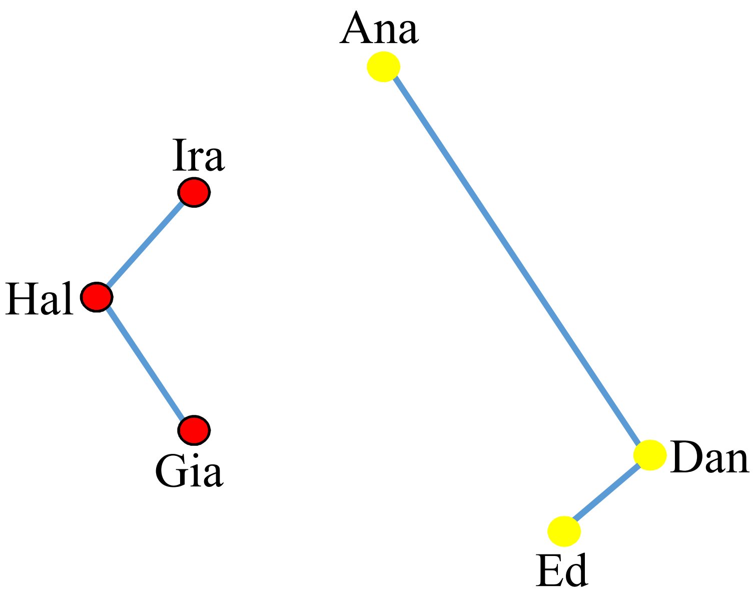
Graph

- **Example --- Cont.:**
  - Putting Hal and Ed on the same team



Graph

- **Example --- Cont.:**
  - Putting {Hal, Ira and Gira} and {Ed, Dan, Anna} on the same team – everyone is happy 



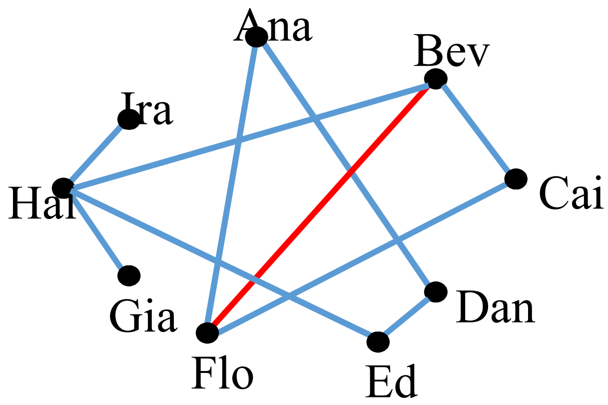
Graph

- **Properties:**

  - As you can see from the drawing, it is possible for two edges to cross at a point that is not a vertex.

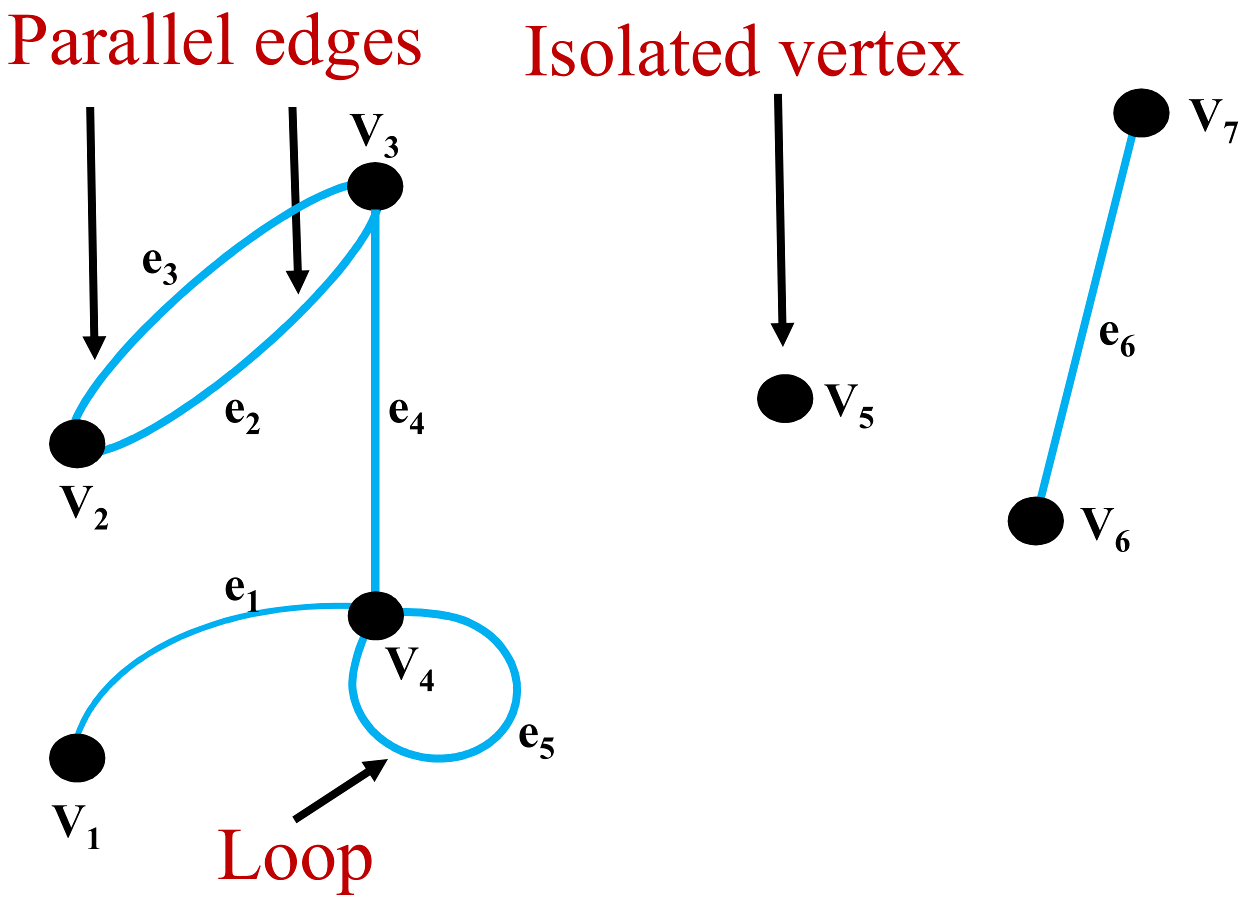
  - Note also that the type of graph described here is quite different from the “graph of an equation” or the “graph of a function.”

  - In general, a graph consists of a set of vertices and a set of edges connecting various pairs of vertices



Graph

- **Properties---Cont.:**
  - The edges may be straight or curved and should either connect one vertex to another or a vertex to itself, as shown below.



Graph Models

- **Social Networks:**
  - Graphs are extensively used to model social structures based on different kinds of relationships between people or groups of people.
  - These social structures, and the graphs that represent them, are known as ***social networks***.

Graph Models

- **Social Networks:**

  - In these graph models, individuals or organizations are represented by vertices; relationships between individuals or organizations are represented by edges.

  - The study of social networks is an extremely active multidisciplinary area, and many different types of relationships between people have been studied using them.

Graph Models

- **Most Commonly Studied Social Networks:**
  - Acquaintanceship and Friendship Graphs
  - Influence Graphs
  - Collaboration Graphs

Graph Models

- **Communication Networks:**

  - We can model different communications networks using vertices to represent devices and edges to represent the particular type of communications links of interest.

  - **For example, Call Graphs**

Graph Models

- **Information Networks:**

  - Graphs can be used to model various networks that link particular types of information.

