15-251

Great Theoretical Ideas in Computer Science

Upcoming Events

Review Session on Saturday
(5 pm, Wean 5409)

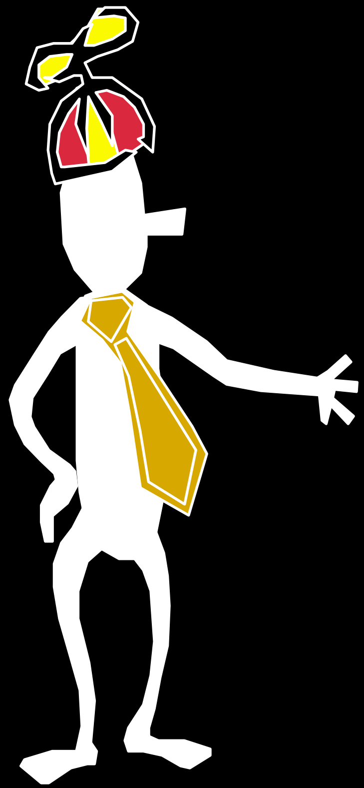
Test on Monday

Election Day

Graphs

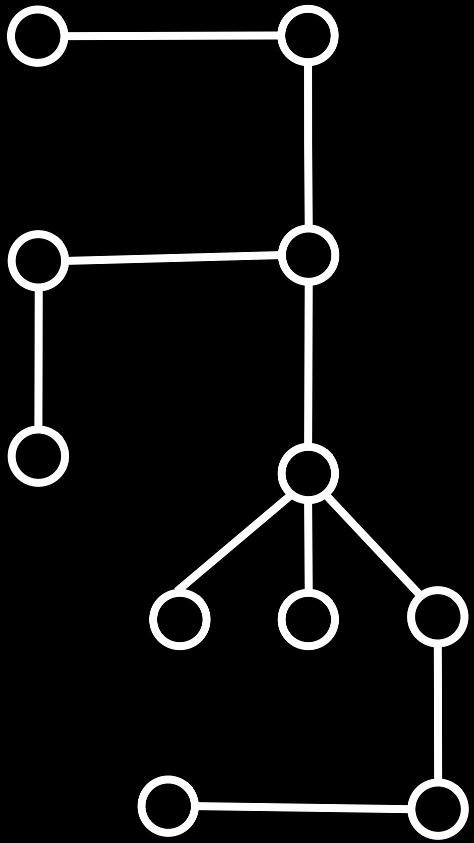
Lecture 18, October 23, 2008

What’s a tree?

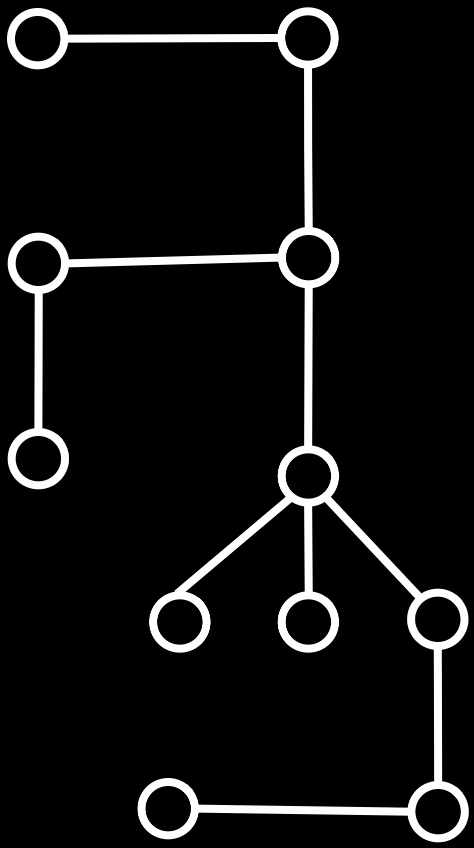


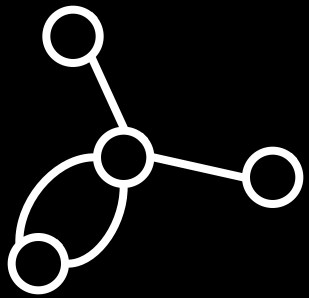
What's a tree?

A **tree** is a connected graph with no cycles

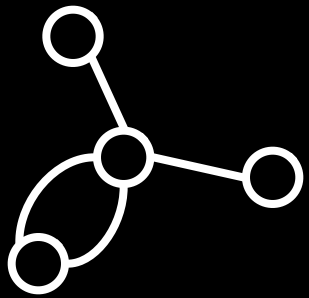


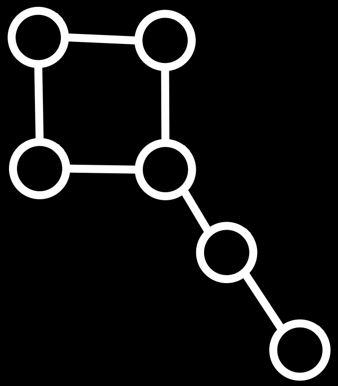
Tree



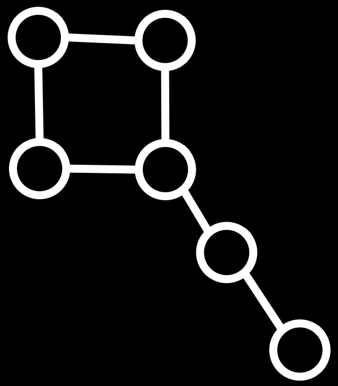


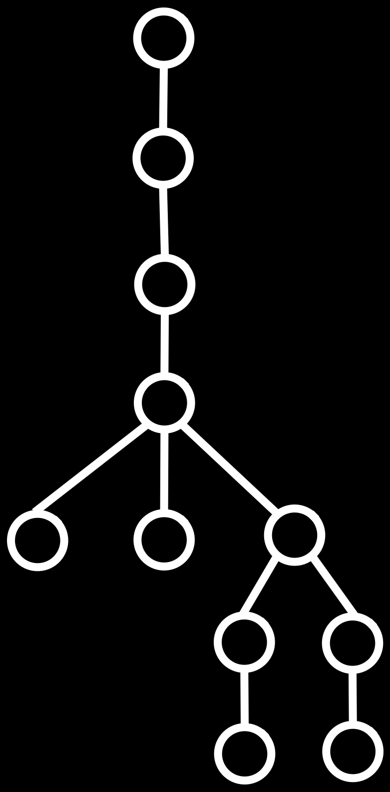
Not aTree



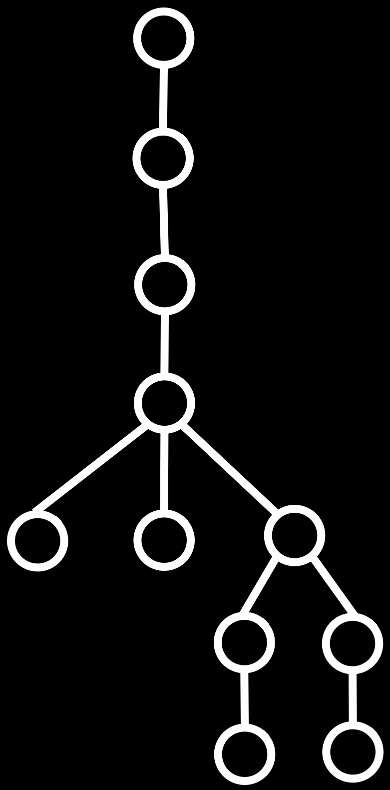


Not a Tree





Tree



How Many n-Node Trees?

How Many n-Node Trees?

1:

How Many n-Node Trees?

1: 

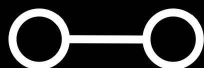
How Many n-Node Trees?

1: 

2:

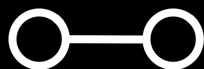
How Many n-Node Trees?

1: 

2: 

How Many n-Node Trees?

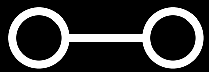
1: 

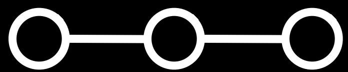
2: 

3:

How Many n-Node Trees?

1: 

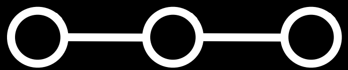
2: 

3: 

How Many n-Node Trees?

1: 

2: 

3: 

4:

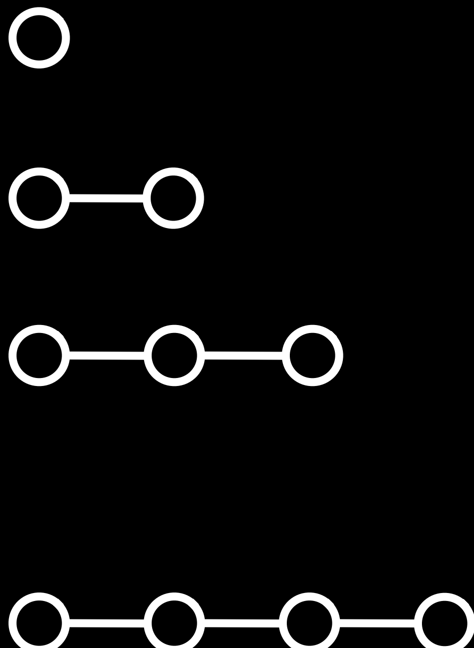
How Many n-Node Trees?

1:

2:

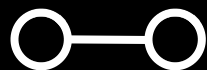
3:

4:



How Many n-Node Trees?

1: 

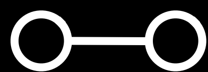
2: 

3: 

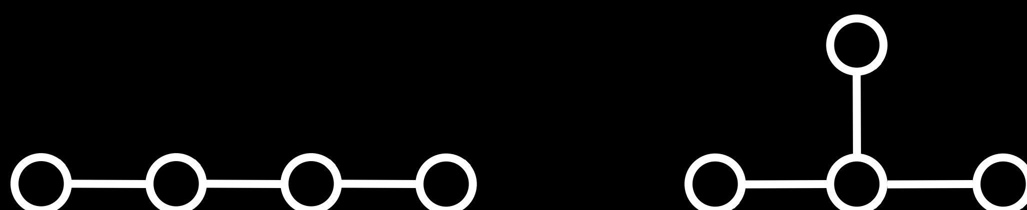
4: 

How Many n-Node Trees?

1: 

2: 

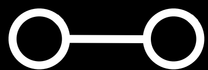
3: 

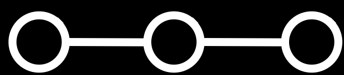
4: 

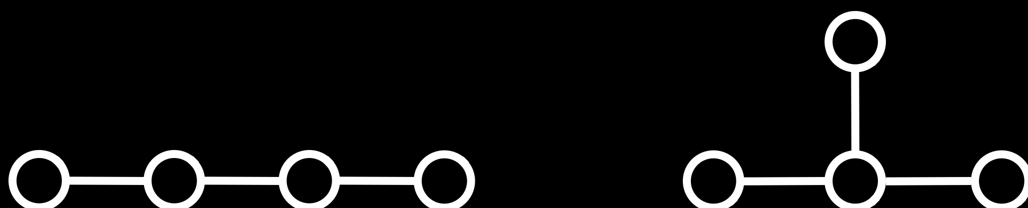
5:

How Many n-Node Trees?

