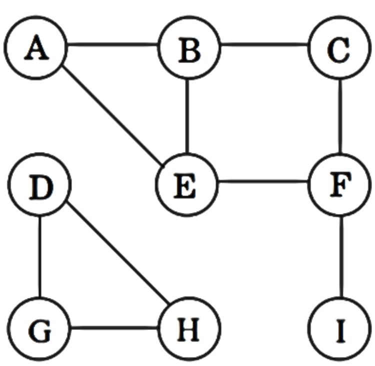
CSCI 2200 FOUNDATIONS OF COMPUTER SCIENCE

David Goldschmidt
goldsd3@rpi.edu
Fall 2022

UNDIRECTED GRAPHS



For given graph $Q = (V, E)$, what are V and E?

Set $V = \{ A, B, C, D, E, F, G, H, I \}$

Set E contains all undirected edges…

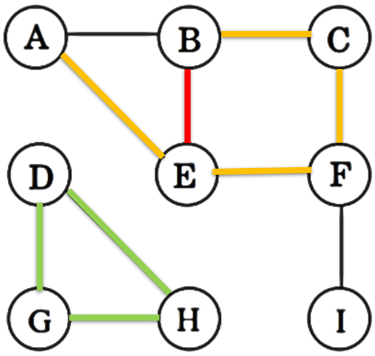
$E = \{ (A,B), (A,E), (B,C), (B,E), (C,F), (D,G), (D,H)\ (E,F), (F,I), (G,H) \}$

Therefore, $|V| = 9$ and $|E| = 10$

We disallow *self-loops*, e.g., edges (A,A) and (B,B), and *multi-graphs* with multiple edges between two vertices

Define *degree* ∂_q as the number of edges that are *incident on* (or *adjacent to*) some vertex q

PATHS, SIMPLE PATHS, AND CYCLES



A *path* is a sequence of vertices with a designated start and end vertex for which we have an edge between each pair of consecutive vertices…

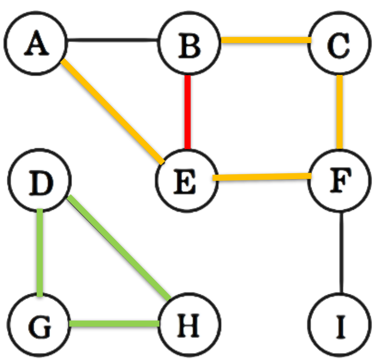
A *simple path* does not repeat vertices

e.g., *AEFCB* is a simple path of length 4 since we traverse 4 edges

e.g., *AEFCBE* is a path of length 5

e.g., *DHGD* is a *cycle* since we start and end on the same vertex — and we do not traverse any edge more than once

CONNECTIVITY AND ISOMORPHISM

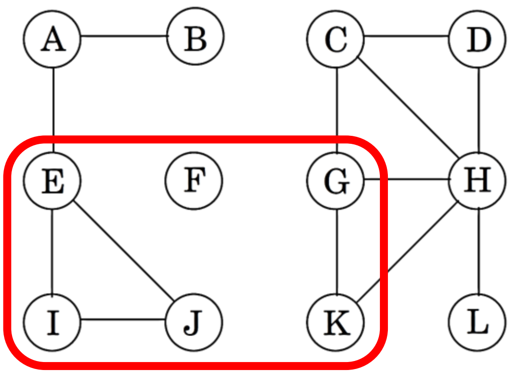


Vertices *u* and *v* are *connected* iff there is a path from vertex *u* to vertex *v*

A graph is *connected* iff every pair of vertices is connected

Two graphs are isomorphic iff both graphs have the same paths…

(INDUCED) SUBGRAPHS



Define *subgraph* $H = (V', E')$ of graph $G = (V, E)$…

…with $V' \subseteq V$ and $E' \subseteq E$ such that all edges of E' are guaranteed to have endpoints in V'

Define *induced subgraph* H' such that all edges of set E that connect vertices of V' are in set E'