- **For example:**

  - The Web Graph

  - Citation Graphs

Graph Models

- **Software Design Application:**
  - Graph models are useful tools in the design of software.
- **For example:**
  - Module Dependency Graphs
  - Precedence Graphs and Concurrent Processing

Graph Models

- **Transportation Network:**
  - We can use graphs to model many different types of transportation networks.
- **For example:**
  - Airline Routes
  - Road Networks

Graph Models

- **Biological Networks:**
  - Many aspects of biological sciences can be modeled using graphs.
- **For example:**
  - Niche Overlap Graphs in Ecology
  - Protein Interaction Graphs

Graph Models

- **Tournaments:**
  - Some examples of how graphs can also be used to model different kinds of tournaments include:
  - Round-Robin Tournaments
  - Single-Elimination Tournaments

Formal Definition of Graphs

- **Definitions:**

  - An **Unordered Pair** is a set $\{a\,, b\}$ representing the two objects a and b.

  - An **Ordered pair** is collection of two objects a and b in order. We denote the ordered pair consisting first of a, then of b as $(a\,, b)$. Two ordered pairs $(a_0\,, b_0)$ and (a_1, b_1) are equal if $a_0 = a_1$ and $b_0 = b_1$.

Formal Definition of Graphs

- **Definitions:**

  - A **Graph** G is an ordered pair $G = (V\ , E)$, where V is set of vertices and E is a set of edges.

  - An **undirected graph** is a graph $G = (V\ , E)$, where E is a set of **unordered** pairs.

  - A **directed graph (or Digraph)** is a graph $G = (V\ , E)$, where E is a set of **ordered** pairs.

Navigating a Graph

- **Definitions:**
  - A **path** in a graph $G = (V, E)$ is a series of nodes $(v_1, v_2, v_3, \dots, v_n)$ where for any $i \in \mathbb{N}$ with $1 \leq i \leq n$, there is an edge form v_i to v_{i+1}.
  - A **simple path** is a path with no repeated nodes.
  - A **cycle** is a path that starts and ends at the same node.
  - A **simple cycle** is a cycle that does not contain any duplicate nodes (except for the very last node) or duplicate edges.

Undirected Graphs

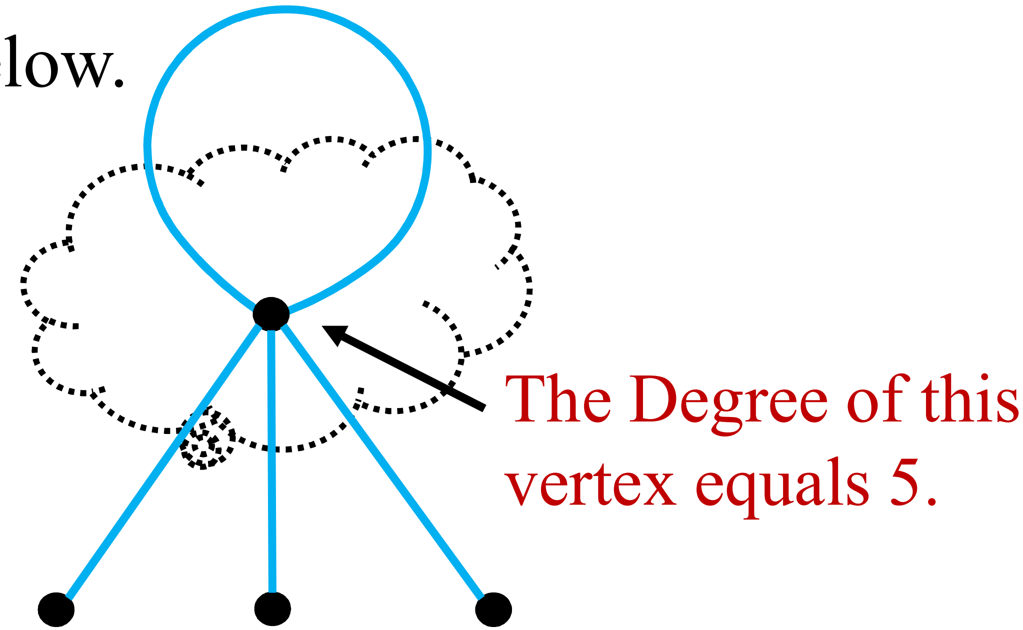
- **The Concept of Degree:**

  - The degree of a vertex is the number of end segments of edges that "stick out of" the vertex.

  - The sum of the degrees of all the vertices in a graph is twice the number of edges of the graph.

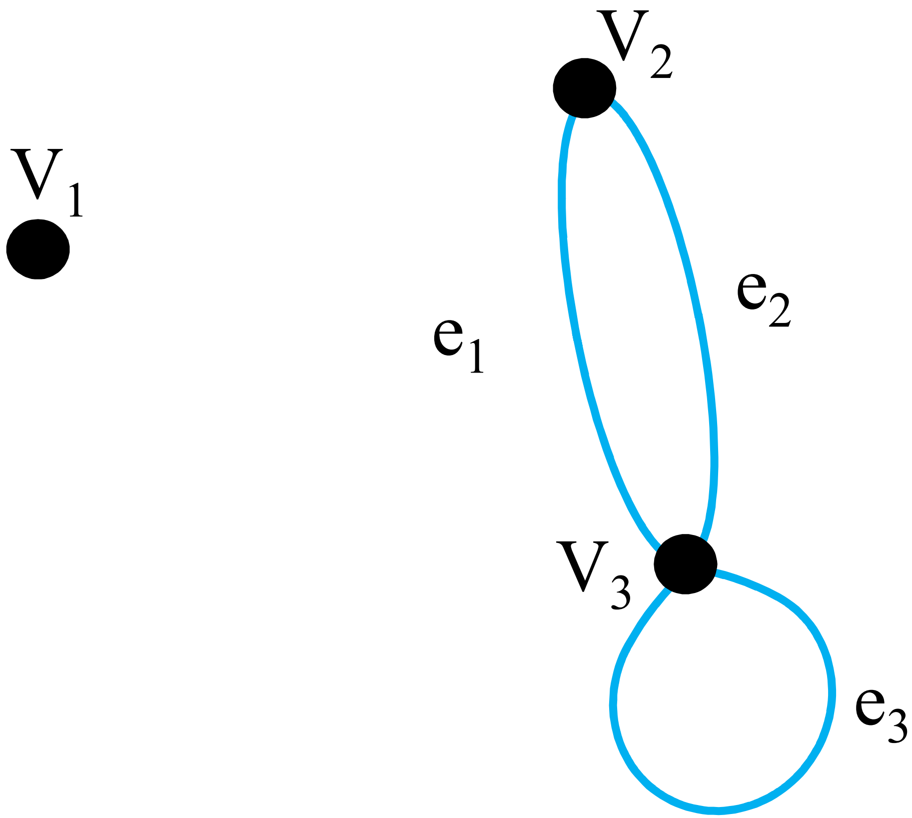
Undirected Graphs

- **The Concept of Degree---Cont.:**
  - Since an edge that is a loop is counted twice, the degree of a vertex can be obtained from the drawing of a graph by counting how many end segments of edges are incident on the vertex.
  - This is illustrated below.



Undirected Graphs

- **The Concept of Degree---Cont.:**
  - Find the degree of each vertex of the graph G shown below. Then find the total degree of G.



Undirected Graphs

- **The Concept of Degree---Cont.:**

  - Note that the total degree of the graph $\boldsymbol{G}$ in our example, which is 6, equals twice the number of edges of $\boldsymbol{G}$, which is 3.

  - Roughly speaking, this is true because each edge has two end segments, and each end segment is counted once toward the degree of some vertex.

  - This result generalizes to any undirected graph.

Undirected Graphs

- **The Concept of Degree---Cont.:**

  - In fact, for any graph without loops, the general result can be explained as follows:

  - ➢ Imagine a group of people at a party. Depending on how social they are, each person shakes hands with various other people.