1: 

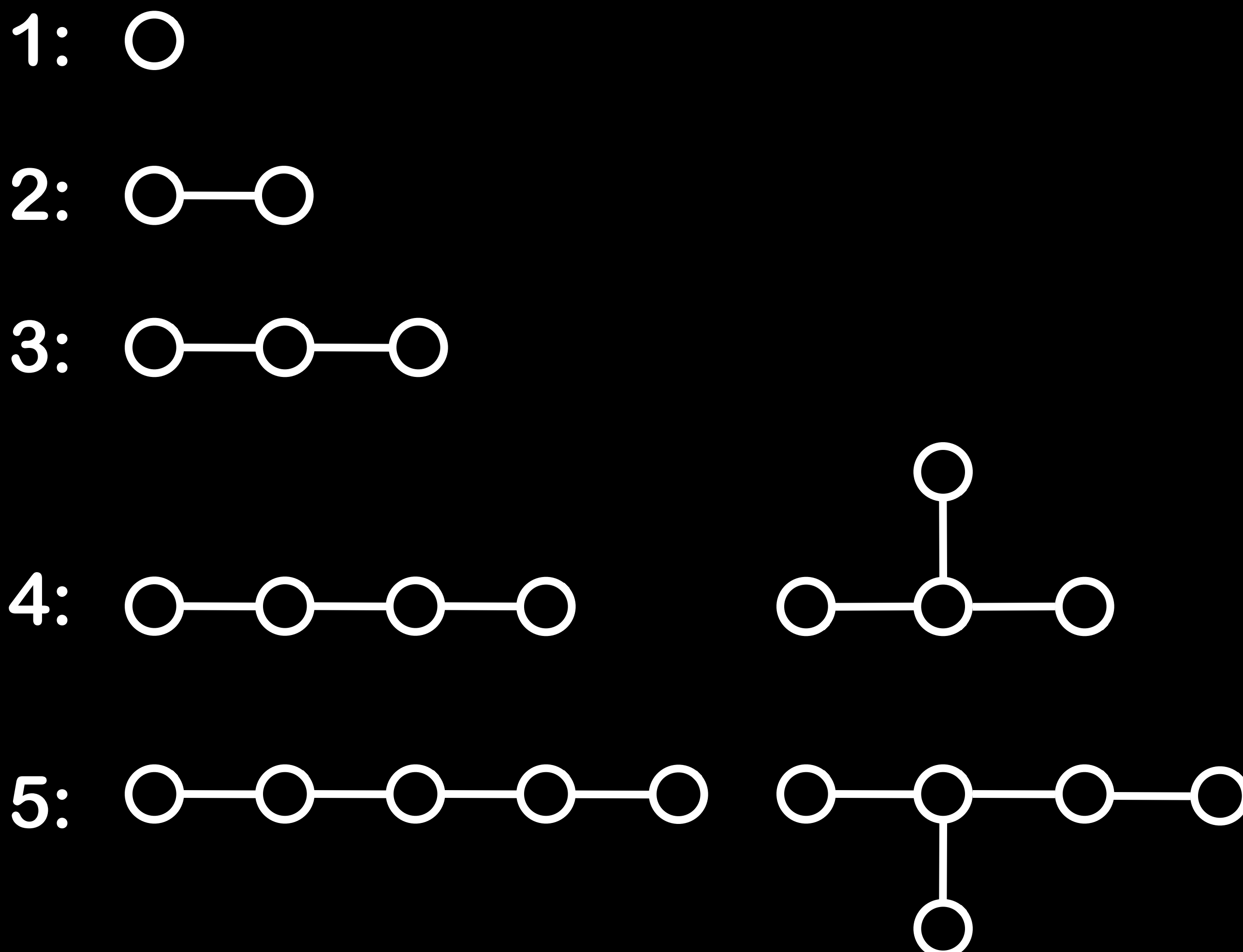
2: 

3: 

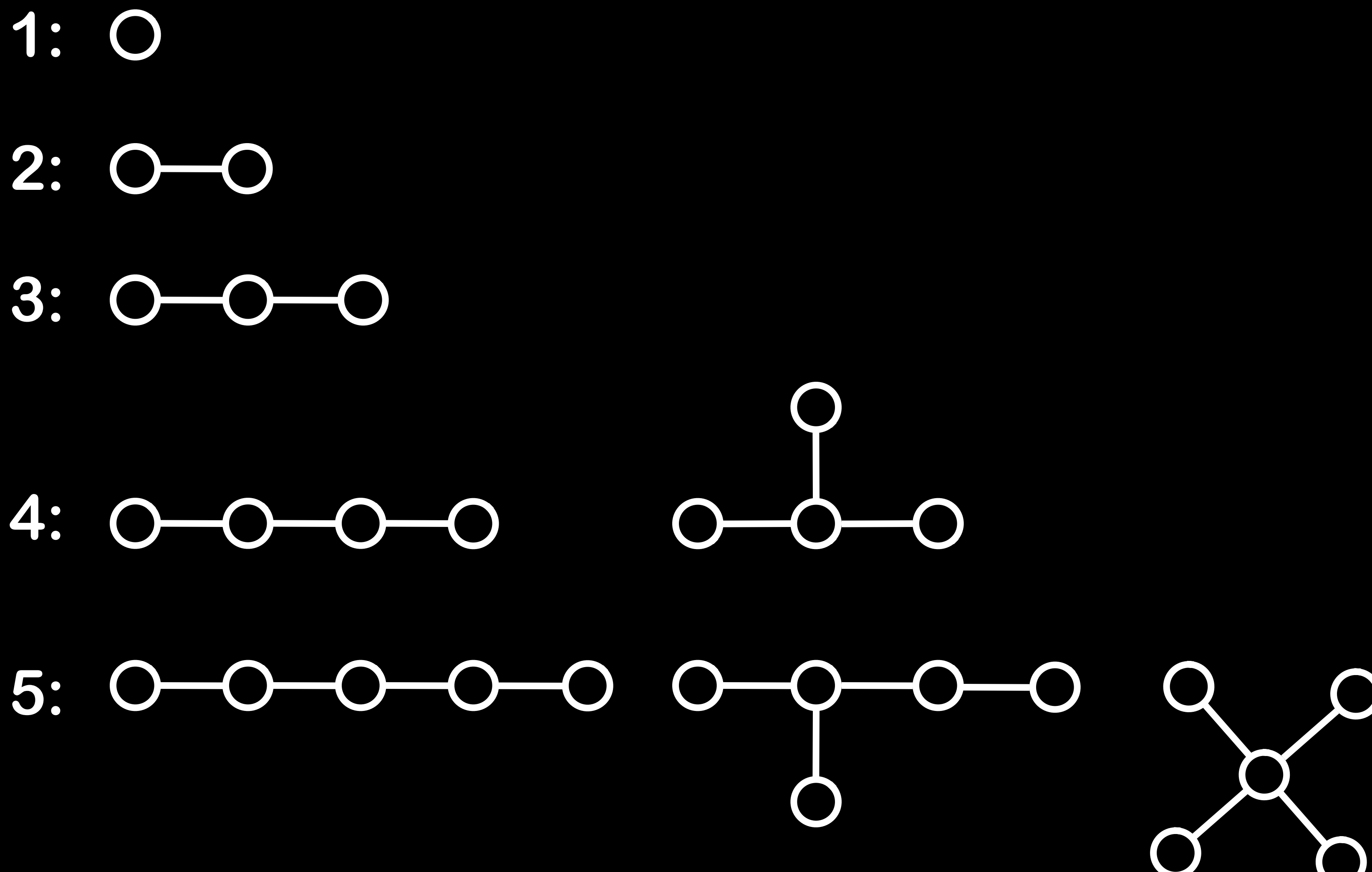
4: 

5: 

How Many n-Node Trees?



How Many n-Node Trees?



We'll pass around a piece of paper. Draw a new 8-node tree, and put your name next to it. (There are 23 of them…)

The Shy People Party

At the shy people party, people enter one-by-one, and as a person comes in, (s)he shakes hand with only one person already at the party.

The Shy People Party

At the shy people party, people enter one-by-one, and as a person comes in, (s)he shakes hand with only one person already at the party.

Prove that at a shy party with n people (n >= 2), at least two people have shaken hands with only one other person.

The Shy People Party

Notation

Notation

In this lecture:

Notation

In this lecture:

n will denote the number of nodes in a graph

Notation

In this lecture:

- n will denote the number of nodes in a graph
- e will denote the number of edges in a graph

Theorem: Let G be a graph with n nodes and e edges

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path
3. G is connected and $n = e + 1$

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path
3. G is connected and $n = e + 1$
4. G is acyclic and $n = e + 1$

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path
3. G is connected and $n = e + 1$
4. G is acyclic and $n = e + 1$
5. G is acyclic and if any two non-adjacent points are joined by a line, the resulting graph has exactly one cycle

To prove this, it suffices to show

$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$

1 => 2

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path

1 => 2

1. G is a tree (connected, acyclic)

2. Every two nodes of G are joined by a unique path

Proof: (by contradiction)

1 => 2

1. G is a tree (connected, acyclic)

2. Every two nodes of G are joined by a unique path

Proof: (by contradiction)

Assume G is a tree that has two nodes connected by two different paths:

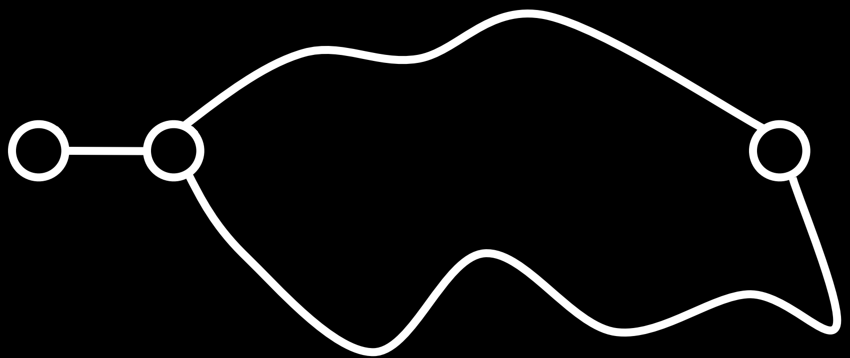
1 => 2

1. G is a tree (connected, acyclic)

2. Every two nodes of G are joined by a unique path

Proof: (by contradiction)

Assume G is a tree that has two nodes connected by two different paths:



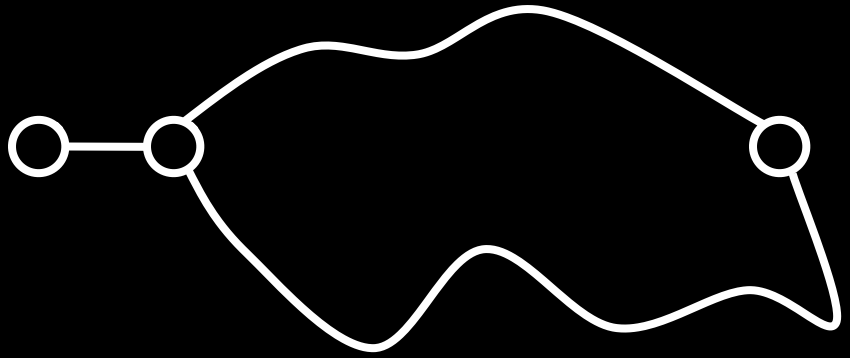
1 => 2

1. G is a tree (connected, acyclic)

2. Every two nodes of G are joined by a unique path

Proof: (by contradiction)

Assume G is a tree that has two nodes connected by two different paths:



Then there exists a cycle!

2 => 3

2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

2 => 3

2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

2 => 3

2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with < n nodes

$2 \Rightarrow 3$

2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with $<$ n nodes

Let G have n nodes and let x and y be adjacent

2 => 3

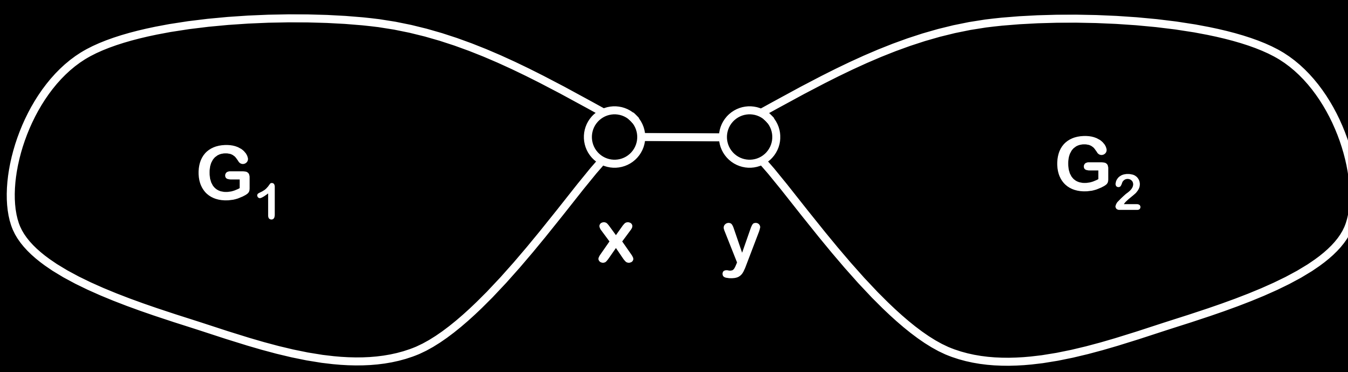
2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with < n nodes

Let G have n nodes and let x and y be adjacent



2 => 3

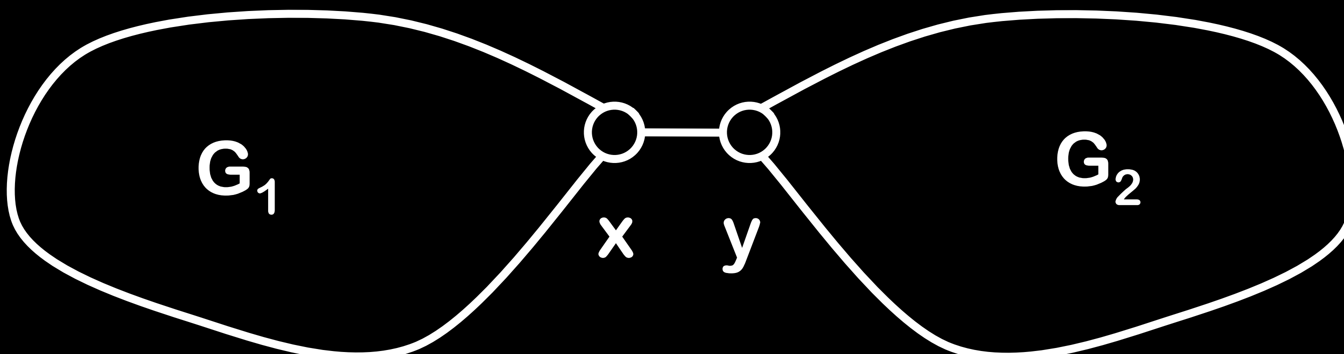
2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with < n nodes