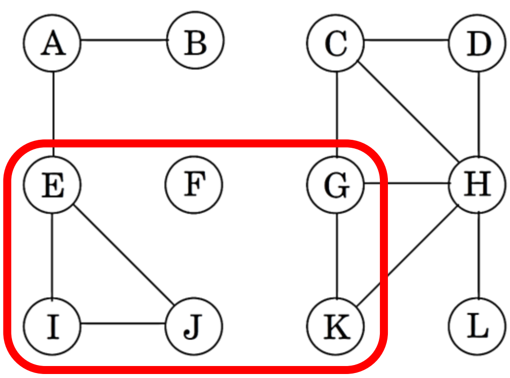
What is the induced subgraph shown in red…?

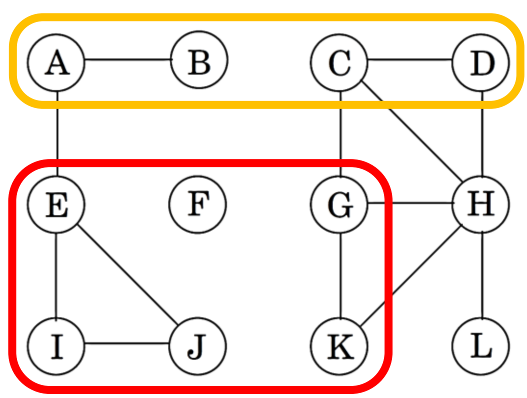
$V' = \{ E, F, G, I, J, K \}$ and

$E' = \{ (E,I), (E,J), (G,K), (I,J) \}$

What is a subgraph of G that is *disjoint* from H…?

(INDUCED) SUBGRAPHS & DISJOINT SUBGRAPHS





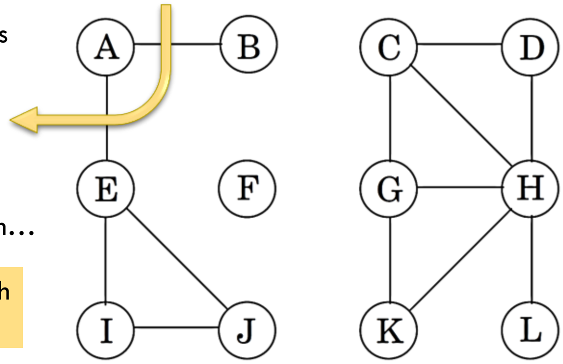
Two graphs are *disjoint* from one another if they do not share any common vertices (or edges)

DEGREE SEQUENCE

Degree sequence ∂ lists all vertex degrees of the graph from highest to lowest…

∂ = [5, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 0]

the degree of the graph itself is therefore 5…



On its own, a degree sequence ∂ is not guaranteed to uniquely describe a graph…

Can you draw two different graphs with ∂ = [2, 2, 2, 2, 2, 2]?

Can you draw two different graphs with ∂ = [3, 3, 2, 1, 1]?

How about ∂ = [3, 3, 3, 2, 1, 1]?

```
degree sequence is [2,2,2,2,2,2]

 O-----O    O                  O---O
 |    /    /|                 /     \
 |   /    / |                /       \
 |  /    /  |     and       O         O
 | /    /   |                \       /
 |/    /    |                 \     /
 O    O-----O                  O---O

                          (these two are isomorphic)

                            O----O----O
                            |         |
                            |         |
                            O----O----O
```

```
degree sequence is [3,3,2,1,1]

            O
           / \
          /   \
         /     \
        O-------O          (no other possible graph!)
       /         \
      /           \
     /             \
    O               O
```

```
degree sequence is [3,3,2,1,1]

       O
      / \
     /   \
    O-------O
   /         \
  /           \
 O             O
```

```
Given: degree sequence is [3,3,2,1,1]
  assign these as vertices A,B,C,D,E

If A is not connected to B, then
 A must be connected to C, D, and E...

...but if A is not connected to B, then
 B must be connected to C, D, and E

That cannot be since degree(D)=degree(E)=1

So, A must be connected to B


     A-------B
```

```
...and A must be connected to two of C, D, E

...and B must be connected to two of C, D, E

So, A and B must both be connected to C

          C
         / \
        /   \
       A-------B
```

```
Vertex D (or E) must be connected to A
 since degree(A)=3...

          C
         / \
        /   \
       A-------B
      /
     /
    D


That leaves E with only one possible
connection...

          C
         / \
        /   \
       A-------B
      /         \
     /           \
    D             E
```

DEGREE SEQUENCE – GRAPH PATTERNS

For each graph pattern, what is $|V|$ and $|E|$…?