  - So each person participates in a certain number of handshakes—perhaps many, perhaps none — but …

Undirected Graphs

- **The Concept of Degree---Cont.:**
  - But because each ***handshake is experienced by two different people***, if the numbers experienced by each person are added together, the sum will equal twice the total number of handshakes.
  - This is such an attractive way of understanding the situation that the following theorem is often called the **handshake lemma or the handshake theorem**.

Undirected Graphs

- **Handshake Theorem: (Theorem 10.1.1)**
  - If $\boldsymbol{G}$ is any undirected graph, then the sum of the degree of all the vertices of $\boldsymbol{G}$ equals twice the number of edges of $\boldsymbol{G}$. Specifically, if the vertices of $\boldsymbol{G}$ are $(v_1, v_2, v_3, \ldots, v_n)$, where n is a non negative integer, then

The Total degree of

$$\boldsymbol{G} = \boldsymbol{deg}(\boldsymbol{v_1}) + \boldsymbol{deg}(\boldsymbol{v_2}) \ldots + \boldsymbol{deg}(\boldsymbol{v_n})$$

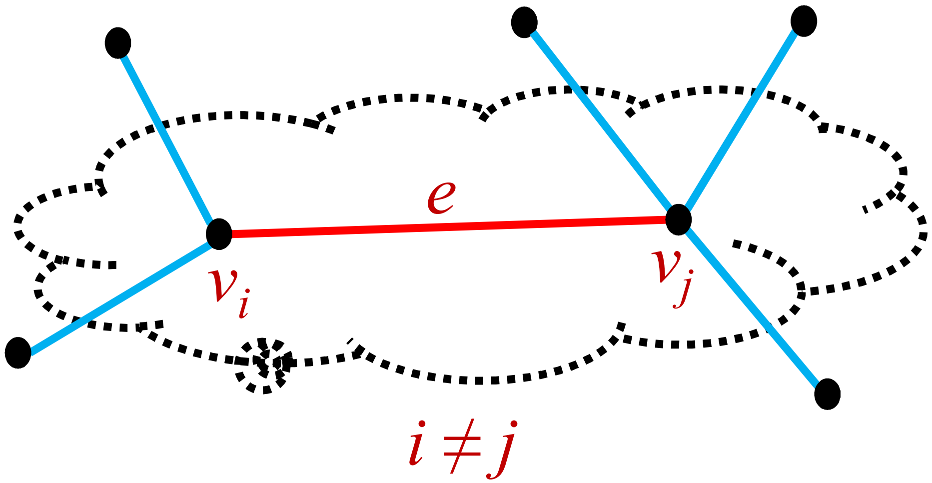
$$= \mathbf{2} * (\boldsymbol{The\ Number\ of\ Edges\ of\ G})$$

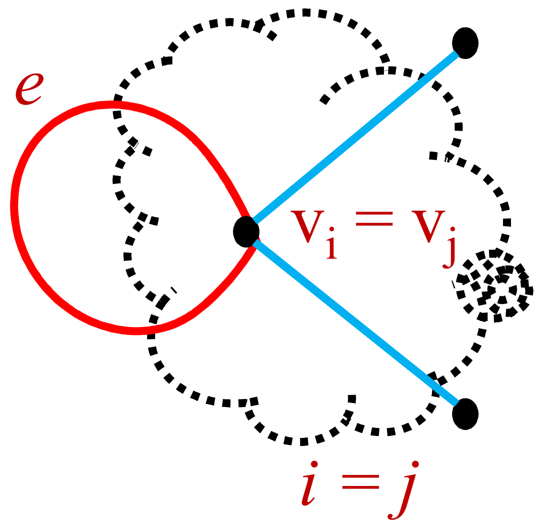
Handshake Theorem

Proof: Let $\boldsymbol{G}$ be a particular but arbitrarily chosen undirected graph, and suppose that $\boldsymbol{G}$ has $\boldsymbol{n}$ vertices $(\boldsymbol{v_1, v_2, v_3, \ldots, v_n})$ and m edges, where $\boldsymbol{n}$ is a positive integer and $\boldsymbol{m}$ is non negative integer. We claim that each edge of $\boldsymbol{G}$ contributes $\mathbf{2}$ to the total degree of $\boldsymbol{G}$.

Handshake Theorem

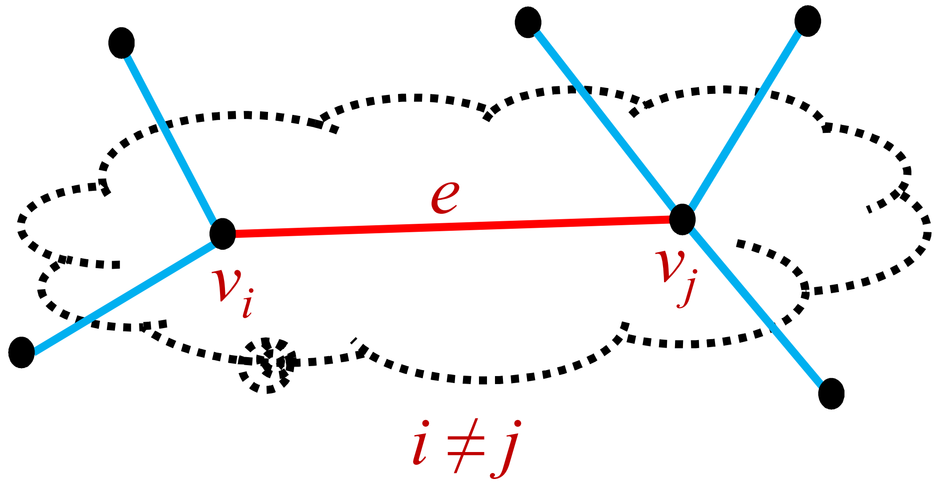
Proof: Let G be a particular but arbitrarily chosen undirected graph, and suppose that G has n vertices $(v_1, v_2, v_3, \ldots, v_n)$ and m edges, where n is a positive integer and m is non negative integer. We claim that each edge of G contributes 2 to the total degree of G. Suppose e is an arbitrarily chosen edge with endpoints v_i **and** v_j. This edge contributes **1** to the degree of v_i and **1** to the degree v_j. As shown below, this is true even if $i = j$, because an edge that is a loop is counted twice in computing the degree of the vertex on which it is incident.

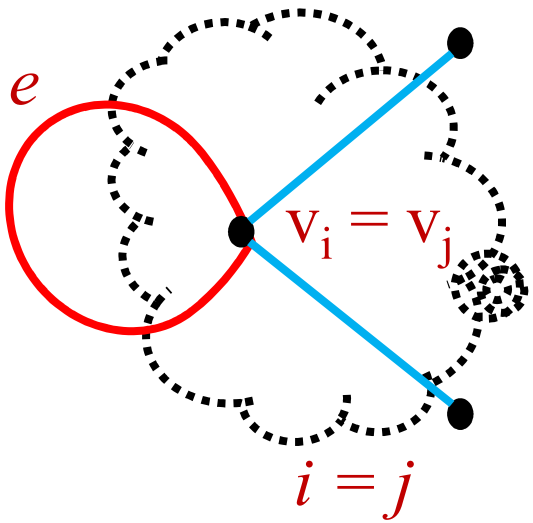




Handshake Theorem

Proof: Let G be a particular but arbitrarily chosen undirected graph, and suppose that G has n vertices $(v_1, v_2, v_3, \dots, v_n)$ and m edges, where n is a positive integer and m is non negative integer. We claim that each edge of G contributes 2 to the total degree of G. Suppose e is an arbitrarily chosen edge with endpoints v_i and v_j. This edge contributes 1 to the degree of v_i and 1 to the degree v_j. As shown below, this is true even if $i = j$, because an edge that is a loop is counted twice in computing the degree of the vertex on which it is incident.