Let G have n nodes and let x and y be adjacent



Let n_1, e_1 be number of nodes and edges in G_1

2 => 3

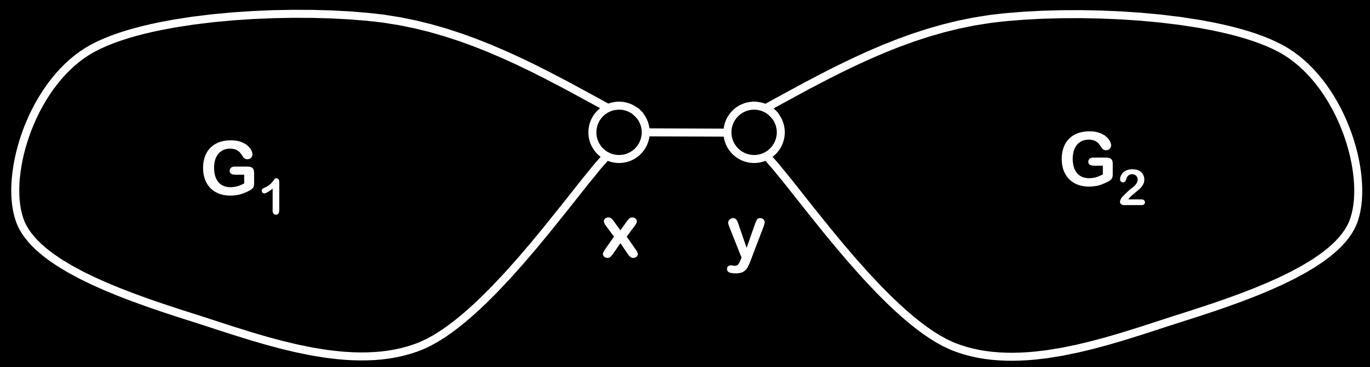
2. Every two nodes of G are joined by a unique path

3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with < n nodes

Let G have n nodes and let x and y be adjacent



Let n_1, e_1 be number of nodes and edges in G_1

Then $n = n_1 + n_2 = e_1 + e_2 + 2 = e + 1$

3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

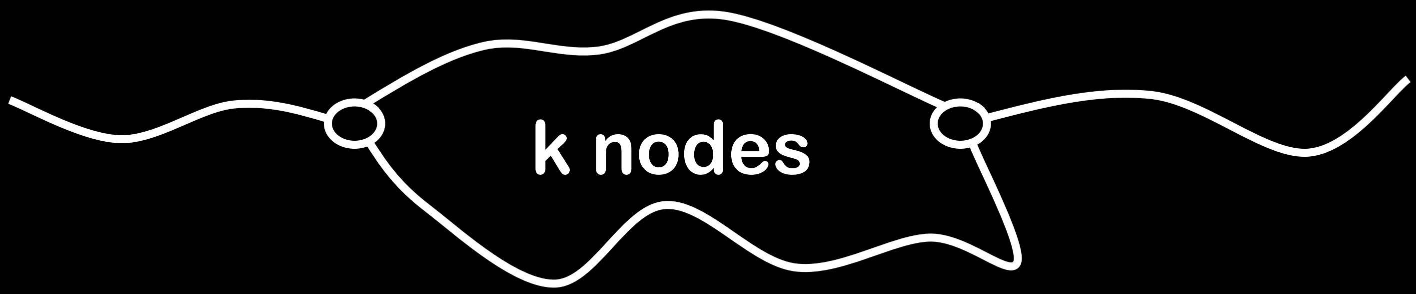
Assume G is connected with $n = e + 1$, and G has a cycle containing k nodes

3 => 4 3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

Assume G is connected with $n = e + 1$, and G has a cycle containing k nodes



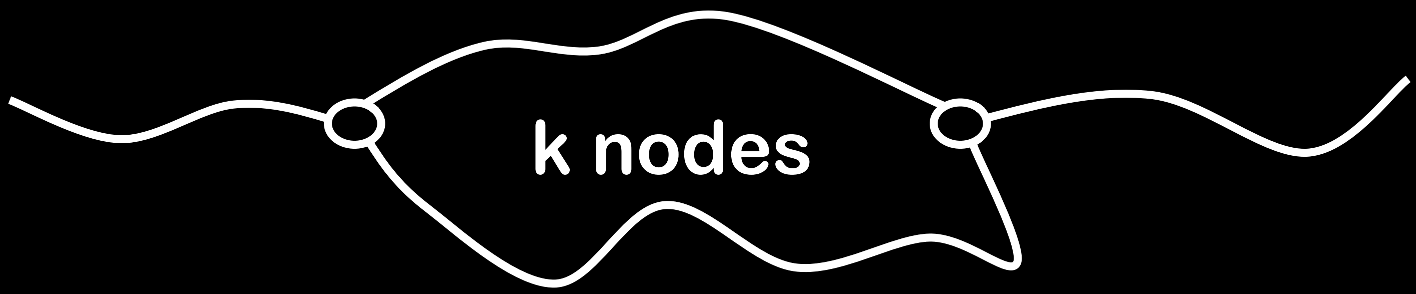
3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

Assume G is connected with $n = e + 1$, and G has a cycle containing k nodes



Note that the cycle has k nodes and k edges

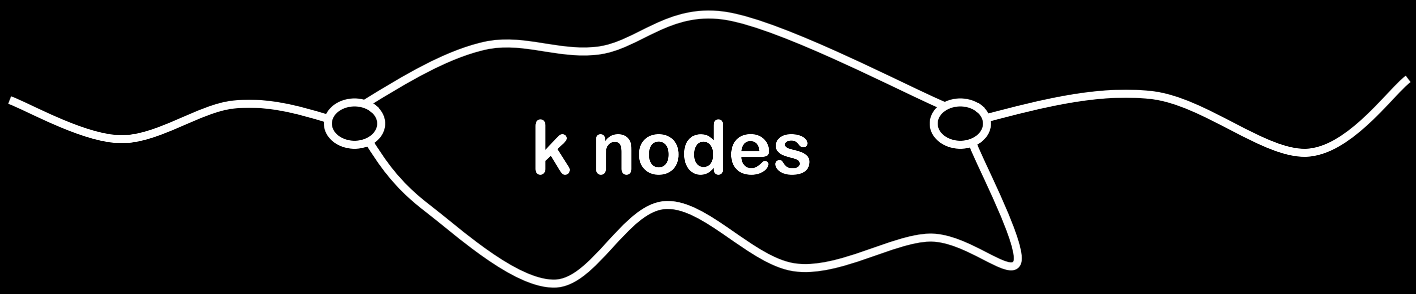
3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

Assume G is connected with $n = e + 1$,
and G has a cycle containing k nodes



Note that the cycle has k nodes and k edges

Start adding nodes and edges until you cover the whole graph

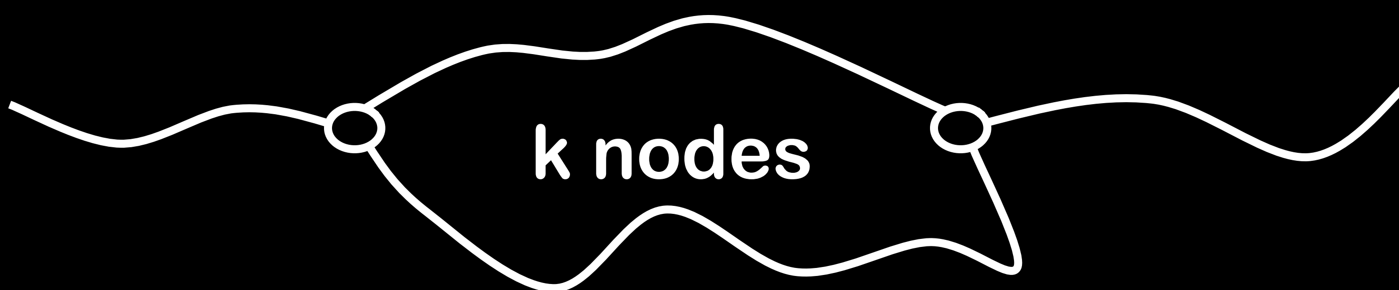
3 => 4

3. G is connected and $n = e + 1$

4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

Assume G is connected with $n = e + 1$, and G has a cycle containing k nodes



Note that the cycle has k nodes and k edges

Start adding nodes and edges until you cover the whole graph

Number of edges in the graph will be at least n

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Proof:

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Proof:

Assume all but one of the points in the tree have degree at least 2

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Proof:

Assume all but one of the points in the tree have degree at least 2

In any graph, sum of the degrees =

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Proof:

Assume all but one of the points in the tree have degree at least 2

In any graph, sum of the degrees = 2e

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

Proof:

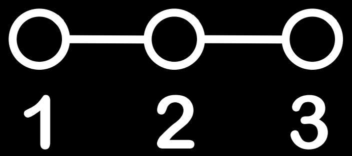
Assume all but one of the points in the tree have degree at least 2

In any graph, sum of the degrees = 2e

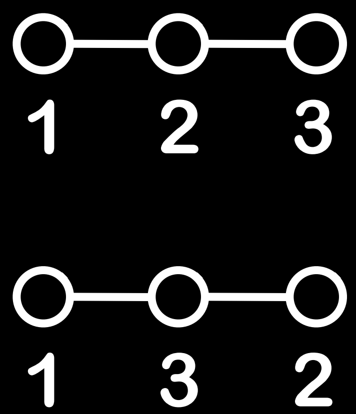
Then the total number of edges in the tree is at least (2n-1)/2 = n - 1/2 > n - 1

How many labeled trees are there with three nodes?

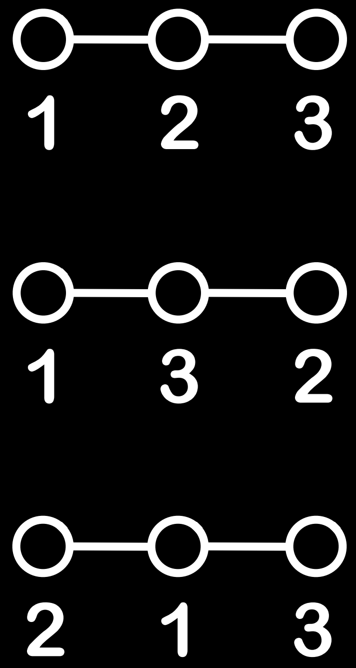
How many labeled trees are there with three nodes?



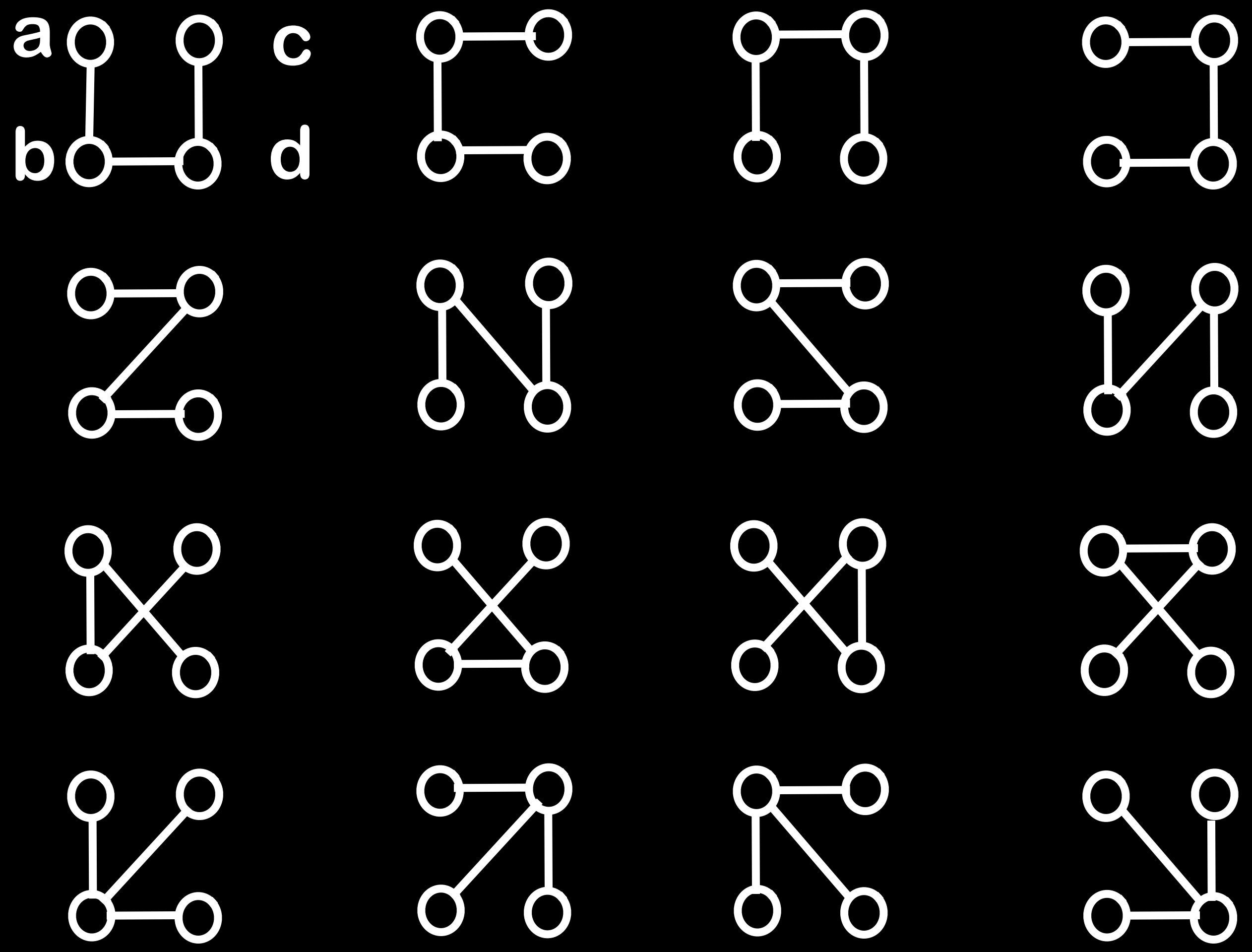
How many labeled trees are there with three nodes?