K_n Complete graph or n-clique

$K_{n,\ell}$ Complete bipartite graph with n left and ℓ right vertices

L_n Line or path with n vertices

C_n Cycle with n vertices

S_{n+1} Star with a central vertex connected to n peripheral vertices, i.e., $K_{1,n}$

W_{n+1} Wheel — a cycle of n vertices with a central vertex

Complete, K_5

$[4, 4, 4, 4, 4]$

Bipartite, $K_{3,2}$

$[3, 3, 2, 2, 2]$

Line, L_5

$[2, 2, 2, 1, 1]$

Cycle, C_5

$[2, 2, 2, 2, 2]$

Star, S_6

$[5, 1, 1, 1, 1, 1]$

Wheel, W_6

$[5, 3, 3, 3, 3, 3]$

DEGREE SEQUENCE – GRAPH PATTERNS

For each graph pattern, what is $|V|$ and $|E|$…?

K_n Complete graph or n-clique

$K_{n,\ell}$ Complete bipartite graph with n left and ℓ right vertices

Complete, K_5

$[4, 4, 4, 4, 4]$

Bipartite, $K_{3,2}$

$[3, 3, 2, 2, 2]$

Line, L_5

$[2, 2, 2, 1, 1]$

```
Complete graph K_n
  |V| = n
  |E| = (n-1) + (n-2) + ... + 1
         n(n-1)
      = --------
           2
```

```
Complete Bipartite graph K_{n,l}
    |E| = (n)(l)
```

$[2, 2, 2, 2, 2]$ $[5, 1, 1, 1, 1, 1]$ $[5, 3, 3, 3, 3, 3]$

REPRESENTING A GRAPH

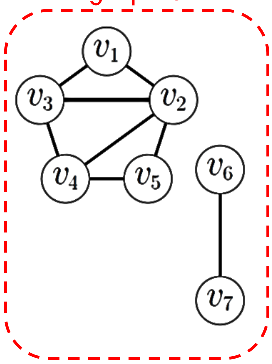
How would these representation schemes change for a directed graph?

For computing on graphs, we need convenient and efficient graph representations

Adjacency List

v_1: v_2, v_3
v_2: v_1, v_3, v_4, v_5
v_3: v_1, v_2, v_4
v_4: v_2, v_3, v_5
v_5: v_2, v_4
v_6: v_7
v_7: v_6

graph G



Adjacency Matrix

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

DEGREE SEQUENCE – HANDSHAKING THEOREM

Can you draw two different graphs with ∂ = [3, 3, 3, 2, 1, 1]?

We cannot construct such a graph because when we add an edge, it has two endpoint vertices and therefore increases the sum of degrees by two…

Prove this theorem using induction…

Theorem. Handshaking Theorem

For any graph the sum of vertex-degrees equals twice the number of edges, $\sum_{i=1}^{n} \delta_i = 2|E|$.

DEGREE SEQUENCE – HANDSHAKING THEOREM

$n = |V|$

Theorem. Handshaking Theorem

For any graph the sum of vertex-degrees equals twice the number of edges, $\sum_{i=1}^{n} \delta_i = 2|E|$.

Proof. We prove that any graph with $m \geq 0$ edges has $\Sigma_{i=1}^{n}\, \partial_i = 2m$ by induction on m.

1. **[Base case]** For $m = 0$, every $\partial_i = 0$, so $\Sigma_i\, \partial_i = 0 = 2m$.
2. **[Induction step]** Assume the claim holds for every graph with m edges.