Therefore, ***e*** contributes 2 to the total degree of G. since ***e*** was arbitrarily chosen, this shows each edge of G contributes 2 to the total degree of G. Thus

The total degree of $G = 2 * (\textit{The Number of Edges of G})$

Undirected Graphs

- **Related Corollary:**
  - The total degree of a graph is even.
- **Proof:**
  - By the previous theorem the total degree of $\boldsymbol{G}$ equals $\underline{\boldsymbol{2\ times}}$ the number of edges, which is an integer, and so the total degree of $\boldsymbol{G}$ is even.

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges.

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

- Because deg(v) is even for $v \in V_1$, the first term on the right hand side of the last equality is even.

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

- Because $\deg(v)$ is even for $v \in V_1$, the first term on the right hand side of the last equality is even.

- The ***second term*** on the right hand side ***is also even*** because it can be written as **the difference of two even numbers.**

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

- Because $\deg(v)$ is even for $v \in V_1$, the first term on the right hand side of the last equality is even.

- The second term on the right hand side is also even because it can be written as the difference of two even numbers.

  - Because all the terms in this sum are odd, there must be an even number of such terms.

Undirected Graphs (Even Number of Vertices with Odd Degree)

- **Proposition:** In any graph there are an even number of vertices of odd degree.

- **Proof:** Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V , E)$ with m edges. Then

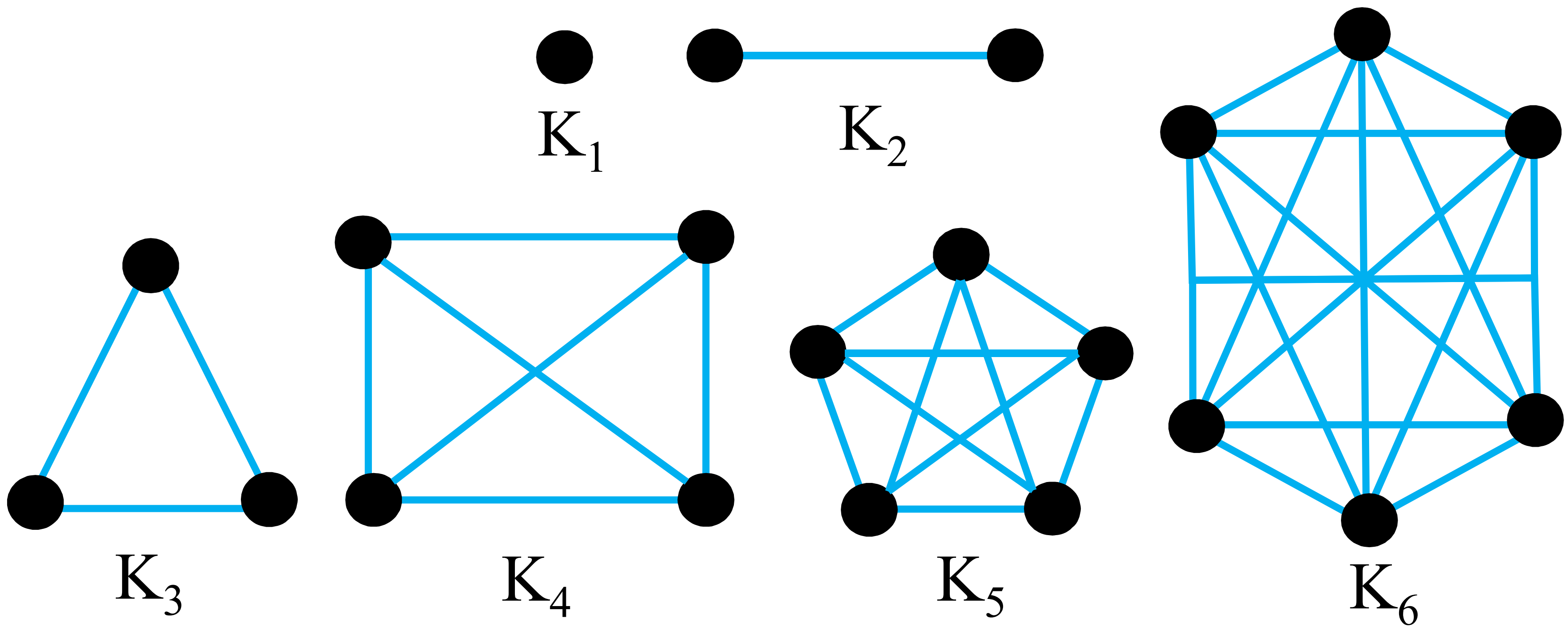
$$\boldsymbol{2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)}$$

- Because $\deg(v)$ is even for $v \epsilon V_1$, the first term on the right hand side of the last equality is even.
- The second term on the right hand side is also even because it can be written as the difference of two even numbers.
  - Because all the terms in this sum are odd, there must be an even number of such terms.

- **Thus, there are an even number of vertices of odd degree.**

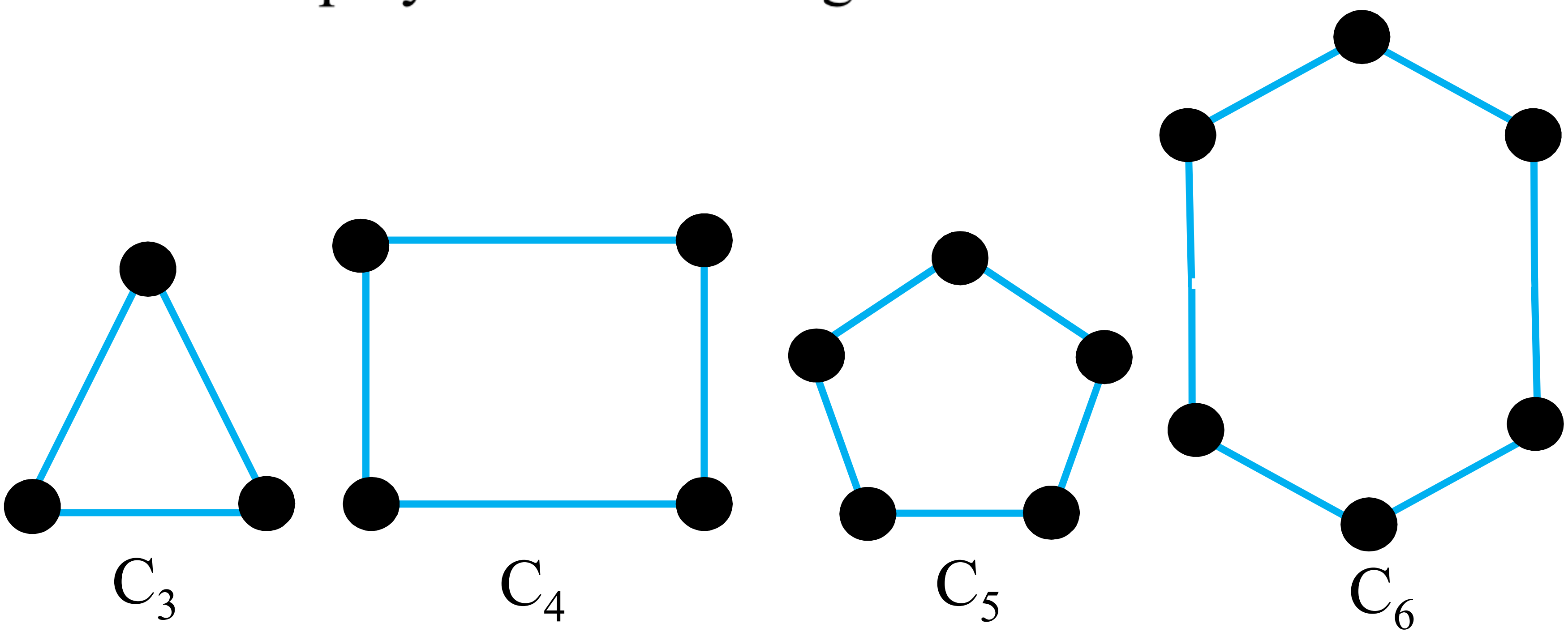
Undirected Graphs

- **Some Special Simple Graphs:**
  - **Complete Graphs:** A complete graph on n vertices denoted by K_n is a simple graph that contains exactly one edge between each pair of distinct vertices. The Graphs $K_n \; for \; 1 \leq n \leq 6$.