How many **labeled** trees are there with three nodes?

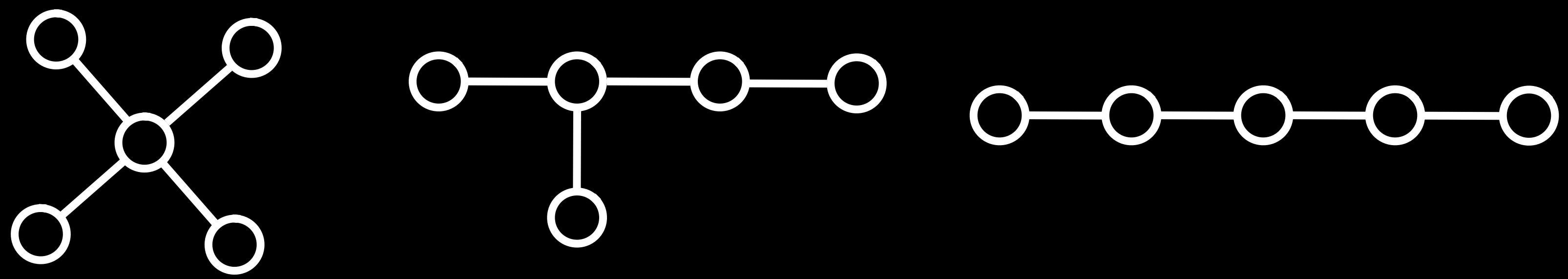


How many **labeled** trees are there with four nodes?

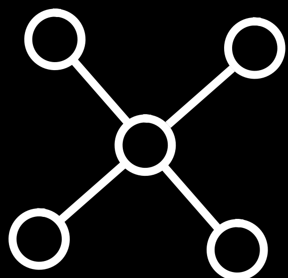


How many labeled trees are there with five nodes?

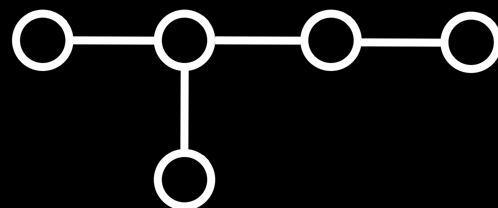
How many labeled trees are there with five nodes?



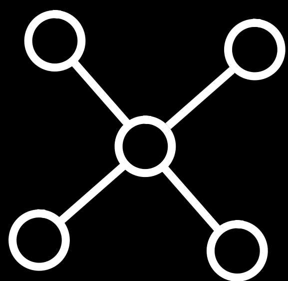
How many labeled trees are there with five nodes?



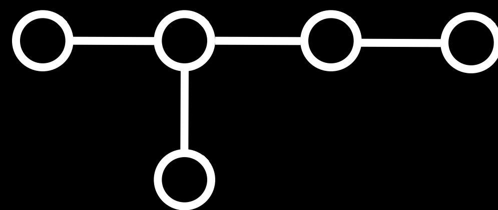
5
labelings



How many labeled trees are there with five nodes?



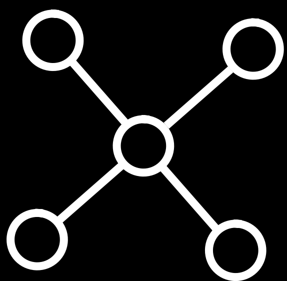
5
labelings

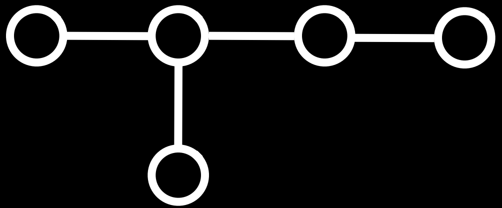


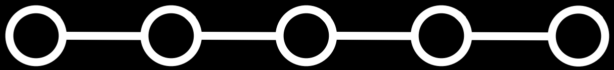
5 x 4 x 3

labelings

How many labeled trees are there with five nodes?







5
labelings

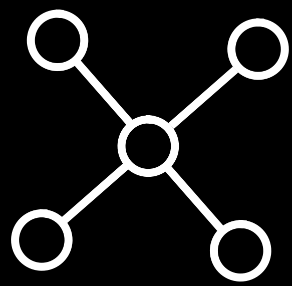
5 x 4 x 3

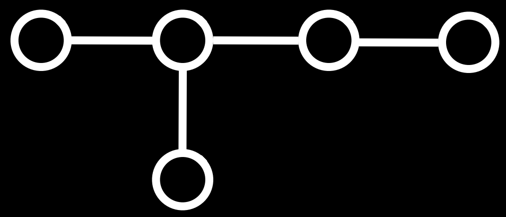
labelings

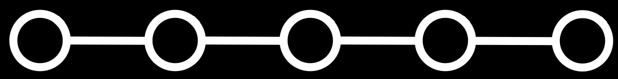
5!/ 2

labelings

How many labeled trees are there with five nodes?







5
labelings

5 x 4 x 3

labelings

5!/ 2

labelings

125 labeled trees

How many labeled trees are there with n nodes?

How many labeled trees are there with n nodes?

3 labeled trees with 3 nodes

How many labeled trees are there with n nodes?

3 labeled trees with 3 nodes

16 labeled trees with 4 nodes

How many labeled trees are there with n nodes?

3 labeled trees with 3 nodes

16 labeled trees with 4 nodes

125 labeled trees with 5 nodes

How many labeled trees are there with n nodes?

3 labeled trees with 3 nodes

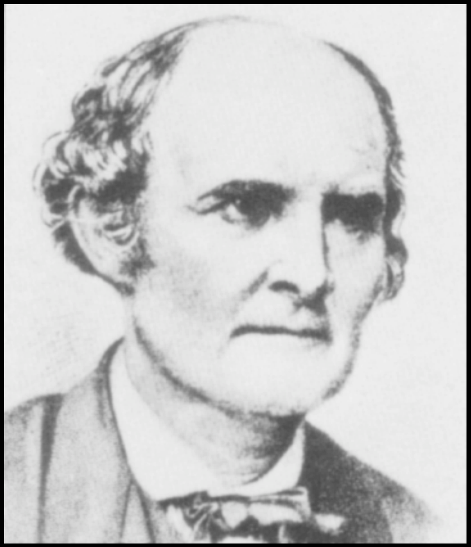
16 labeled trees with 4 nodes

125 labeled trees with 5 nodes

n^{n-2} labeled trees with n nodes

Cayley's Formula

The number of labeled trees on n nodes is n^{n-2}



The proof will use the correspondence principle

The proof will use the correspondence principle

Each labeled tree on n nodes

corresponds to

A sequence in $\{1,2,\ldots,n\}^{n-2}$ (that is, n-2 numbers, each in the range [1..n])

How to make a sequence from a tree?

How to make a sequence from a tree?

Loop through i from 1 to $n-2$

How to make a sequence from a tree?

Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

How to make a sequence from a tree?

Loop through i from 1 to $n-2$

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

How to make a sequence from a tree?

Loop through i from 1 to $n-2$

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

How to make a sequence from a tree?

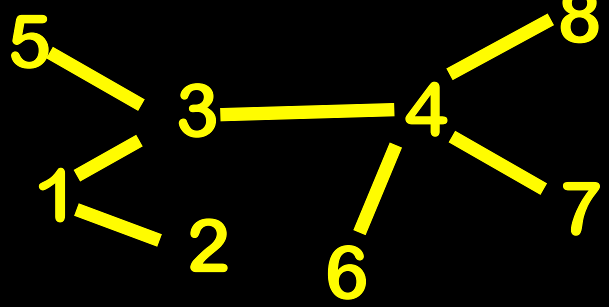
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



How to make a sequence from a tree?

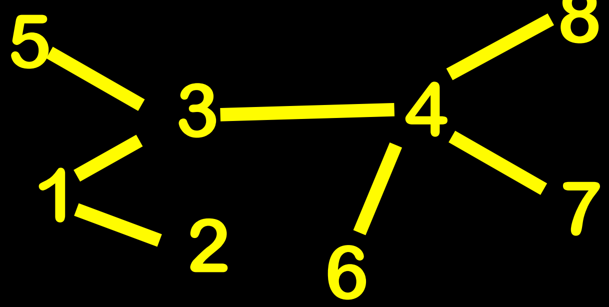
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1

How to make a sequence from a tree?

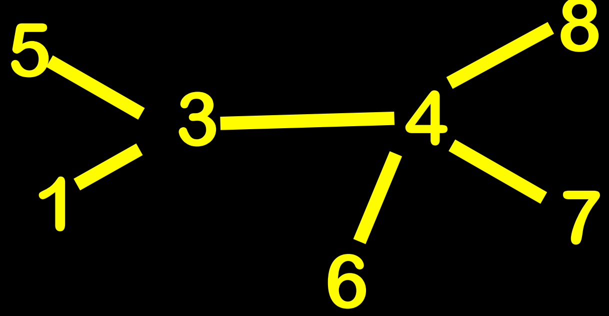
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1

How to make a sequence from a tree?

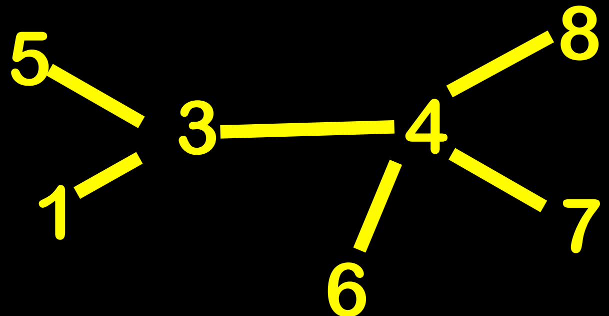
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3

How to make a sequence from a tree?

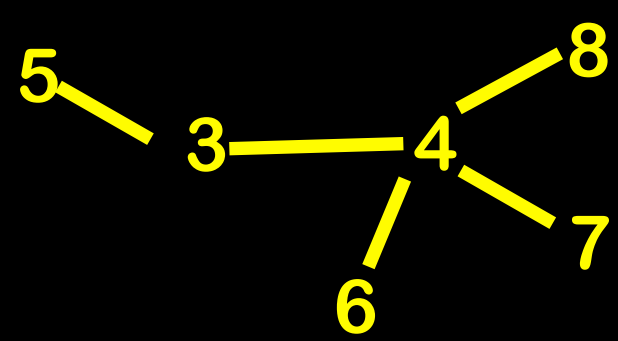
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3

How to make a sequence from a tree?

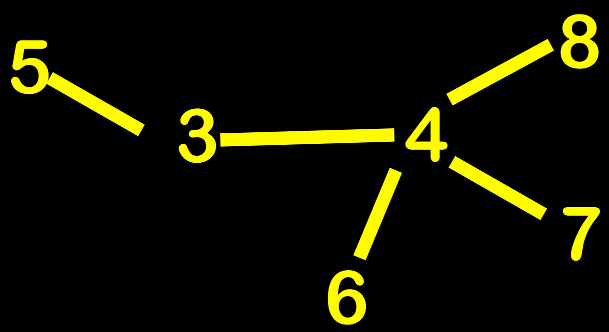
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3 3

How to make a sequence from a tree?

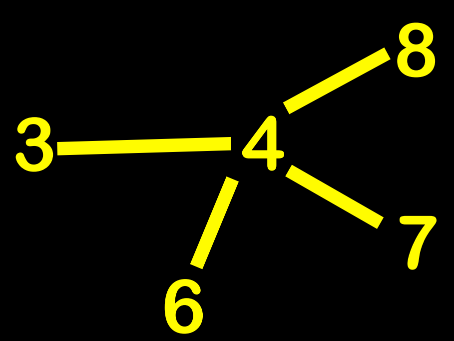
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the ith element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3 3

How to make a sequence from a tree?

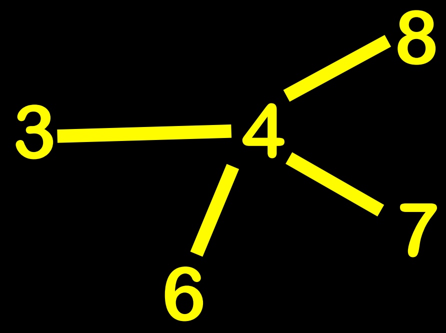
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3 3 4

How to make a sequence from a tree?

