CSE 191

Introduction to Discrete Structures

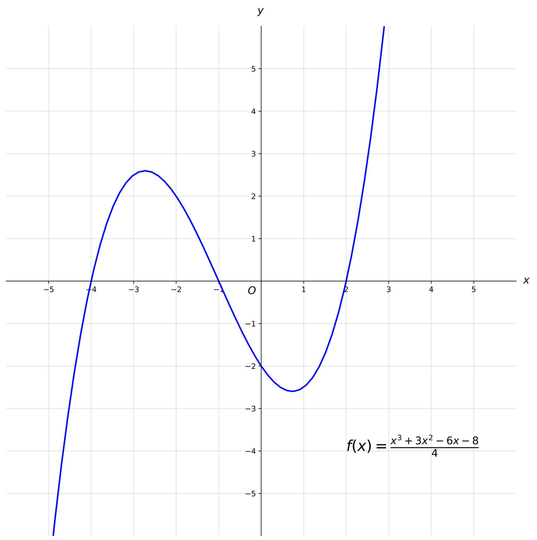
Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Graphs

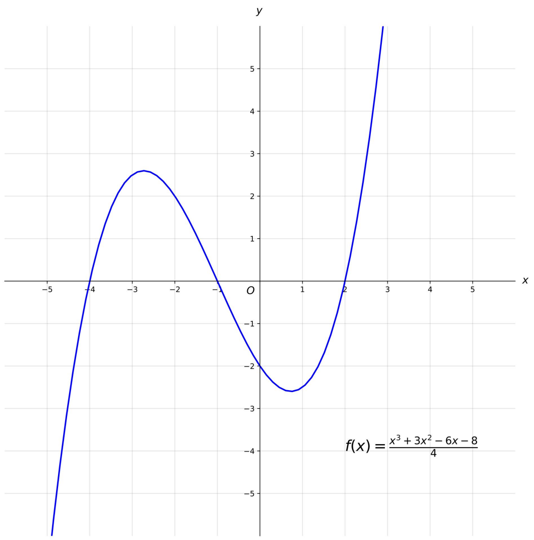
Outline

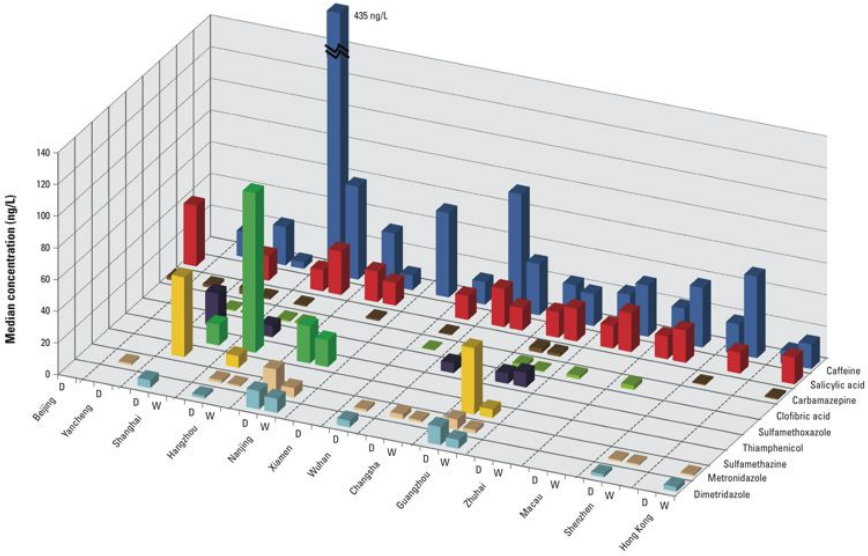
- **What are graphs?**
- Graph Examples
- Graph Representation
- Handshake Theorem
- Graph Isomorphism
- Connectivity
- Graph Coloring
- Trees

What are Graphs?

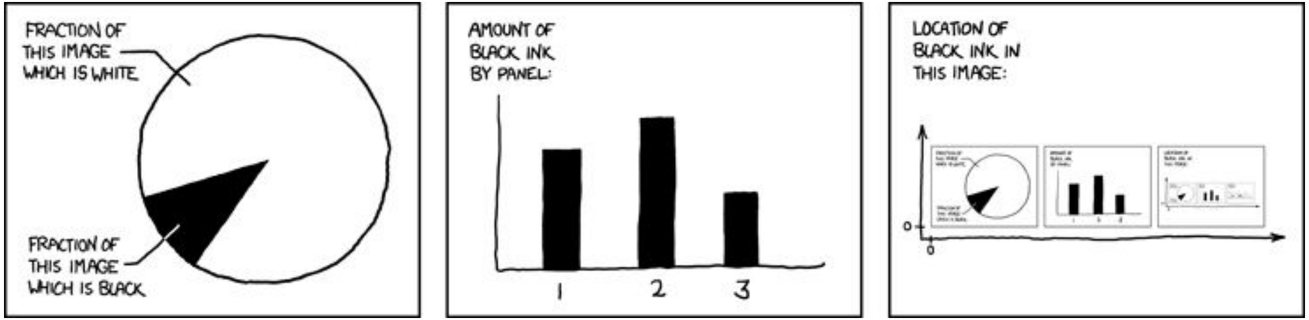


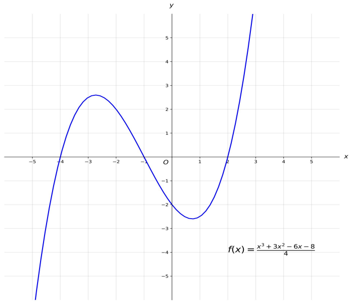
What are Graphs?

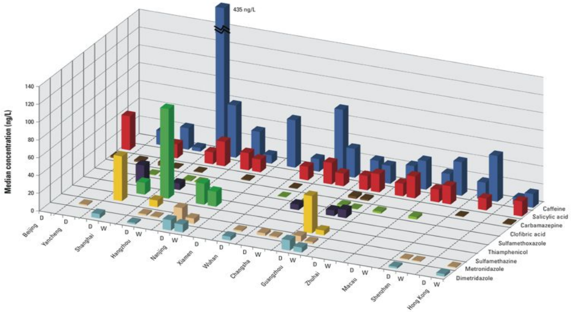




What are Graphs?







Source: https://xkcd.com/688/, https://en.wikipedia.org/wiki/Graph_of_a_function, http://stackoverflow.com/questions/23261760/how-to-generate-3-d-bar-graph-in-r

What are Graphs?

Graphs model relationships between pairs of objects

- Each object is represented as a **node** or **vertex** in the graph
- Relationships between objects are represented as **edges**

What are Graphs?

Graphs model relationships between pairs of objects

- Each object is represented as a **node** or **vertex** in the graph
- Relationships between objects are represented as **edges**

What kinds of things can we represent as graphs?

What are Graphs?

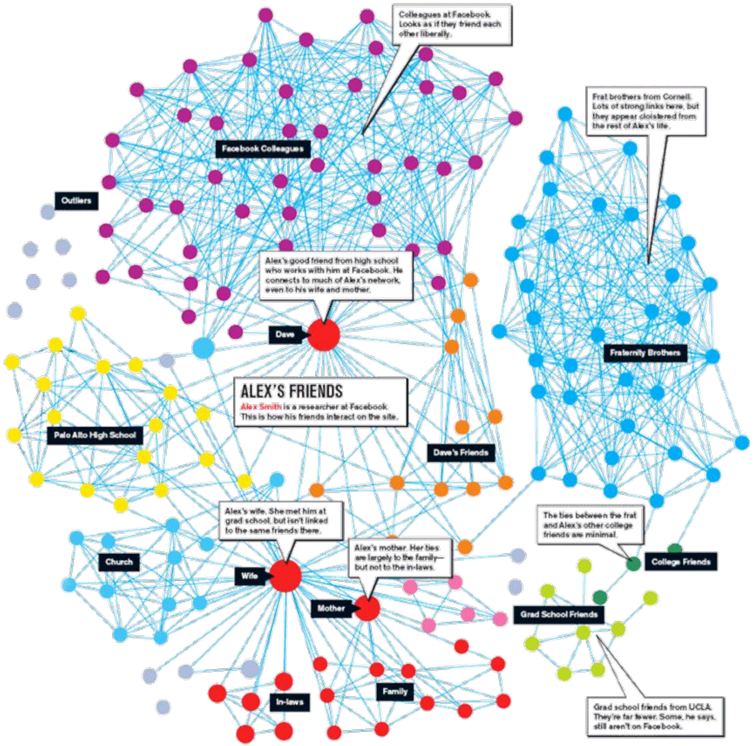
The flight map for JetBlue is a graph

- Cities are vertices
- Edges represent direct flights that JetBlue makes

What are Graphs?

All kinds of aspects of social media can be represented as graphs

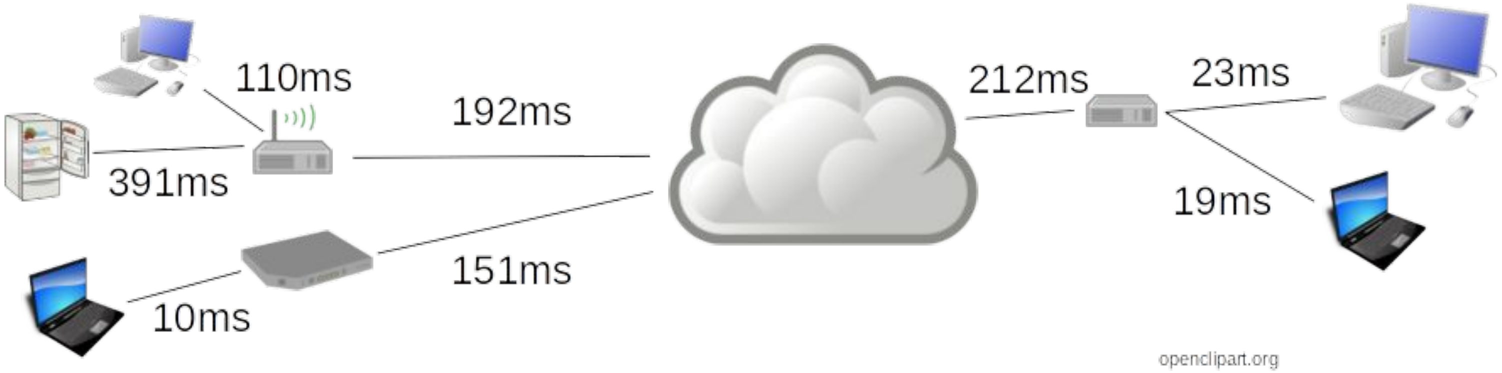
- Vertices can represent users, posts, images, comments, etc
- Edge can represent likes, interactions, friendships, etc



What are Graphs

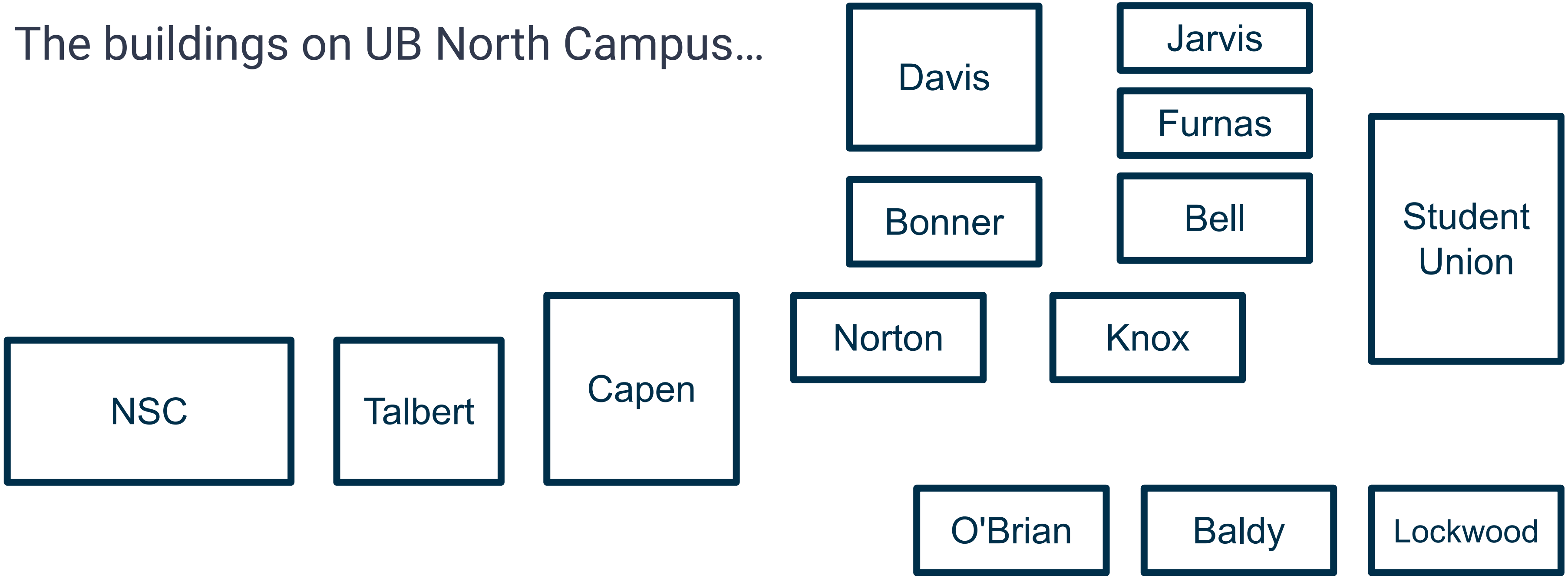
Computer networks can be represented as a graph (...the internet as well)

- Vertices might represent devices
- Edges represent channels through which devices could communicate



What are Graphs?

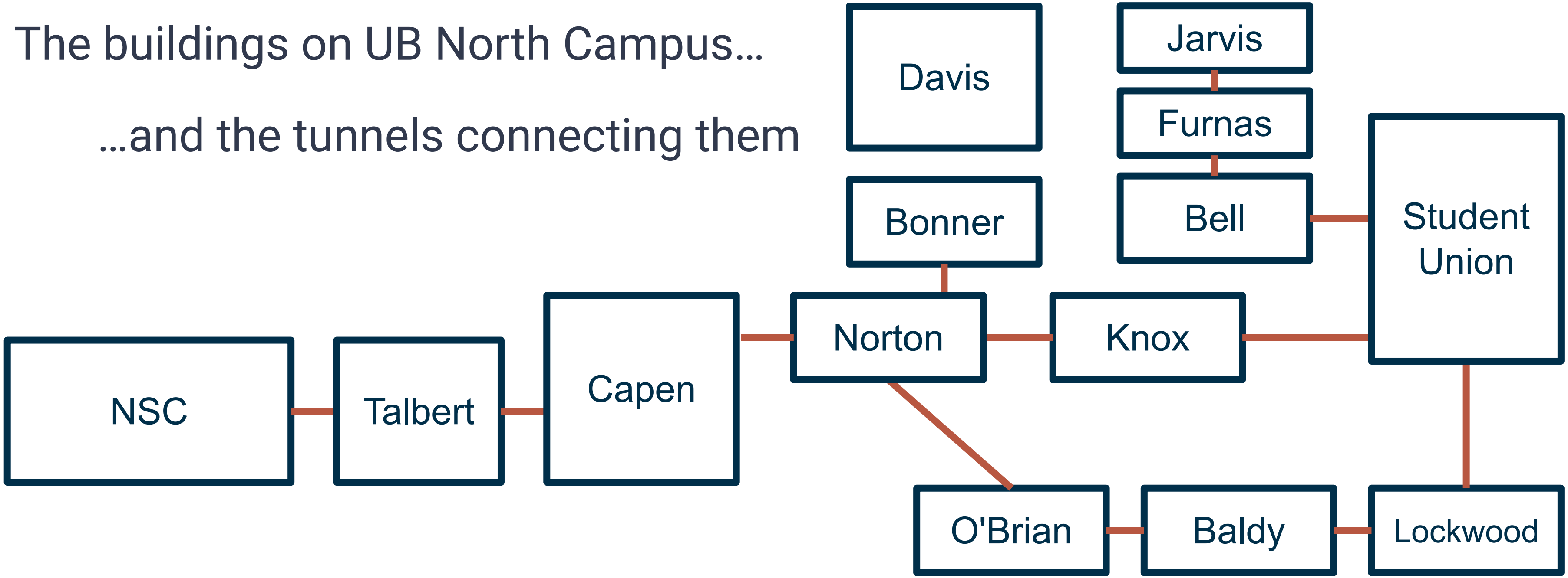
The buildings on UB North Campus...



What are Graphs?

The buildings on UB North Campus...

...and the tunnels connecting them



What are Graphs?

More Examples

- Maps (ie google maps)
- The internet (webpages are nodes, links are edges)
- Interactions between molecules
- Dependencies among tasks (ie GNU make)
- Moves in a game (ie Chess)
- Many, many, ….many more

Why Care About Graphs?

Once we have represented our objects in graph form there are many algorithms we can use to learn about relations between these objects

- Find out if two nodes are related (is there an edge between them)
- Find out if two nodes are connected via some path
  - If they are, can we find the cheapest/shortest path between them
    - What the fastest I can get to Wegmans?
    - What is the cheapest way to fly from Buffalo to California
- Find vulnerabilities/weak points
- Paths that go through a set of nodes
  - ie schedule the stops for a UPS driver

Why Care About Graphs?

Once we have represented our objects in graph form there are many algorithms we can use to learn about relations between these objects

- Find out if two nodes are related (is there an edge between them)
- Find out if two nodes are connected via some path
  - If they are, can we find the cheapest/shortest path between them
    - What the fastest I can get to Wegmans?
    - What is the cheapest way to fly from Buffalo to California
- Find vulnerabilities/weak points
- Paths that go through a set of nodes
  - ie schedule the stops for a UPS driver

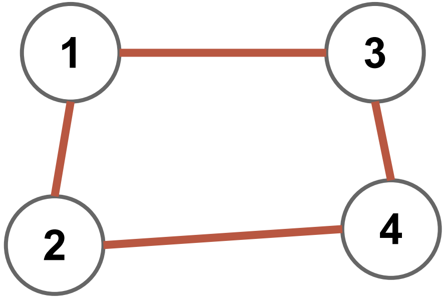
…and yes, many more

Graph Definition

An **(undirected) graph** ***G*** **= (*V*, *E*)** consists of ***V***, a nonempty set of **vertices** (or **nodes**) and a set of **edges**.

- Each edge has one or two vertices associated with it, called ***endpoints***
- An edge, **{*u*, *v*}** is said to **connect** its endpoints ***u*** and ***v***

Vertices (V)
Edges (E)

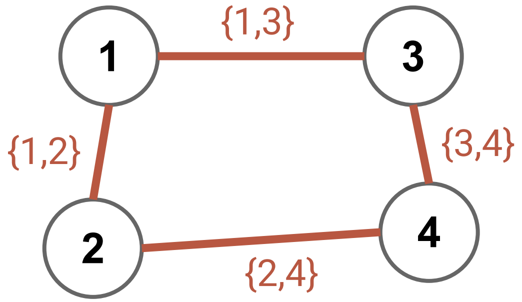


Graph Definition

An <u>(undirected) graph</u> ***G* = (*V*, *E*)** consists of ***V***, a nonempty set of **vertices** (or **nodes**) and a set of **edges**.

- Each edge has one or two vertices associated with it, called ***endpoints***
- An edge, **{*u*, *v*}** is said to **connect** its endpoints ***u*** and ***v***

Vertices (V)
Edges (E)



Undirected vs Directed

Undirected Edge: $\{u, v\}$ represented as a set

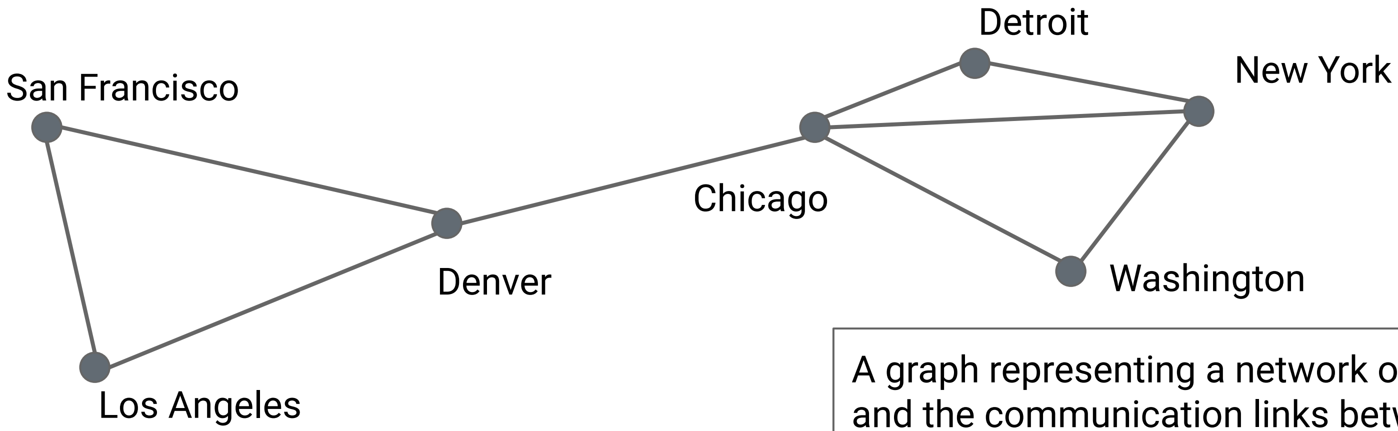
- Order doesn't matter ($\{u, v\} = \{v, u\}$)
- Represents a symmetric relationship

Directed Edge: (u, v) represented as a tuple

- Order does matter, (u, v) and (v, u) are not the same
- Both edges may not exist
- Represents an asymmetric relationship (ie a one-way street)

Simple Graph

A **Simple Graph** is a graph in which every edge connects two **different** vertices and where no two edges connect the same pair of vertices

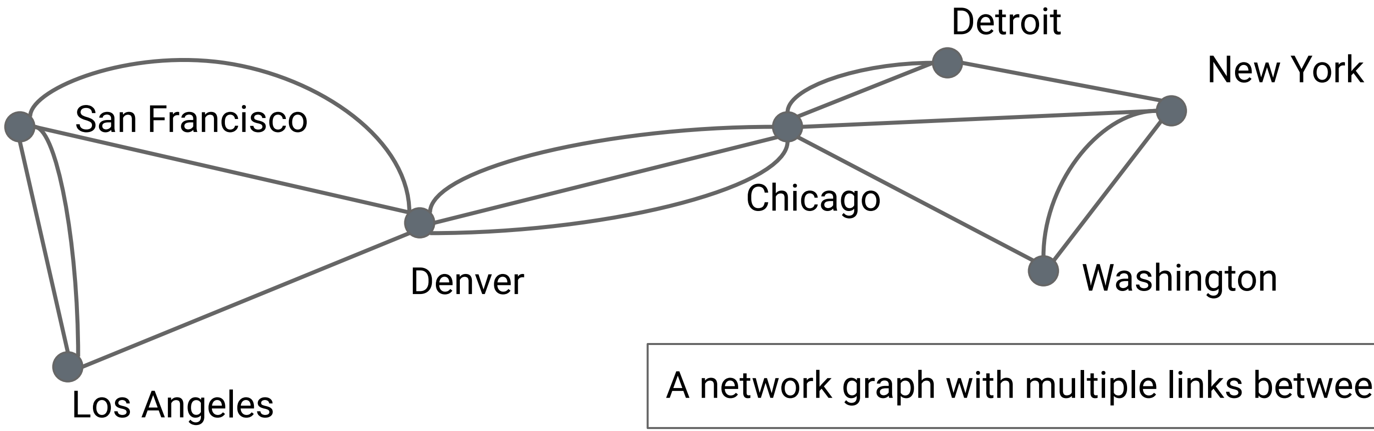


A graph representing a network of data centers and the communication links between them

Multigraph

A graph that may have **multiple edges** connected the same vertices are called **<u>multigraphs</u>**.

- When there are ***m*** different edges associated with the same pair of vertices, **{*u*, *v*}**, we say that edge **{*u*, *v*}** has **<u>multiplicity</u> *m*.**

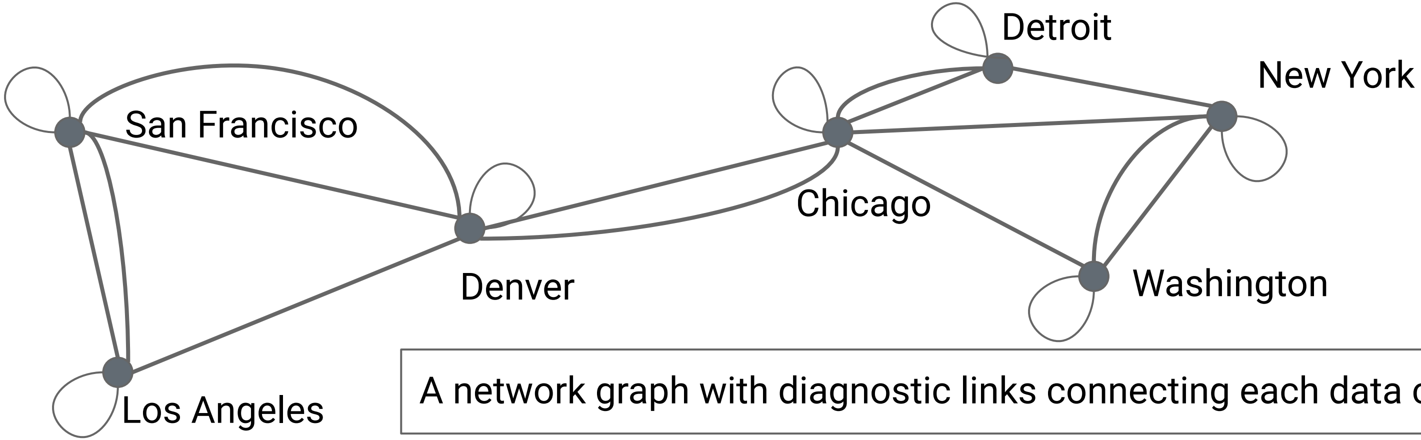


A network graph with multiple links between computers

Graphs with Loops

The edges that connect a given vertex to itself are called **<u>loops</u>** or sometimes **<u>self-loops</u>**.

- Graphs that may include loops and/or multiple edges between the same pair of vertices are sometimes called **<u>pseudographs</u>**.