   We must prove that the claim holds for arbitrary graph G with $m + 1$ edges.

DEGREE SEQUENCE – HANDSHAKING THEOREM

$n = |V|$

Theorem. Handshaking Theorem

For any ... $= 2|E|$.

Proo ... n on m.

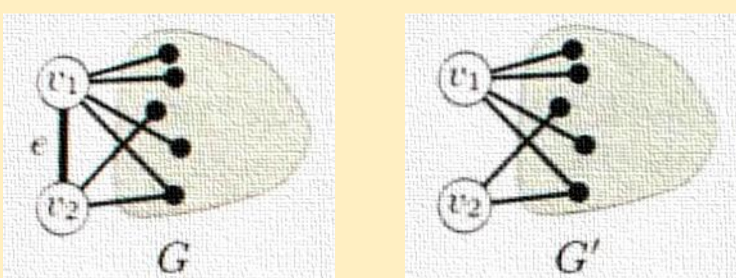
1. [I

2. [I

V

Graph G has $m + 1$ edges, including arbitrary edge e = (v_1, v_2)

Graph G′ has m edges and is otherwise identical to G



DEGREE SEQUENCE – HANDSHAKING THEOREM

$n = |V|$

Theorem. Handshaking Theorem

For any graph the sum of vertex-degrees equals twice the number of edges, $\sum_{i=1}^{n} \delta_i = 2|E|$.

Proof. We prove that any graph with $m \geq 0$ edges has $\sum_{i=1}^{n} \partial_i = 2m$ by induction on m.

1. **[Base case]** For $m = 0$, every $\partial_i = 0$, so $\sum_i \partial_i = 0 = 2m$.
2. **[Induction step]** Assume the claim holds for every graph with m edges.

   We must prove that the claim holds for arbitrary graph G with $m + 1$ edges.

   Graph G has $m + 1$ edges, including arbitrary edge $e = (v_1, v_2)$.

   Removing e gives us G′ with m edges. Assume remaining vertices are $v_3, v_4, \ldots, v_n$.

   Let ∂_i and ∂_i' be vertex degrees for G and G′ — then $\partial_i = \partial_i'$ for $j \geq 3$.

Theorem. Handshaking Theorem

For any graph the sum of vertex-degrees equals twice the number of edges, $\sum_{i=1}^{n} \delta_i = 2|E|$.

Proof. We prove that any graph with $m \geq 0$ edges has $\sum_{i=1}^{n} \partial_i = 2m$ by induction on m.

1. **[Base case]** For $m = 0$, every $\partial_i = 0$, so $\sum_i \partial_i = 0 = 2m$.
2. **[Induction step]** Assume the claim holds for every graph with m edges.

   We must prove that the claim holds for arbitrary graph G with $m + 1$ edges.

   Graph G has $m + 1$ edges, including arbitrary edge $e = (v_1, v_2)$.

   Removing e gives us G′ with m edges. Assume remaining vertices are $v_3, v_4, \ldots, v_n$.

   Let ∂_i and ∂_i' be vertex degrees for G and G′ — then $\partial_i = \partial_i'$ for $j \geq 3$.

   From our induction hypothesis, G′ has $\sum_i \partial_i' = 2m$.

   Rewrite this as $2m = \sum_i \partial_i' = \partial_1' + \partial_2' + \sum_{i=3}^{n} \partial_i' = \partial_1' + \partial_2' + \sum_{i=3}^{n} \partial_i$ (see above).

   Since $\partial_1' = \partial_1 - 1$ and $\partial_2' = \partial_2 - 1$, we have $2m = \sum_{i=1}^{n} \partial_i - 2$.