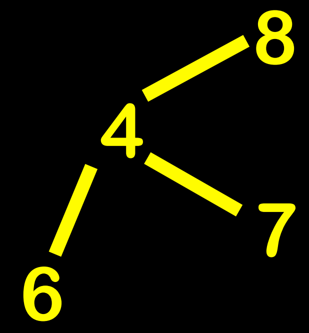
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the ith element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



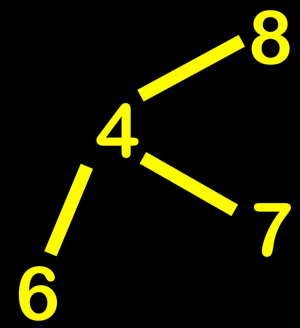
1 3 3 4

How to make a sequence from a tree?

Loop through i from 1 to n-2

- Let L be the degree-1 node with the lowest label
- Define the i^{th} element of the sequence as the label of the node adjacent to L
- Delete the node L from the tree

Example:



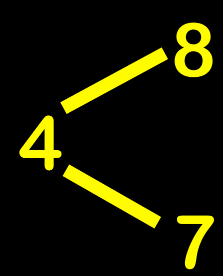
1 3 3 4 4

How to make a sequence from a tree?

Loop through i from 1 to n-2

- Let L be the degree-1 node with the lowest label
- Define the i^{th} element of the sequence as the label of the node adjacent to L
- Delete the node L from the tree

Example:



1 3 3 4 4

How to make a sequence from a tree?

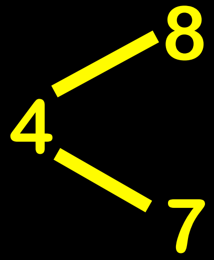
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3 3 4 4 4

How to make a sequence from a tree?

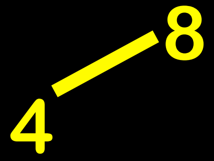
Loop through i from 1 to n-2

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:



1 3 3 4 4 4

How to make a sequence from a tree?

Loop through i from 1 to $n-2$

Let L be the degree-1 node with the lowest label

Define the i^{th} element of the sequence as the label of the node adjacent to L

Delete the node L from the tree

Example:

1 3 3 4 4 4

How to reconstruct the unique tree from a sequence S:

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

Loop until S is empty

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

Loop until S is empty

- Let i = smallest # in I but not in S
- Let s = first label in sequence S
  - Add edge {i, s} to the tree
  - Delete i from I
  - Delete s from S

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

Loop until S is empty

- Let i = smallest # in I but not in S
- Let s = first label in sequence S
  - Add edge {i, s} to the tree
  - Delete i from I
  - Delete s from S

Add edge {a,b}, where I = {a,b}

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

Loop until S is empty

  Let i = smallest # in I but not in S

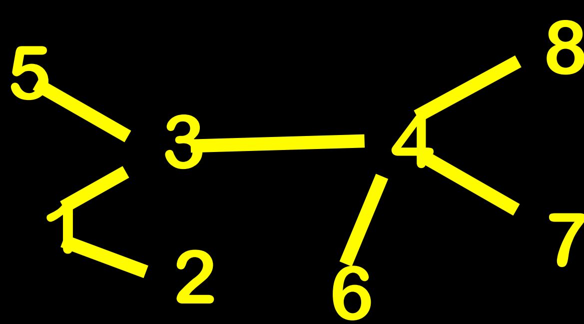
  Let s = first label in sequence S

    Add edge {i, s} to the tree

    Delete i from I

    Delete s from S

Add edge {a,b}, where I = {a,b}



1 3 3 4 4 4

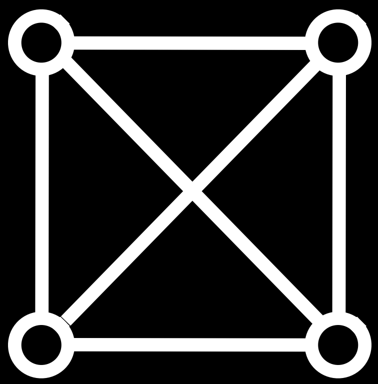
Spanning Trees

Spanning Trees

A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G

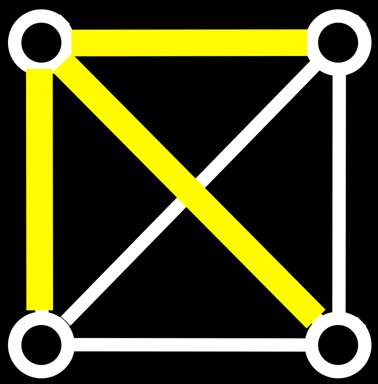
Spanning Trees

A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G



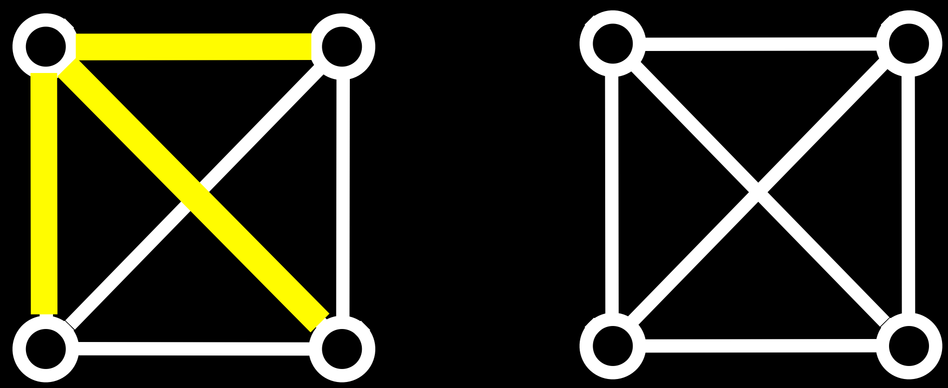
Spanning Trees

A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G



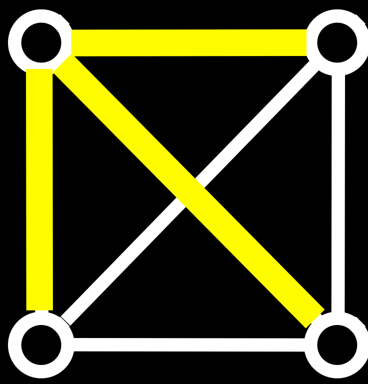
Spanning Trees

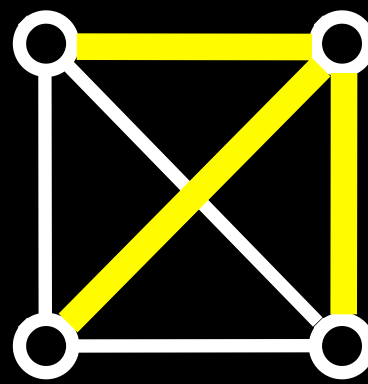
A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G



Spanning Trees

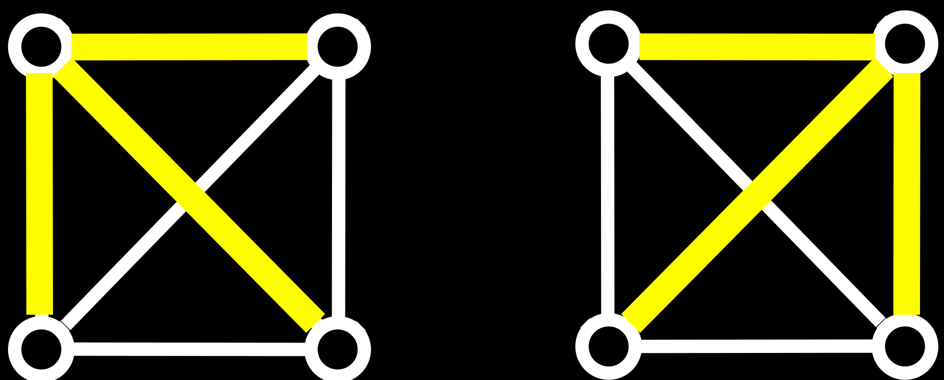
A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G