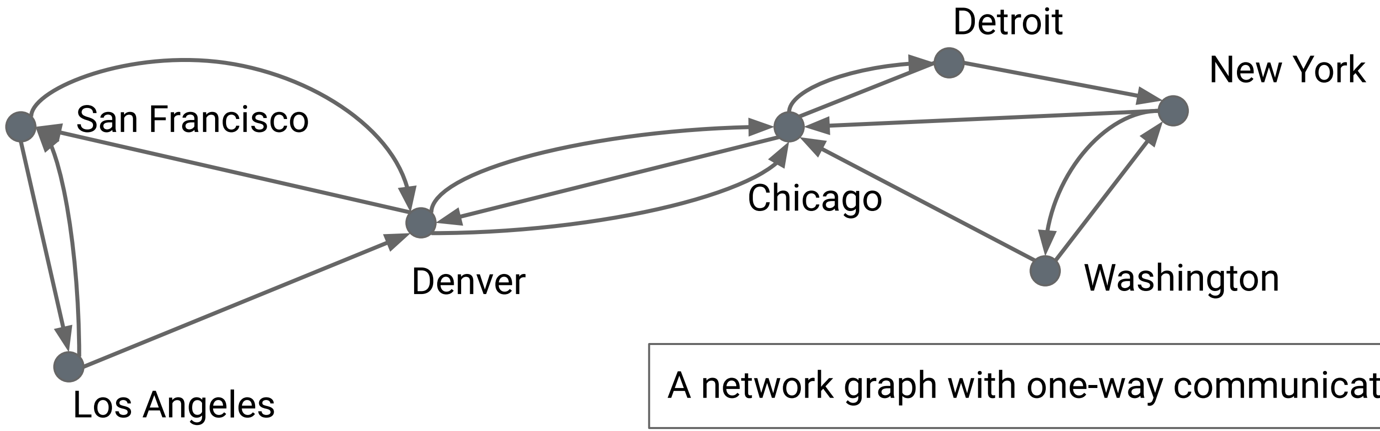


A network graph with diagnostic links connecting each data center to itself

Directed Graph

A **<u>directed graph</u>** (or **<u>digraph</u>**) **(*V*, *E*)** consists of a nonempty set of vertices ***V*** and a set of ***directed edges*** (or ***arcs***) ***E***.

- Each directed edge is associated with an ordered pair of vertices
- The directed edge **(*u*, *v*)** is said to start at ***u*** and end at ***v***



A network graph with one-way communication links

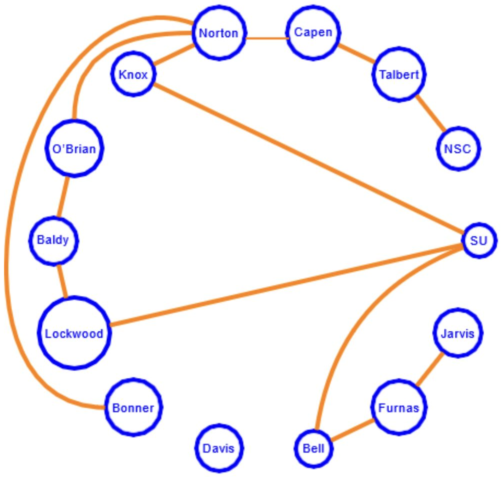
Graph Drawing

Note: One graph can be drawn in many ways. How we decide to draw the graph can help to convey important information clearly.

We can always draw a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ on a plane

- Place every vertex as a point
- Place every edge as an arc connecting its endpoints

Recall our graph of campus buildings →

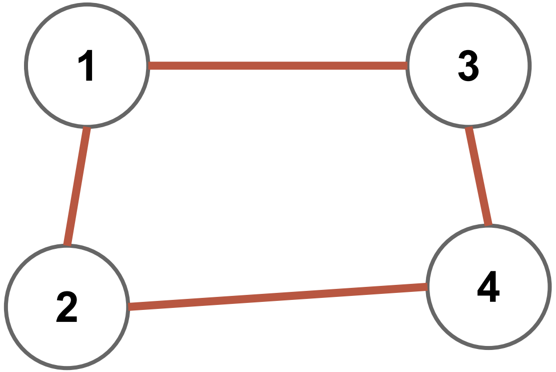


Outline

- What are graphs?
- **Graph Examples**
- Graph Representation
- Handshake Theorem
- Graph Isomorphism
- Connectivity
- Graph Coloring
- Trees

Graph Examples

Let $G_1 = (V_1, E_q)$ be the following graph:

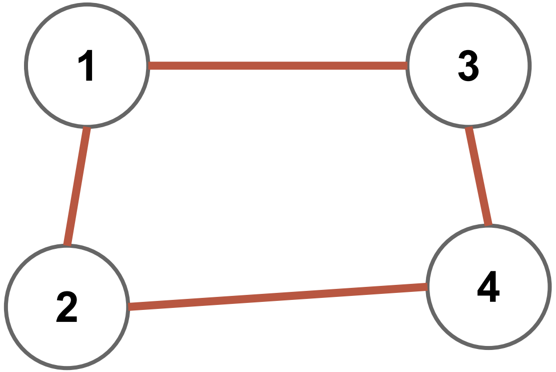


What are the vertices of $\boldsymbol{G_1}$ (what is $\boldsymbol{V_1}$)?

What are the edges of $\boldsymbol{G_1}$ (what is $\boldsymbol{E_1}$)?

Graph Examples

Let $G_1 = (V_1, E_q)$ be the following graph:



What are the vertices of $\boldsymbol{G_1}$ (what is $\boldsymbol{V_1}$)? {1, 2, 3, 4 }

What are the edges of $\boldsymbol{G_1}$ (what is $\boldsymbol{E_1}$)? {{1,2}, {1,3}, {2,4}, {3,4}}

Graph Examples

Let $G_2 = (V_2, E_2)$ be the following graph:

$V_2 = \{A, B, C, D, E\}$

$E_2 = \{\{A,B\}, \{A,C\}, \{B,C\}, \{B,D\}, \{C,E\}, \{D,E\}\}$

Draw G_2

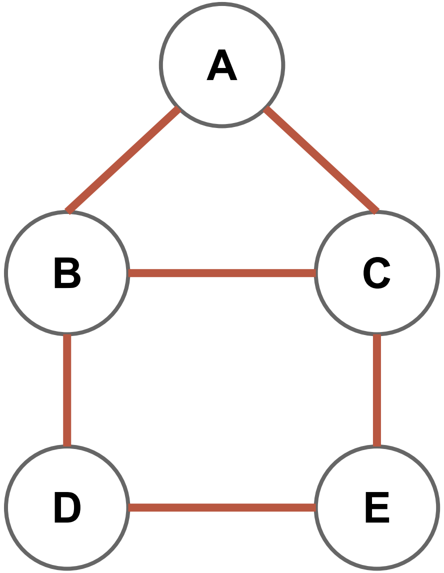
Graph Examples

Let $G_2 = (V_2, E_2)$ be the following graph:

$V_2 = \{A, B, C, D, E\}$

$E_2 = \{\{A,B\}, \{A,C\}, \{B,C\}, \{B,D\}, \{C,E\}, \{D,E\}\}$

Draw G_2



More Terminology

In an undirected graph:

Two vertices are **adjacent** (or **neighbors**) if they are endpoints of an edge

An edge is **incident with** (or **connecting**) its endpoints

The **degree of a vertex** is *v*, denoted by **deg(*v*)** is the number of edges incident with ***v***

Note: *A **loop** is an edge of the form **{v,v}**, and it adds to the degree of **v** twice*

More Terminology

The set of all neighbors of a vertex ***v*** of ***G* = (*V*, *E*)**, denoted by ***N*(*v*)**, is called the **<u>neighborhood</u>** of ***v***

If ***A*** is a subset of ***V***, then the neighborhood of ***A***, or ***N*(*A*)** is the set of all vertices that are adjacent to at least one vertex in ***A***. So $N(A) = \bigcup_{v \in A} N(v)$

Graph Examples

Recall G_1:

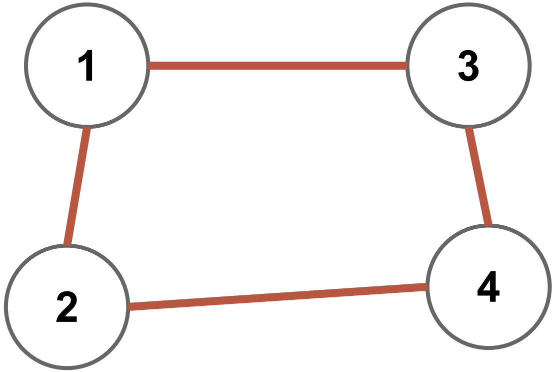
What are the neighbors of 1?

What is $\boldsymbol{N}(\{1,4\})$?

What is $\boldsymbol{N}(\{1,3\})$?

What vertices are incident to edge {2,4}?

What is the degree of 4?



Graph Examples

Recall G_1:

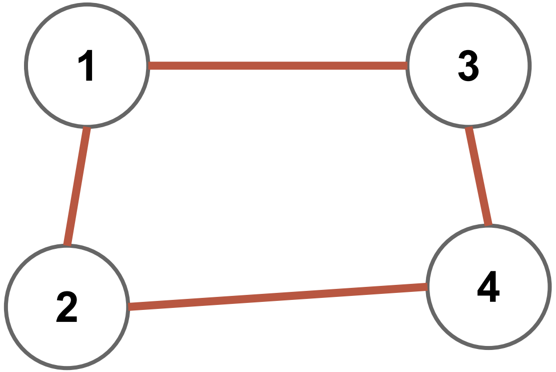
What are the neighbors of 1? **2 and 3**

What is $\boldsymbol{N}$({1,4})? **{2,3}**

What is $\boldsymbol{N}$({1,3})? **{1,2,3,4}**

What vertices are incident to edge {2,4}? **2 and 4**

What is the degree of 4? **deg(4) = 2**



Graph Examples

Recall G_1:

What are the neighbors of 1? **2 and 3**

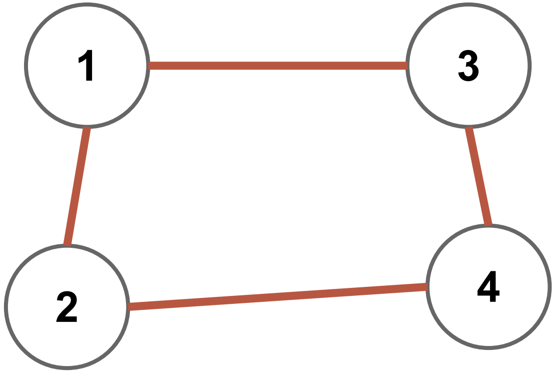
What is ***N***({1,4})? **{2,3}**

What is ***N***({1,3})? **{1,2,3,4}**

These are sets of vertices

What vertices are incident to edge {2,4}? **2 and 4**

What is the degree of 4? **deg(4) = 2**



Graph Examples

Recall G_1:

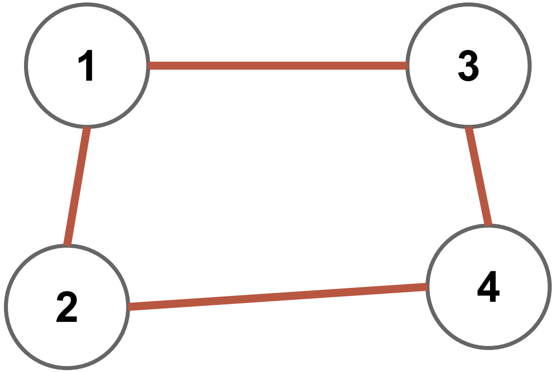
What are the neighbors of 1? **2 and 3**

What is $N(\{1,4\})$? $\mathbf{\{2,3\}}$

What is $N(\{1,3\})$? $\mathbf{\{1,2,3,4\}}$

What vertices are incident to edge $\{2,4\}$? **2 and 4**

What is the degree of 4? $\mathbf{deg(4) = 2}$



This is an edge

Graph Examples

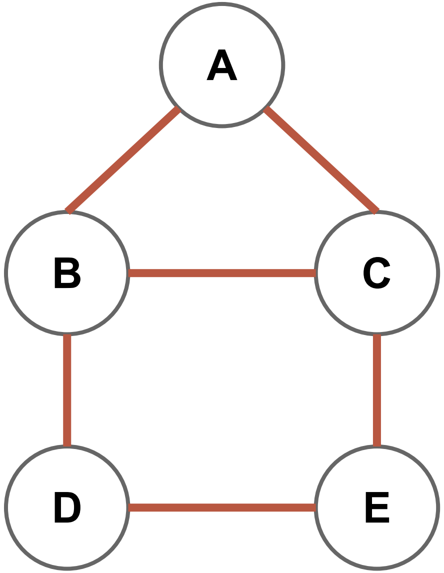
Recall G_2

What is the neighborhood of ***B***?

What is ***N***(***E***)?

Is vertex ***B*** adjacent to vertex ***E***?

What is the degree of vertex ***C***?



Graph Examples

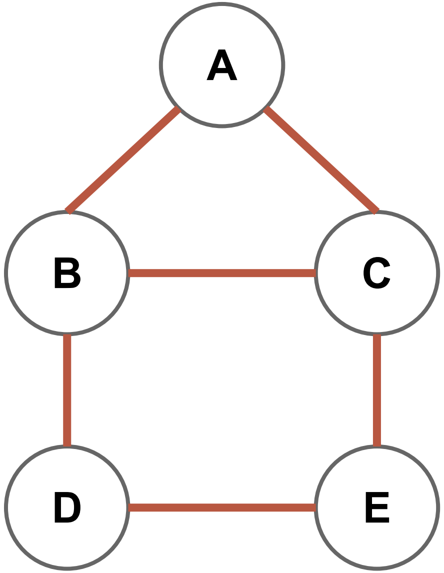
Recall G_2

What is the neighborhood of B? $\{A, C, D\}$

What is $N(E)$? $\{C, D\}$

Is vertex B adjacent to vertex E? **No**

What is the degree of vertex C? $\mathbf{deg}(C) = 3$



Graph Examples

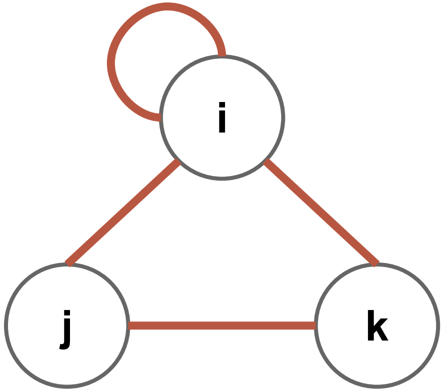
Let G_3 = (V_3, E_3):

What are the neighbors of ***i***?

What are the neighbors of ***k***?

What is the degree of vertex ***i***?

Is G_3 a simple graph?



Graph Examples

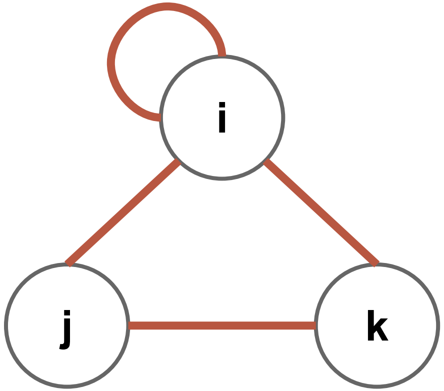
Let $G_3 = (V_3, E_3)$:

What are the neighbors of *i*? ***i*, *j*, and *k***

What are the neighbors of *k*? ***i* and *j***

What is the degree of vertex *i*? **deg(*i*) = 4**

Is G_3 a simple graph? **No, there is a loop**



More Graph Terminology

In a graph with directed edges

The **<u>in-degree</u>** of a vertex ***v***, denoted by $\deg^-(\boldsymbol{v})$, is the number of edges with ***v*** as their terminal (or ending) vertex

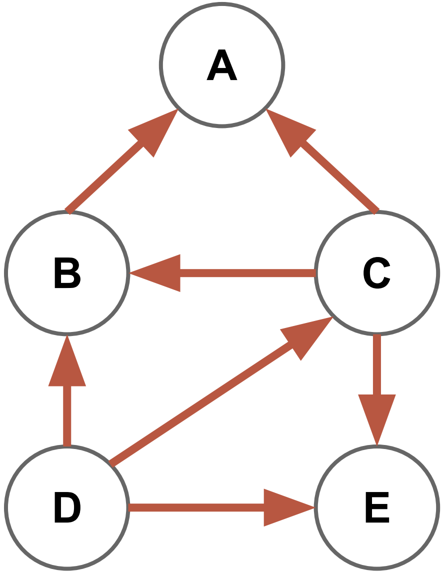
The **<u>out-degree</u>** of a vertex ***v***, denoted by $\deg^+(\boldsymbol{v})$, is the number of edges with ***v*** as their initial (or starting) vertex

Note: *A loop at vertex* ***v*** *contributes 1 to both the in and out degree of* ***v***

Graph Examples

Let $G_4 = (V_4, E_4)$:

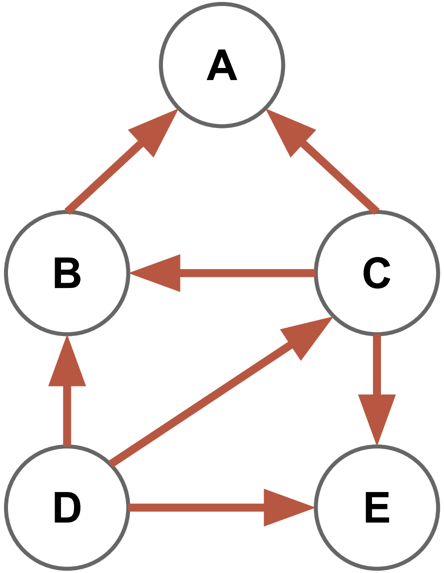
$\deg^-(A)$	$\deg^+(A)$
$\deg^-(B)$	$\deg^+(B)$
$\deg^-(C)$	$\deg^+(C)$
$\deg^-(D)$	$\deg^+(D)$
$\deg^-(E)$	$\deg^+(E)$



Graph Examples

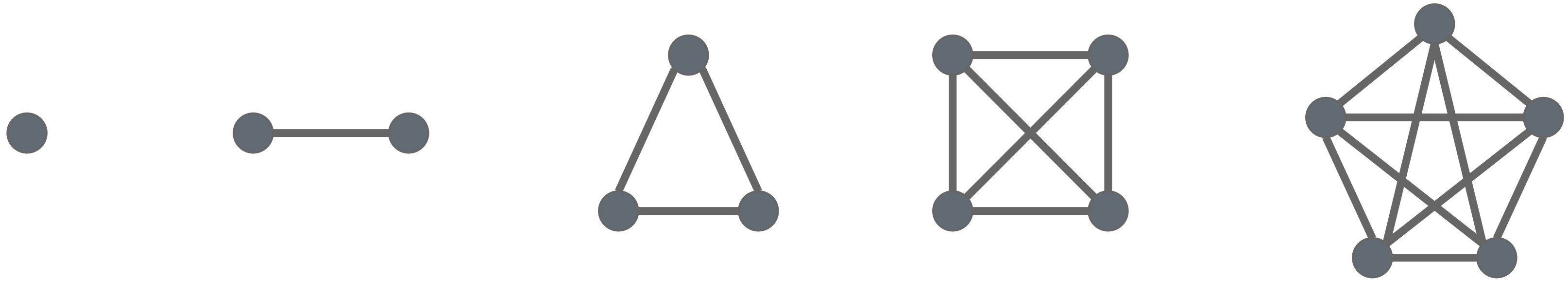
Let $G_4 = (V_4, E_4)$:

$\deg^-(A)$ **2**	$\deg^+(A)$ **0**
$\deg^-(B)$ **2**	$\deg^+(B)$ **1**
$\deg^-(C)$ **1**	$\deg^+(C)$ **3**
$\deg^-(D)$ **0**	$\deg^+(D)$ **3**
$\deg^-(E)$ **2**	$\deg^+(E)$ **0**



Special Simple Graphs

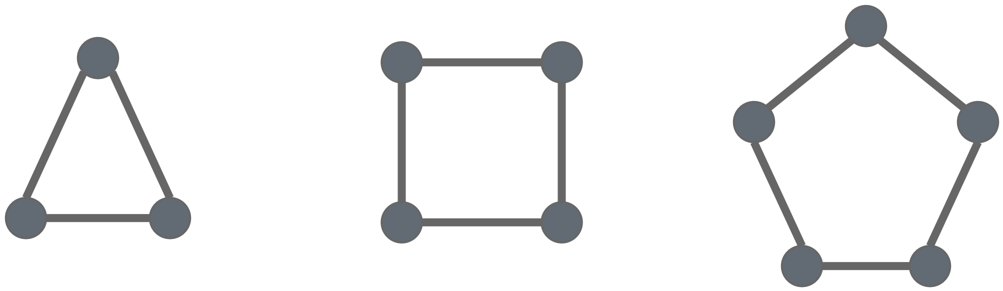
A **complete graph on *n* vertices**, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.



K_1 K_2 K_3 K_4 K_5

Special Simple Graphs

A **cycle**, C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$



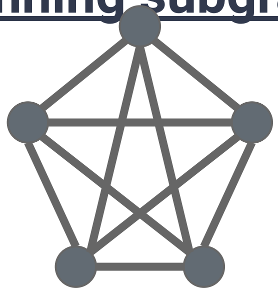
C_3 C_4 C_5

Subgraphs

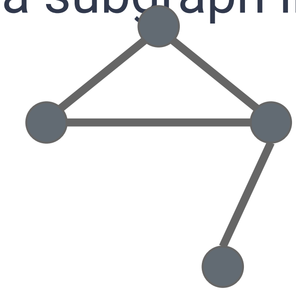
A **subgraph** of a graph ***G*=(*V*,*E*)** is a graph ***H*=(*W*,*F*)** where $W \subseteq V$ and $F \subseteq E$.

A subgraph, ***H*** of ***G*** is a **proper subgraph** of ***G*** if $H \neq G$.

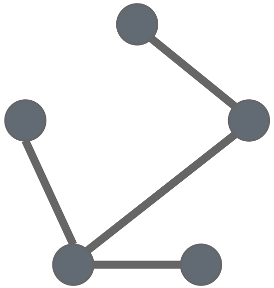
A **spanning subgraph** is a subgraph in which $W = V$



K_5



A subgraph of K_5



A spanning subgraph of K_5

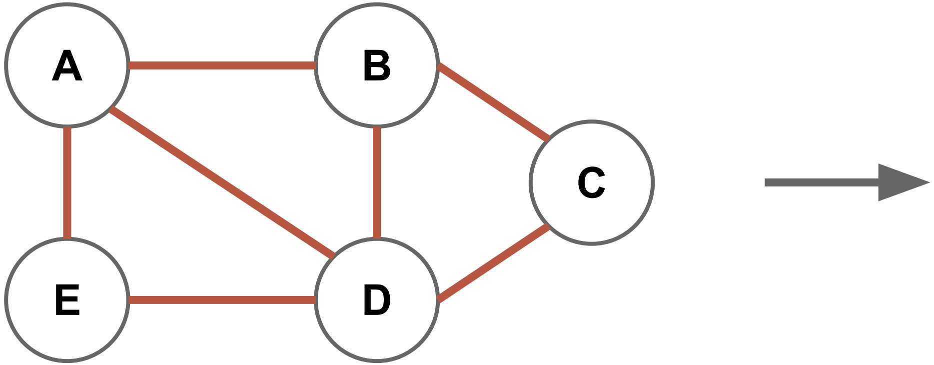
Outline

- What are graphs?
- Graph Examples
- **Graph Representation**
- Handshake Theorem
- Graph Isomorphism
- Connectivity
- Graph Coloring
- Trees

Adjacency List

In the **<u>Adjacency List</u>** representation of a graph, each vertex has a list of all of its neighbors.

Note: *if the graph is undirected, then if **a** is in **b**'s list of neighbors, **b** must also be in **a**'s list of neighbors*

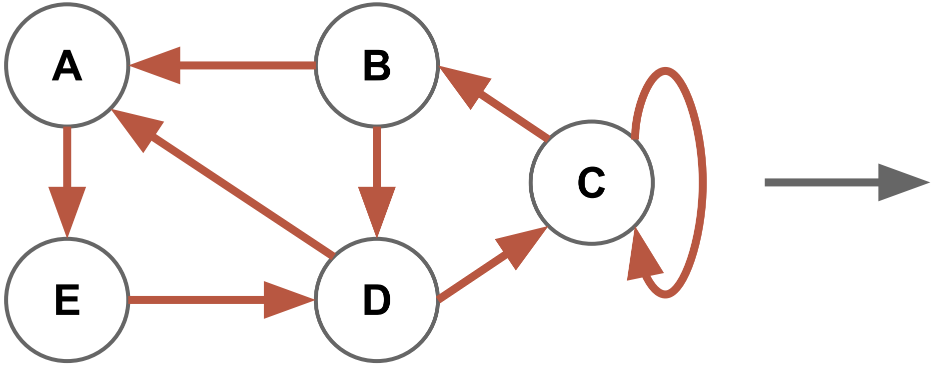


Vertex	Adjacent Vertices
A	B,D,E
B	A,D,C
C	B,D
D	A,B,C,E
E	A,D

Adjacency List

In the **Adjacency List** representation of a graph, each vertex has a list of all of its neighbors.

Note: *if the graph is directed, then the adjacency list shows which vertices we can get to from a given vertex*

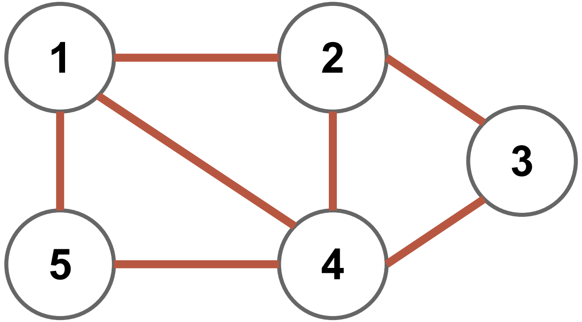


Vertex	Adjacent Vertices
A	E
B	A, D
C	B, C
D	A, C
E	D

Adjacency Matrix

The **Adjacency Matrix** for a graph ***M*** with ***n*** vertices is an ***n*** by ***n*** matrix, whose entries are 0 or 1, indicating if an edge is present.

- $\boldsymbol{M_{i,j}}$ **= 1** iff **{*i*, *j*}** is an edge in the graph
- The vertices of ***M*** are labeled with integers in the range **1** to ***n***
- If ***M*** is undirected, $\boldsymbol{M_{i,j} = M_{j,i}}$ because edge **{*i*, *j*)** = edge **{*j*, *i*}**

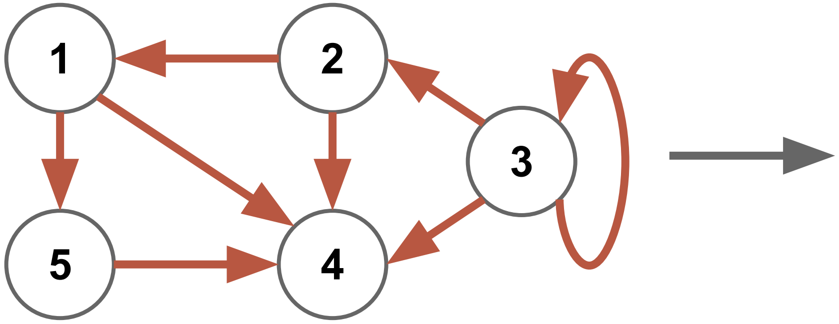


	1	2	3	4	5
1	0	1	0	1	1
2	1	0	1	1	0
3	0	1	0	1	0
4	1	1	1	0	1
5	1	0	0	1	0

Adjacency Matrix

The **Adjacency Matrix** for a graph ***M*** with ***n*** vertices is an ***n*** by ***n*** matrix, whose entries are 0 or 1, indicating if an edge is present.