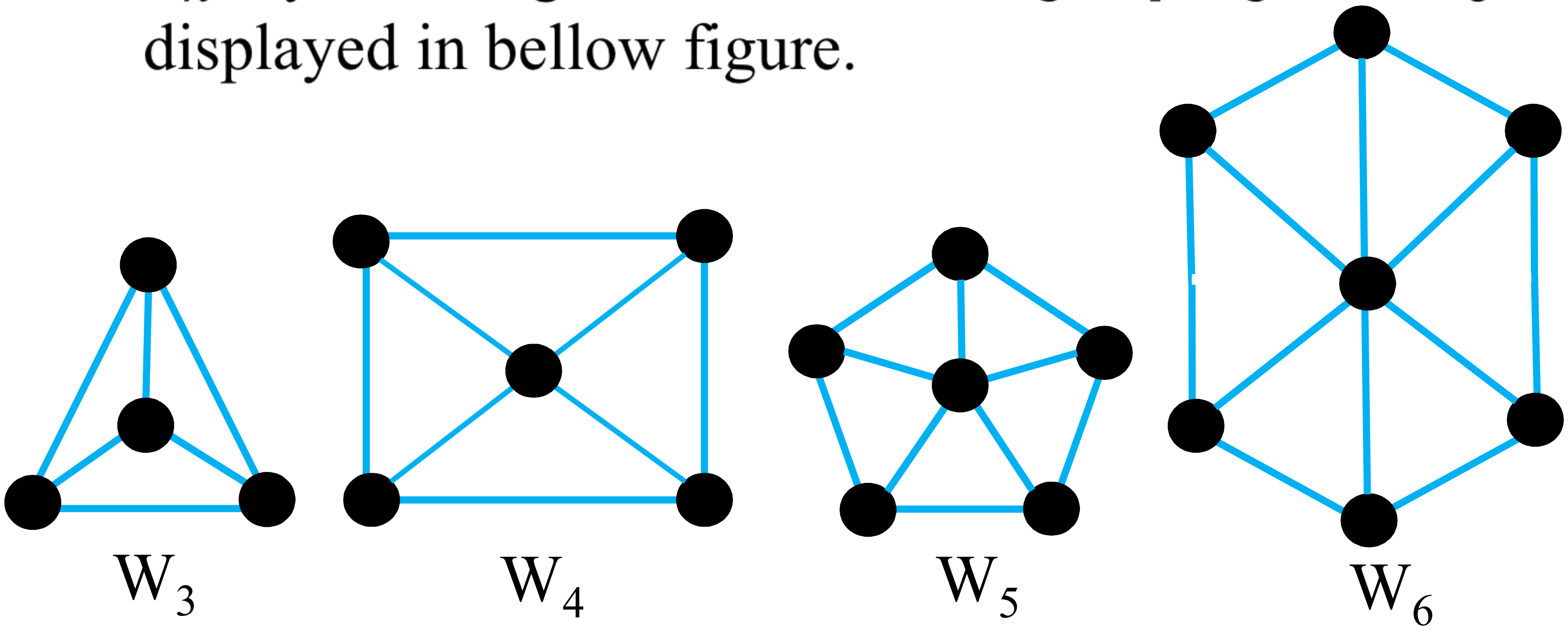
Undirected Graphs

- **Some Special Simple Graphs---Cont.:**
  - **Cycles:** Cycle C_n , n ≥ 3 , consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, …. , $\{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$. The cycles C_3, C_4, C_5, and C_6 are displayed in bellow figure.



Undirected Graphs

- **Some Special Simple Graphs---Cont.:**
  - **Wheels:** We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges. The wheels W_3 W_4 W_5 and W_6 are displayed in bellow figure.



Undirected Graphs

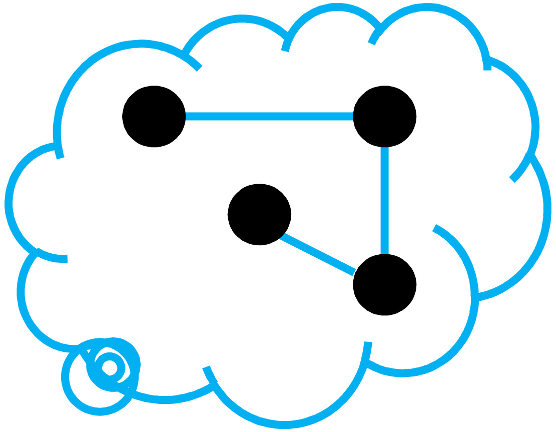
- **Graphs Connectivity:**

  - What two US states have no interstate highways?

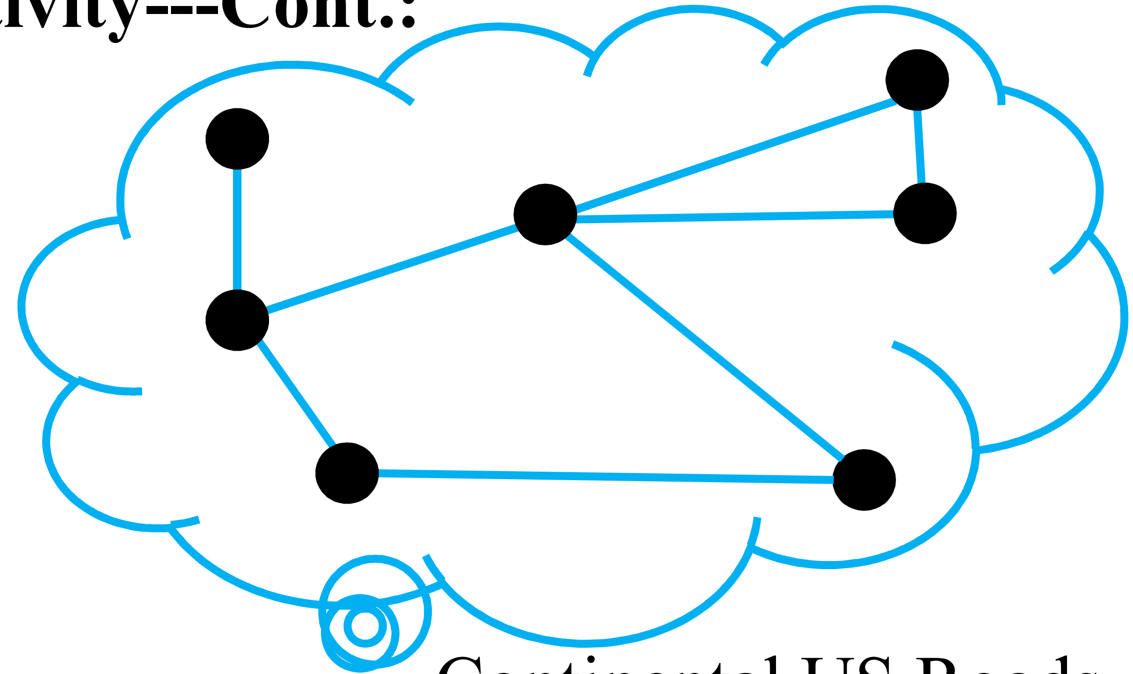
  - In order for such states to exist, what should be the case?