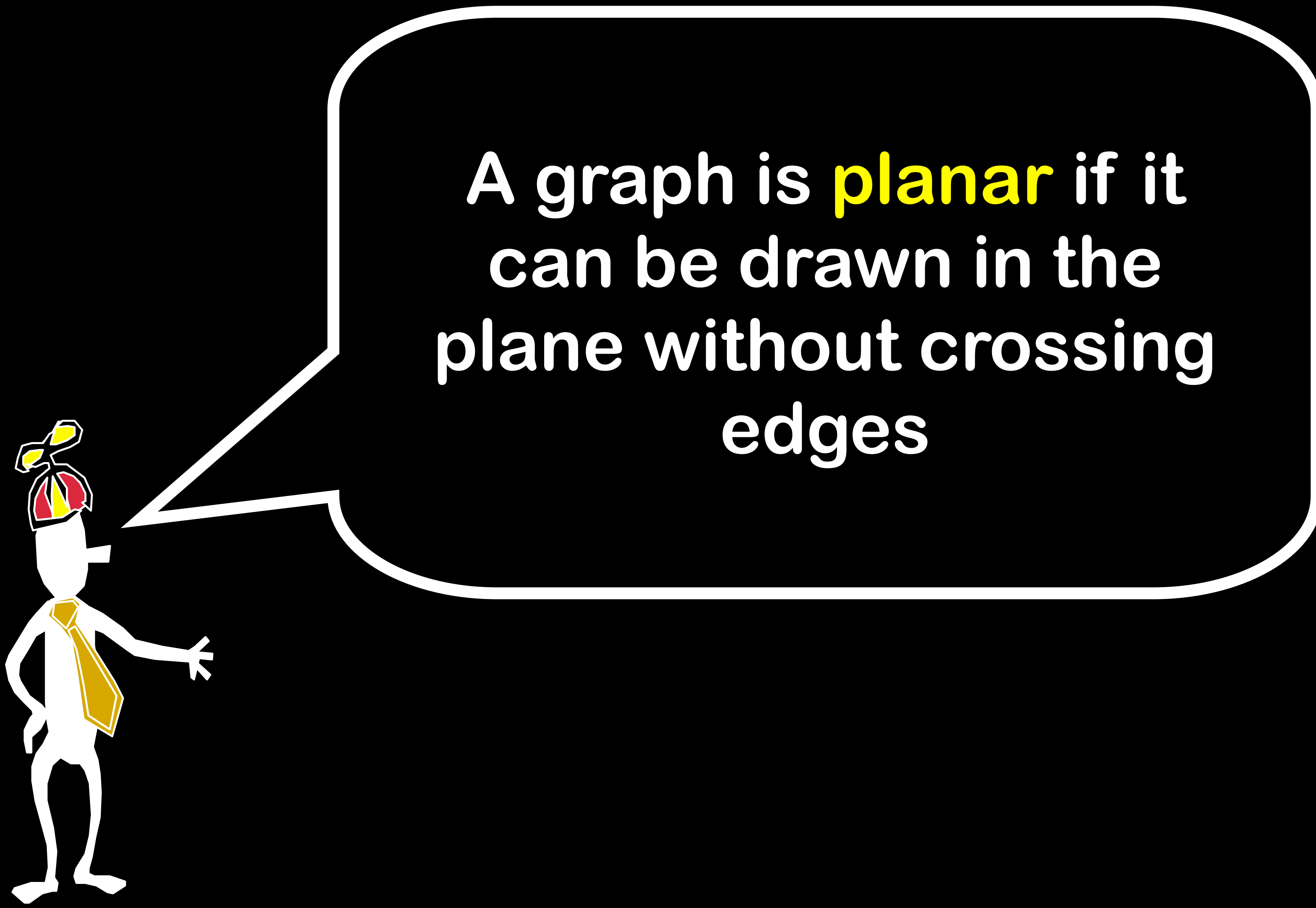


Spanning Trees

A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G

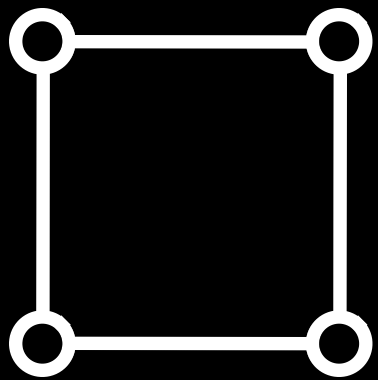


Every connected graph has a spanning tree

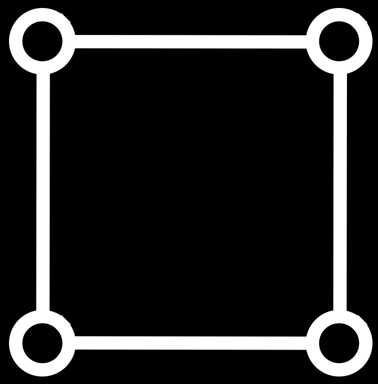

A graph is planar if it can be drawn in the plane without crossing edges

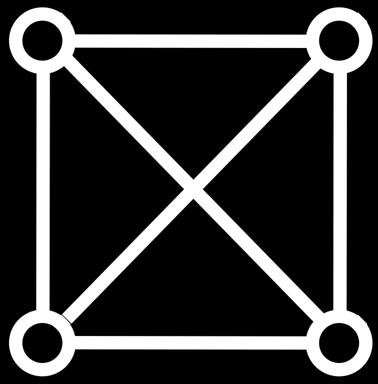
Examples of Planar Graphs

Examples of Planar Graphs

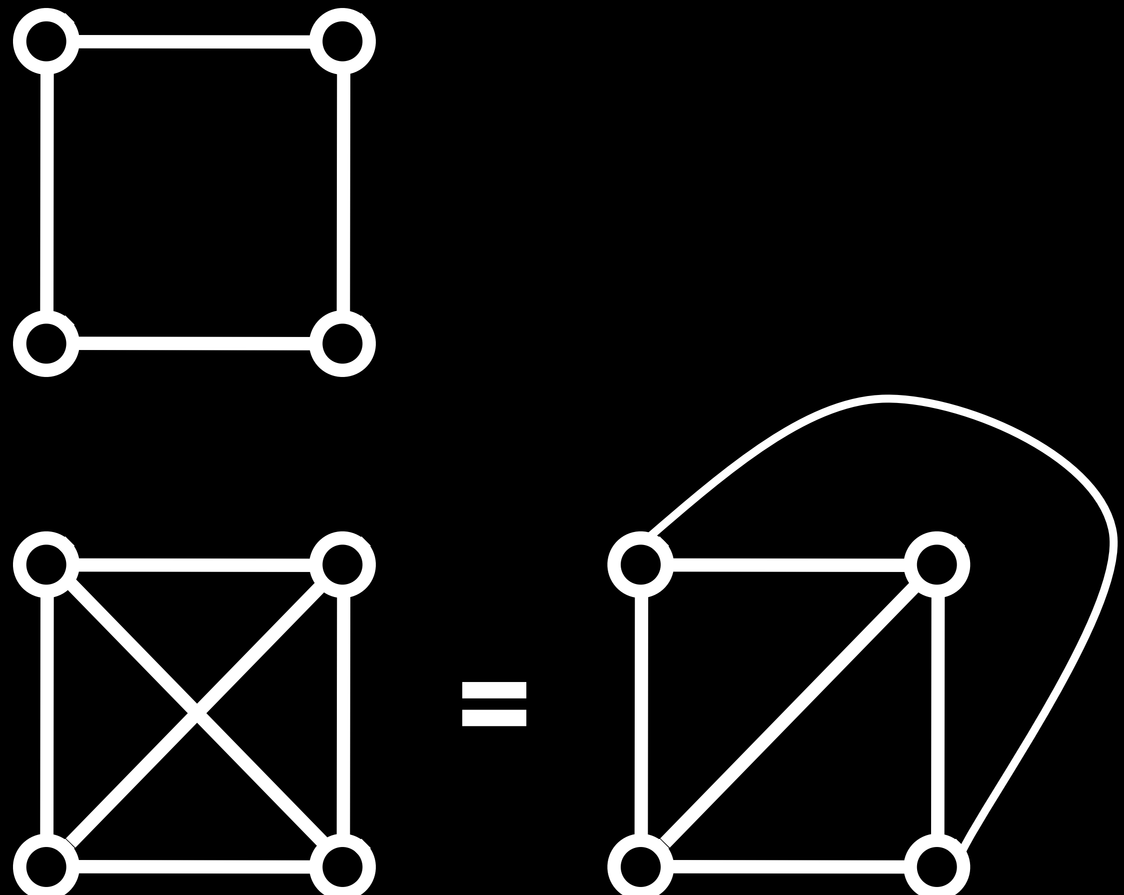


Examples of Planar Graphs



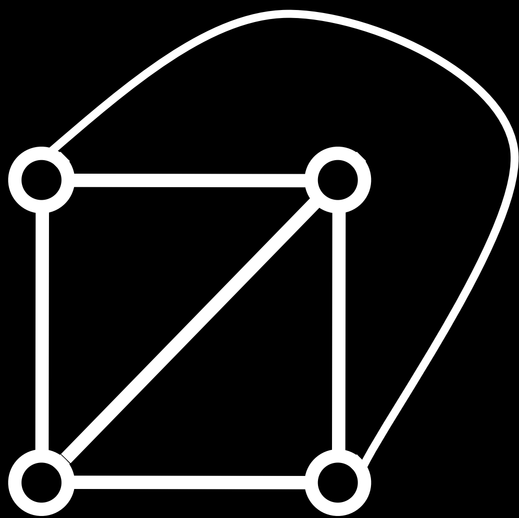


Examples of Planar Graphs



http://www.planarity.net

Faces

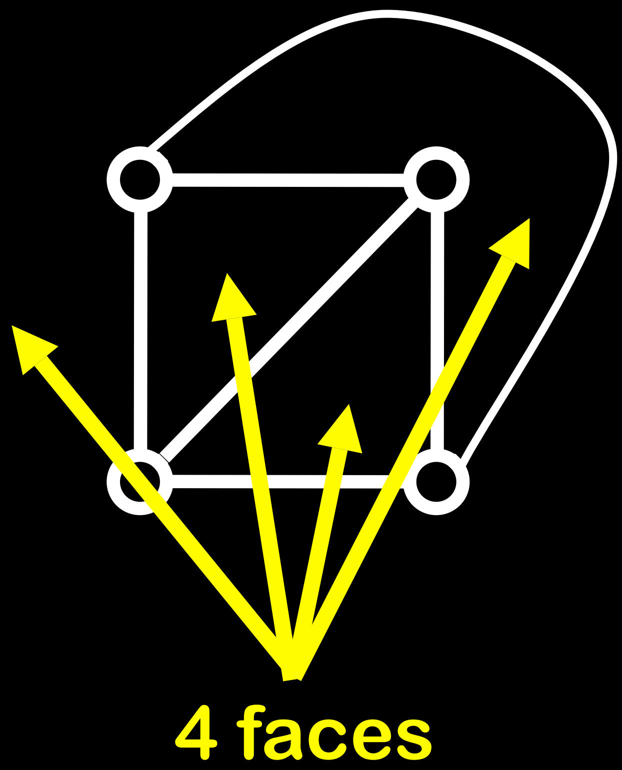


Faces

A planar graph splits the plane into disjoint faces

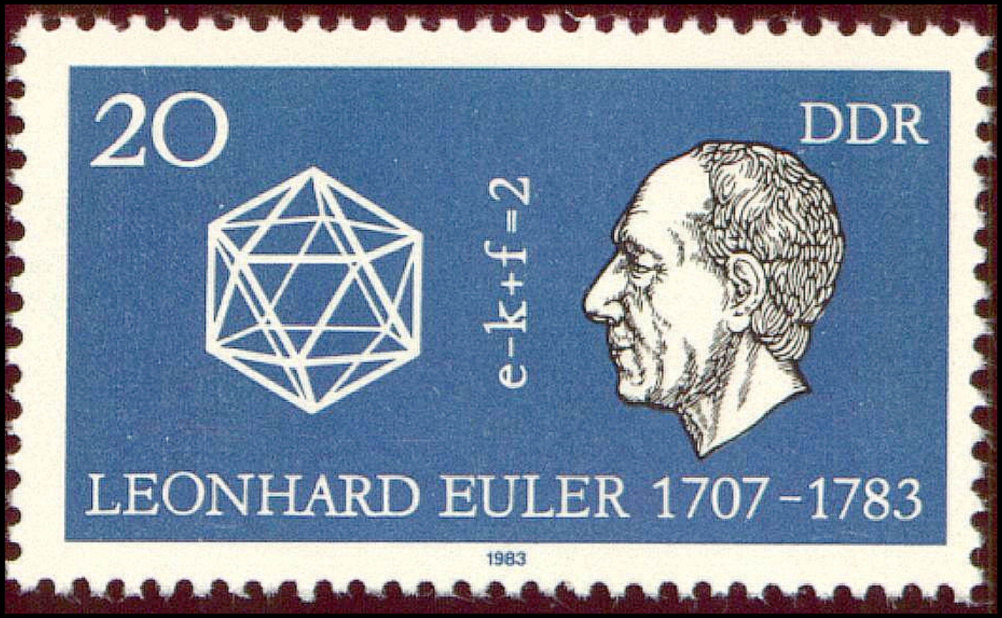
Faces

A planar graph splits the plane into disjoint faces



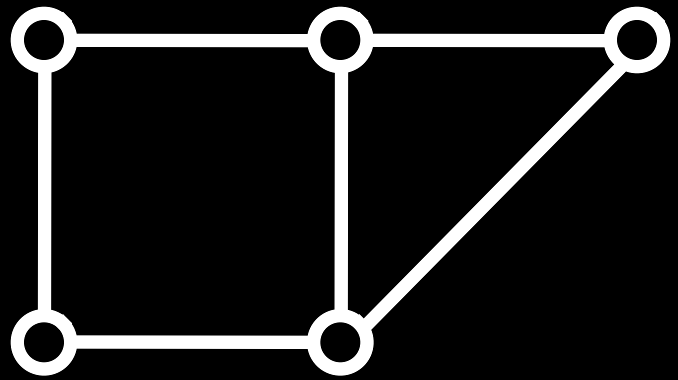
Euler’s Formula

If G is a connected planar graph with n vertices, e edges and f faces, then $n - e + f = 2$

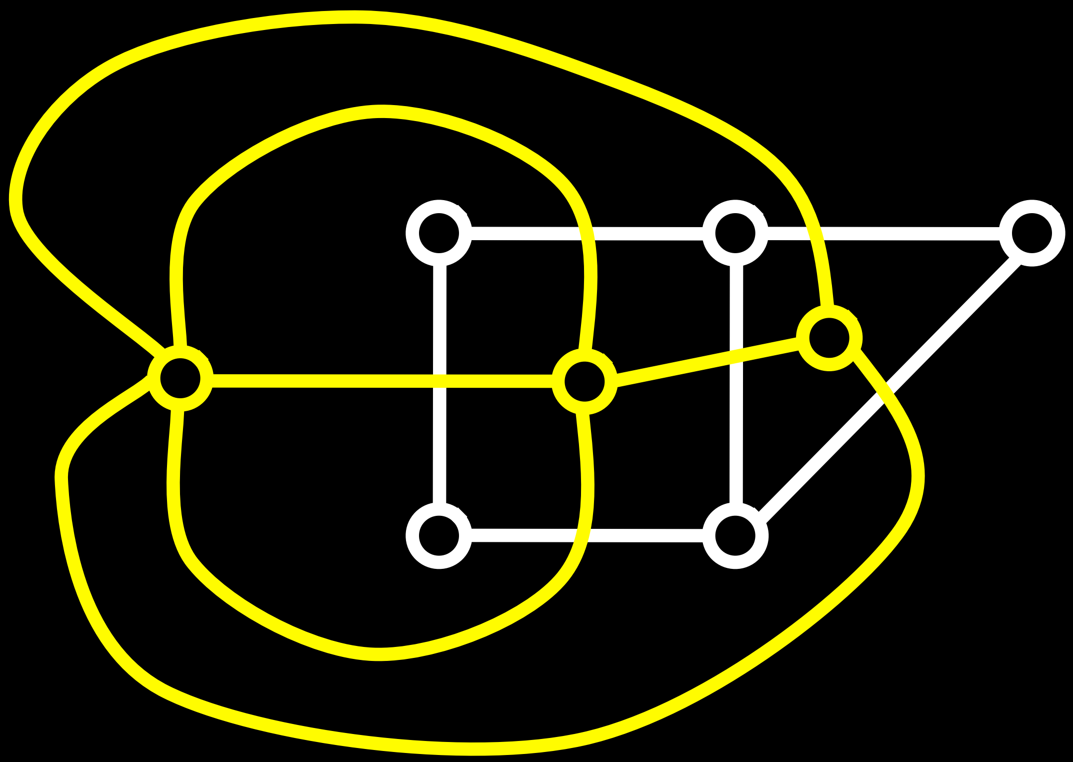


Rather than using induction, we'll use the important notion of the dual graph

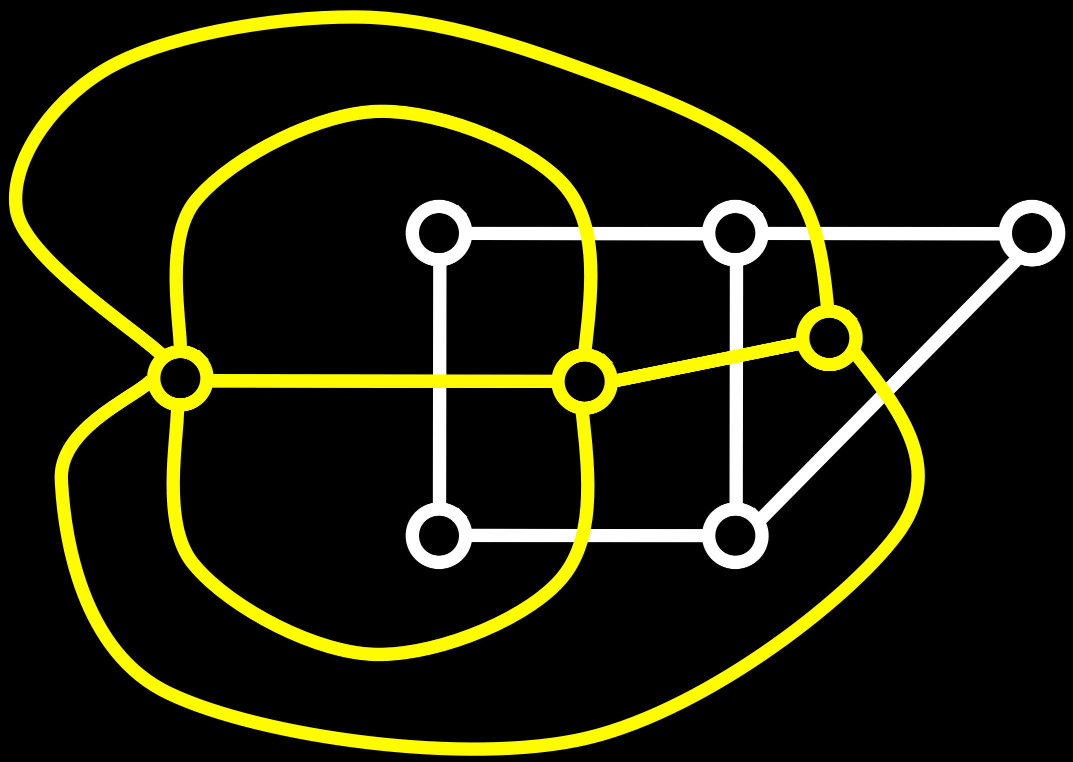
Rather than using induction, we'll use the important notion of the dual graph