- $\boldsymbol{M_{i,j}}$ **= 1** iff **(*i*, *j*)** is an edge in the graph
- The vertices of ***M*** are labeled with integers in the range **1** to ***n***
- If ***M*** is directed, $\boldsymbol{M_{i,j}}$ does not necessarily equal $\boldsymbol{M_{j,i}}$

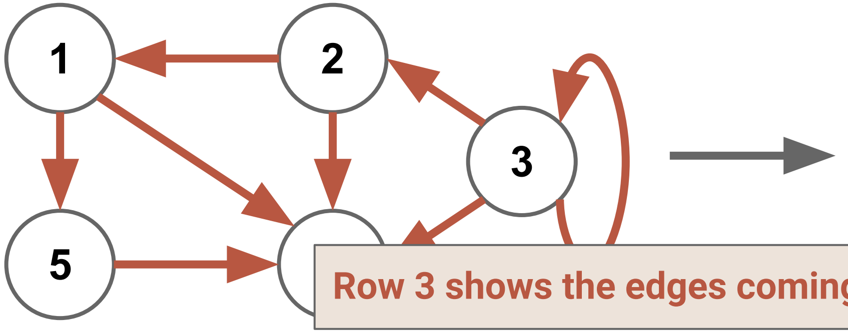


	1	2	3	4	5
1	0	0	0	1	1
2	1	0	0	1	0
3	0	1	1	1	0
4	0	0	0	0	0
5	0	0	0	1	0

Adjacency Matrix

The **Adjacency Matrix** for a graph ***M*** with ***n*** vertices is an ***n*** by ***n*** matrix, whose entries are 0 or 1, indicating if an edge is present.

- $\boldsymbol{M_{i,j}} = \mathbf{1}$ iff $(\boldsymbol{i}, \boldsymbol{j})$ is an edge in the graph
- The vertices of ***M*** are labeled with integers in the range **1** to ***n***
- If ***M*** is directed, $\boldsymbol{M_{i,j}}$ does not necessarily equal $\boldsymbol{M_{j,i}}$



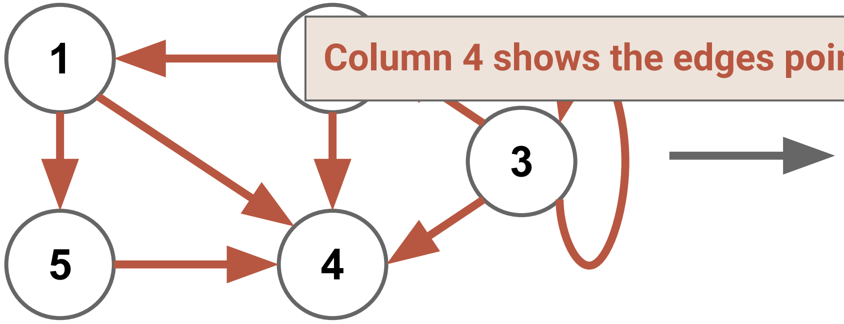
Row 3 shows the edges coming FROM 3

	1	2	3	4	5
1	0	0	0	1	1
2	1	0	0	1	0
3	0	1	1	1	0
4	0	0	0	0	0
5	0	0	0	1	0

Adjacency Matrix

The **Adjacency Matrix** for a graph ***M*** with ***n*** vertices is an ***n*** by ***n*** matrix, whose entries are 0 or 1, indicating if an edge is present.

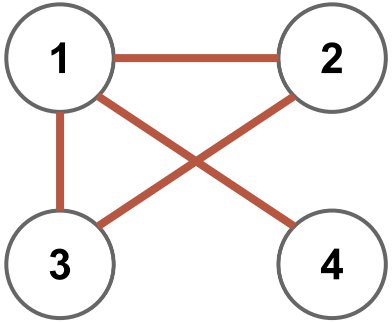
- $M_{i,j} = 1$ iff (i, j) is an edge in the graph
- The vertices of ***M*** are labeled with integers in the range **1** to ***n***
- If ***M*** is directed, $M_{i,j}$ does not necessarily equal $M_{j,i}$



$$
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & \boxed{4} & 5 \\
\hline
1 & 0 & 0 & 0 & 1 & 1 \\
2 & 1 & 0 & 0 & 1 & 0 \\
3 & 0 & 1 & 1 & 1 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 \\
\hline
\end{array}
$$

Graph Representation Examples

Write out the adjacency matrix:

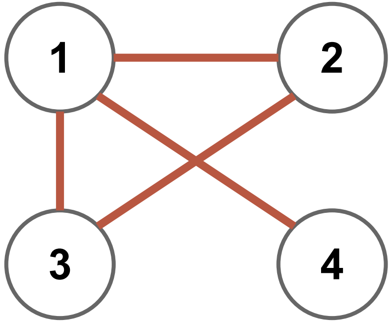


Draw the graph:

$$
\begin{array}{c|cccc|}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 1 & 1 & 0 \\
2 & 1 & 0 & 0 & 1 \\
3 & 1 & 0 & 0 & 1 \\
4 & 0 & 1 & 1 & 0 \\
\hline
\end{array}
$$

Graph Representation Examples

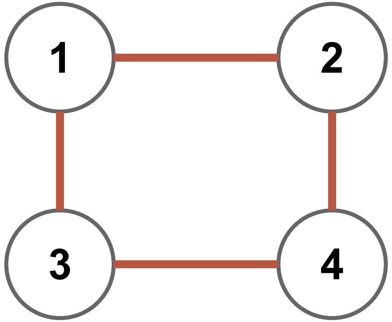
Write out the adjacency matrix:



	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0
4	1	0	0	0

Draw the graph:

	1	2	3	4
1	0	1	1	0
2	1	0	0	1
3	1	0	0	1
4	0	1	1	0



Outline

- What are graphs?
- Graph Examples
- Graph Representation
- **Handshake Theorem**
- Graph Isomorphism
- Connectivity
- Graph Coloring
- Trees

Handshake Theorem

Theorem

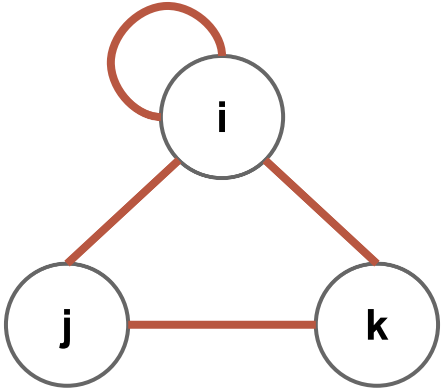
For any undirected graph, ***G* = (*V*, *E*)**: $\sum_{v \in V} deg(v) = 2|E|$

Note: *If* ***V*** *=* *{****v****$_1$, **v**$_2$, …, **v**$_n$} then*

$$\sum_{v \in V} deg(v) = deg(v_1) + deg(v_2) + \ldots + deg(v_n)$$

Handshake Theorem

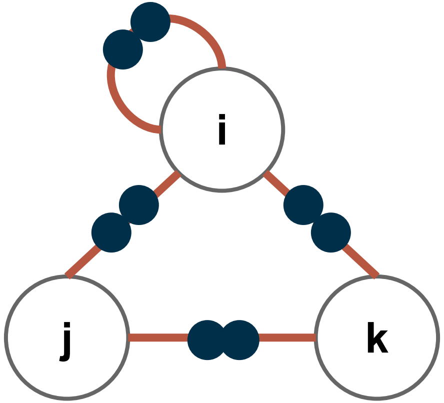
Consider this graph, ***G***, with 3 vertices and 4 edges:



According to the handshake theorem, $\mathbf{2|\mathit{E}| = deg(\mathit{i}) + deg(\mathit{j}) + deg(\mathit{k})}$

Handshake Theorem

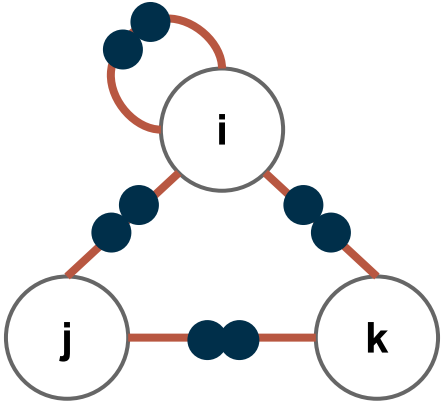
Consider putting 2 dots on every edge



How many dots are there?

Handshake Theorem

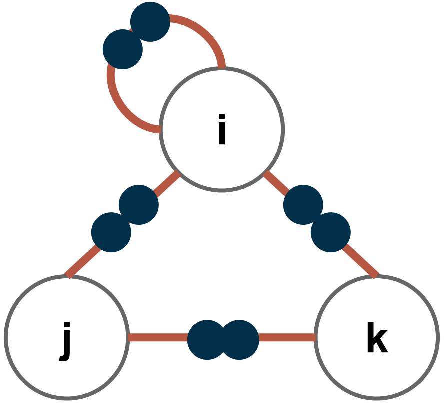
Consider putting 2 dots on every edge



How many dots are there? **2 dots * 4 edges = 8 (aka 2|*E*|)**

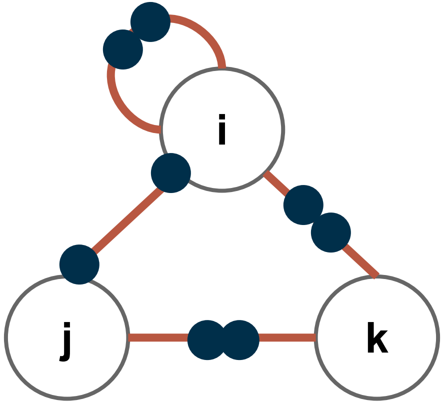
Handshake Theorem

Now move each dot to the incident vertices



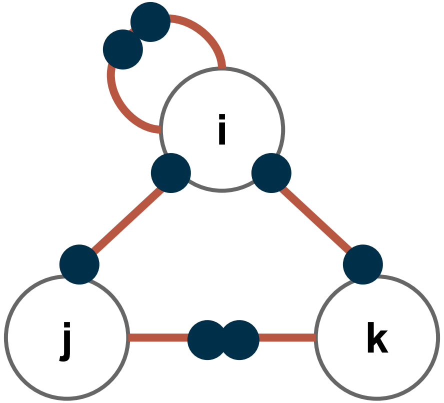
Handshake Theorem

Now move each dot to the incident vertices



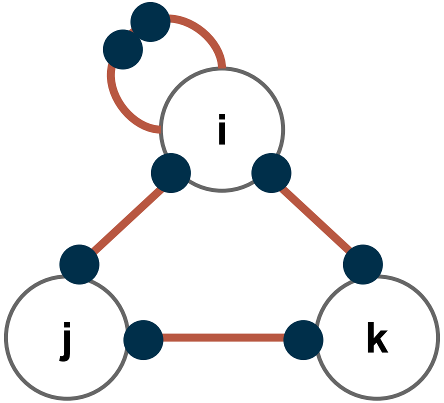
Handshake Theorem

Now move each dot to the incident vertices



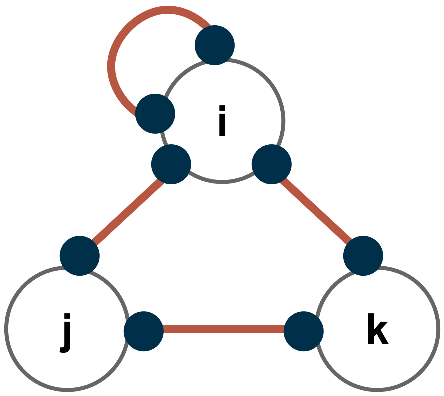
Handshake Theorem

Now move each dot to the incident vertices



Handshake Theorem

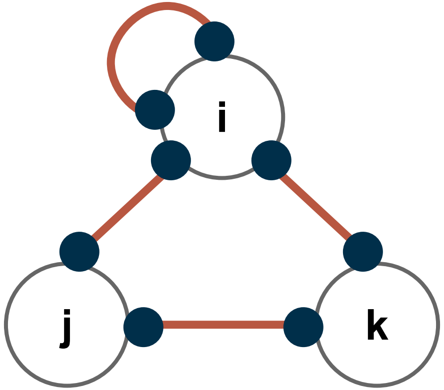
Now move each dot to the incident vertices



Handshake Theorem

Now move each dot to the incident vertices

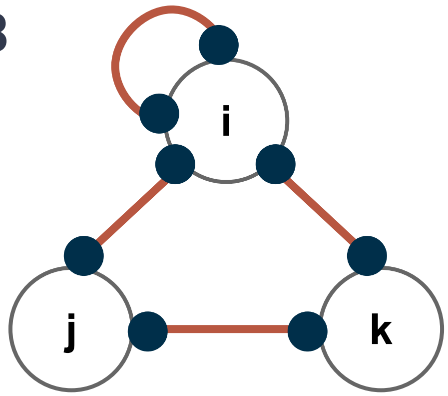
How many dots now?



Handshake Theorem

Now move each dot to the incident vertices

How many dots now? **Still 8**



Handshake Theorem

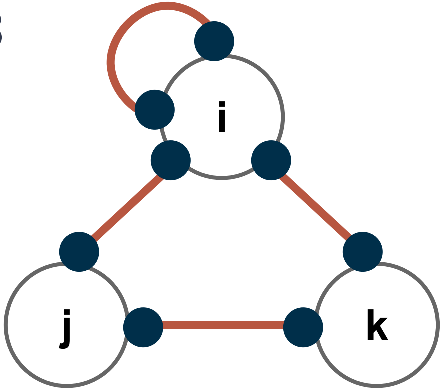
Now move each dot to the incident vertices

How many dots now? **Still 8**

How many touch ***i***?

How many touch ***j***?

How many touch ***k***?



Handshake Theorem

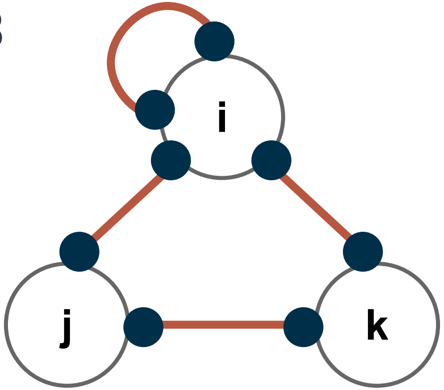
Now move each dot to the incident vertices

How many dots now? **Still 8**

How many touch ***i***? **4**

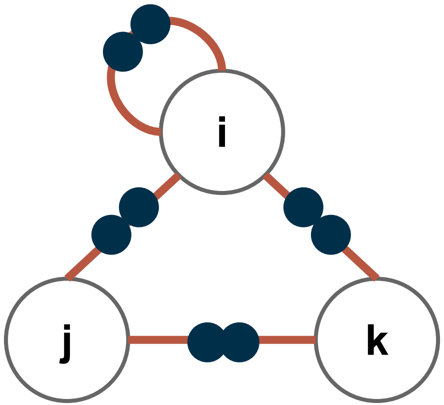
How many touch ***j***? **2**

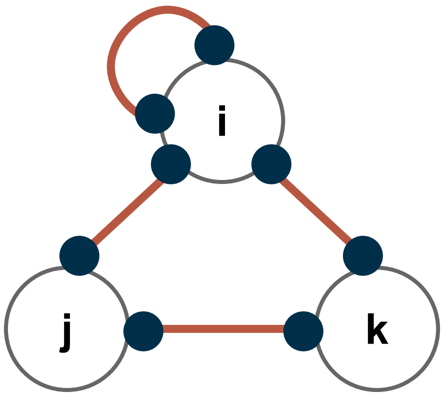
How many touch ***k***? **2**



Handshake Theorem

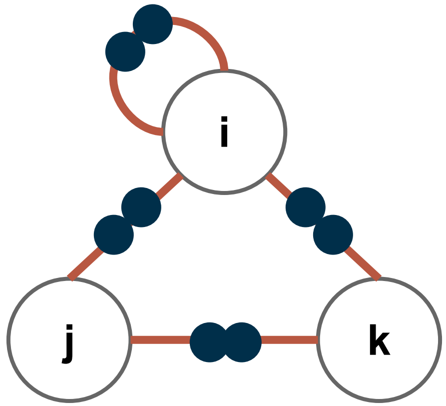
Depending on how the dots are arranged we associate them differently

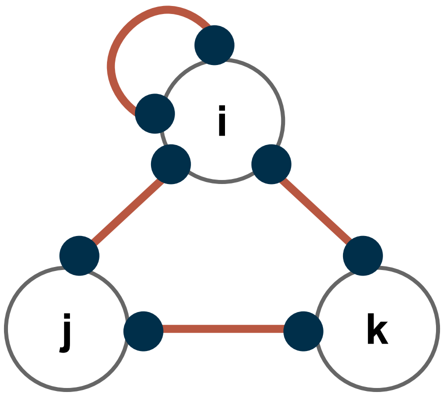




Handshake Theorem

Depending on how the dots are arranged we associate them differently





$$2|E| = 8 = 4 + 2 + 2 = deg(i) + deg(j) + deg(k) = \sum_{v \in V} deg(v)$$

Handshake Theorem Proof Sketch

This is a sketch for the complete proof of the handshake theorem

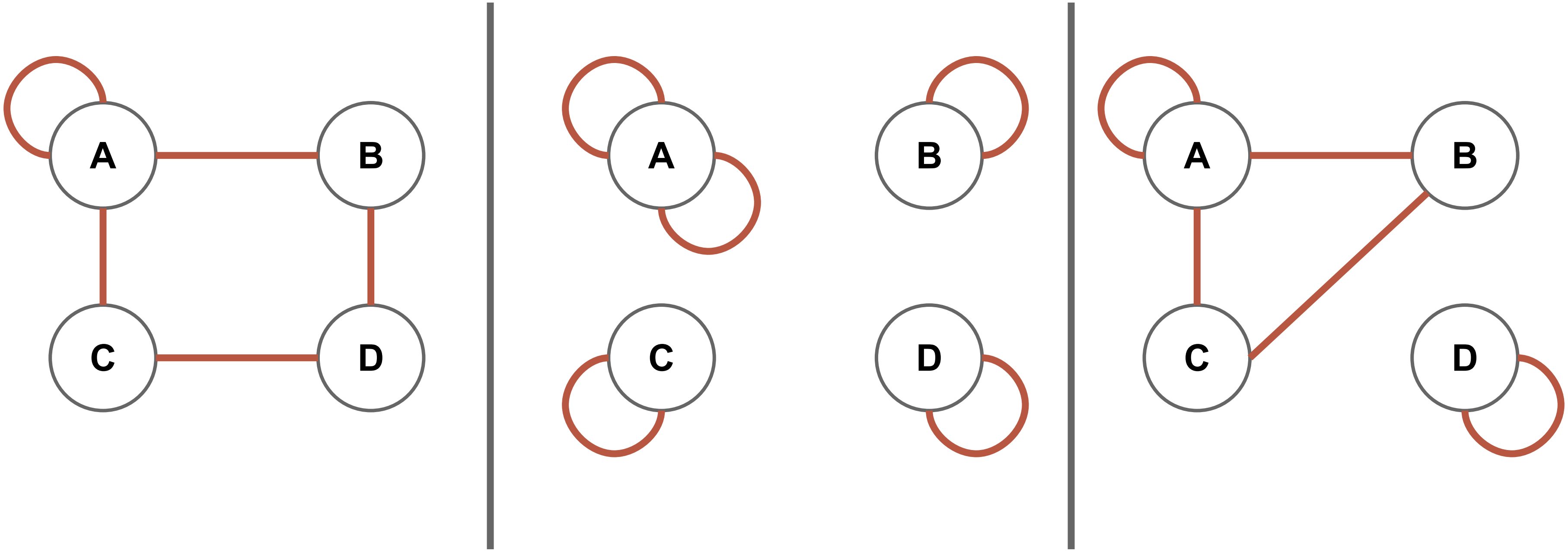
- We only demonstrated it for a single graph
- The full/general proof follows the same kind of idea
  - Proof by induction on the number of edges in the graph
- **Useful application:** Runtime analysis for graph algorithms

Handshake Theorem Application

What could a graph with vertices of degrees 2, 2, 2, and 4 look like?

Handshake Theorem Application

What could a graph with vertices of degrees 2, 2, 2, and 4 look like?



Handshake Theorem Application

Suppose $\boldsymbol{G = (V, E)}$ is a simple, connected graph, with $\boldsymbol{|V| = 5}$

1. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 5}$?
2. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 20}$?
3. If $\boldsymbol{\Sigma_{v \in V} \deg(v) = 14}$, what is the smallest value $\boldsymbol{\deg(v)}$ can have for any vertex $\boldsymbol{v \in V}$?
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

Handshake Theorem Application

Suppose **G = (V, E)** is a simple, **connected** graph, with **$|V|$ = 5**

1. Is it possible for $\sum_{v \in V} \deg(v) = 5$?
2. Is it possible for $\sum_{v \in V} \deg(v) = 20$?
3. If $\sum_{v \in V} \deg(v) = 14$, what is the smallest value $\deg(v)$ can have for any vertex $v \in V$?
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

A connected graph is one where every vertex has a path of edges leading to every other vertex

Handshake Theorem Application

Suppose $\boldsymbol{G = (V, E)}$ is a simple, connected graph, with $\boldsymbol{|V| = 5}$

1. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 5}$?
2. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 20}$?
3. If $\boldsymbol{\Sigma_{v \in V} \deg(v) = 14}$, what is the smallest value $\boldsymbol{\deg(v)}$ can have for any vertex $\boldsymbol{v \in V}$?
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

Handshake Theorem Application

Suppose $\boldsymbol{G = (V, E)}$ is a simple, connected graph, with $\boldsymbol{|V| = 5}$

1. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 5}$? **No.** $\boldsymbol{\Sigma_{v \in V} \deg(v) = 2\,|E|}$**...can't be odd**
2. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 20}$?
3. If $\boldsymbol{\Sigma_{v \in V} \deg(v) = 14}$, what is the smallest value $\boldsymbol{\deg(v)}$ can have for any vertex $\boldsymbol{v \in V}$?
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

Handshake Theorem Application

Suppose **G = (V, E)** is a simple, connected graph, with **$|V| = 5$**

1. Is it possible for **$\Sigma_{v \in V}\deg(v) = 5$**? **No. $\Sigma_{v \in V}\deg(v) = 2\,|E|$...can't be odd**
2. Is it possible for **$\Sigma_{v \in V}\deg(v) = 20$**? **Yes. If $|E| = 10$, $\Sigma_{v \in V}\deg(v) = 20$**
3. If **$\Sigma_{v \in V}\deg(v) = 14$**, what is the smallest value **$\deg(v)$** can have for any vertex **$v \in V$**?
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

Handshake Theorem Application

Suppose $\boldsymbol{G = (V, E)}$ is a simple, connected graph, with $\boldsymbol{|V| = 5}$

1. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 5}$? **No.** $\boldsymbol{\Sigma_{v \in V} \deg(v) = 2\,|E|}$**...can't be odd**
2. Is it possible for $\boldsymbol{\Sigma_{v \in V} \deg(v) = 20}$? **Yes. If** $\boldsymbol{|E| = 10, \Sigma_{v \in V} \deg(v) = 20}$
3. If $\boldsymbol{\Sigma_{v \in V} \deg(v) = 14}$, what is the smallest value $\boldsymbol{\deg(v)}$ can have for any vertex $\boldsymbol{v \in V}$? **1.** $\boldsymbol{|E| = 7}$**, degrees could be 1, 3, 3, 3, 4 for example**
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**?

Handshake Theorem Application

Suppose **$G = (V, E)$** is a simple, connected graph, with **$|V| = 5$**

1. Is it possible for $\sum_{v \in V} \deg(v) = 5$? **No. $\sum_{v \in V} \deg(v) = 2\,|E|$...can't be odd**
2. Is it possible for $\sum_{v \in V} \deg(v) = 20$? **Yes. If $|E| = 10$, $\sum_{v \in V} \deg(v) = 20$**
3. If $\sum_{v \in V} \deg(v) = 14$, what is the smallest value $\deg(v)$ can have for any vertex $v \in V$? **1. $|E| = 7$, degrees could be 1, 3, 3, 3, 4 for example**
4. How many edges are there in a graph that has **10** vertices, with each vertex having degree **6**? **$\sum_{v \in V} \deg(v) = 60 = 2|E|$. $|E| = 30$**

Outline

- What are graphs?
- Graph Examples
- Graph Representation
- Handshake Theorem
- **Graph Isomorphism**
- Connectivity
- Graph Coloring
- Trees

Isomorphism of Graphs

The simple graphs $\boldsymbol{G_1} = (\boldsymbol{V_1}, \boldsymbol{E_1})$ and $\boldsymbol{G_2} = (\boldsymbol{V_2}, \boldsymbol{E_2})$ are **isomorphic** if there exists a one-to-one and onto function f from $\boldsymbol{V_1}$ to $\boldsymbol{V_2}$ with the property that $\boldsymbol{a}$ and $\boldsymbol{b}$ are adjacent in $\boldsymbol{G_1}$ if and only if $f(\boldsymbol{a})$ and $f(\boldsymbol{b})$ are adjacent in $\boldsymbol{G_2}$ for all $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\boldsymbol{V_1}$

- Such a function is called an **isomorphism**
- Two simple graphs that are not isomorphic are called **nonisomorphic**

Isomorphism of Graphs

A property is said to be **preserved under isomorphism** if whenever two graphs are isomorphic, one graph has the property iff the other graph also has that property.

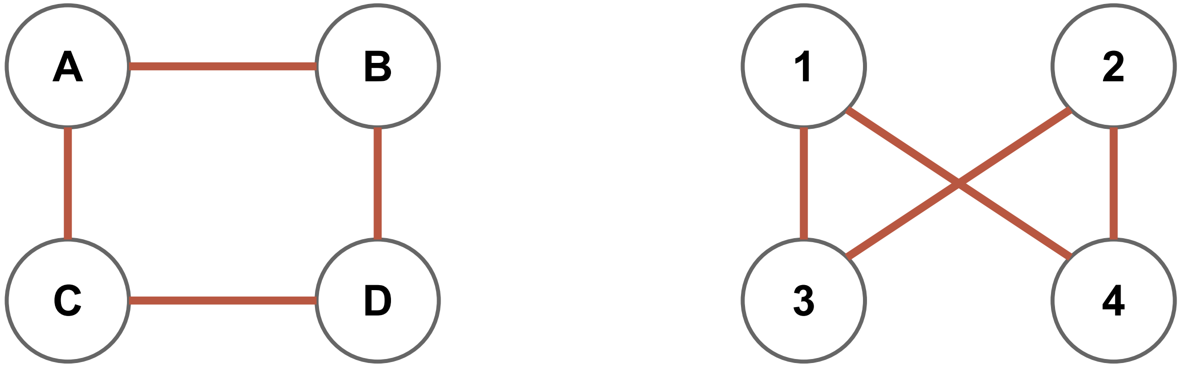
A property preserved by isomorphism of graphs is called a **graph invariant**

Isomorphic simple graphs must also have:

- The same number of vertices
- The same number of edges
- The same degree for all vertices

Isomorphism Examples

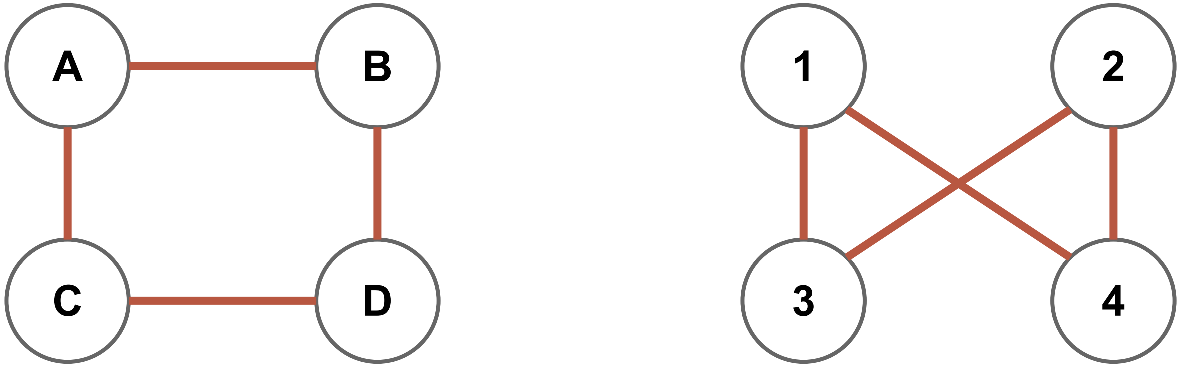
Are these graphs isomorphic?