Undirected Graphs

- **Graphs Connectivity---Cont.:**



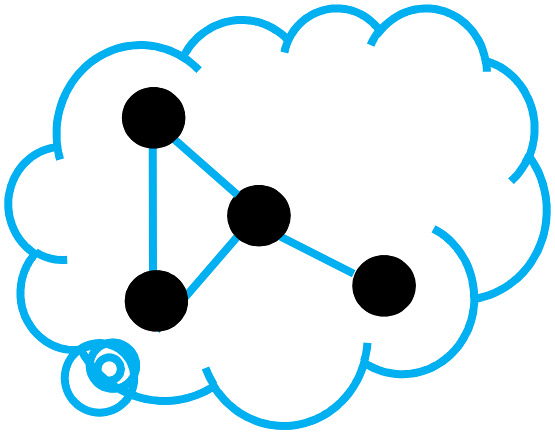
Alaskan Roads



Continental US Roads

- Different pieces but still one graph.

Let's come up with a definition for such Components of a graph!



Hawaiian Roads

Undirected Graphs

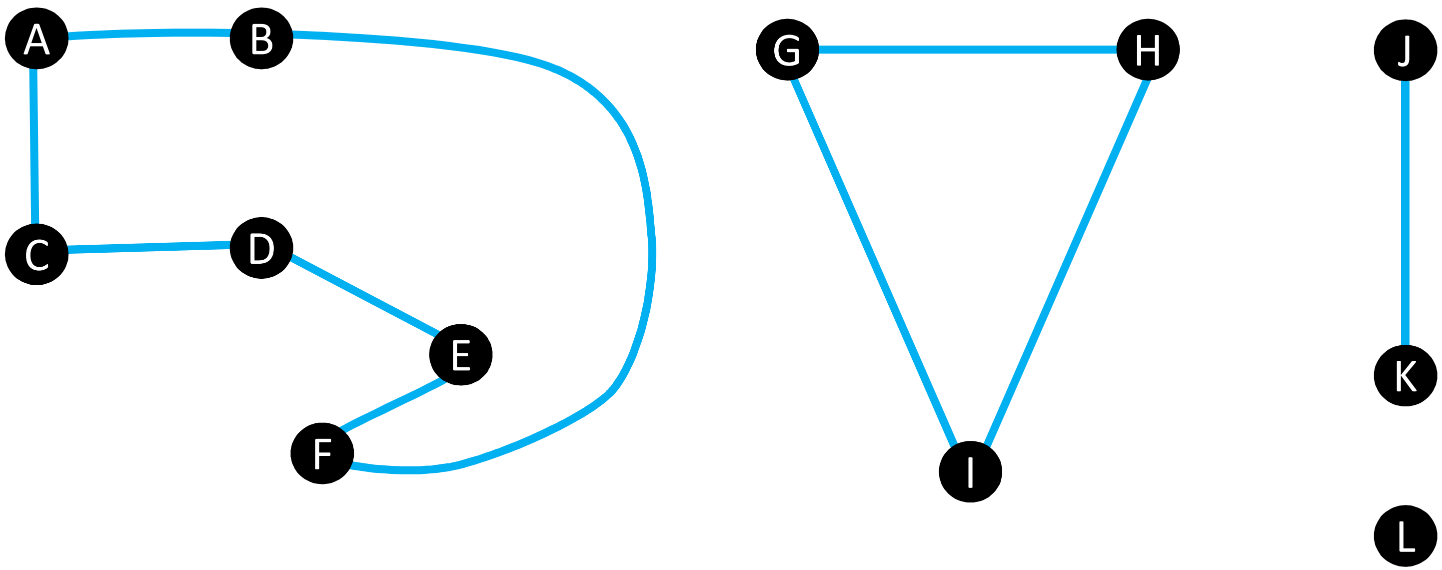
- **Connected Nodes:**

  - Let G be an undirected graph. Two nodes u and v are called **connected** if there is a path from u to v in G. if u and v are connected, we denote this by writing $u \Leftrightarrow v$. If u and v are not connected, we denoted this by writing $u \nLeftrightarrow v$.

  - **Note:** Sometimes, the term "walk" is used instead of "path".

Undirected Graphs

- **Connected Components:**
  - For Example; Consider the following graph.



  - $A \leftrightarrow B$ and $A \leftrightarrow E$.
  - What about L?

Undirected Graphs

- **Connectivity Between Nodes:**

  - Connectivity between nodes have some properties, which are summarized below

- **Theorem:** Let $G = (V, E)$ be an undirected graph. Then:

  - If $v \in V$, then $v \leftrightarrow v$
  - If $u\ , v \in V$, and $u \leftrightarrow v$, then $v \leftrightarrow u$
  - If $u, v, w \in V$, then if $u \leftrightarrow v$, and $v \leftrightarrow w, then$ $u \leftrightarrow w$.

Connectivity Between Nodes

- **Theorem:** Let $G = (V, E)$ be an undirected graph. Then:
  1. If $v \in V$, then $v \leftrightarrow v$
  2. If u , $v \in V$, and $u \leftrightarrow v$, then $v \leftrightarrow u$
  3. If $u, v, w \in V$, then if $u \leftrightarrow v$,and$v \leftrightarrow w$, $then$ $u \leftrightarrow w$.
- **Proof:** We prove each part independently.
  - **To prove (1),** note that for any $v \in V$, ***the trivial path*** (v) is a path from v to itself. Thus $v \leftrightarrow v$

Connectivity Between Nodes

- **Theorem:** Let $G = (V, E)$ be an undirected graph. Then:
  1. If $v \in V$, then $v \leftrightarrow v$
  2. If $u, v \in V$, and $u \leftrightarrow v$, then $v \leftrightarrow u$
  3. If $u, v, w \in V$, then if $u \leftrightarrow v$,and$v \leftrightarrow w, then\ u \leftrightarrow w$.
- **Proof:** We prove each part independently.
  - **To prove (1),** note that for any $v \in V$, the trivial path (v) is a path from v to itself. Thus $v \leftrightarrow v$
  - **To prove (2),** consider any $u, v \in V$ where $u \leftrightarrow v$. Then there must be some path $(u, x_1, x_2, \ldots, x_n, v)$. Since G is an undirected graph, this means $v, x_1, x_2, \ldots, x_n, u$ is a path from v to u. Thus $v \leftrightarrow u$.

Connectivity Between Nodes

- **Theorem:** Let $G = (V, E)$ be an undirected graph. Then:
  1. If $v \in V$, then $v \leftrightarrow v$
  2. If $u, v \in V$, and $u \leftrightarrow v$, then $v \leftrightarrow u$
  3. If $u, v, w \in V$, then if $u \leftrightarrow v$, and $v \leftrightarrow w$, then $u \leftrightarrow w$.

- **Proof:** We prove each part independently.

  - **To prove (1),** note that for any $v \in V$, the trivial path (v) is a path from v to itself. Thus $v \leftrightarrow v$

  - **To prove (2),** consider any $u, v \in V$ where $u \leftrightarrow v$. Then there must be some path $(u, x_1, x_2, \dots, x_n, v)$. Since G is an undirected graph, this means $v, x_1, x_2, \dots, x_n, u$ is a path from v to u. Thus $v \leftrightarrow u$.

  - **To prove (3),** consider any $u, v, w \in V$ where $u \leftrightarrow v$, and $v \leftrightarrow w$. Then there must be paths $(u, x_1, x_2, \dots, x_n, v)$ and $(v, y_1, y_2, \dots, y_n, w)$. Consequently, $(u, x_1, x_2, \dots, x_n, v, y_1, y_2, \dots, y_n, w)$ is a path from u to w. Thus $u \leftrightarrow w$.