   Adding 2 to both sides gives us $2(m + 1) = \sum_{i=1}^{n} \partial_i$ — as was to be shown. ∎

ROOTED BINARY TREE

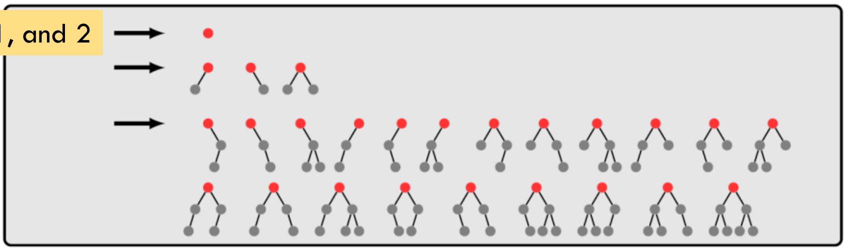
Write a recursive definition that generates RBTs of arbitrary height…

A *rooted binary tree* (*RBT*) is a graph with $|V| \geq 0$ vertices and the following properties:

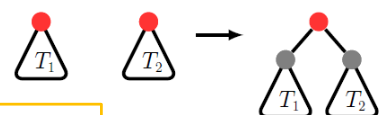
1. One vertex is identified as the *root* of the tree (indicated in red)
2. There is exactly one path from the root to any other vertex in the tree

   (Here, a *path* is defined as a sequence of edge traversals…)
3. **Each vertex has at most two children!**

RBTs of height 0, 1, and 2

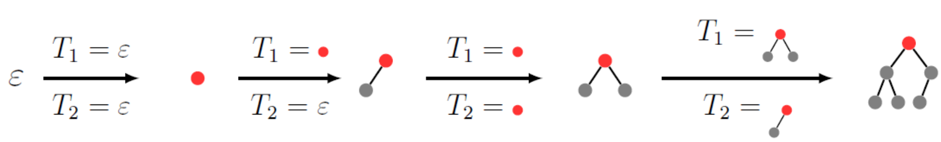


ROOTED BINARY TREE



Recursive definition of Rooted Binary Tree (RBT).

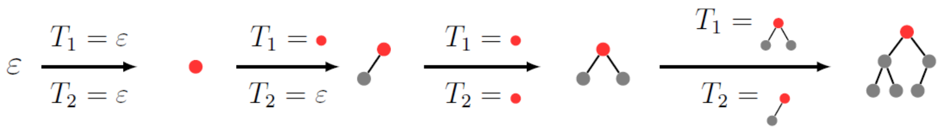
1. The empty tree ε is an RBT (no root). **[basis]**
2. If T_1 and T_2 are disjoint RBTs with roots r_1 and r_2, then linking r_1 and r_2 to a new root r produces a new RBT with root r. **[constructor]**



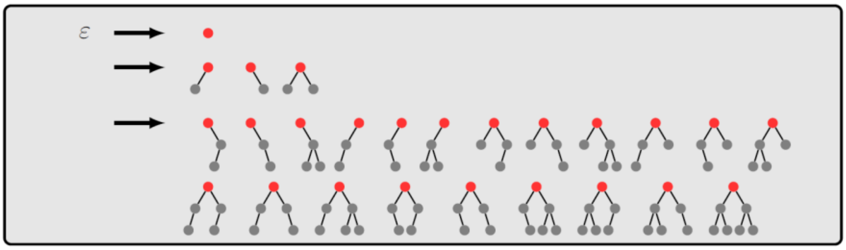
ROOTED BINARY TREE

Recursive definition of Rooted Binary Tree (RBT).

1. The empty tree ε is an RBT (no root).
2. If T_1 and T_2 are disjoint RBTs with roots r_1 and r_2, then linking r_1 and r_2 to a new root r produces a new RBT with root r.



RBTs of height 0, 1, and 2



Can you prove that an RBT with $n \geq 1$ vertices must have $n - 1$ edges…?

Recursive definition of Rooted Binary Tree (RBT).

1. The empty tree ε is an RBT (no root).
2. If T_1 and T_2 are disjoint RBTs with roots r_1 and r_2, then linking r_1 and r_2 to a new root r produces a new RBT with root r.

STRUCTURAL INDUCTION

Our claim is $P(T)$: if T is an RBT with $n \geq 1$ vertices, then T has $n - 1$ edges.

Proof. We use structural induction to prove claim $P(T)$.