Isomorphism Examples

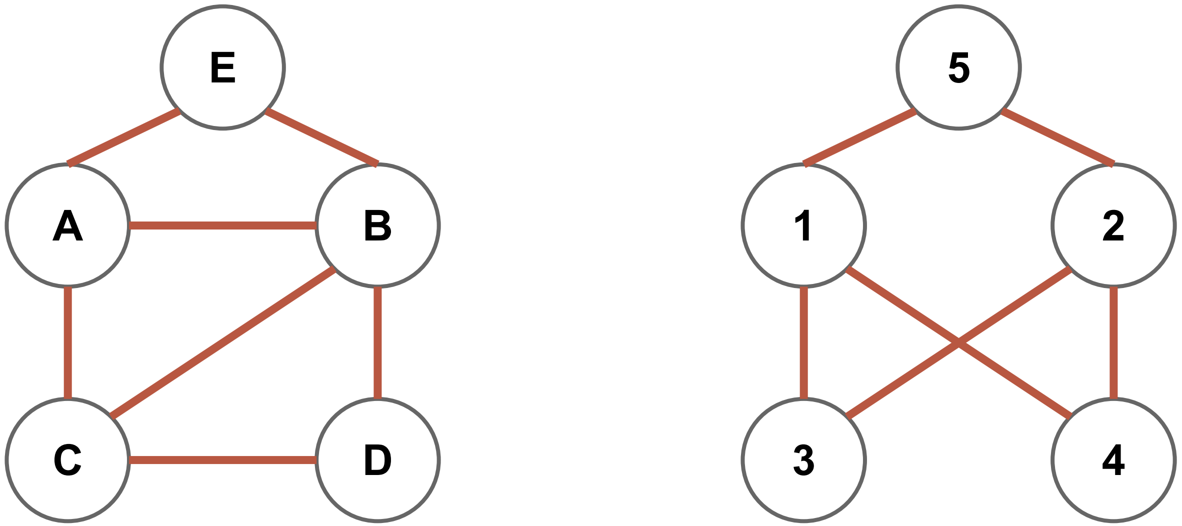
Are these graphs isomorphic?

Yes. A possible isomorphism: {(A,1), (D,2), (C,3), (B,4)}



Isomorphism Examples

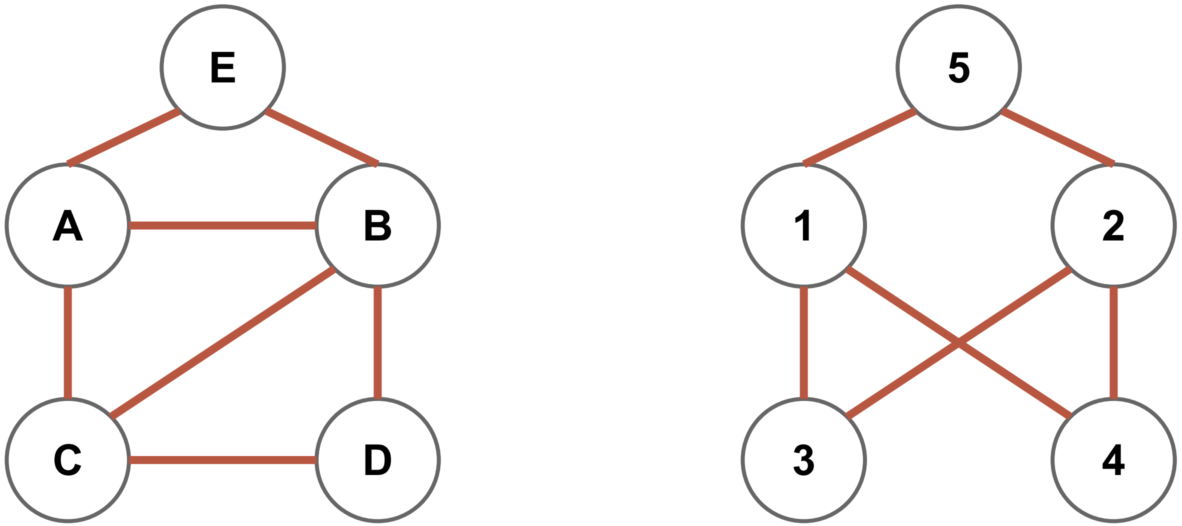
Are these graphs isomorphic?



Isomorphism Examples

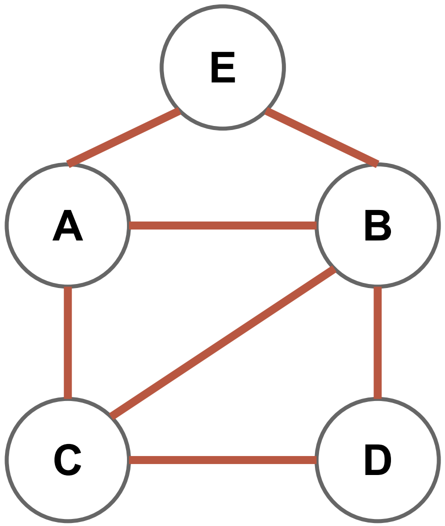
Are these graphs isomorphic?

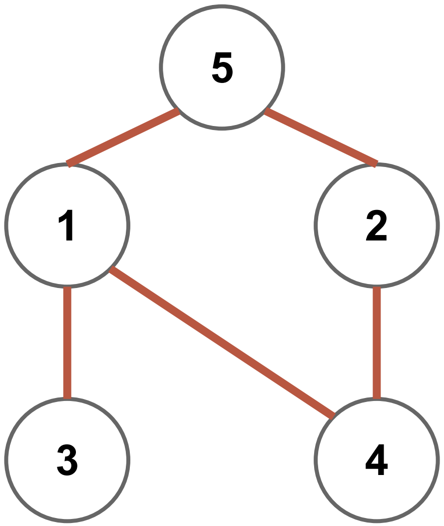
No. The left graph has a vertex with degree 4, the right does not...



Isomorphism Examples

Are these graphs isomorphic?

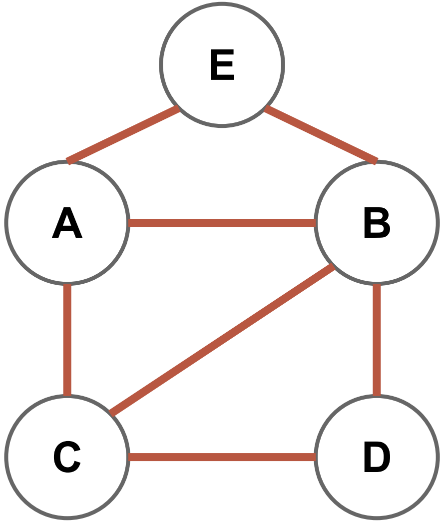


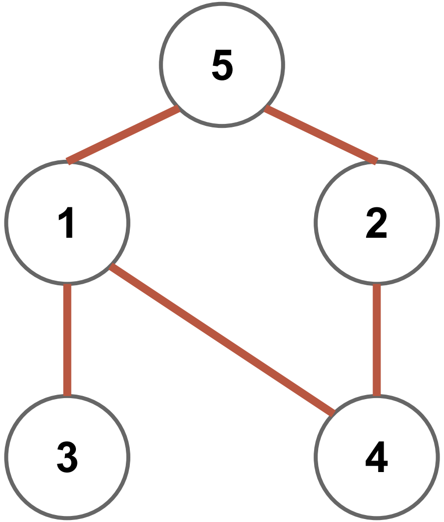


Isomorphism Examples

Are these graphs isomorphic?

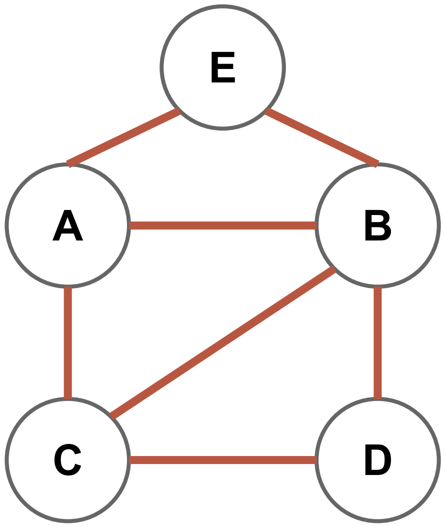
No. They don't have the same number of edges

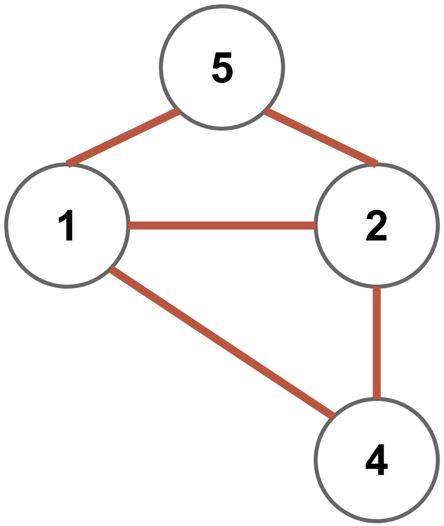




Isomorphism Examples

Are these graphs isomorphic?

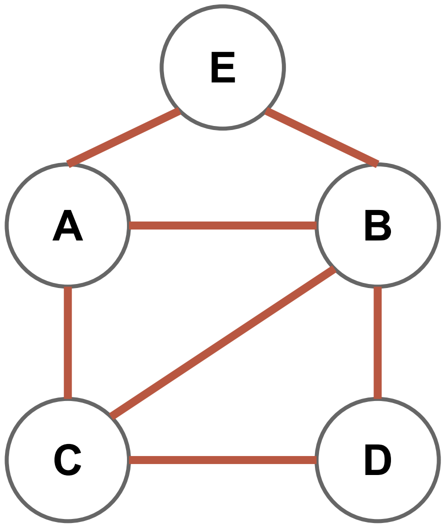


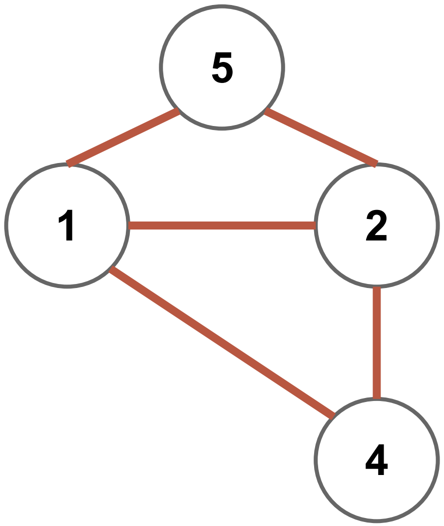


Isomorphism Examples

Are these graphs isomorphic?

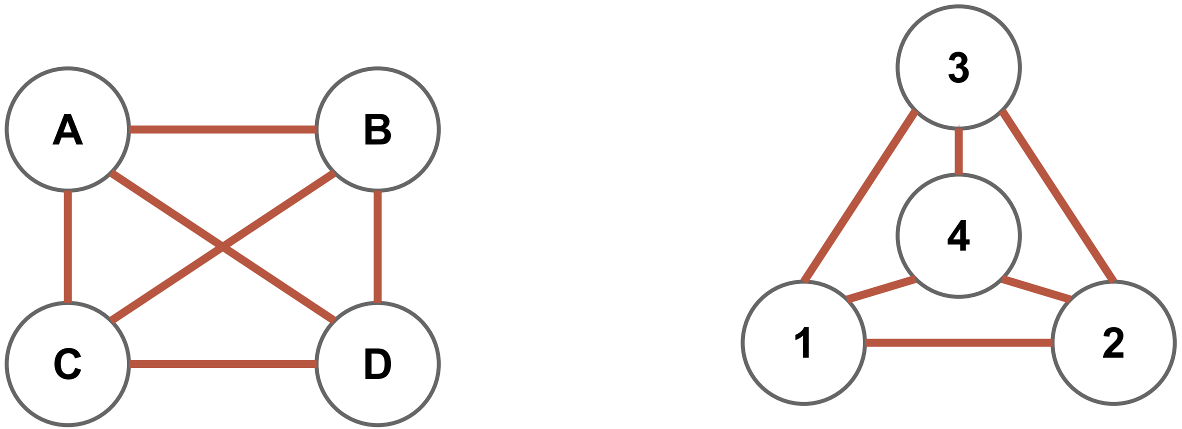
No. They don't have the same number of vertices (or edges, or degrees)





Isomorphism Examples

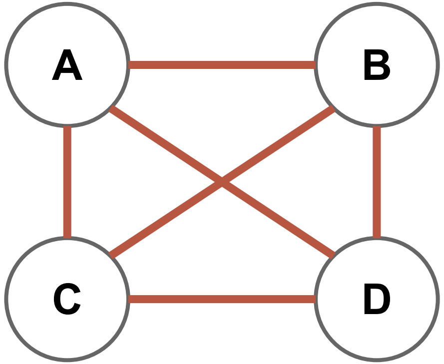
Are these graphs isomorphic?

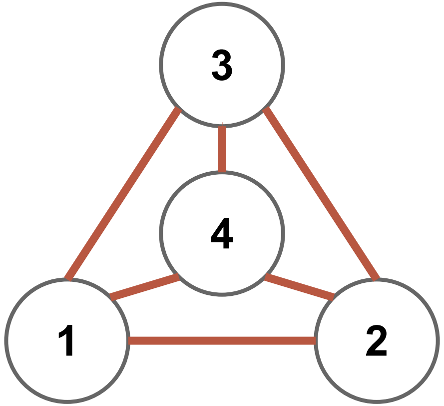


Isomorphism Examples

Are these graphs isomorphic?

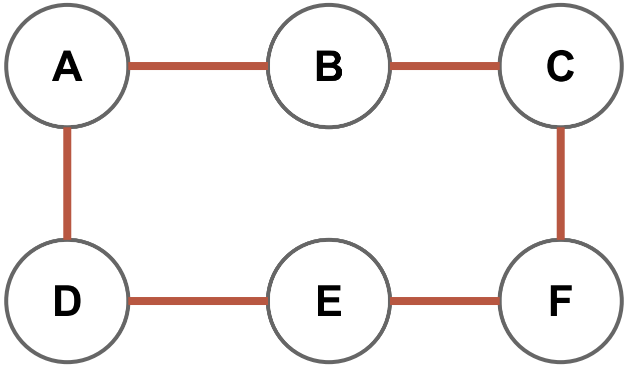
Yes. A possible isomorphism: {(A,1), (B,2), (C,3), (D,4)}

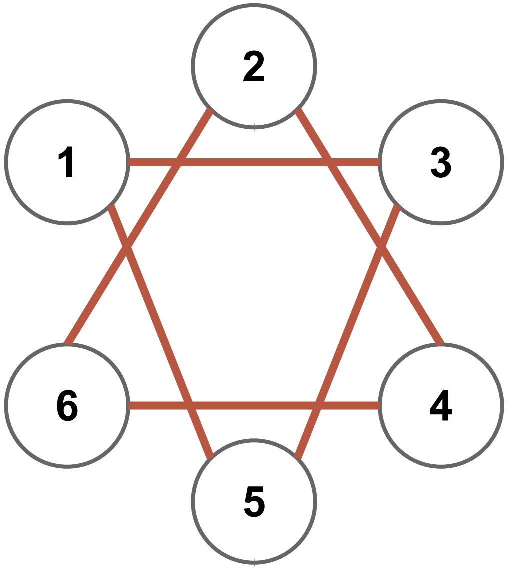




Isomorphism Examples

Are these graphs isomorphic?

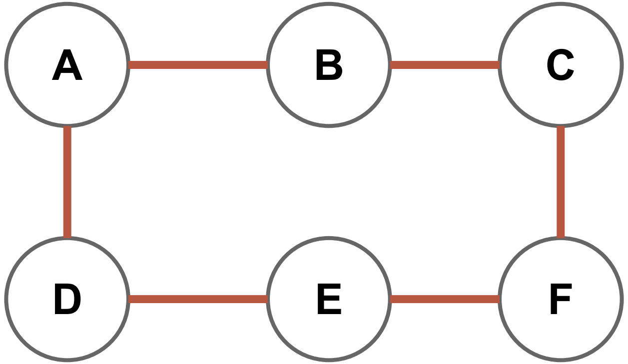


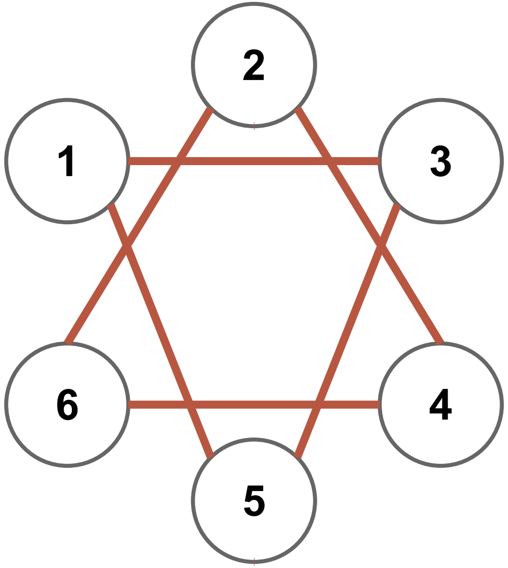


Isomorphism Examples

Are these graphs isomorphic?

No. Same number of vertices, edges, and same degrees…but no bijective function that preserves adjacency





Outline

- What are graphs?
- Graph Examples
- Graph Representation
- Handshake Theorem
- Graph Isomorphism
- **Connectivity**
- Graph Coloring
- Trees

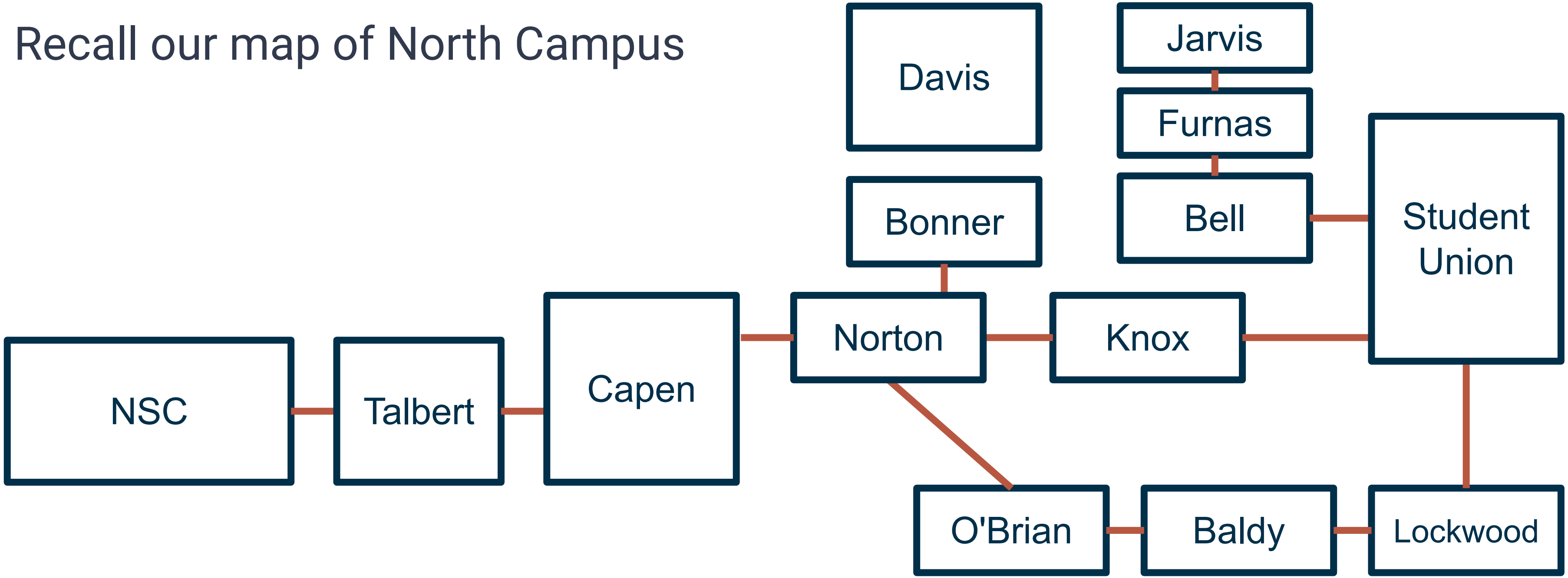
Paths

Consider a graph with nodes representing locations, and edges representing whether or not you can walk between two locations…

A common question we may ask is can we walk from point ***A*** to point ***B***?

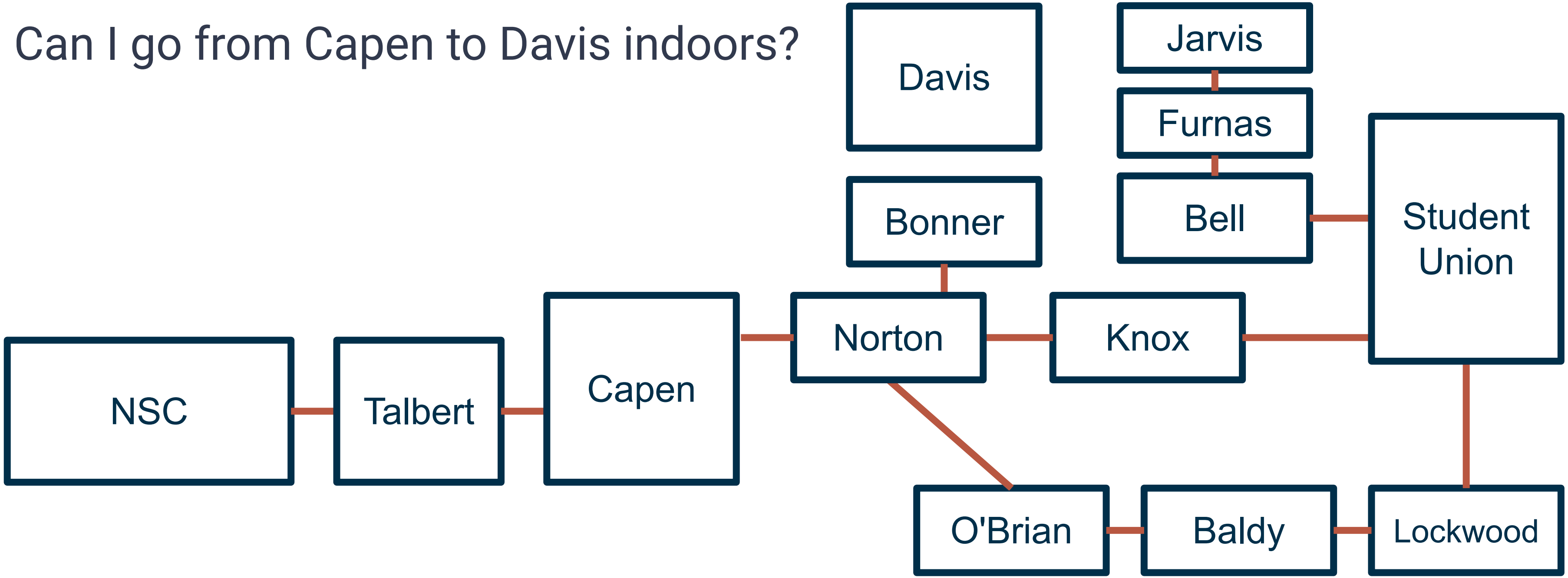
Paths

Recall our map of North Campus



Paths

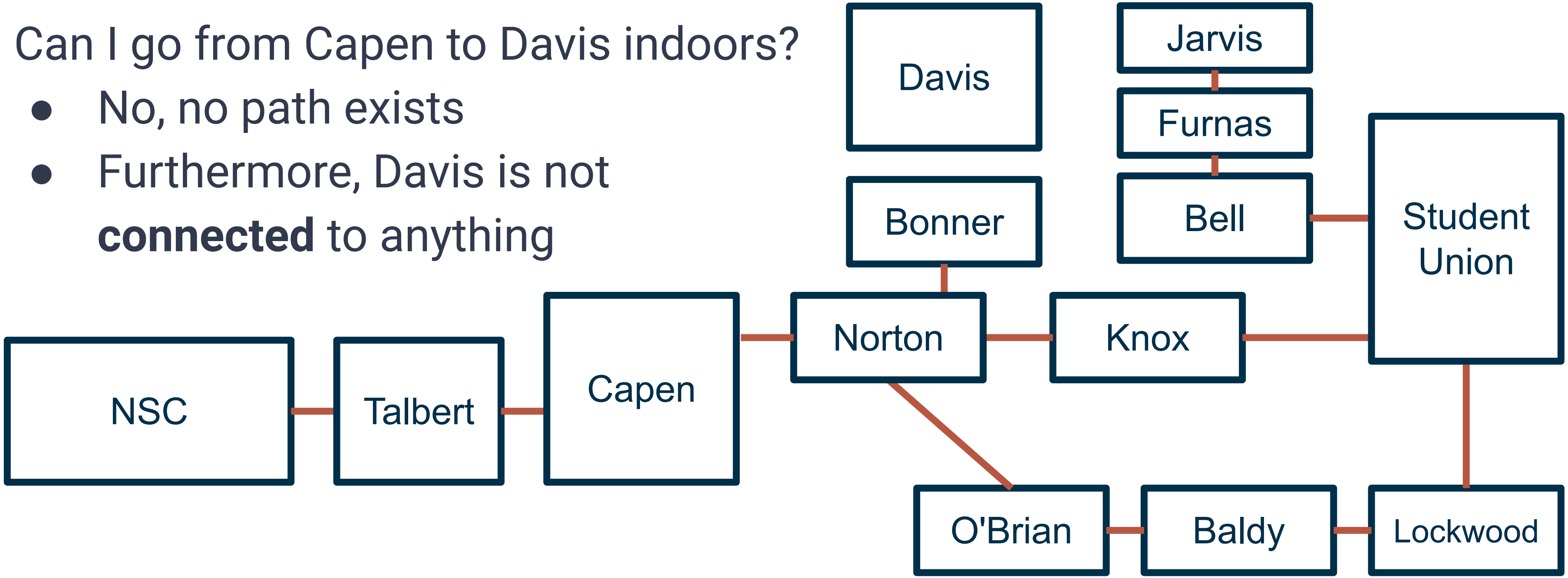
Can I go from Capen to Davis indoors?