Undirected Graphs

- **Connected Graph:**

  - 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.

  - **This allows us to see if a graph is in just one piece or not.**

Undirected Graphs

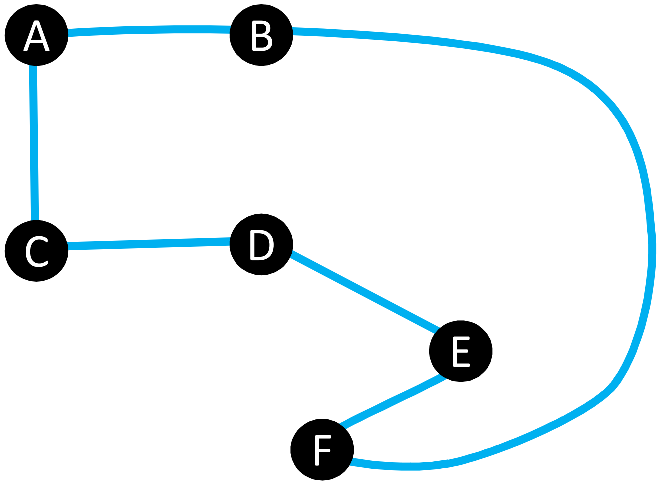
- **Connected Component:**

  - Can we say? -- “A connected component of a graph is a bunch of nodes that are all connected with each other”

  - What do you think?

Undirected Graphs

- **Connected Component---Cont.:**



- As per the definition, *A, B, C* and *D* are all connected, but do not form a connected component of the graph.
- *E* and *F* are also in the same component as them.
- Thus the definition is not enough or incomplete. How to complete it?

Undirected Graphs

- **Connected Component---Cont.:**
  - Let $G = (V, E)$ be an undirected graph. A connected component of G is a nonempty set of nodes C (That is, $C \subseteq V$), such that

    1. For any u, $v \in C$, we have $u \leftrightarrow v$
    2. For any $u \in C$ and $v \in V - C$, we have $u \not\leftrightarrow v$

Final Topic for Today

- **Two Important Questions:**

1. How do we know that connected components even exist?

2. How do we know if there is just one way of breaking any graph into its connected components?

How to Answer these Questions?

- We will answer by proving two important theorems.

a. First, we will prove that connected components cannot overlap.

b. Second, we will show that it is always possible to break a graph apart into its connected components.

(a) No Overlapping

- **Theorem**

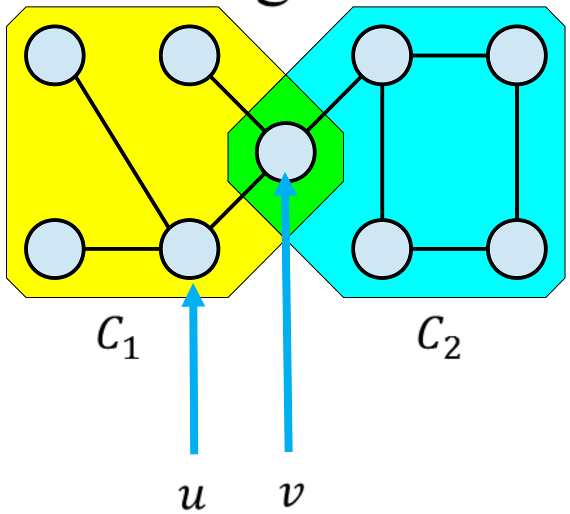
  - Let G be an undirected graph and let C_1 and C_2 be connected components of G. If $C_1 \neq C_1$ Then $C_1 \cap C_2 \neq \emptyset$.

(a) No Overlapping

- **Theorem:** Let G be an undirected graph and let C_1 and C_2 be connected components of G. If If $C_1 \neq C_1$ then $C_1 \cap C_2 \neq \emptyset$.

(a) No Overlapping

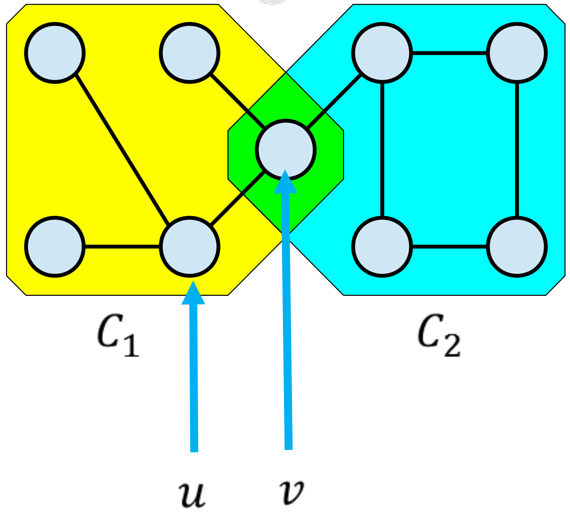
- **Theorem:** Let G be an undirected graph and let C_1 and C_2 be connected components of G. If If $C_1 \neq C_1$ then $C_1 \cap C_2 \neq \emptyset$.
- **Proof:** Let's try contradiction – Imagine two overlapping connected components



$$C_1 \neq C_2, u \in C_1, v \in C_1 \text{ and } v \in C_2$$

(a) No Overlapping

- **Theorem:** Let G be an undirected graph and let C_1 and C_2 be connected components of G. If If $C_1 \neq C_1$ then $C_1 \cap C_2 \neq \emptyset$.
- **Proof:** Let's try contradiction – Imagine two overlapping connected components



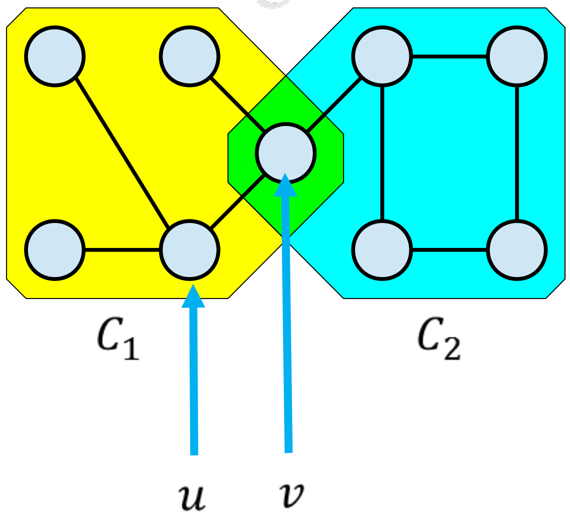
$C_1 \neq C_2$, $u \in C_1, v \in C_1$ and $v \in C_2$

- As per the definition of connected component

$$u \leftrightarrow v$$

(a) No Overlapping

- **Theorem:** Let G be an undirected graph and let C_1 and C_2 be connected components of G. If If $C_1 \neq C_1$ then $C_1 \cap C_2 \neq \emptyset$.
- **Proof:** Let's try contradiction – Imagine two overlapping connected components



$C_1 \neq C_2$, $u \in C_1$, $v \in C_1$ and $v \in C_2$

- As per the definition of connected component

$$u \leftrightarrow v$$

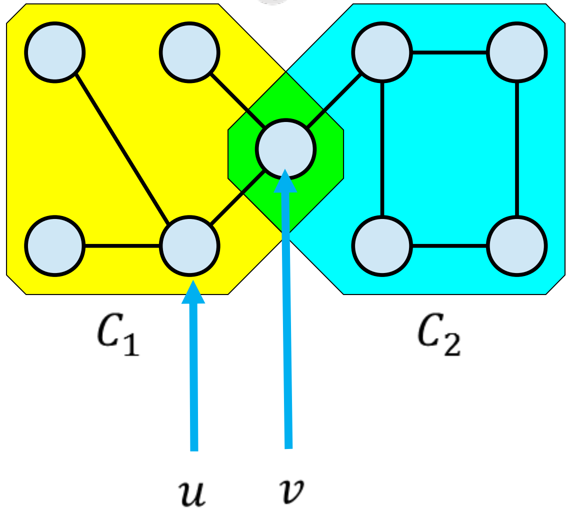
- But as per the same definition it is also true that

$$u \not\leftrightarrow v$$

(a) No Overlapping

- **Theorem:** Let G be an undirected graph and let C_1 and C_2 be connected components of G. If If $C_1 \neq C_1$ then $C_1 \cap C_2 \neq \emptyset$.
- **Proof:** Let's try contradiction – Imagine two overlapping connected components

$C_1 \neq C_2, u \in C_1, v \in C_1$ and $v \in C_2$



- As per the definition of connected component

$$u \leftrightarrow v$$

- But as per the same definition it is also true that

$$u \not\leftrightarrow v$$

- Which is a contradiction. Hence the components must be disjoint.