1. **[Base case]** For $P(\varepsilon)$, the claim is **T** since ε is not an RBT with $n \geq 1$…
2. **[Induction step]** We must prove that the constructor preserves P. For our constructor, let parent T_1 have v_1 vertices and e_1 edges, and let parent T_2 have v_2 vertices and e_2 edges. Assume $P(T_1) \wedge P(T_2)$ is **T** (i.e., our induction hypothesis). Let the constructed RBT T_{new} have v_{new} vertices and e_{new} edges. We must show $e_{new} = v_{new} - 1$.

   Case 1. $T_1 = T_2 = \varepsilon$. Here, tree T_{new} has $v_{new} = 1$, $e_{new} = 0$, and $e_{new} = v_{new} - 1$.

   Case 2. $T_1 = \varepsilon$; $T_2 \neq \varepsilon$. Here, $v_{new} = v_2 + 1$ and $e_{new} = e_2 + 1$. Applying the induction hypothesis for e_2, we have $e_{new} = e_2 + 1 = (v_2 - 1) + 1 = v_2 = v_{new} - 1$.

Our claim is $P(T)$: if T is an RBT with $n \geq 1$ vertices, then T has $n - 1$ edges.

Proof. We use structural induction to prove claim $P(T)$.

1. **[Base case]** For $P(\varepsilon)$, the claim is **T** since ε is not an RBT with $n \geq 1$…
2. **[Induction step]** We must prove that the constructor preserves P. For our constructor, let parent T_1 have v_1 vertices and e_1 edges, and let parent T_2 have v_2 vertices and e_2 edges. Assume $P(T_1) \wedge P(T_2)$ is **T** (i.e., our induction hypothesis). Let the constructed RBT T_{new} have v_{new} vertices and e_{new} edges. We must show $e_{new} = v_{new} - 1$.

   here, the constructor preserves property $P(T)$, i.e., $e_{new} = v_{new} - 1$

   Case 1. $T_1 = T_2 = \varepsilon$. Here, tree T_{new} has $v_{new} = 1$, $e_{new} = 0$, and $e_{new} = v_{new} - 1$.

   Case 2. $T_1 = \varepsilon$; $T_2 \neq \varepsilon$. Here, $v_{new} = v_2 + 1$ and $e_{new} = e_2 + 1$. Applying the induction hypothesis, we have $e_{new} = e_2 + 1 = (v_2 - 1) + 1 = v_2 = v_{new} - 1$.

   Case 3. $T_1 \neq \varepsilon$; $T_2 = \varepsilon$. Here, $v_{new} = v_1 + 1$ and $e_{new} = e_1 + 1$. Applying the induction hypothesis, we have $e_{new} = e_1 + 1 = (v_1 - 1) + 1 = v_1 = v_{new} - 1$.

   Case 4. $T_1 \neq \varepsilon$; $T_2 \neq \varepsilon$. Here, $v_{new} = v_1 + v_2 + 1$ and $e_{new} = e_1 + e_2 + 2$. Applying the induction hypothesis, we have $e_{new} = e_1 + e_2 + 2 = (v_1 - 1) + (v_2 - 1) + 2$. Thus, $e_{new} = v_1 + v_2 = v_{new} - 1$.
3. By structural induction, $P(T)$ is **T** for all $T \in$ RBT. ■

Prove that an n-vertex graph with more than $n - 1$ edges must have a cycle…

TREES

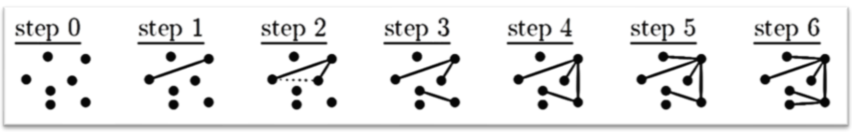
Removing both the *binary* and *rooted* constraints of RBTs, we define a *tree* as a graph with the following properties:

1. The graph is connected…

   …meaning there is exactly one path from any vertex to any other vertex

2. There are no cycles — the graph is *acyclic*

isomorphic trees



How many trees can we construct from $n = 7$ vertices…?