Paths

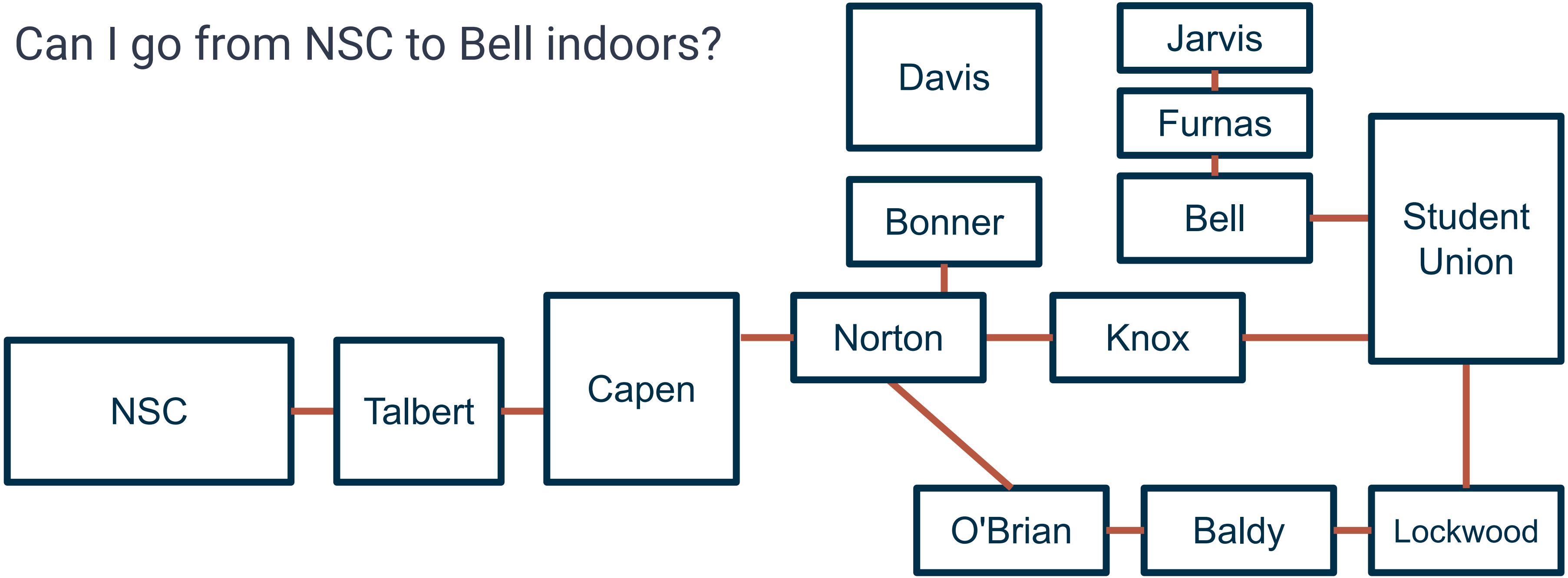
Can I go from Capen to Davis indoors?

- No, no path exists
- Furthermore, Davis is not **connected** to anything



Paths

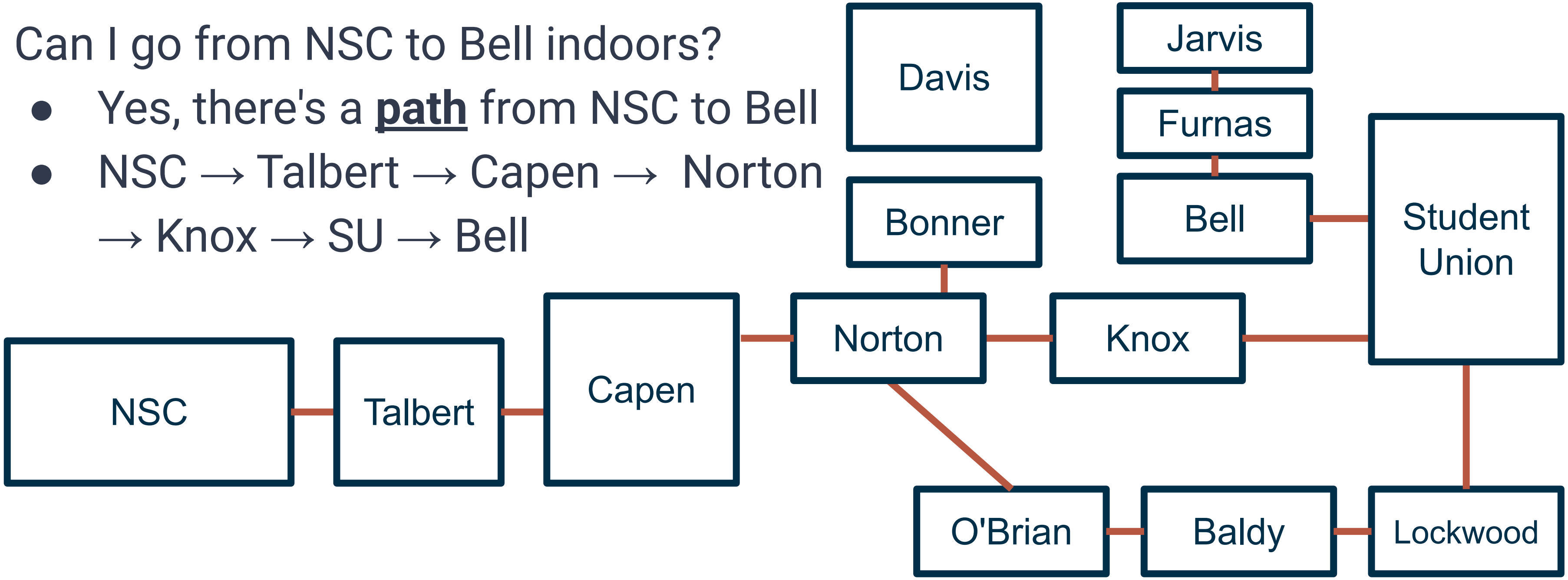
Can I go from NSC to Bell indoors?



Paths

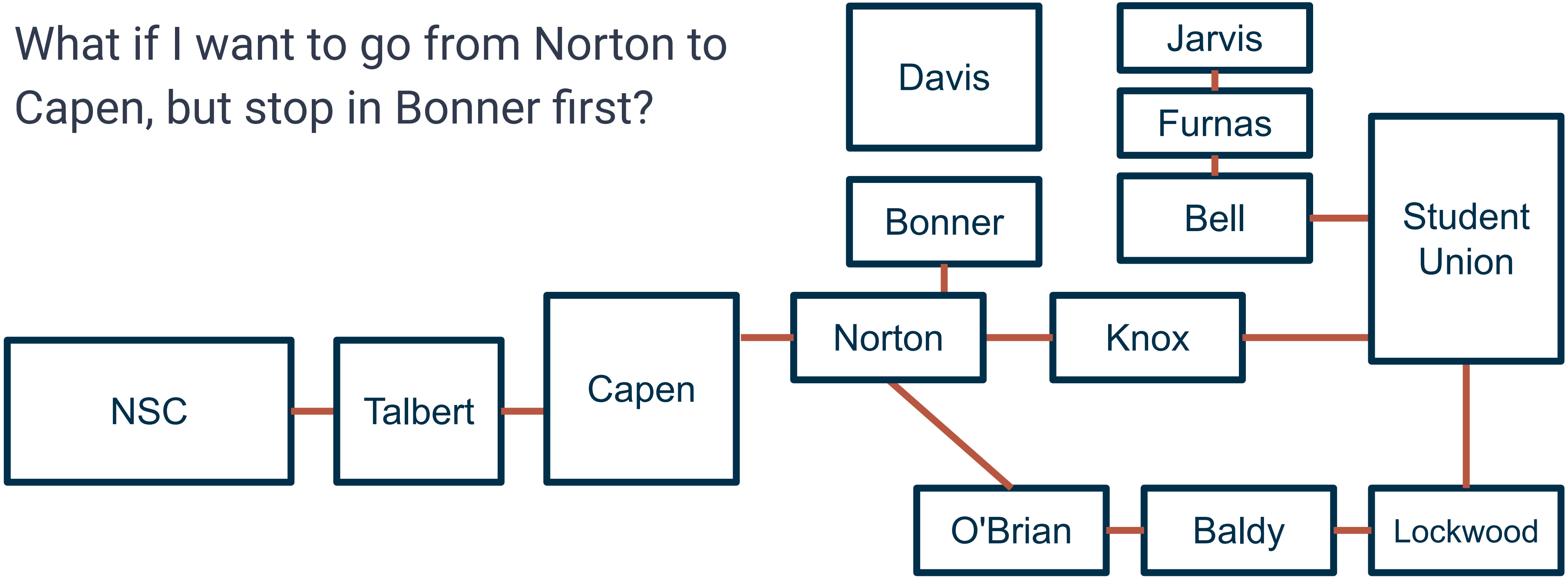
Can I go from NSC to Bell indoors?

- Yes, there's a **path** from NSC to Bell
- NSC → Talbert → Capen → Norton → Knox → SU → Bell



Paths

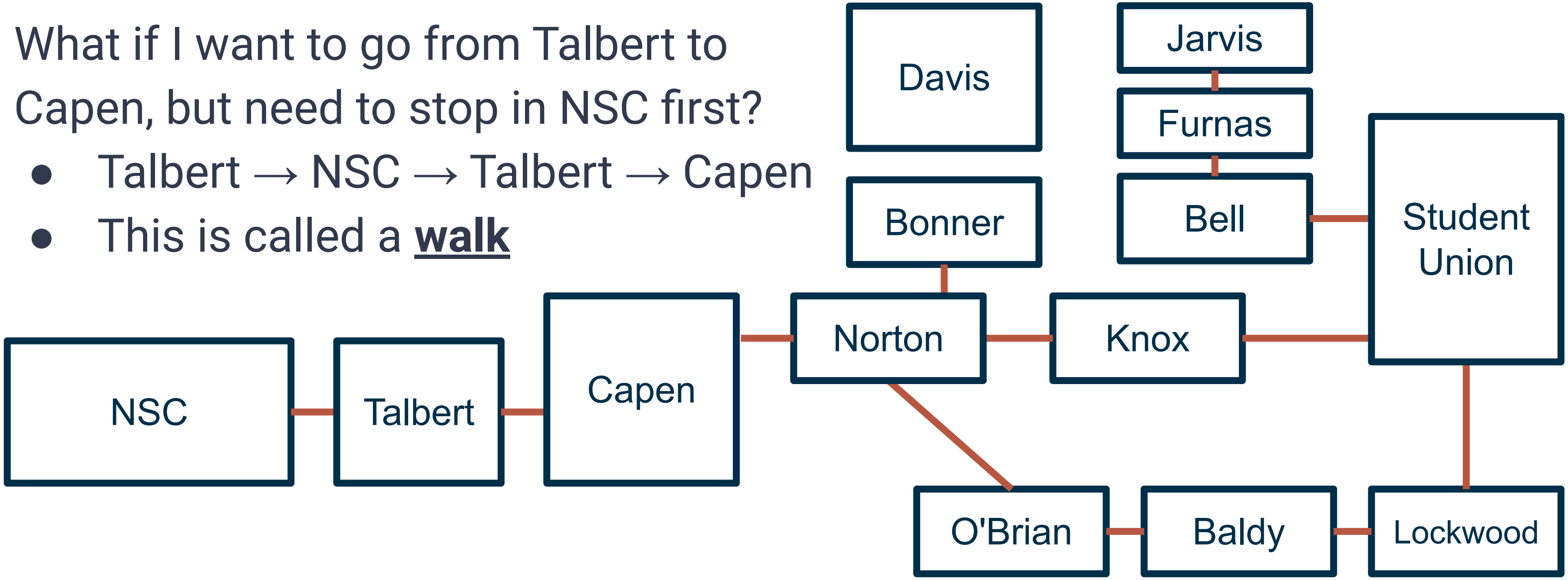
What if I want to go from Norton to Capen, but stop in Bonner first?



Paths

What if I want to go from Talbert to Capen, but need to stop in NSC first?

- Talbert → NSC → Talbert → Capen
- This is called a **walk**



Walks, Trails, and Paths

A **walk** from ***s*** to ***t*** is a sequence of vertices $s, v_1, v_2, \ldots, v_k, t$ such that there is an edge between any two consecutive vertices in the list

- If the first and last vertices are different it's an **open walk**
- If they are the same, then it's a **closed walk**

A **trail** from ***s*** to ***t*** is an open walk that has no repeated edges

A **path** from ***s*** to ***t*** is a trail from ***s*** to ***t*** where all vertices are unique

- Note that a path is a special case of a trail

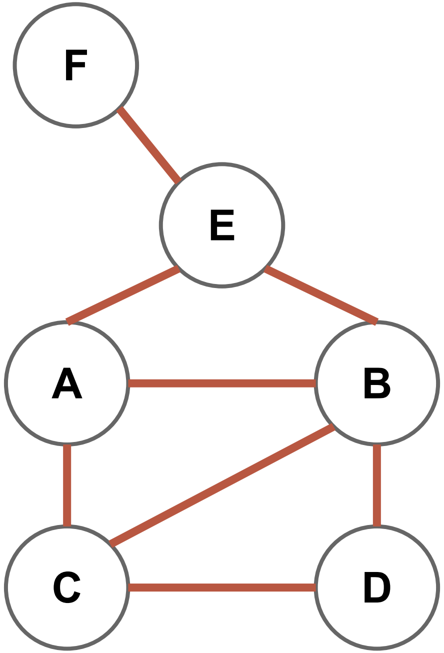
Circuits and Cycles

A **circuit** is a closed walk that has no repeated edges

A **cycle** is a circuit with length at least three such that there are no repeated vertices other than the first and the last.

- A closed path is a cycle

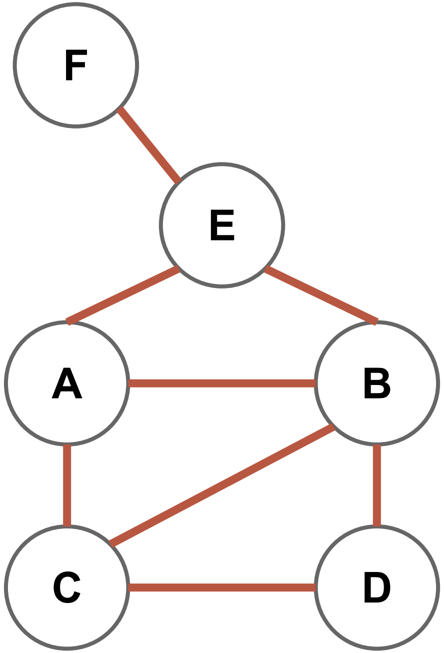
Example



Is the following a walk, trail, path, cycle, and/or circuit?

A, B, D, A

Example

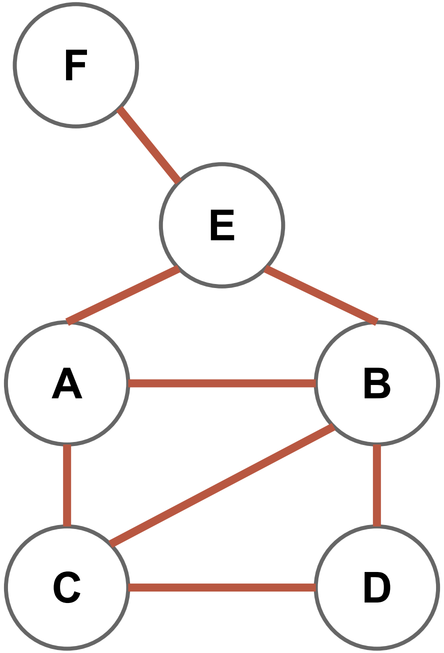


Is the following a walk, trail, path, cycle, and/or circuit?

A, B, D, A

None of the above (there is no edge from D to A)

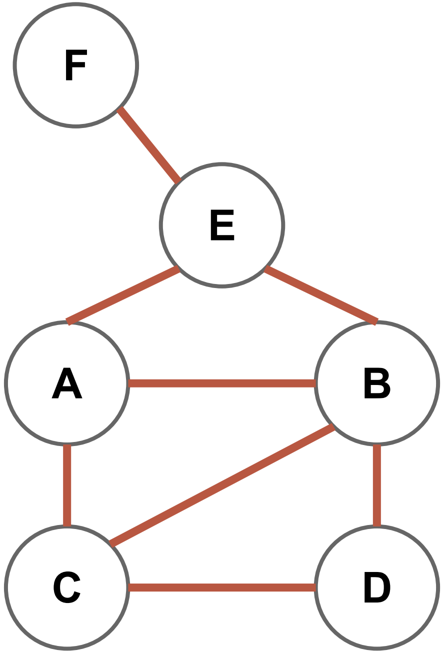
Example



Is the following a walk, trail, path, cycle, and/or circuit?

A, B, A, C, B

Example



Is the following a walk, trail, path, cycle, and/or circuit?

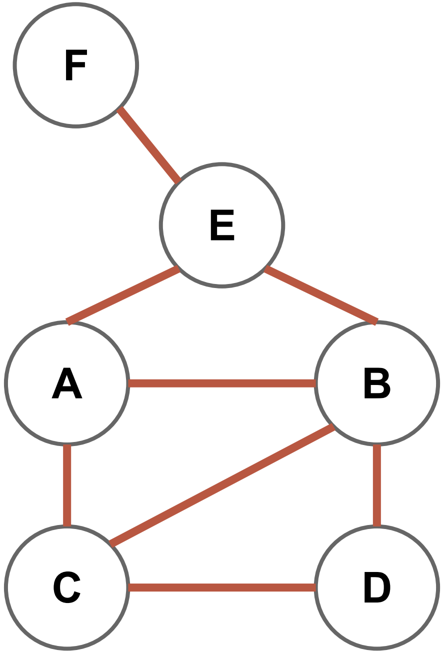
A, B, A, C, B

It is a walk

(not a trail, we take {A,B} twice)

(not a cycle or circuit because it doesn't A ≠ B)

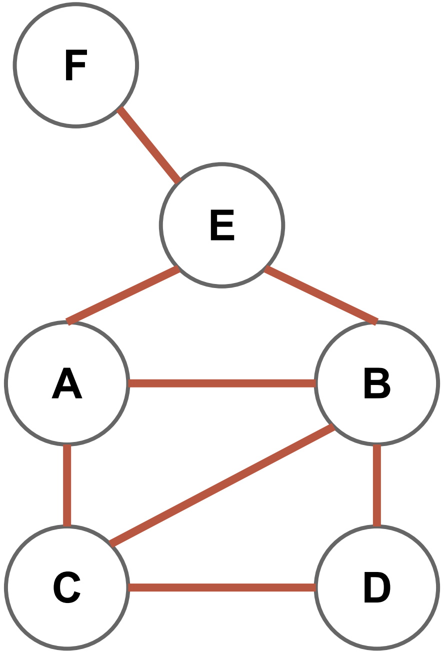
Example



Is the following a walk, trail, path, cycle, and/or circuit?

E, B, A, E, F

Example



Is the following a walk, trail, path, cycle, and/or circuit?

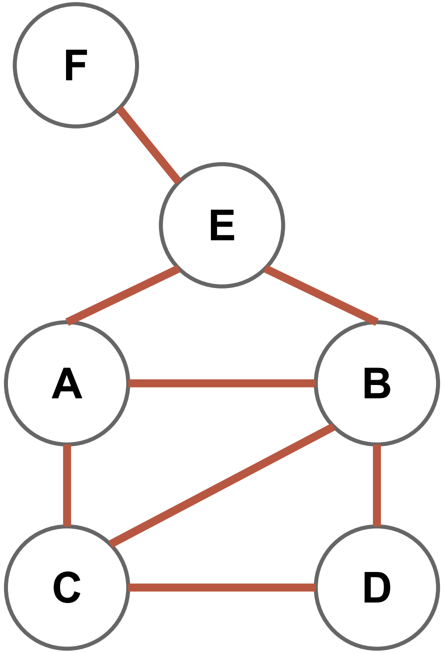
E, B, A, E, F

walk and trail

(not a path, it visits E twice)

(not a cycle or circuit, E ≠ F)

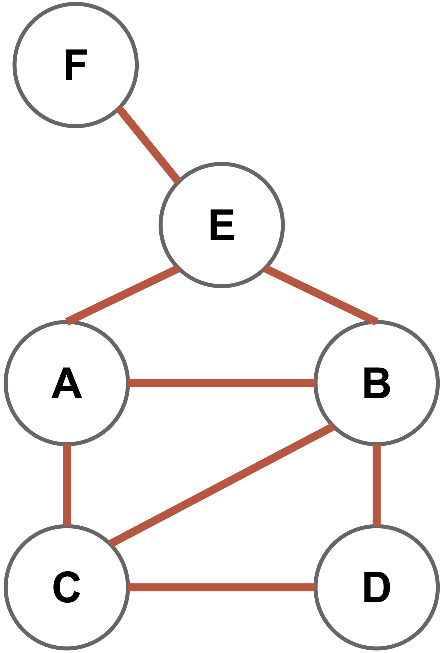
Example



Is the following a walk, trail, path, cycle, and/or circuit?

F, E, B, C, A

Example



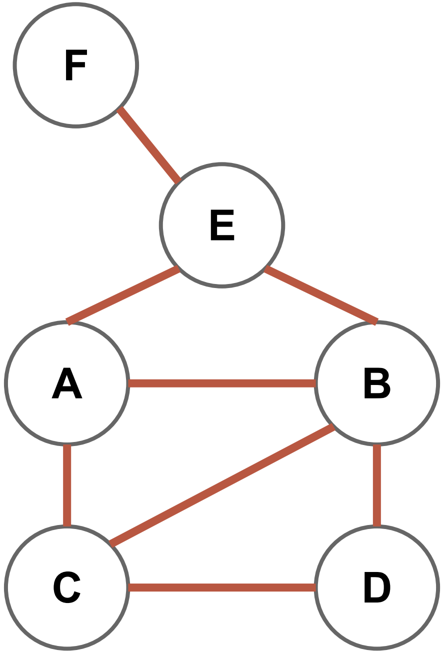
Is the following a walk, trail, path, cycle, and/or circuit?

F, E, B, C, A

walk, trail, and path

(not a circuit or cycle, F ≠ A)

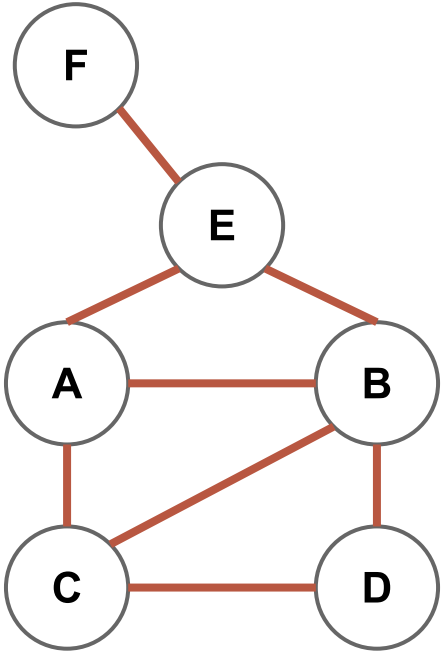
Example



Is the following a walk, trail, path, cycle, and/or circuit?

E, B, C, A, B, E

Example



Is the following a walk, trail, path, cycle, and/or circuit?

E, B, C, A, B, E

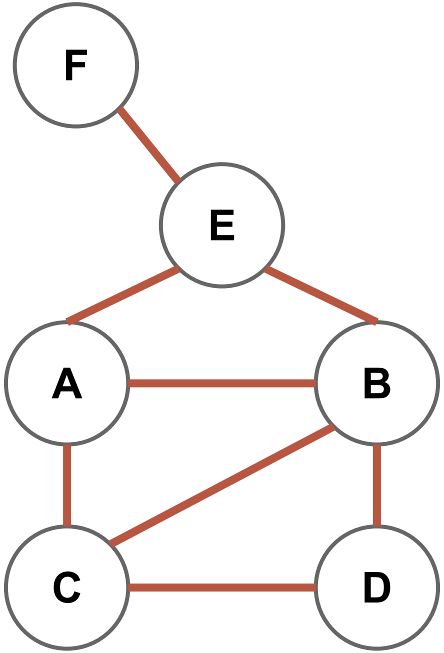
a closed walk

(not a trail, because it is closed)

(not a circuit because {B,E} repeats)

(not a path or cycle because B repeats)

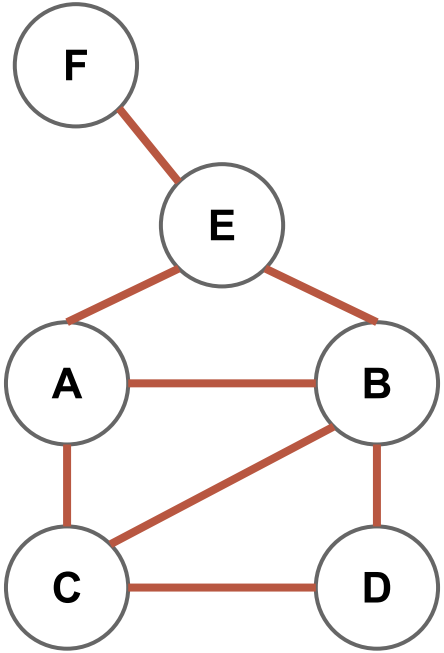
Example



Is the following a walk, trail, path, cycle, and/or circuit?

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

Example



Is the following a walk, trail, path, cycle, and/or circuit?

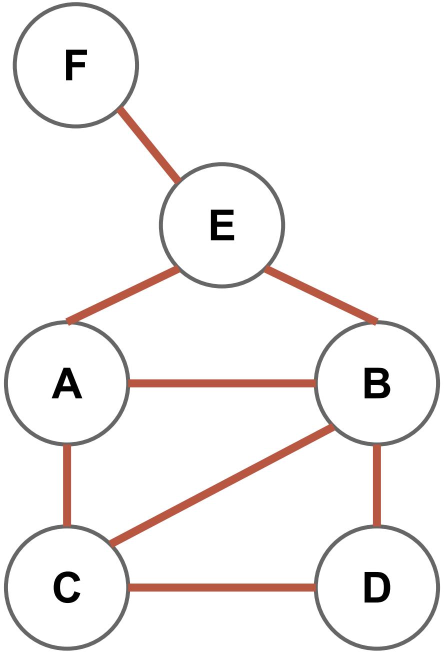
E, B, D, C, B, A, E

a closed walk, and a circuit (no repeated edges)

(not a trail, because it is closed)

(not a path or cycle because B repeats)

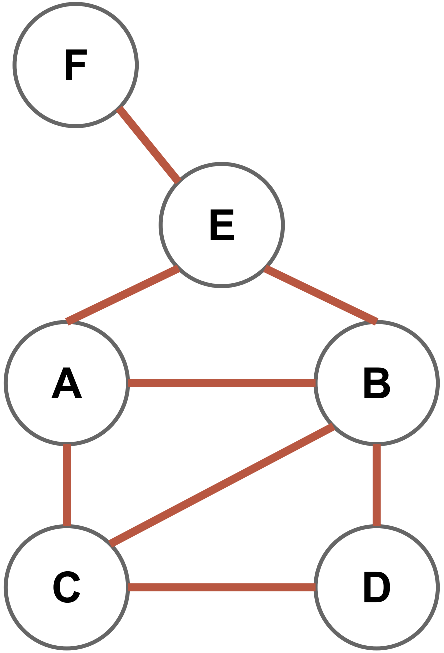
Example



Is the following a walk, trail, path, cycle, and/or circuit?