(a) No Overlapping

- **What does this Mean???**

  - If we split a graph into connected components, we can guarantee that those components do not overlap.

  - In other words, this theorem establishes the fact that "***each node (vertex) in a graph belongs to at most one connected component***"

Now Let's Prove (b)?

a. First, we will prove that connected components cannot overlap.

b. Second, we will show that it is always possible to break a graph apart into its connected components.

(b) Existence

- **Proving Existence of Connected Components**

  - This is equivalent of showing that "***each node in a graph belongs to at least one connected component***"

  - How to prove this?

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.

- **Proof:** let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.
- **Proof:** Let $G = (V, E)$ be any undirected graph and let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.

- First we prove that $v \in C$. To see this, note that by construction, $v \in C$ iff $v \leftrightarrow v$. As proven earlier, $v \leftrightarrow v$ is always true. Consequently, $v \in C$.

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.
- **Proof:** Let $G = (V, E)$ be any undirected graph and let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.
- First we prove that $v \in C$. To see this, note that by construction, $v \in C$ *iff* $v \leftrightarrow v$. As proven earlier, $v \leftrightarrow v$ is always true. Consequently, $v \in C$.
- Next, we prove that C is a connected component. This proof proceeds in two steps:

1. First we prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$;
2. Second, we prove that for any $u_1 \in C$ and $u_2 \in V - C$ that $u_1 \not\leftrightarrow u_2$.

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.
- **Proof:** Let $G = (V, E)$ be any undirected graph and let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.
- First we prove that $v \in C$. To see this, note that by construction, $v \in C$ *iff* $v \leftrightarrow v$. As proven earlier, $v \leftrightarrow v$ is always true. Consequently, $v \in C$.
- Next, we prove that C is a connected component. This proof proceeds in two steps:
- First we prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$;
- Second, we prove that for any $u_1 \in C$ and $u_2 \in V$ that $u_1 \nleftrightarrow u_2$.
- To prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$, consider any $u_1, u_2 \in C$. By construction, this means that $u_1 \leftrightarrow v$ and $u_2 \leftrightarrow v$. Or $u_1 \leftrightarrow v$ and $v \leftrightarrow u_2$ this means that $u_1 \leftrightarrow u_2$.

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.
- **Proof:** Let $G = (V, E)$ be any undirected graph and let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.
- First we prove that $v \in C$. To see this, note that by construction, $v \in C$ *iff* $v \leftrightarrow v$. As proven earlier, $v \leftrightarrow v$ is always true. Consequently, $v \in C$.
- Next, we prove that C is a connected component. This proof proceeds in two steps:
- First we prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$;
- Second, we prove that for any $u_1 \in C$ and $u_2 \in V$ that $u_1 \not\leftrightarrow u_2$.
- To prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$, consider any $u_1, u_2 \in C$. By construction, this means that $u_1 \leftrightarrow v$ and $u_2 \leftrightarrow v$. As proven earlier, since $u_2 \leftrightarrow v$, we know that $v \leftrightarrow u_2$. Also as proven earlier, since $u_1 \leftrightarrow v$ and $v \leftrightarrow u_2$ this means that $u_1 \leftrightarrow u_2$.
- Finally, to prove that for any $u_1 \in C$ and $u_2 \in V - C$, that $u_1 \not\leftrightarrow u_2$, consider any $u_1 \in C$ and $u_2 \in V - C$. Assume for the sake of contradiction that $u_1 \leftrightarrow u_2$. Since $u_1 \in C$, we know that $u_1 \leftrightarrow v$. Since $u_1 \leftrightarrow u_2$, Therefore $u_2 \leftrightarrow v$. Thus by definition of C, this means that $u_2 \in C$ contradicting the fact that $u_1 \in V - C$. We have reached a contradiction, so our assumption must have been wrong. Thus $u_1 \not\leftrightarrow u_2$

(b) Existence

- **Formal Proof: Theorem:** Let $G = (V, E)$ be an undirected graph. Then for any $v \in V$, there is a connected component C such that $v \in C$.
- **Proof:** Let $G = (V, E)$ be any undirected graph and let $v \in V$ be an arbitrary node in the graph. Consider the set $C = \{u \in V \mid u \leftrightarrow v\}$. We will prove C is a connected component containing v.
- First we prove that $v \in C$. To see this, note that by construction, $v \in C$ *iff* $v \leftrightarrow v$. As proven earlier, $v \leftrightarrow v$ is always true. Consequently, $v \in C$.
- Next, we prove that C is a connected component. This proof proceeds in two steps:
- First we prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$;
- Second, we prove that for any $u_1 \in C$ and $u_2 \in V$ that $u_1 \nleftrightarrow u_2$.
- To prove that for any $u_1, u_2 \in C$, that $u_1 \leftrightarrow u_2$, consider any $u_1, u_2 \in C$. By construction, this means that $u_1 \leftrightarrow v$ and $u_2 \leftrightarrow v$. As proven earlier, since $u_2 \leftrightarrow v$, we know that $v \leftrightarrow u_2$. Also as proven earlier, since $u_1 \leftrightarrow v$ and $v \leftrightarrow u_2$ this means that $u_1 \leftrightarrow u_2$.
- Finally, to prove that for any $u_1 \in C$ and $u_2 \in V$, that $u_1 \nleftrightarrow u_2$, consider any $u_1 \in C$ and $u_2 \in V$. Assume for the sake of contradiction that $u_1 \leftrightarrow u_2$. Since $u_1 \in C$, we know that $u_1 \leftrightarrow v$. Since $u_1 \leftrightarrow u_2$, we know that $u_2 \leftrightarrow u_1$. Therefore, since $u_2 \leftrightarrow u_1$ and $u_1 \leftrightarrow v$, we have that $u_2 \leftrightarrow v$. Thus by definition of C, this means that $u_2 \in C$ contradicting the fact that $u_1 \in V - C$. We have reached a contradiction, so our assumption must have been wrong. Thus $u_1 \nleftrightarrow u_2$.
- Thus C is a connected component containing v. since our choice of v and G ere arbitrary, any node in any graph belongs to at least one connected component.

Thus Proven

a. Connected components cannot overlap.

b. Second, it is always possible to break a graph apart into its connected components.

Why did we prove (a) and (b)

- **Two Important Questions:**

1. How do we know that connected components even exist?

2. How do we know if there is just one way of breaking any graph into its connected components?

So did we answer?

- **(a) and (b)** can be put together to show that :
  - **Theorem:** Every node in an undirected graph belongs to **exactly one** connected component.
- So have we found the answers to the following?

1. How do we know that connected components even exist?
2. How do we know if there is just one way of breaking any graph into its connected components?