Prove that an n-vertex graph with more than $n - 1$ edges must have a cycle…

SPANNING TREES

How many spanning trees can we construct for the connected graph on the left…?

A graph is *connected* if every pair of vertices is connected…

…and two vertices v_1 and v_2 are *connected* if there is a path between v_1 and v_2

Given graph $G = (V, E)$, a *spanning tree* is a subgraph $H = (V, E')$ such that H is connected and acyclic

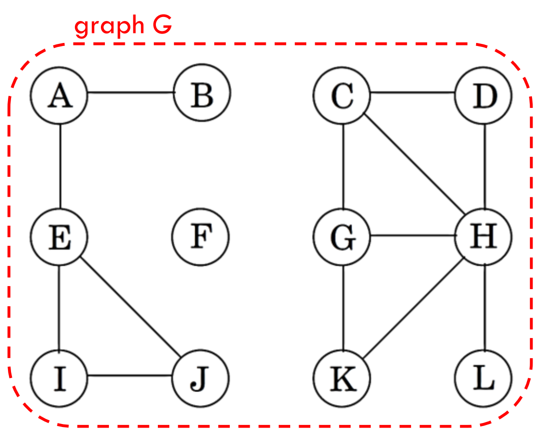
If $E = E'$, then G is a tree — and if G and H are weighted graphs, we are often interested in finding a *minimum spanning tree*…

Prove that an n-vertex graph with more than $n - 1$ edges must have a cycle…

SPANNING TREES AND COMPONENTS

A *component* (or *connected component*) is a graph or induced subgraph in which every vertex is connected to every other vertex…

In a connected component, there is a path from any vertex to any other vertex



How are components and spanning trees related?

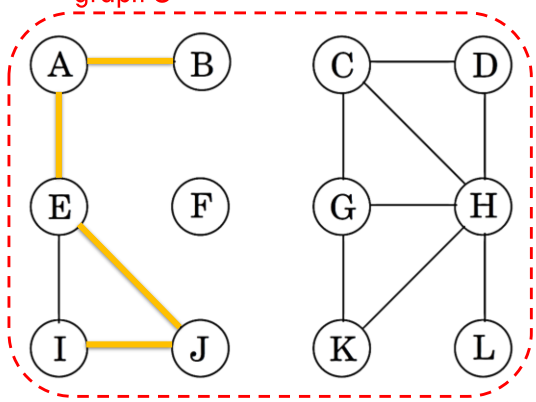
For each component in a graph, we can construct one or more spanning trees…

How many spanning trees are there in each component of the given graph…?

Prove that an n-vertex graph with more than $n-1$ edges must have a cycle…

SPANNING TREES AND COMPONENTS

graph G



To construct spanning tree $T = (V, E)$ for G, we start with arbitrary vertex v_1 — $V = \{v_1\}$

As long as there is an untraversed edge e that leads to vertex $v_i \notin V$, we add e to E and also add v_i to V

e.g., $V = \{ A, B, E, J, I \}$

$E = \{ (A,B), (A,E), (E,J), (I,J) \}$

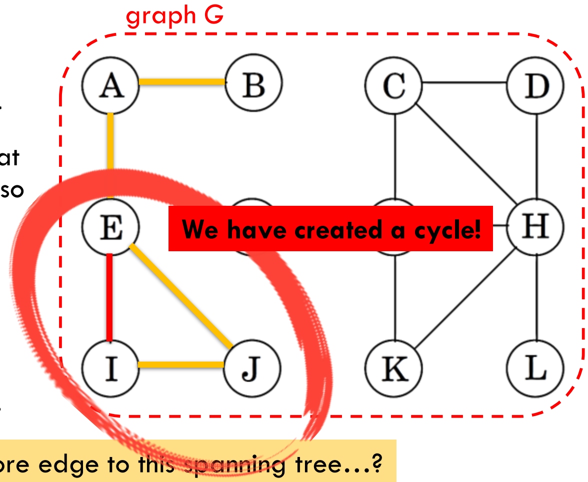
In general, this construction produces a component of G using only $|V| - 1$ edges…

What happens if we add one more edge to this spanning tree…?