A, B, C, A

Example



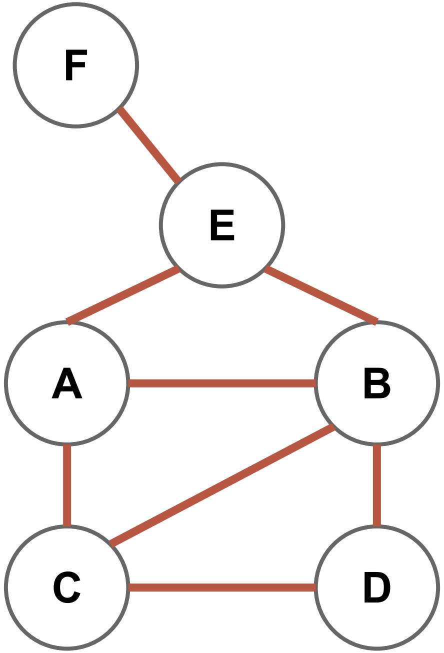
Is the following a walk, trail, path, cycle, and/or circuit?

A, B, C, A

a closed walk, circuit, path, cycle

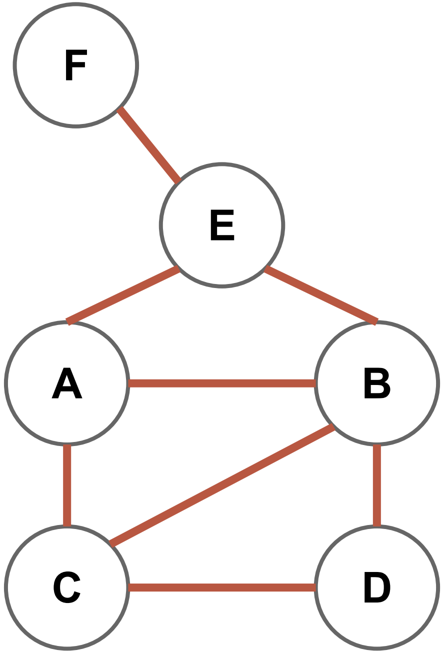
(not a trail, because it is closed)

Example



What's the longest path in this graph?

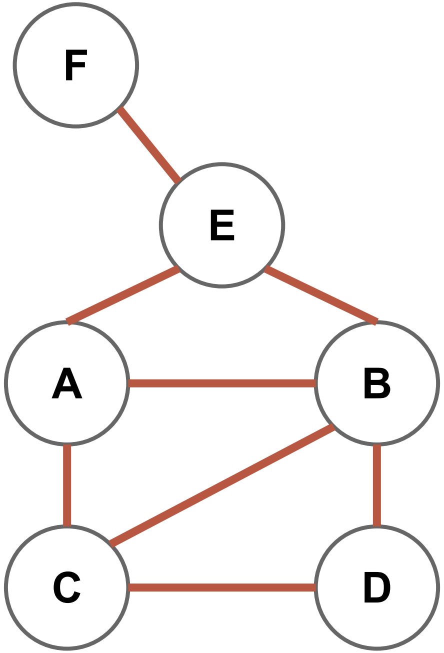
Example



What's the longest path in this graph?

F, E, A, C, B, D (others possible too)

Example

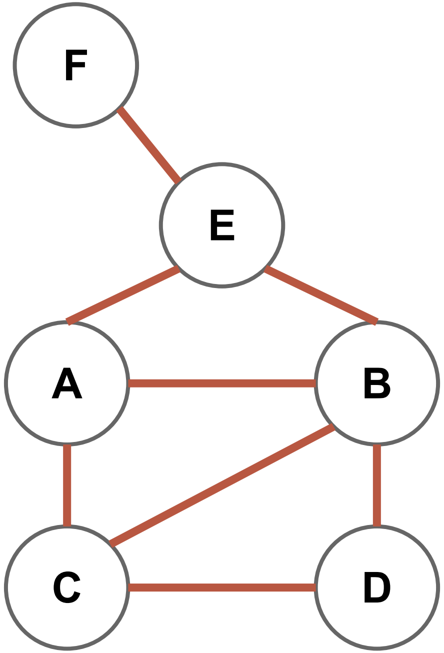


What's the longest path in this graph?

F, E, A, C, B, D (others possible too)

What's the largest cycle?

Example



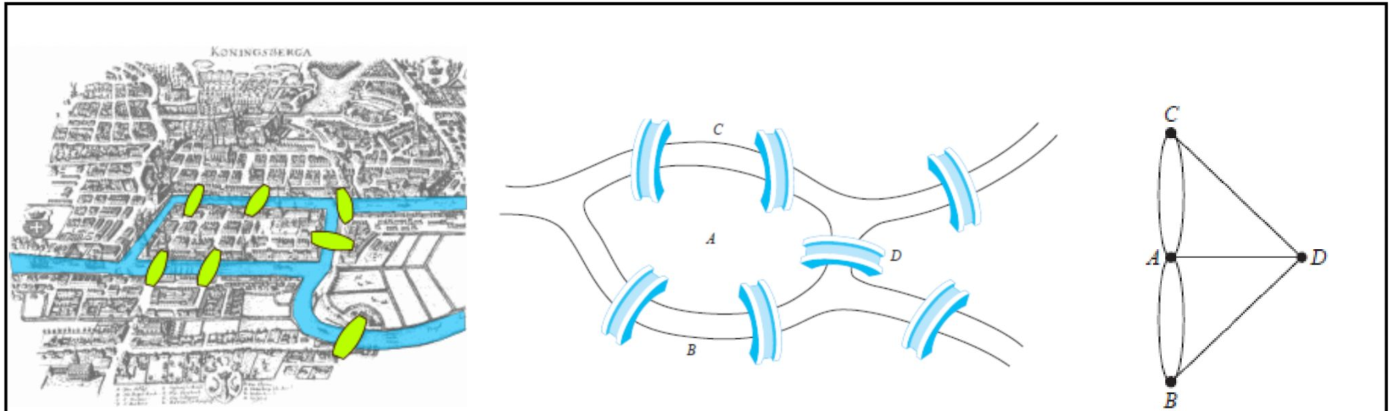
What's the longest path in this graph?

F, E, A, C, B, D (others possible too)

What's the largest cycle?

E, B, D, C, A, E (others possible too)

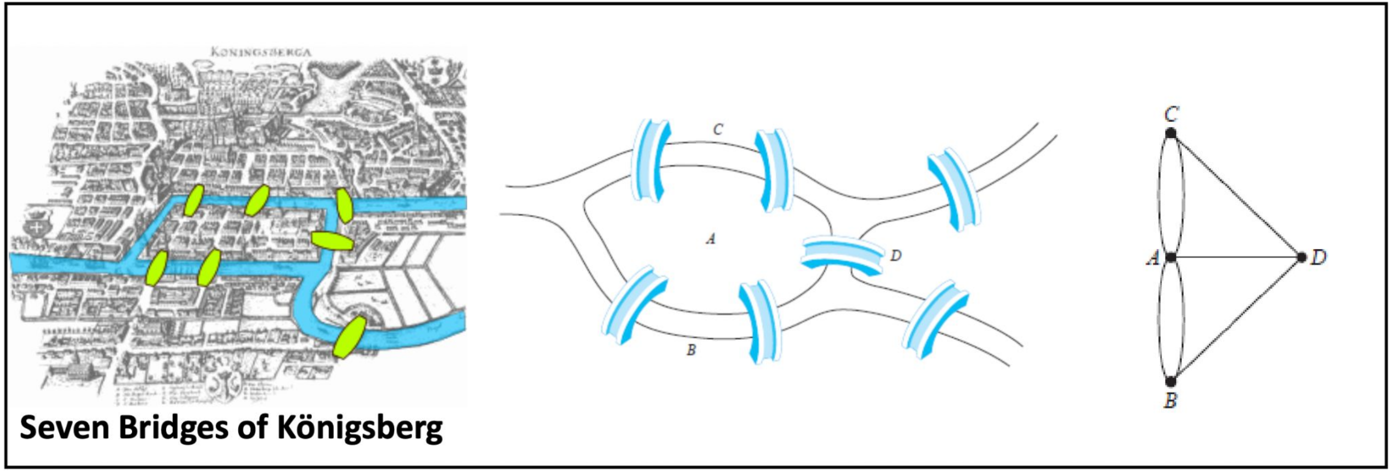
Euler Trail and Circuit



Seven Bridges of Königsberg

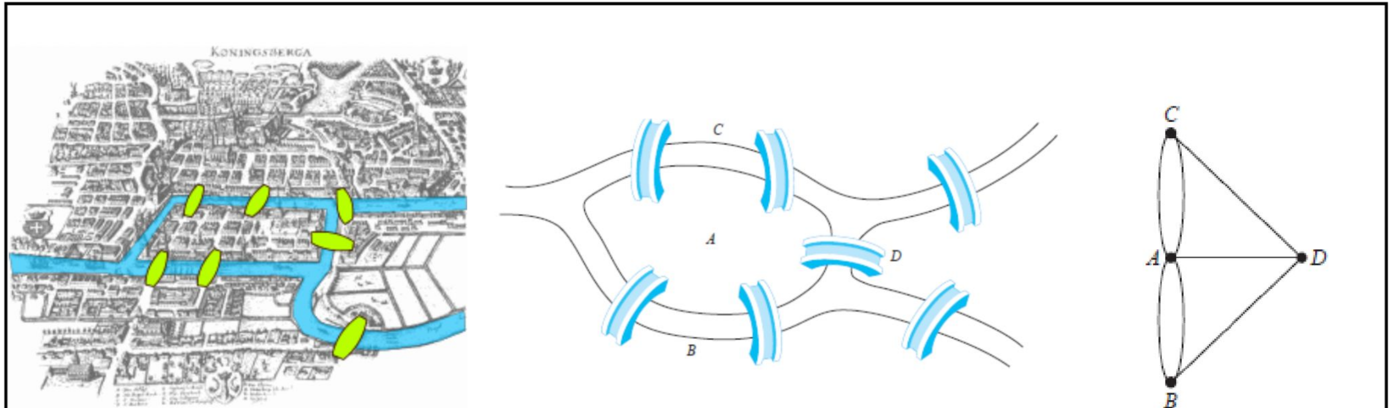
Can we cross every bridge exactly once?

Euler Trail and Circuit



Can we cross every bridge exactly once? **NO!**

Euler Trail and Circuit



Seven Bridges of Königsberg

Leonhard Euler

Can we cross every bridge exactly once? **NO!**

Solved by Swiss mathematician Leonhard Euler in 1736 (laid foundation of Graph Theory!)

Euler Trail and Circuit

An **Euler circuit** in a graph ***G*** is a simple circuit containing every edge of ***G***

An **Euler trail** in a graph ***G*** is a trail that visits every edge of ***G*** exactly once

Euler Trail and Circuit

An **Euler circuit** in a graph ***G*** is a simple circuit containing every edge of ***G***

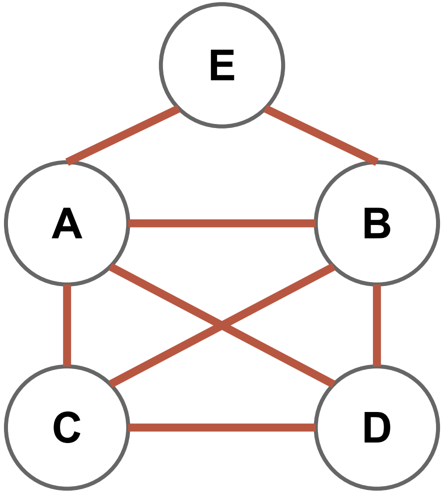
An **Euler trail** in a graph ***G*** is a trail that visits every edge of ***G*** exactly once

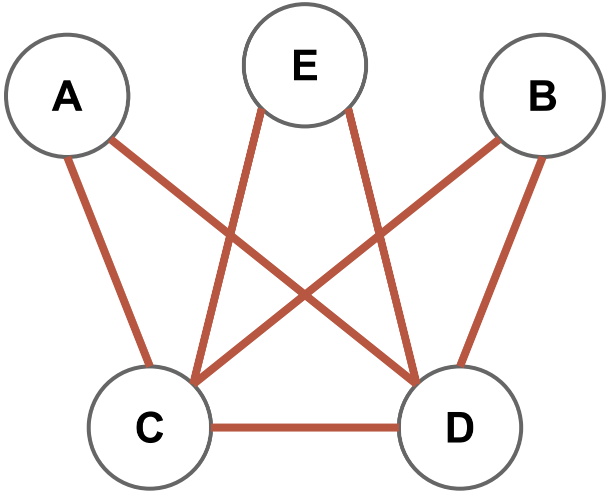
Theorem

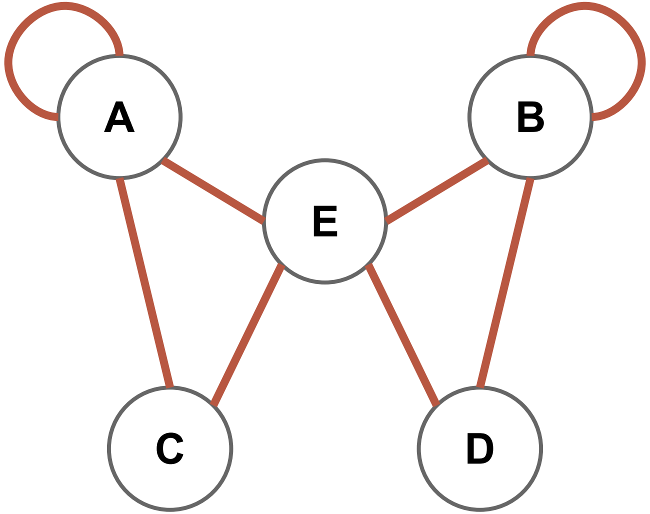
A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has an even degree

Euler Trail and Circuit

Find an Euler circuit in each graph, or state why there isn't one

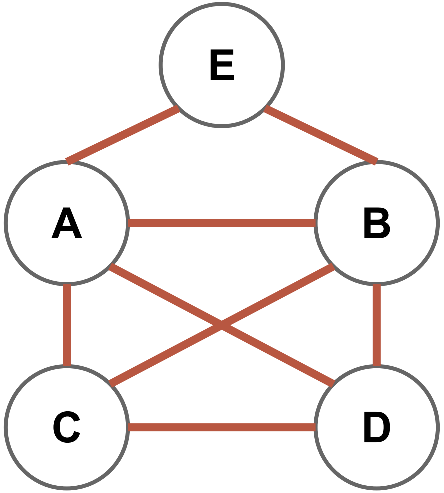




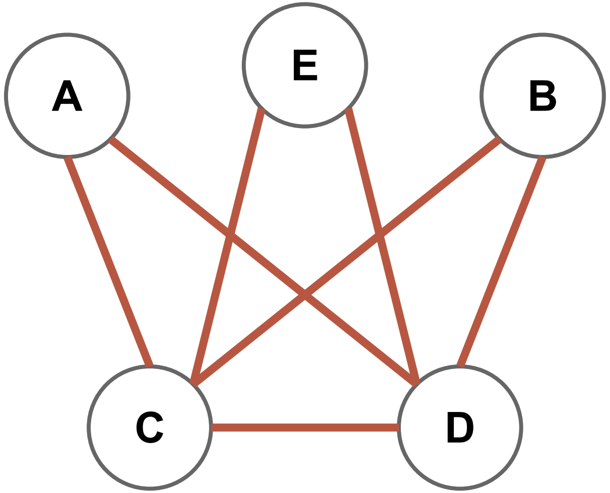


Euler Trail and Circuit

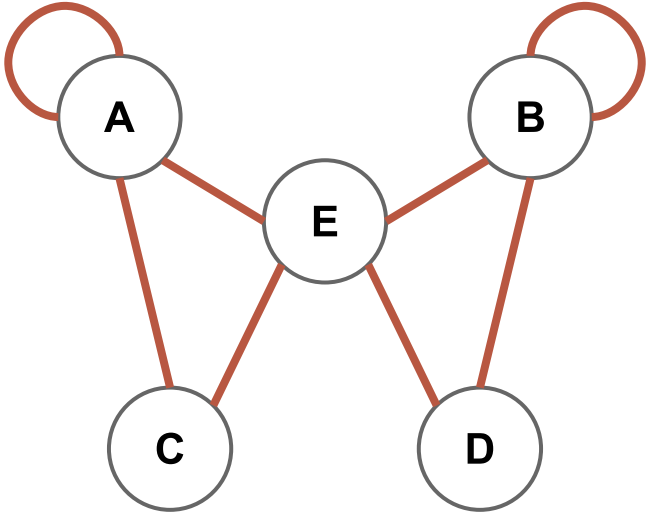
Find an Euler circuit in each graph, or state why there isn't one



None (deg(C), deg(D) = 3



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

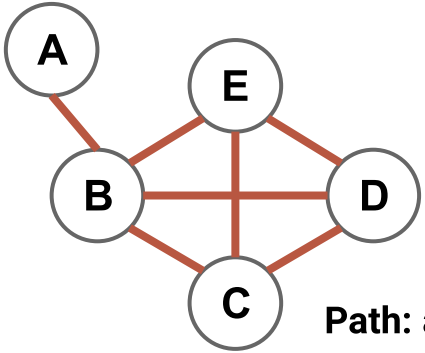


A,A,C,E,D,B,B,E,A

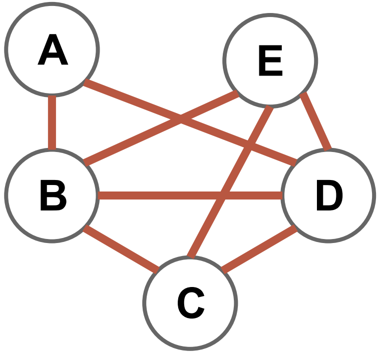
Hamiltonian Path and Cycle

A path in a graph ***G*** that passes through every vertex exactly once is called a **Hamiltonian path**.

A cycle in a graph ***G*** that passes through every vertex exactly once is called a **Hamiltonian cycle**.



Path: a,b,c,d,e



Cycle: a,b,c,e,d,a
Path: a,b,c,d,e

Connectivity

A node ***s*** is **<u>connected</u>** to ***t*** if there is a path from ***s*** to ***t***

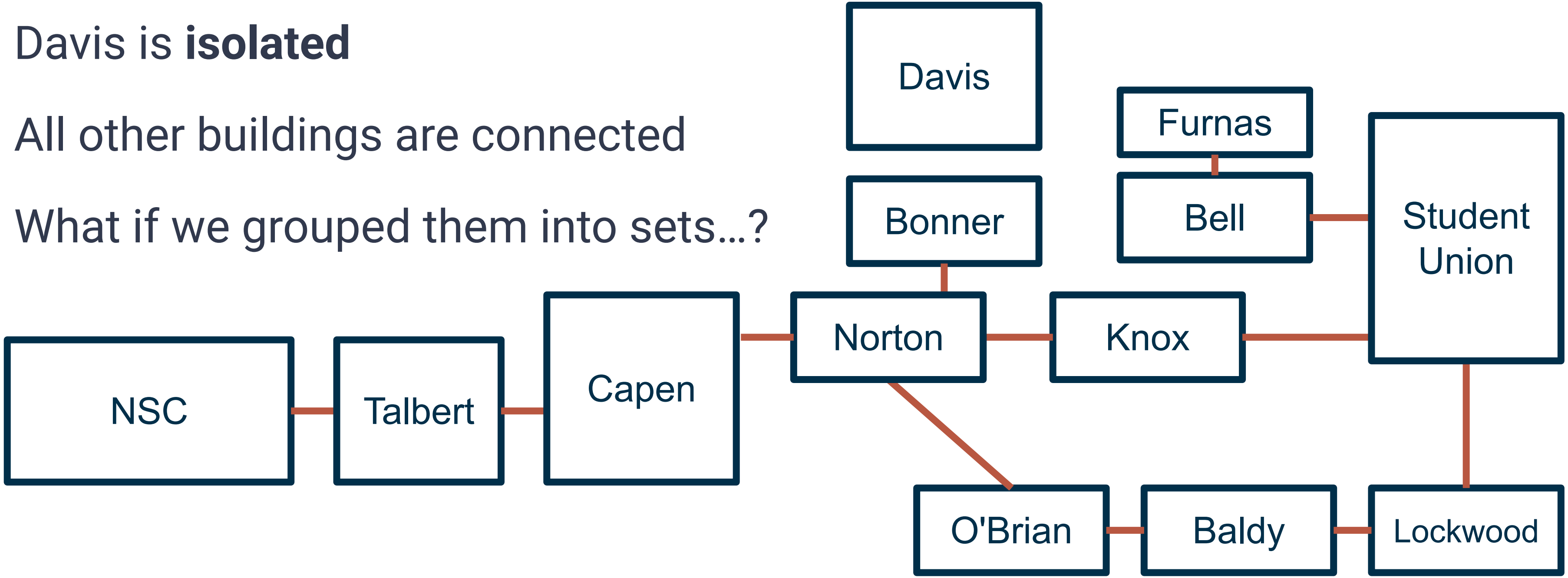
A node ***s*** is **<u>isolated</u>** if there is no other vertex connected to ***s***

Connectivity

Davis is **isolated**

All other buildings are connected

What if we grouped them into sets…?



Connected Components

A set of vertices in a graph is **connected** if every pair of vertices in the set is connected

A graph is said to be **connected** if the entire set of vertices is connected

A graph that is not connected is **disconnected**

A **connected component** is a ***maximal*** set of connected vertices

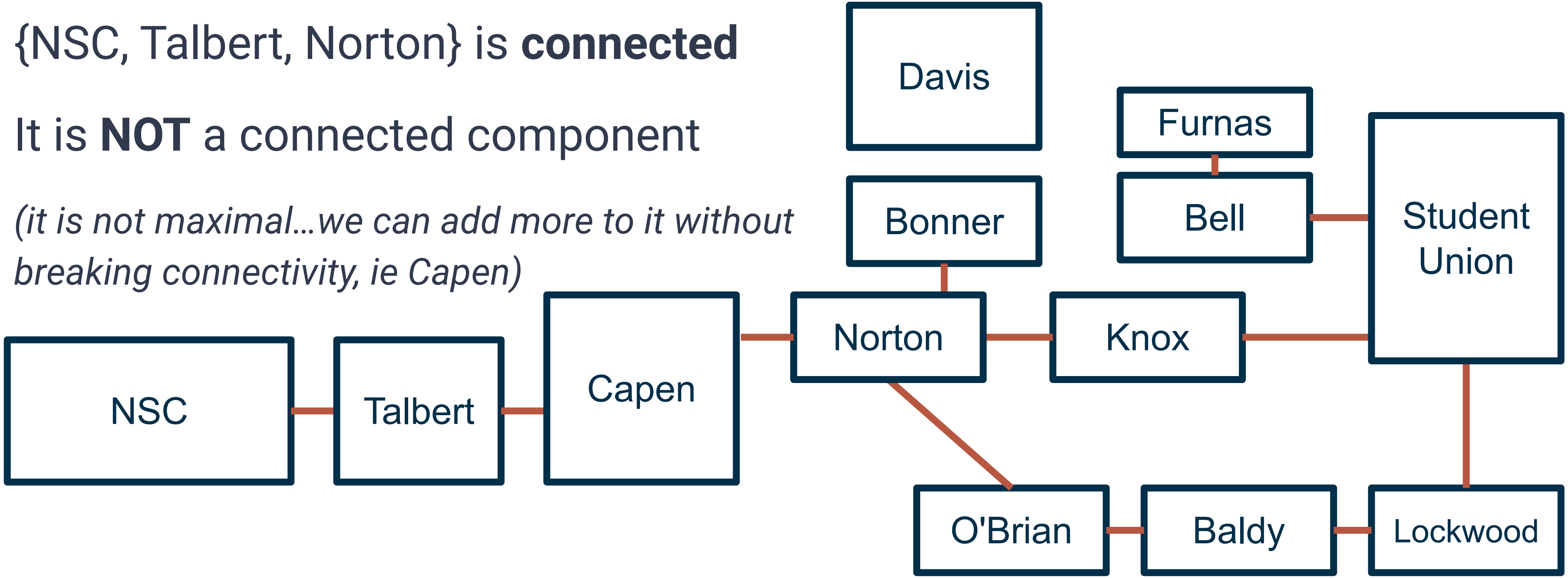
Note: *A disconnected graph can be split into more than one connected component*

Connected Components

{NSC, Talbert, Norton} is **connected**

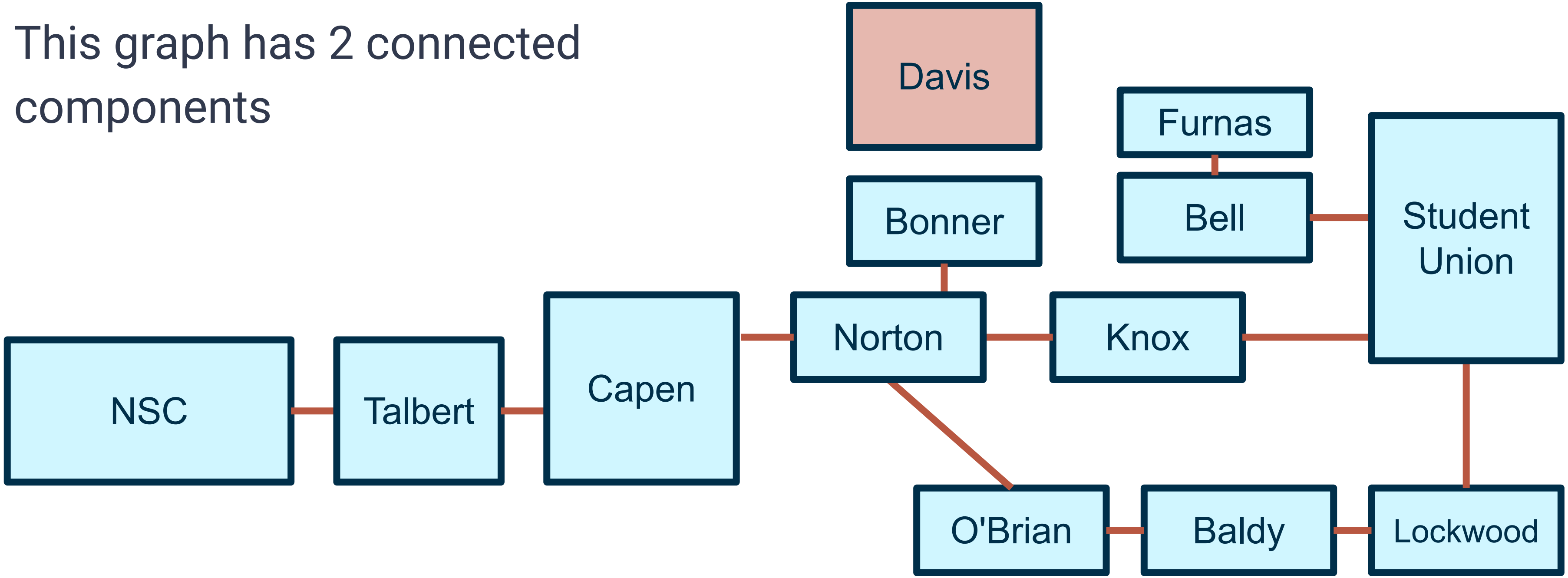
It is **NOT** a connected component

(it is not maximal…we can add more to it without breaking connectivity, ie Capen)



Connected Components

This graph has 2 connected components



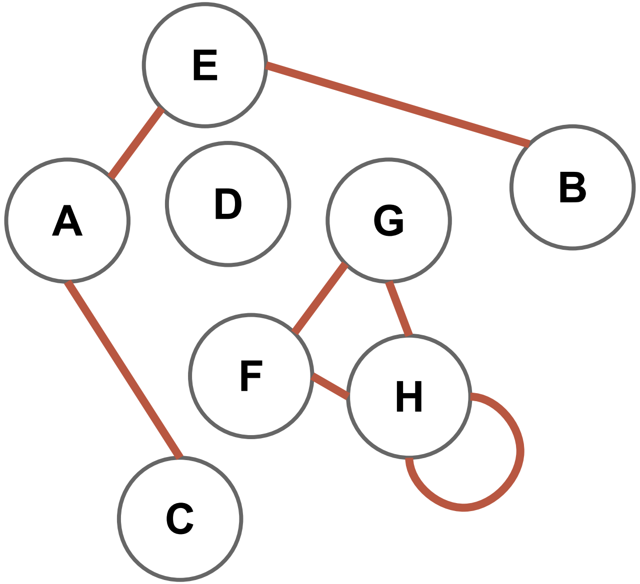
Connected Components

How many connected components does this graph have?