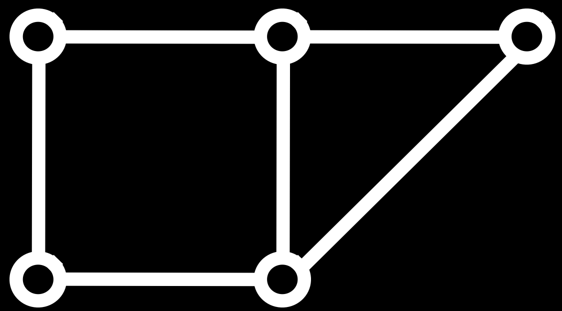
Rather than using induction, we'll use the important notion of the dual graph

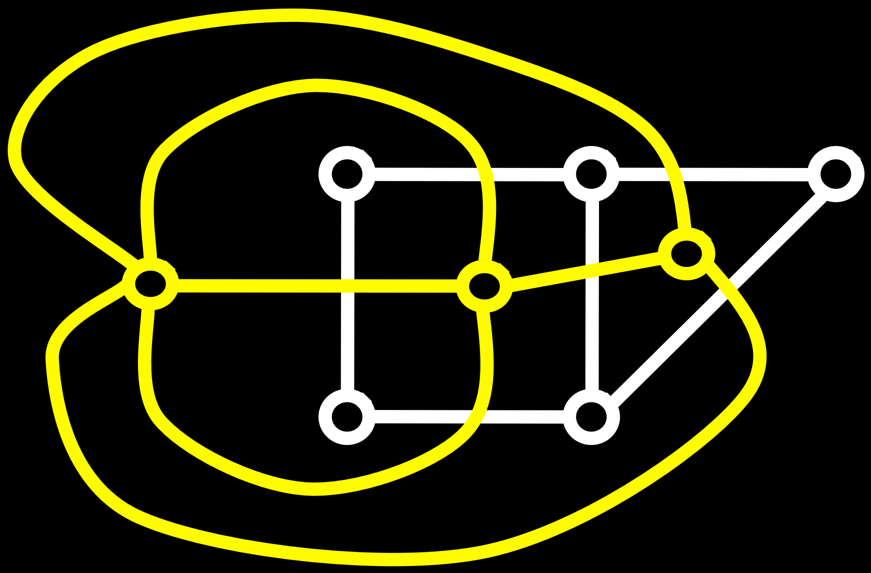


Rather than using induction, we'll use the important notion of the **dual graph**

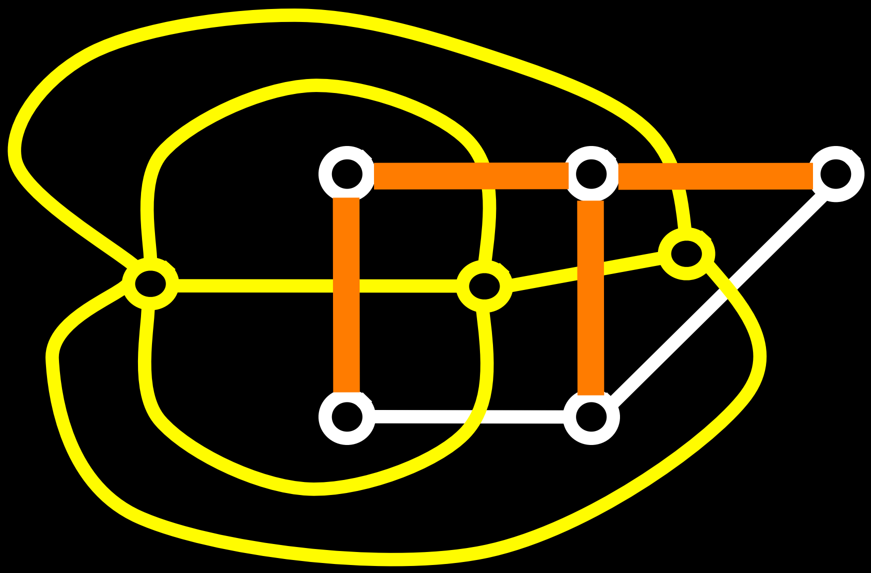


Dual = put a node in every face, and an edge between every adjacent face



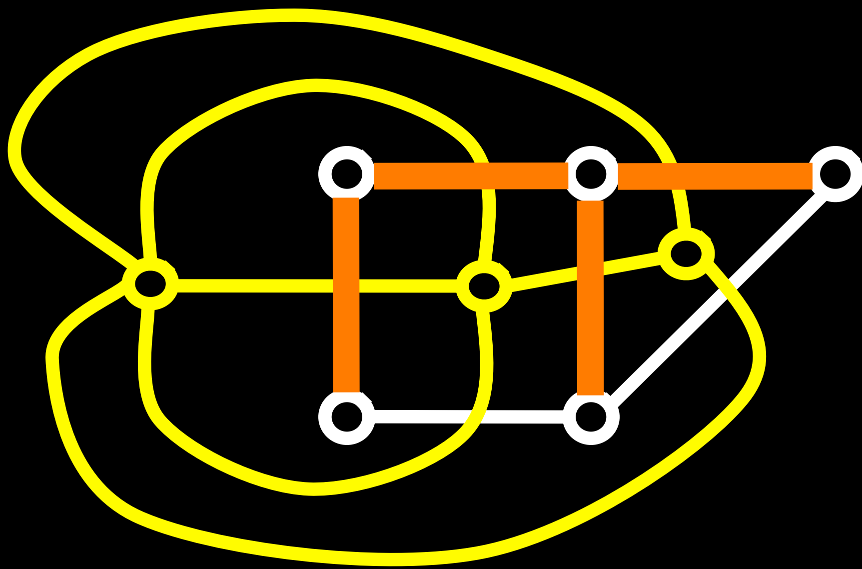


Let G* be the dual graph of G



Let G* be the dual graph of G

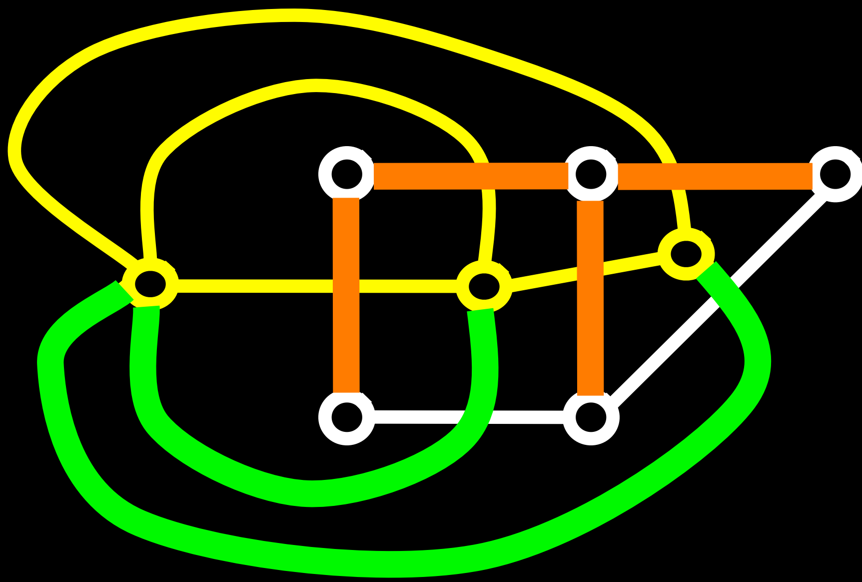
Let T be a spanning tree of G



Let G* be the dual graph of G

Let T be a spanning tree of G

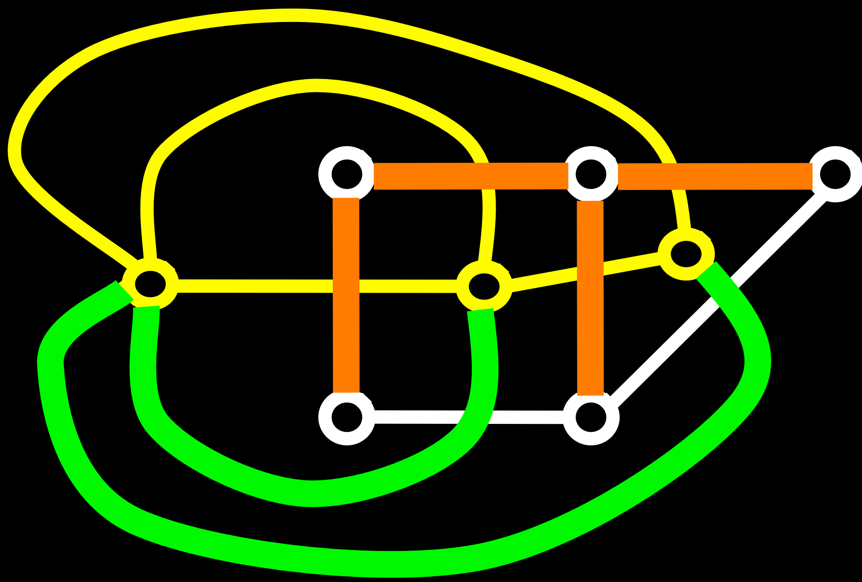
Let T* be the graph where there is an edge in dual graph for each edge in G – T