Prove that an n-vertex graph with more than $n-1$ edges must have a cycle…

SPANNING TREES AND COMPONENTS

graph G



To construct spanning tree $T = (V, E)$ for G, we start with arbitrary vertex v_1 — $V = \{v_1\}$

As long as there is an untraversed edge e that leads to vertex $v_i \notin V$, we add e to E and also add v_i to V

e.g., $V = \{ A, B, E, J, I \}$

$E = \{ (A,B), (A,E), (E,J), (I,J) \}$

In general, this construction produces a component of G using only $|V| - 1$ edges…

What happens if we add one more edge to this spanning tree…?

Prove that an n-vertex graph with more than $n - 1$ edges must have a cycle…

SPANNING TREES AND COMPONENTS

To construct spanning tree $T = (V, E)$ for G, we start with arbitrary vertex v_1 — $V = \{v_1\}$

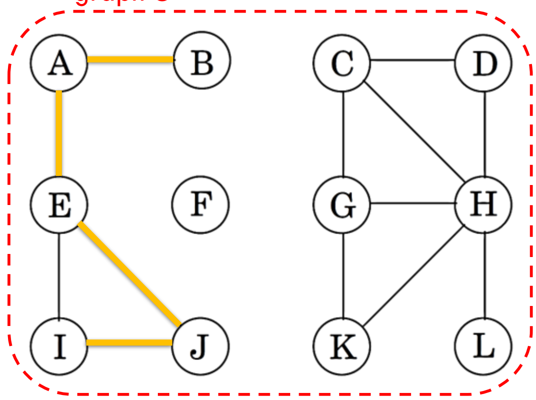
As long as there is an untraversed edge e that leads to vertex $v_i \notin V$, we add e to E and also add v_i to V

e.g., $V = \{ A, B, E, J, I \}$

$E = \{ (A,B), (A,E), (E,J), (I,J) \}$

In general, this construction produces a component of G using only $|V| - 1$ edges…

graph G



Apply this same technique with starting vertex C…

Prove that an n-vertex graph with more than $n - 1$ edges must have a cycle…

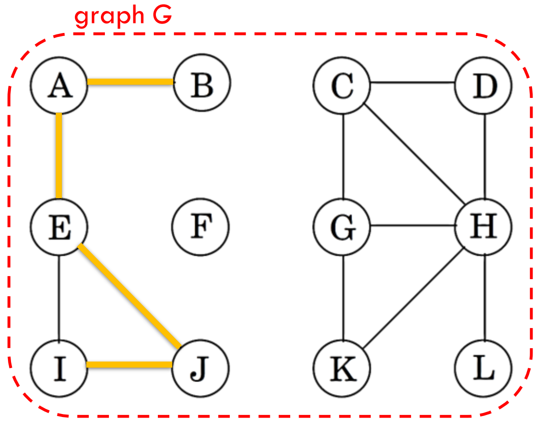
SPANNING TREES AND COMPONENTS

Use strong induction to write a formal proof to the above claim that an n-vertex graph with more than $n - 1$ edges must have a cycle

For the induction step, take any graph with n vertices and $n + k$ edges ($k \geq 0$), remove an edge e, then consider two cases…

Case 1. If the graph remains connected, show that adding e back to the graph will create a cycle

graph G



Case 2. If the graph becomes disconnected, show that there must be a cycle — and adding e back certainly will not remove that cycle!

See Exercise 11.6…

WHAT NEXT…?

Exam 2 is on Wednesday, November 9 — email me if you have extra-time accommodations and we have yet to schedule a make-up for this exam!

Problem Set 6 will be posted later today…

- …and due in your recitations **next week** on Wednesday, November 9

Earning late days has still not been tallied, so still assume you have earned them even though you do not yet see them in Submitty…

Practice! Tinker! Practice! Tinker! Practice! Tinker! Practice! Tinker! Practice!