Is {A,B,C} connected?

Is it a connected component?

Is {D} a connected component?



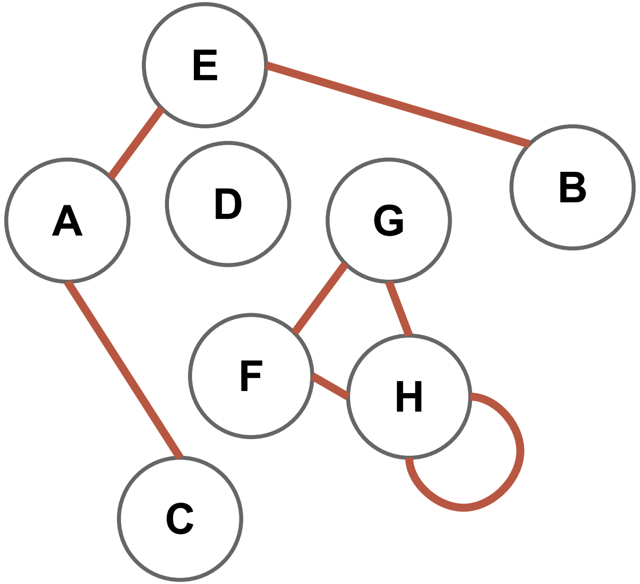
Connected Components

How many connected components does this graph have? **3**

Is {A,B,C} connected? **Yes**

Is it a connected component? **No**

Is {D} a connected component? **Yes**



Outline

- What are graphs?
- Graph Examples
- Graph Representation
- Handshake Theorem
- Graph Isomorphism
- Connectivity
- **Graph Coloring**
- Trees

Graph Coloring

Suppose we want to schedule final exams for the following seven courses such that students are not taking two exams at the same time:

CSE115, CSE116, CSE191, MTH141, CSE241, CSE305, CSE396

Graph Coloring

Suppose we want to schedule final exams for the following seven courses such that students are not taking two exams at the same time:

CSE115, CSE116, CSE191, MTH141, CSE241, CSE305, CSE396

The following pairs are pairs of courses with common students

(CSE115, CSE191), (CSE115, MTH141), (CSE115, CSE241), (CSE115, CSE305), (CSE116, CSE191), (CSE116, CSE241), (CSE116, CSE305), (CSE116, CSE396), (CSE191, MTH141), (CSE191, CSE241), (CSE241, MTH141), (CSE241, CSE396), (CSE305, MTH141), (CSE305, CSE396), (CSE396, MTH141)

Graph Coloring

Suppose we want to schedule final exams for the following seven courses such that students are not taking two exams at the same time:

CSE115, CSE116, CSE191, MTH141, CSE241, CSE305, CSE396

The follow n students

How can we relate this to graphs?

(CSE115, CSE191), (CSE115, MTH141), (CSE115, CSE241), (CSE115, CSE305), (CSE116, CSE191), (CSE116, CSE241), (CSE116, CSE305), (CSE116, CSE396), (CSE191, MTH141), (CSE191, CSE241), (CSE241, MTH141), (CSE241, CSE396), (CSE305, MTH141), (CSE305, CSE396), (CSE396, MTH141)

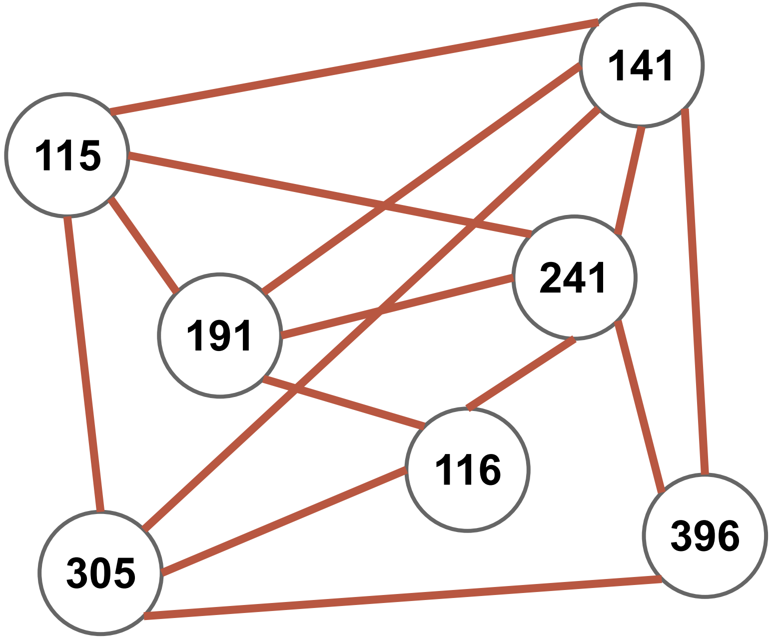
Graph Coloring

Let the courses be the vertices, and the pairs be edges

Two course are adjacent if they have common students

Color the vertices so that no two adjacent vertices have the same color

Colors = Time Slots



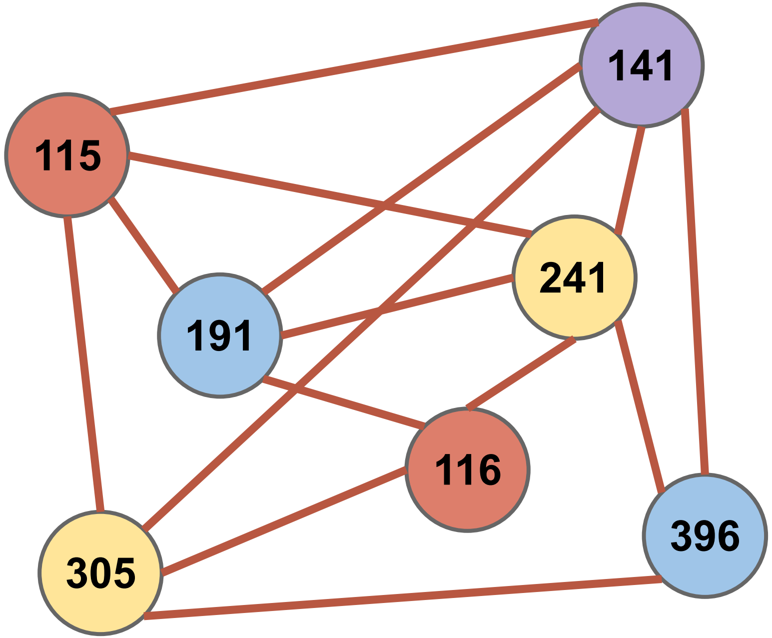
Graph Coloring

Let the courses be the vertices, and the pairs be edges

Two course are adjacent if they have common students

Color the vertices so that no two adjacent vertices have the same color

Colors = Time Slots



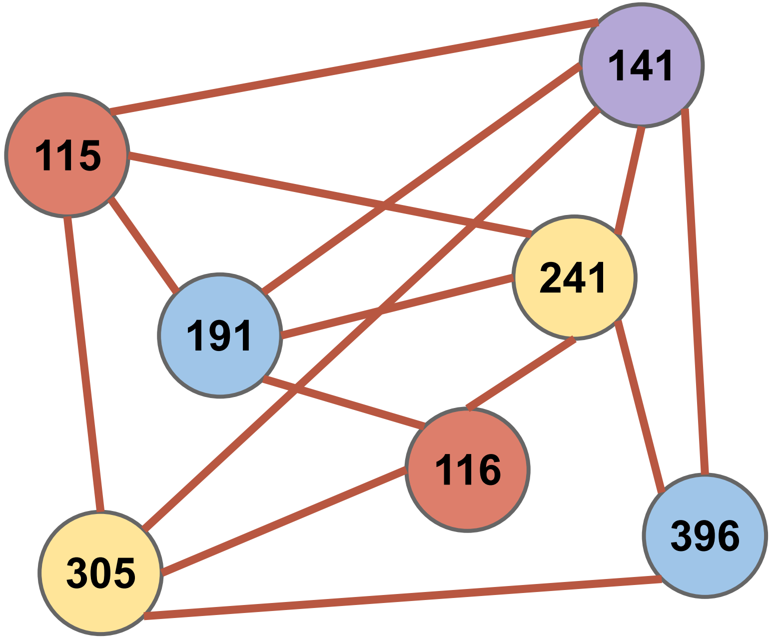
Graph Coloring

Let the courses be the vertices, and the pairs be edges

Thus a possible scheduling is:

1. CSE115, CSE116
2. CSE191, CSE396
3. CSE305, CSE241
4. MTH141

Can we have fewer time slots (colors)?

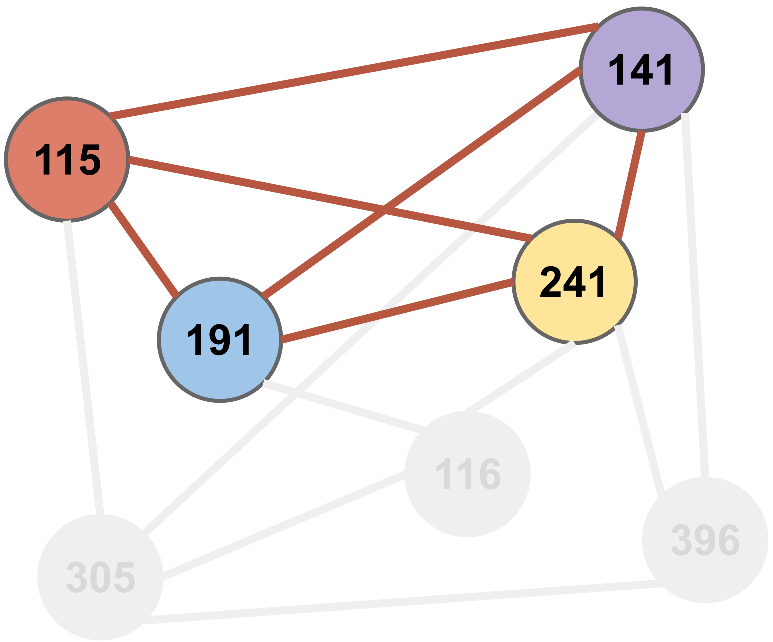


Graph Coloring

Let the courses be the vertices, and the pairs be edges

Thus a possible scheduling is:

115, 141, 191, and 241 are all adjacent to each other, so no we cannot color this graph with less than 4 colors

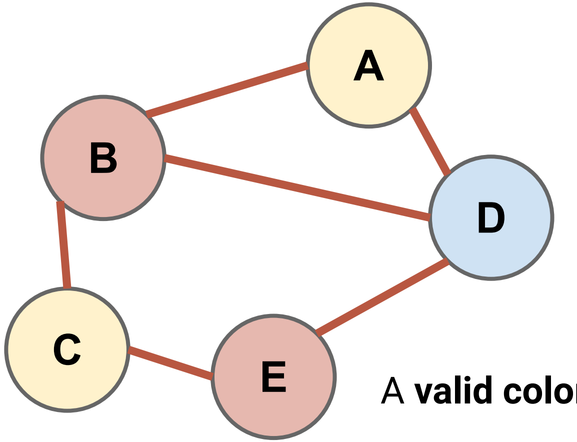


Can we have fewer time slots (colors)?

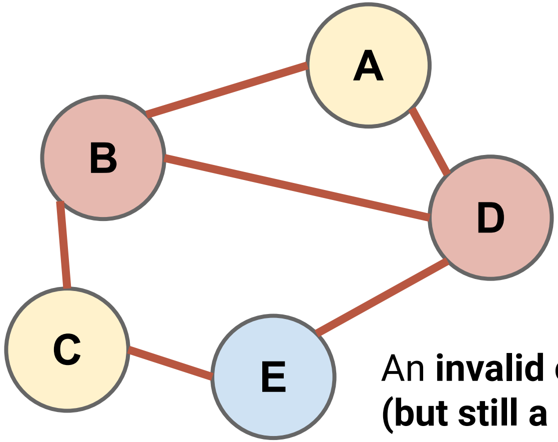
Graph Coloring

Let $G = (V, E)$ be an undirected graph and C be a finite set of colors. A **valid coloring** of G is a function $c: V \to C$ such that for $\forall \{x, y\} \in E, c(x) \neq c(y)$

- If $|C| = k$, then c is a ***k*-coloring** of G



A **valid coloring**



An **invalid coloring**
(but still a coloring)

Graph Coloring

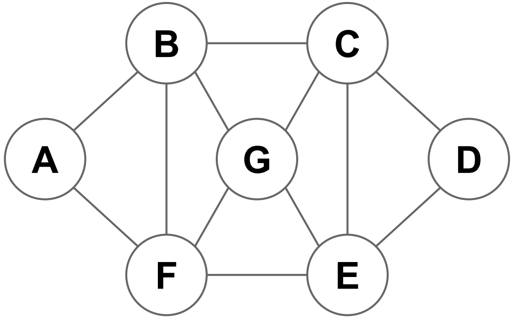
The **chromatic number** of a graph $\boldsymbol{G}$ (denoted $\boldsymbol{\chi(G)}$) is the smallest $\boldsymbol{k}$ such that there is a valid $\boldsymbol{k}$**-coloring** of $\boldsymbol{G}$

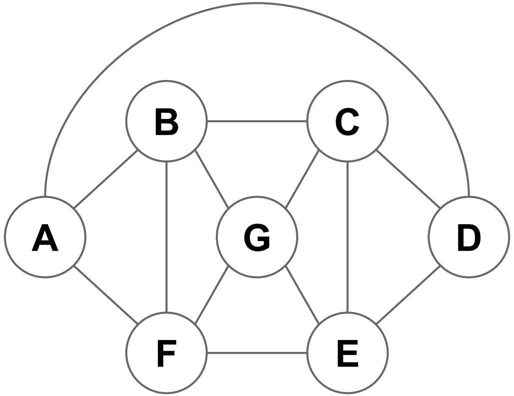
Theorem

Let $\boldsymbol{G}$ be an undirected graph. Let $\boldsymbol{\Delta(G)}$ be the maximum degree of any vertex in $\boldsymbol{G}$. Then $\boldsymbol{\chi(G) \leq \Delta(G) + 1}$

Graph Coloring

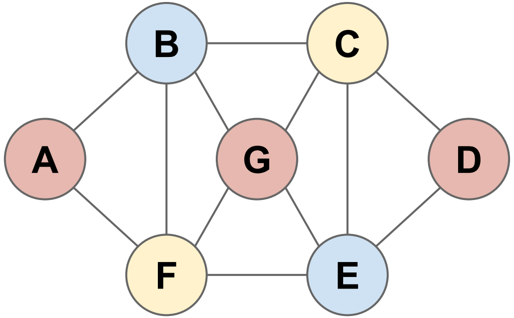
What are the chromatic numbers of these graphs?



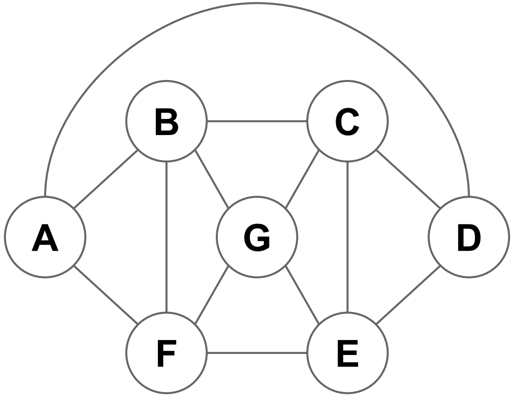


Graph Coloring

What are the chromatic numbers of these graphs?

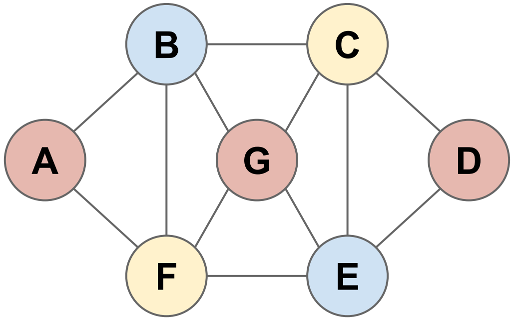


$\chi(G) = 3$

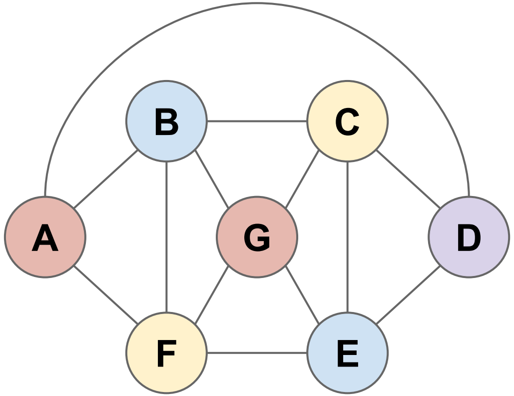


Graph Coloring

What are the chromatic numbers of these graphs?



$\chi(G) = 3$



$\chi(G) = 4$

Graph Coloring

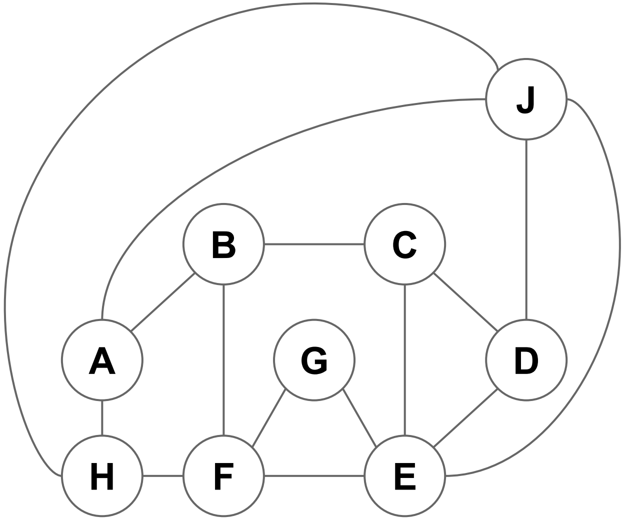
Greedy Coloring Algorithm

1. Number the set of possible colors (assume there is a large supply of possible colors, even if not all get used)
2. Consider each vertex ***v*** (in arbitrary order)
   a. Assign ***v*** a color that is different from the colors of ***v***'s. Use the lowest numbered color possible.

Greedy Graph Coloring Example

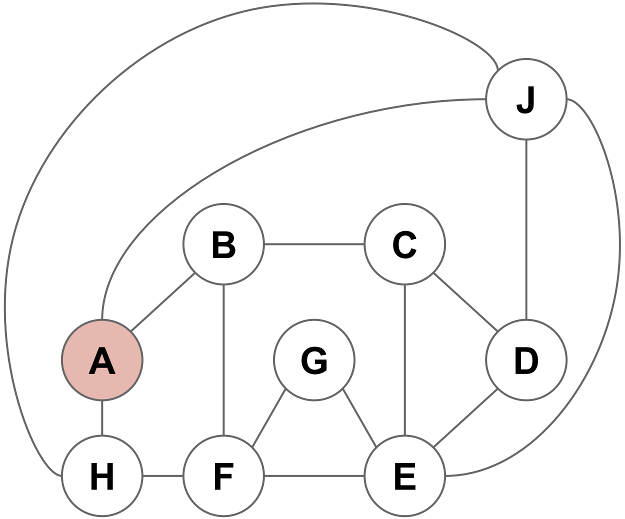
1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

Vertex Order: A C E B D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

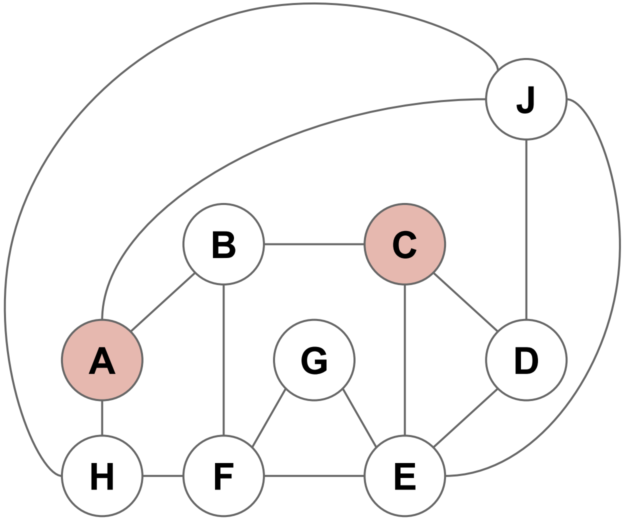


Vertex Order: <u>A</u> C E B D J H F G

Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

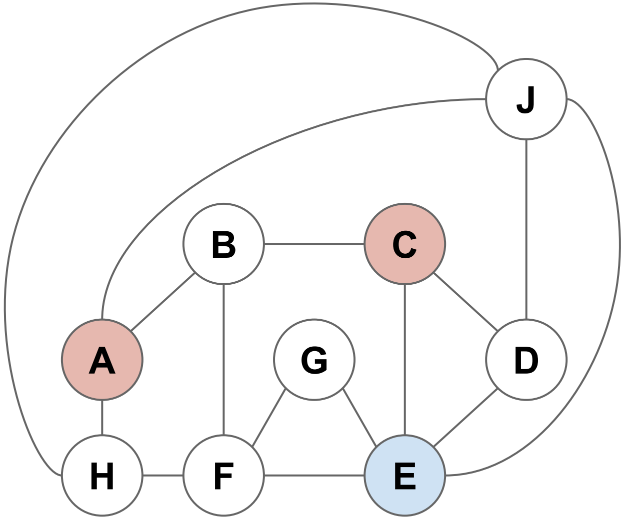
Vertex Order: A C E B D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

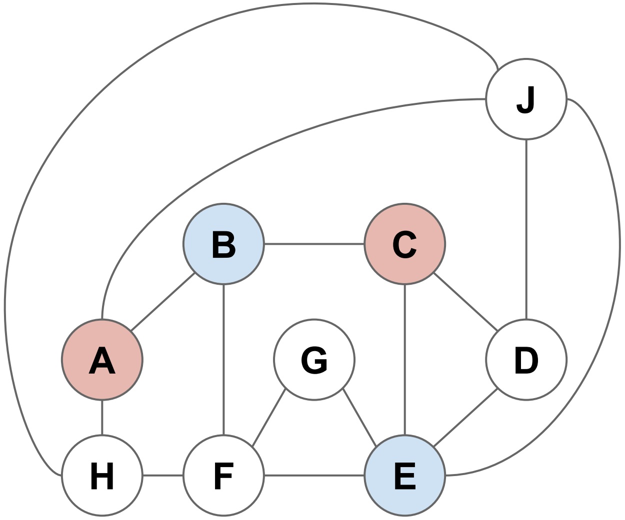
Vertex Order: A C E B D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

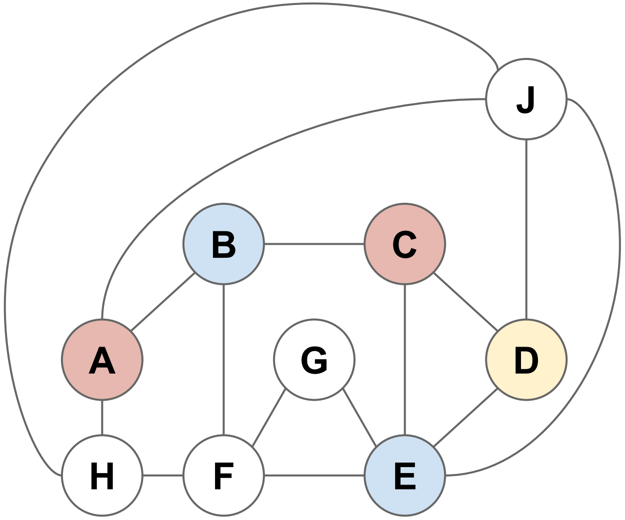
Vertex Order: A C E <u>B</u> D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

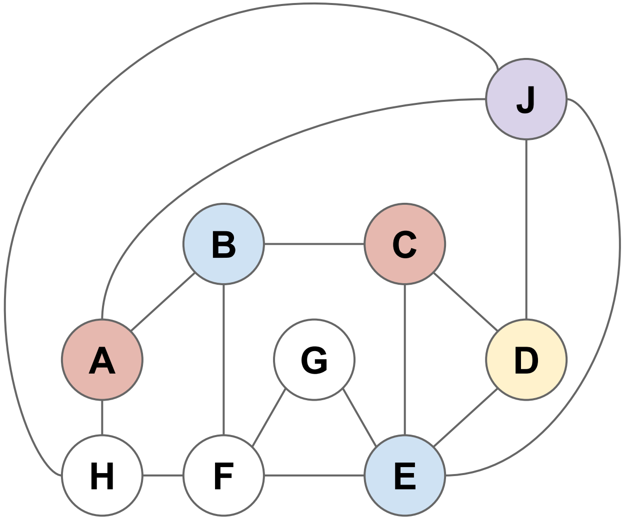
Vertex Order: A C E B <u>D</u> J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