Let G* be the dual graph of G

Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

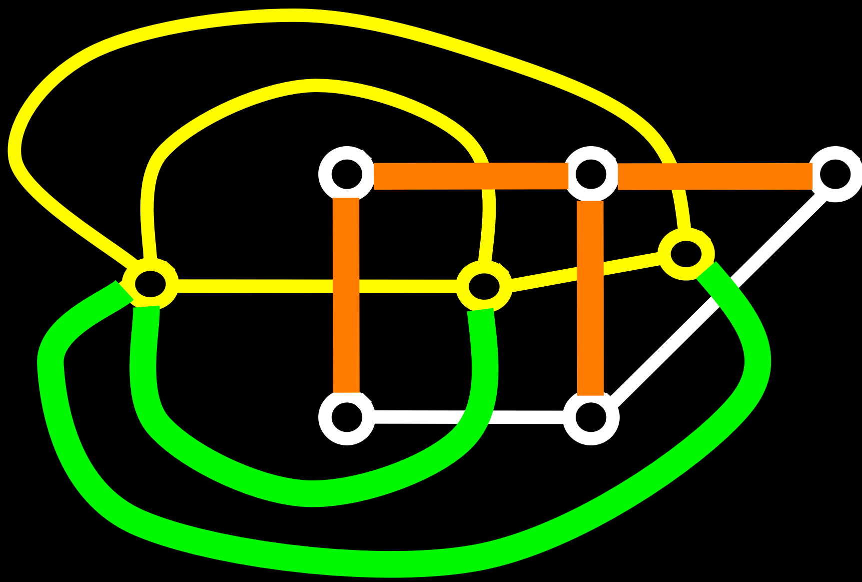


Let G* be the dual graph of G

Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

Then T* is a spanning tree for G*



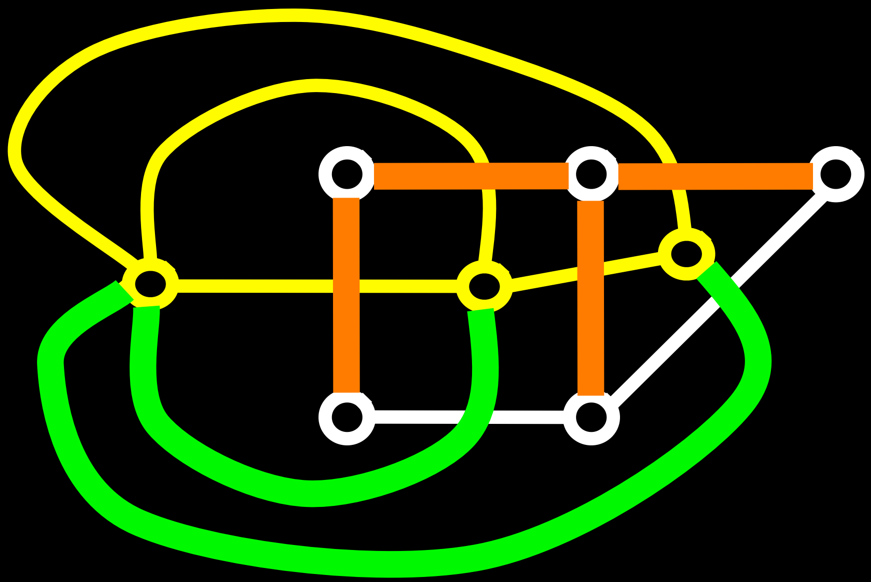
Let G* be the dual graph of G

Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

Then T* is a spanning tree for G*

$$n = e_T + 1$$



Let G* be the dual graph of G

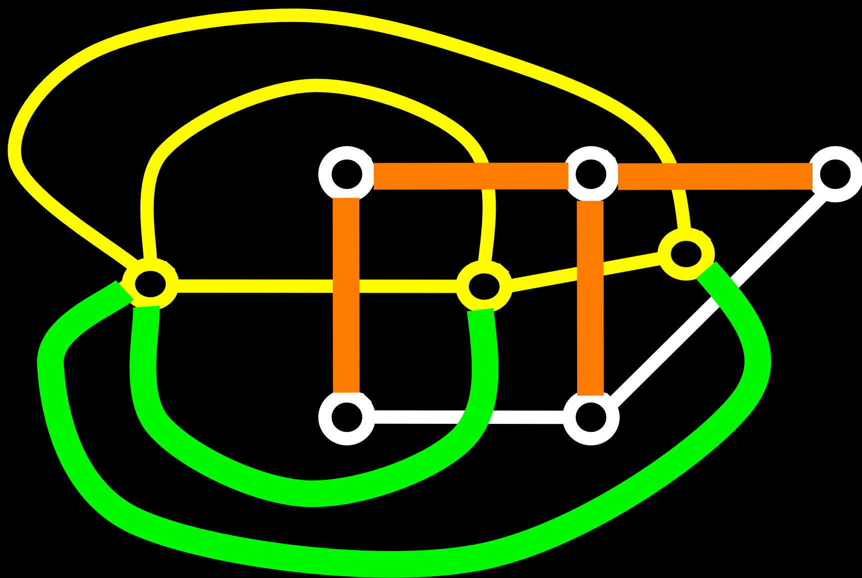
Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

Then T* is a spanning tree for G*

$$n = e_T + 1$$

$$f = e_{T*} + 1$$



Let G* be the dual graph of G

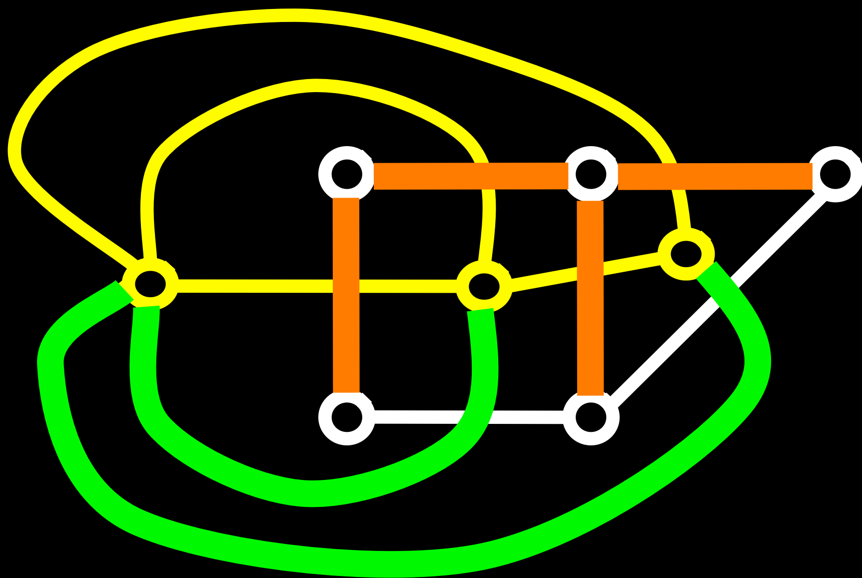
Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

Then T* is a spanning tree for G*

$$n = e_T + 1 \qquad n + f = e_T + e_{T^*} + 2$$

$$f = e_{T^*} + 1$$



Let G* be the dual graph of G

Let T be a spanning tree of G

Let T* be the graph where there is an edge in dual graph for each edge in G – T

Then T* is a spanning tree for G*

$$n = e_T + 1 \qquad n + f = e_T + e_{T^*} + 2$$

$$f = e_{T^*} + 1 \qquad = e + 2$$

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Proof of 1:

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Proof of 1:

In any graph, (sum of degrees) = $2e$

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Proof of 1:

In any graph, (sum of degrees) = $2e$

Assume all vertices have degree ≥ 6

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5

2. G has at most $3n - 6$ edges

Proof of 1:

In any graph, (sum of degrees) = $2e$

Assume all vertices have degree ≥ 6

Then $e \geq 3n$

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Proof of 1:

In any graph, (sum of degrees) = $2e$

Assume all vertices have degree ≥ 6

Then $e \geq 3n$

Furthermore, since G is simple, $3f \leq 2e$

Corollary: Let G be a simple planar graph with $n > 2$ vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most $3n - 6$ edges

Proof of 1:

In any graph, (sum of degrees) = $2e$

Assume all vertices have degree ≥ 6

Then $e \geq 3n$

Furthermore, since G is simple, $3f \leq 2e$

So $3n + 3f \leq 3e \Rightarrow 3(n-e+f) \leq 0$, contradiction.

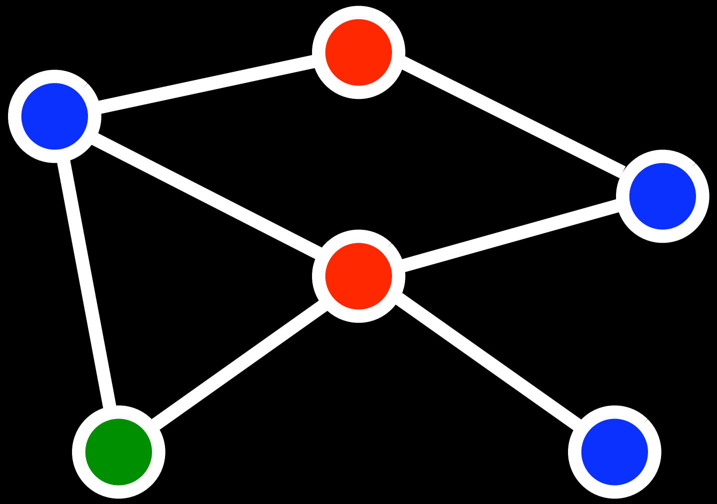
Graph Coloring

Graph Coloring

A coloring of a graph is an assignment of a color to each vertex such that no neighboring vertices have the same color

Graph Coloring

A coloring of a graph is an assignment of a color to each vertex such that no neighboring vertices have the same color



Graph Coloring

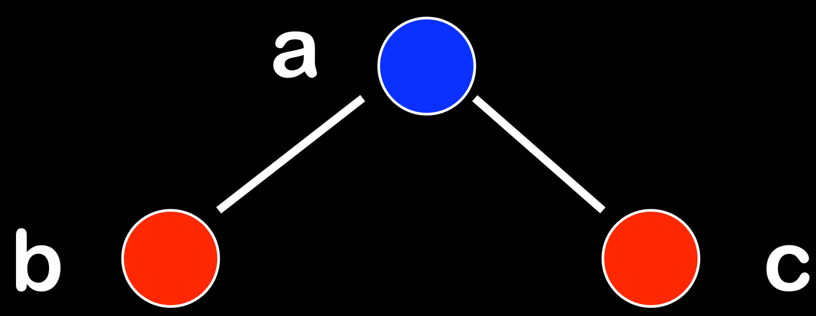
Arises surprisingly often in CS

Graph Coloring

Arises surprisingly often in CS

Register allocation: assign temporary variables to registers for scheduling instructions. Variables that interfere, or are simultaneously active, cannot be assigned to the same register

Instructions	Live variables
	a
b = a+2	
	a,b
c = b*b	
	a,c
b = c+1	
	a,b
return a*b	



Theorem: Every planar graph can be 6-colored

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Assume every planar graph with less than n vertices can be 6-colored

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Assume every planar graph with less than n vertices can be 6-colored

Assume G has n vertices

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Assume every planar graph with less than n vertices can be 6-colored

Assume G has n vertices

Since G is planar, it has some node v with degree at most 5

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Assume every planar graph with less than n vertices can be 6-colored

Assume G has n vertices

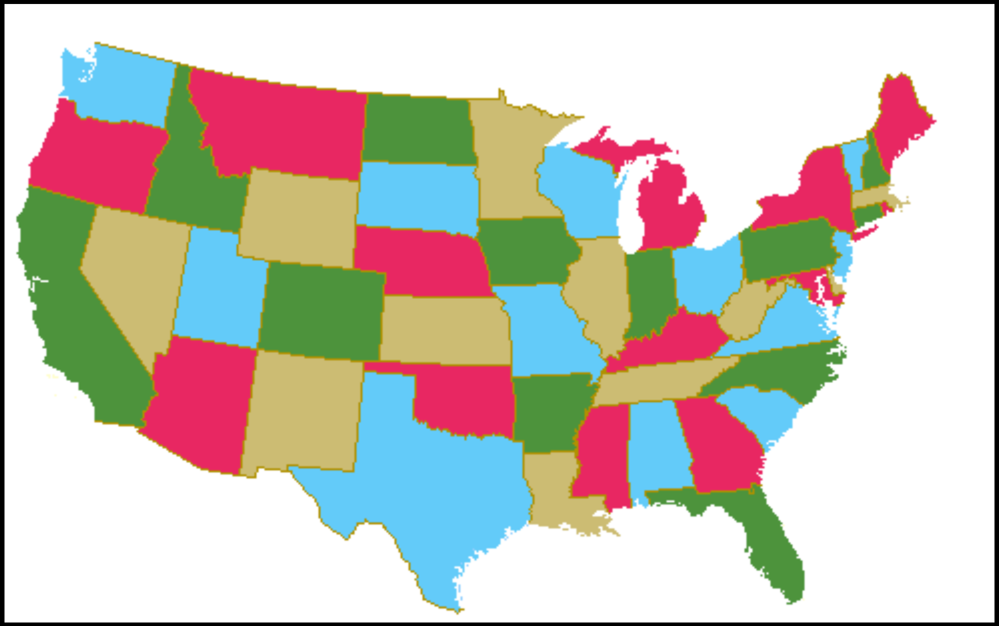
Since G is planar, it has some node v with degree at most 5

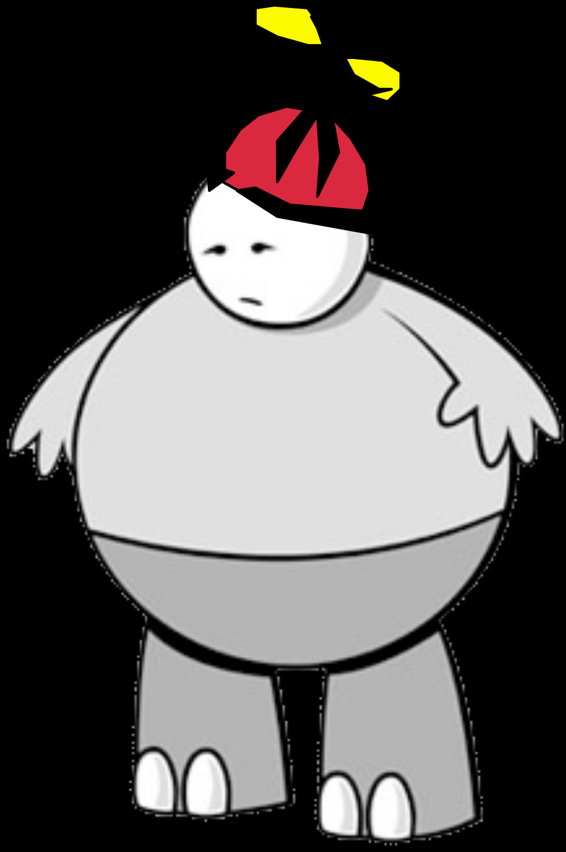
Remove v and color by Induction Hypothesis

Not too difficult to give an inductive proof of 5-colorability, using same fact that some vertex has degree ≤ 5

Not too difficult to give an inductive proof of 5-colorability, using same fact that some vertex has degree ≤ 5

4-color theorem remains challenging!





Here's What You Need to Know…

Trees

- Counting Trees
- Different Characterizations
- Cayley's formula

Planar Graphs

- Definition
- Euler's Theorem
- Coloring Planar Graphs