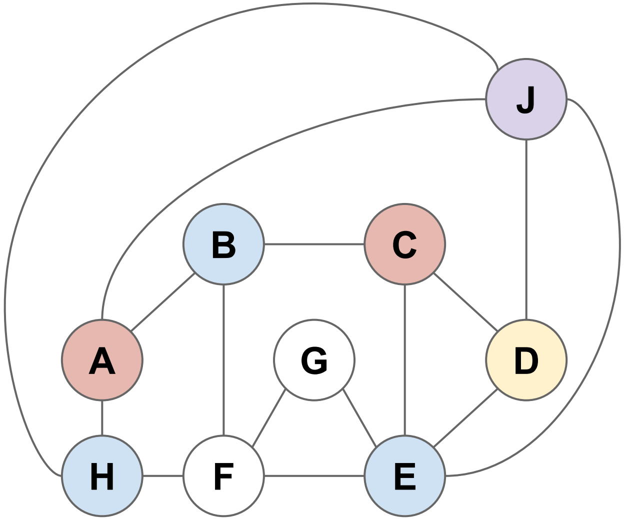
Vertex Order: A C E B D <u>J</u> H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

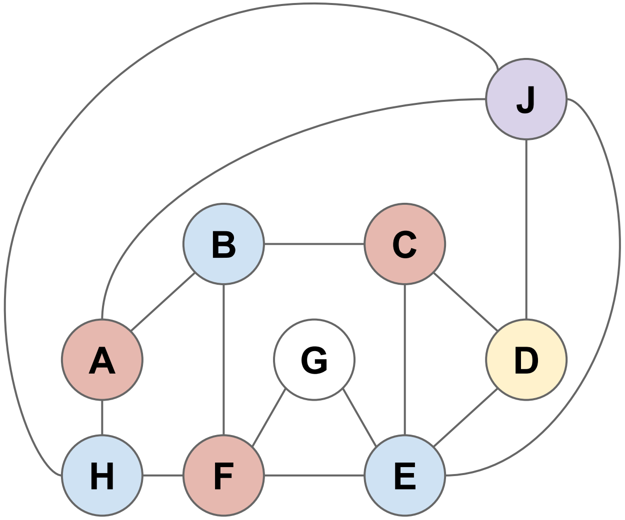
Vertex Order: A C E B D J <u>H</u> F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

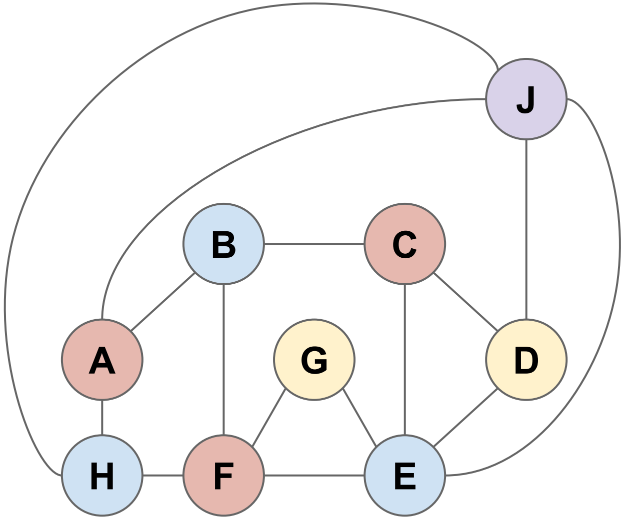
Vertex Order: A C E B D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

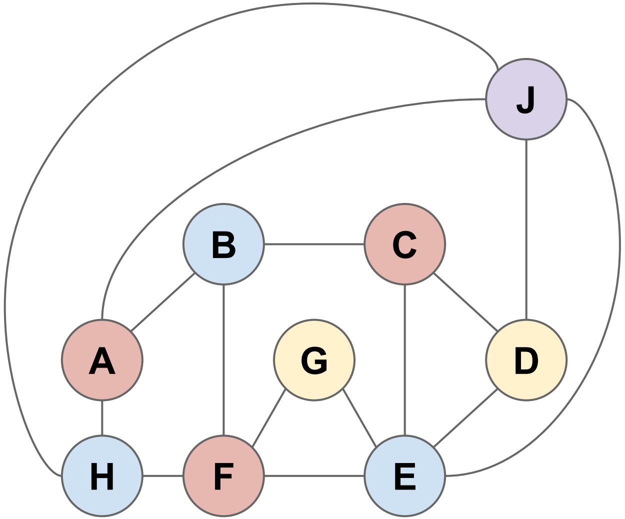
Vertex Order: A C E B D J H F G



Greedy Graph Coloring Example

1. Red
2. Blue
3. Yellow
4. Purple
5. Green
6. Orange
7. …

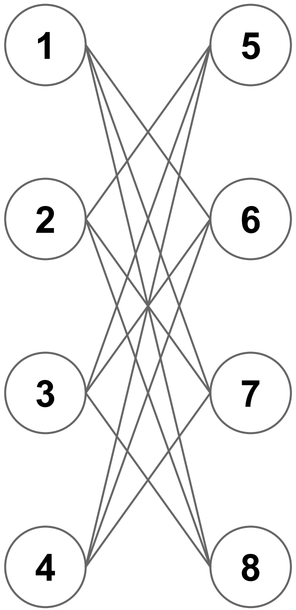
Does the greedy algorithm find the chromatic number?

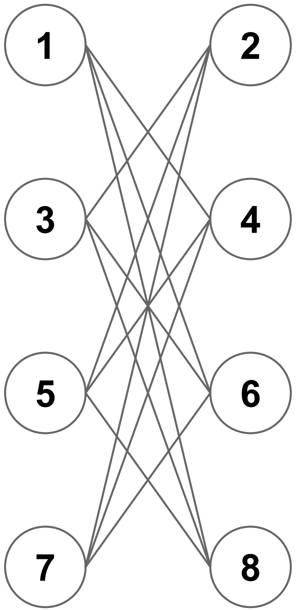


Vertex Order: A C E B D J H F G

Greedy Graph Coloring Exercise

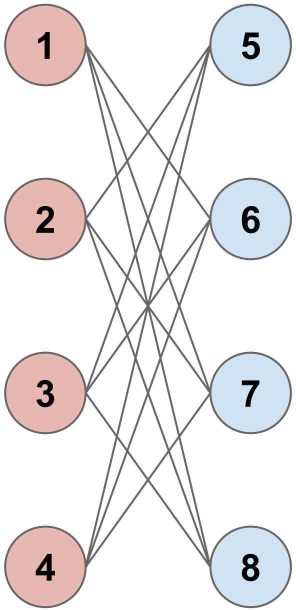
Use the greedy algorithm to color these two isomorphic graphs in ascending order





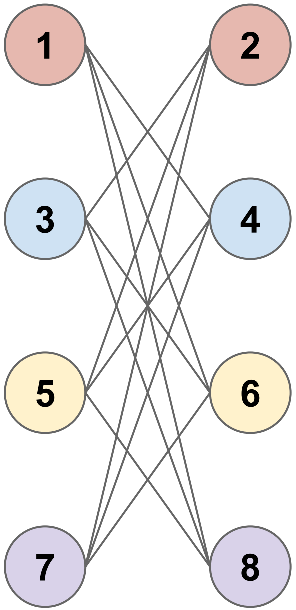
Greedy Graph Coloring Exercise

Use the greedy algorithm to color these two isomorphic graphs in ascending order



Does the greedy algorithm find the chromatic number?

No :(



Outline

- What are graphs?
- Graph Examples
- Graph Representation
- Handshake Theorem
- Graph Isomorphism
- Connectivity
- Graph Coloring
- **Trees**

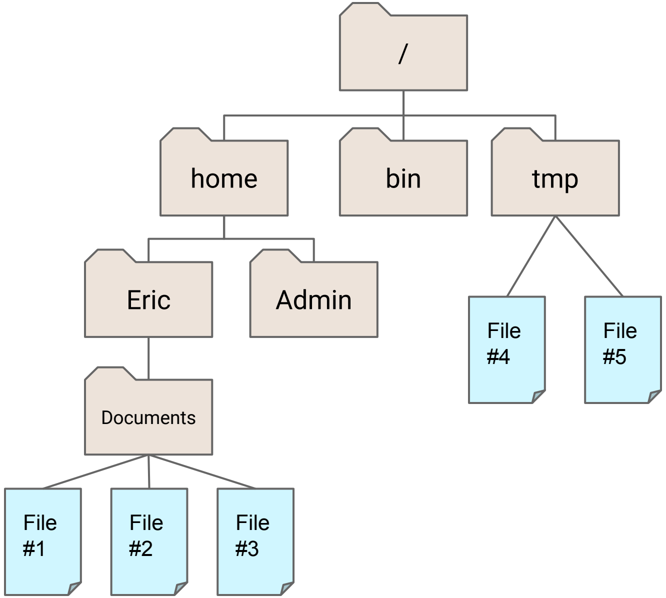
Trees

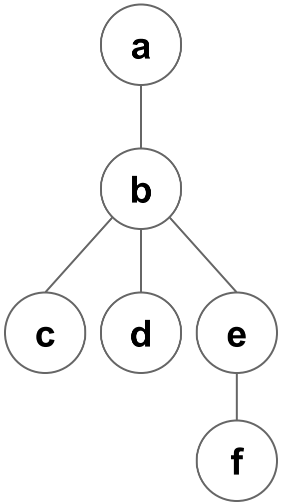
A **tree** is an undirected graph that is ***connected*** and has ***no cycles***

A **free tree** has no organization of vertices and edges

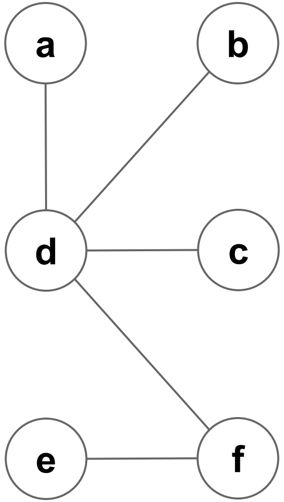
A **rooted tree** has a designated root at the top
(ie the "/" and "**a**" vertices →)

A free tree can be made rooted by choosing a root



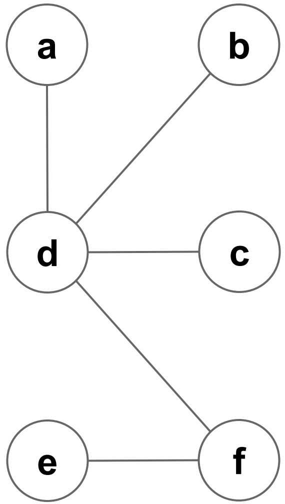


Tree Examples

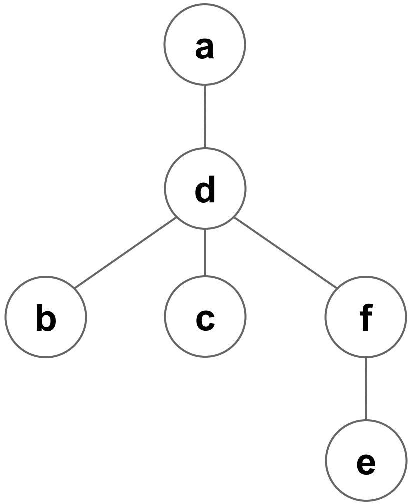


A free tree, T_1

Tree Examples

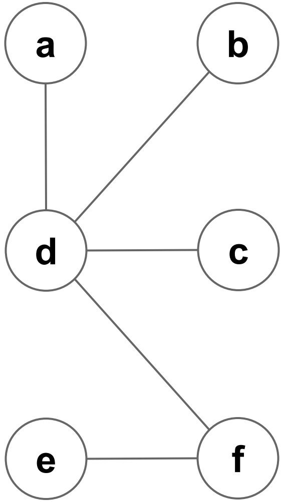


A free tree, $\boldsymbol{T_1}$

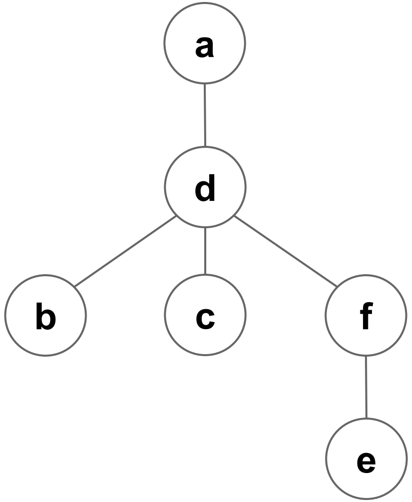


A rooted tree, $\boldsymbol{T_2}$, which is $\boldsymbol{T_1}$ rooted at ***a***

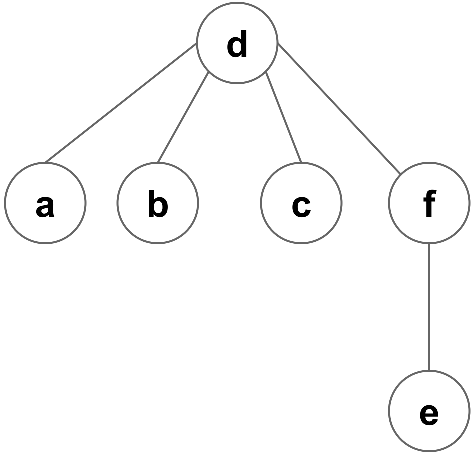
Tree Examples



A free tree, $\boldsymbol{T_1}$



A rooted tree, $\boldsymbol{T_2}$, which is $\boldsymbol{T_1}$ rooted at ***a***



A rooted tree, $\boldsymbol{T_3}$, which is $\boldsymbol{T_1}$ rooted at ***d***

Rooted Tree Terminology

Depth of a vertex: Distance from the root to that vertex

Height of a tree: maximum depth of any vertex

Parent of a vertex: the node above that vertex (towards the root)

- When ***u*** is the parent of ***v***, then ***v*** is a **child** of ***u***
- Vertices with the same parent are called **siblings**

Leaf: a vertex with no children

Rooted Tree Terminology

The **ancestors** of a vertex (other than the root) are the vertices in the path from the root to the vertex (including the root, but excluding the vertex). The root has no ancestors.

The **descendants** of a vertex, ***v***, are all vertices that have ***v*** as an ancestor

If ***a*** is a vertex in a tree, the **subtree rooted at *a*** is the subgraph of the tree consisting of ***a***, all of its descendants, and all edges incident to those descendants

Rooted Tree Examples

For T_2 and T_3:

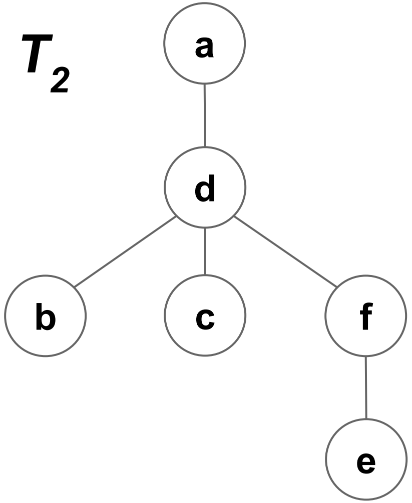
Height?

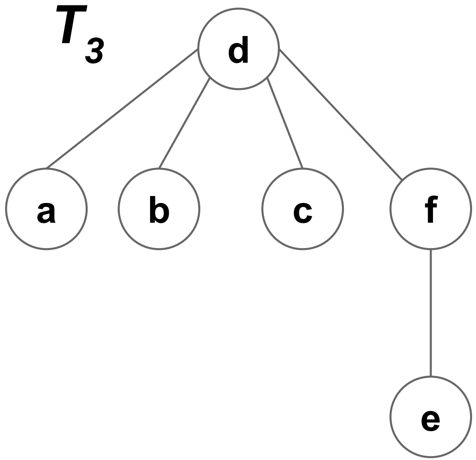
*Ancestors of **f**?*

*Descendants of **f**?*

*Siblings of **f**?*

Leaves?





Rooted Tree Examples

For T_2 and T_3:

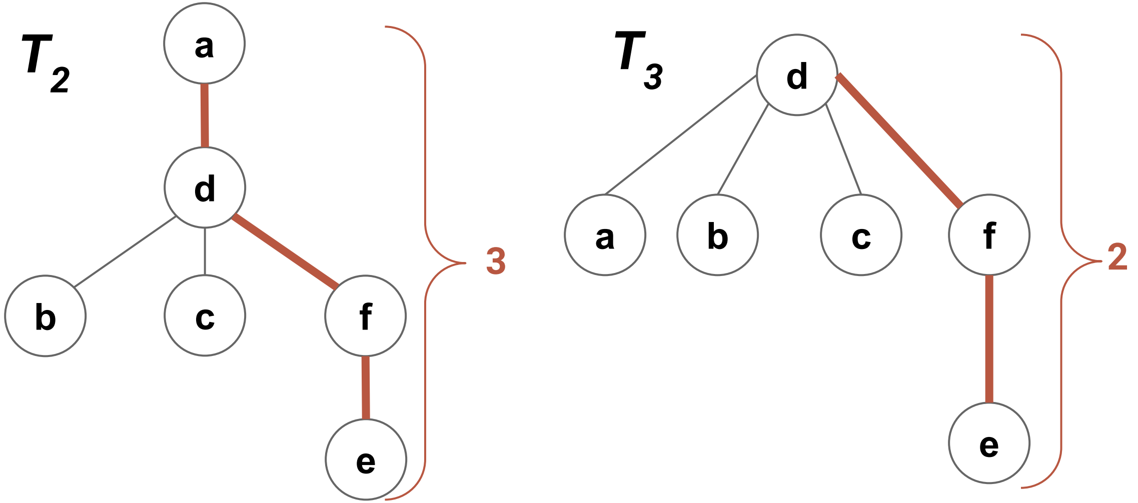
Height? **3 and 2**

*Ancestors of **f**?*

*Descendants of **f**?*

*Siblings of **f**?*

Leaves?



Rooted Tree Examples

For T_2 and T_3:

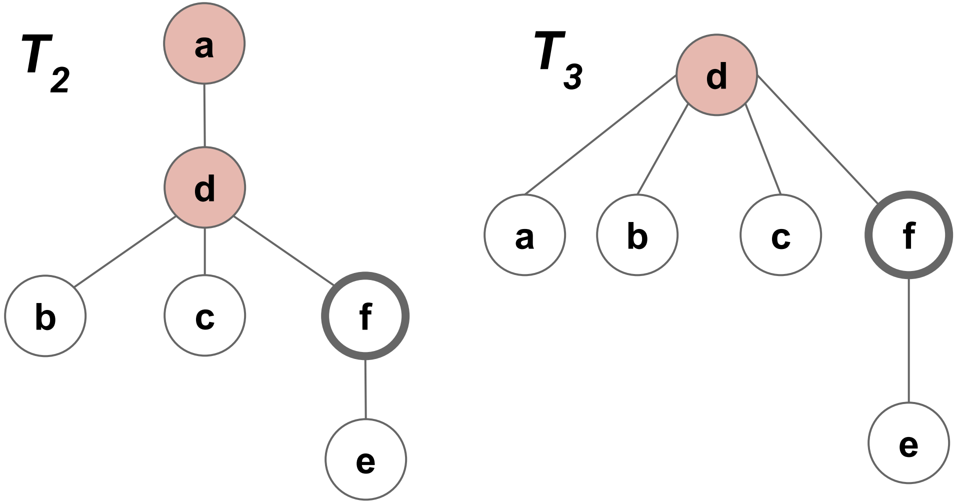
Height? **3 and 2**

*Ancestors of **f**?* **{*d*, *a*} and {*d*}**

*Descendants of **f**?*

*Siblings of **f**?*

Leaves?



Rooted Tree Examples

For T_2 and T_3:

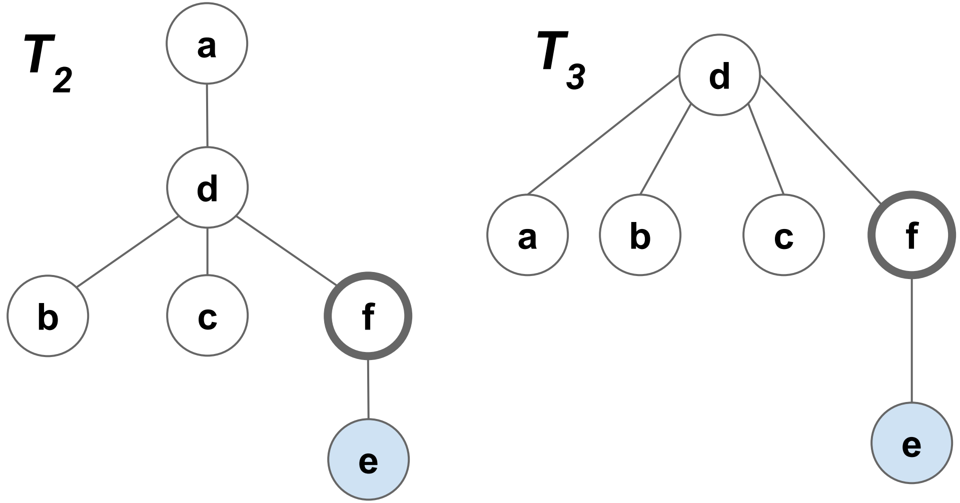
Height? **3 and 2**

*Ancestors of **f**?* **{*d, a*} and {*d*}**

*Descendants of **f**?* **{e} and {e}**

*Siblings of **f**?*

Leaves?



Rooted Tree Examples

For T_2 and T_3:

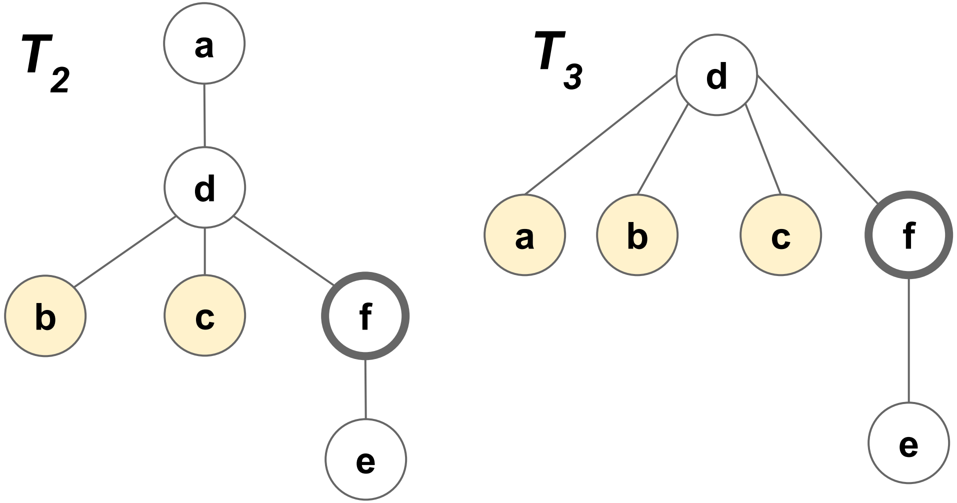
Height? **3 and 2**

Ancestors of ***f****?* **{*d, a*} and {*d*}**

Descendants of ***f****?* **{e} and {e}**

Siblings of ***f****?* **{*c, b*} and {*a,b,c*}**

Leaves?



Rooted Tree Examples

For T_2 and T_3:

Height? **3 and 2**

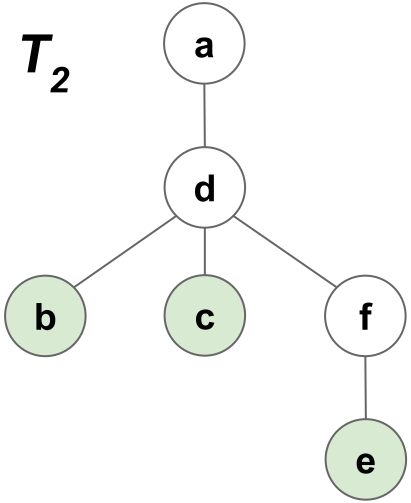
Ancestors of ***f****?* **{*d, a*} and {*d*}**

Descendants of ***f****?* **{e} and {e}**

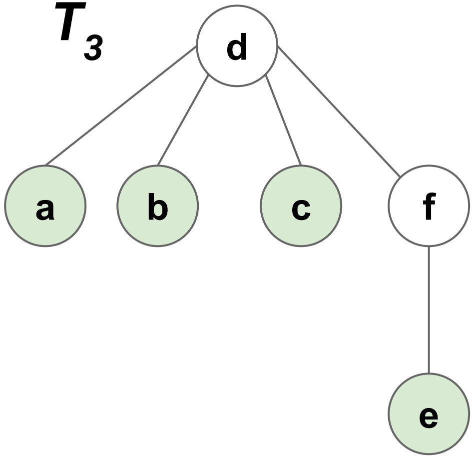
Siblings of ***f****?* **{*c, b*} and {*a,b,c*}**

Leaves? **{*b,c,e*} and {*a,b,c,e*}**

T_2



T_3



Tree Properties

Theorem

There is a unique path of vertices between every pair of nodes in a tree

Tree Properties

Theorem

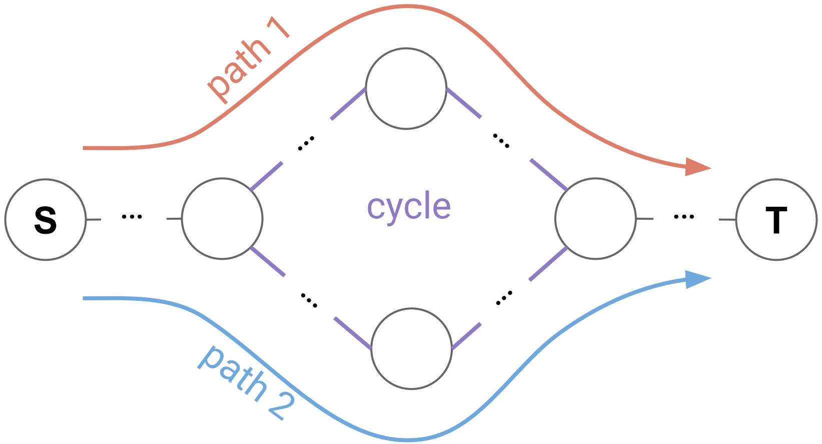
There is a unique path of vertices between every pair of nodes in a tree

Proof by contradiction

Assume there is a pair of nodes with two different paths between them

Because of this, there must be a cycle

Therefore the graph is not a tree (contradiction)



Tree Properties

Theorem

A tree with ***n*** nodes has ***n* - 1** edges

Tree Properties

<u>Theorem</u>

A tree with ***n*** nodes has ***n* - 1** edges

Proof by induction…

Proof by Induction

Property: ***P*(*n*)**: A tree with ***n*** nodes has ***n* - 1** edges

Base Case: ***P*(1)**: A tree with 1 node has 0 edges ✔

Inductive Case:

- Assume "***P*(*k*)**: A tree with ***k*** nodes has ***k*-1** edges" is true for ***k*** ≥ 1
- Want to prove "***P*(*k* + 1)**: A tree with ***k* + 1** nodes has ***k*** edges" is true
  - Let ***T*** be a tree with ***k* + 1** vertices, and let ***v*** be a leaf in ***T***
  - Remove ***v*** and the edge to its parent, we now have a tree with ***k*** vertices
  - By our inductive assumption, this means it has ***k* - 1** edges
  - Since we removed one edge then our original tree must have ***k*** edges ✔