CSCI 5451 Fall 2015

Week 6 Notes

Professor Ellen Gethner

September 20, 2015

Graph Theory and Graph Algorithms

- Last week, we solved the party problem at the whiteboard:

Graph Theory and Graph Algorithms

- Last week, we solved the party problem at the whiteboard:
- Among any six people, there are always three who are all strangers or are all friends.

Graph Theory and Graph Algorithms

- Last week, we solved the party problem at the whiteboard:
- Among any six people, there are always three who are all strangers or are all friends.
- Let's learn enough tools to model other problems by way of graph theory.

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,
- together with a set of **edges**, $E(G)$ such that

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,
- together with a set of **edges**, $E(G)$ such that
- the elements of the set $E(G)$ are either ordered or unordered pairs of vertices.

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,
- together with a set of **edges**, $E(G)$ such that
- the elements of the set $E(G)$ are either ordered or unordered pairs of vertices.
- That is, elements of $E(G)$ look like (v_i, v_j) (or simply $v_i v_j$) where $v_i, v_j \in V(G)$.

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,
- together with a set of **edges**, $E(G)$ such that
- the elements of the set $E(G)$ are either ordered or unordered pairs of vertices.
- That is, elements of $E(G)$ look like (v_i, v_j) (or simply $v_i v_j$) where $v_i, v_j \in V(G)$.
- If the elements of $E(G)$ are ordered pairs, then G is called a **directed graph**.

Graph Theory: Some Basics

- A **finite graph** G is a set of **vertices**, $V(G) = \{v_1, v_2, \ldots, v_n\}$,
- together with a set of **edges**, $E(G)$ such that
- the elements of the set $E(G)$ are either ordered or unordered pairs of vertices.
- That is, elements of $E(G)$ look like (v_i, v_j) (or simply $v_i v_j$) where $v_i, v_j \in V(G)$.
- If the elements of $E(G)$ are ordered pairs, then G is called a **directed graph**.
- If the elements of $E(G)$ are unordered pairs, then G is called an **undirected graph**.

Graph Theory: Some Basics

- **Vocabulary.** If $v_i v_j \in E(G)$ then

Graph Theory: Some Basics

- **Vocabulary.** If $v_i v_j \in E(G)$ then
- we say that vertex v_i is **adjacent** to vertex v_j.

Graph Theory: Some Basics

- **Vocabulary.** If $v_i v_j \in E(G)$ then
- we say that vertex v_i is **adjacent** to vertex v_j.
- Alternatively, we say that v_i and v_j are **neighbors**.

Graph Theory Basics, Example

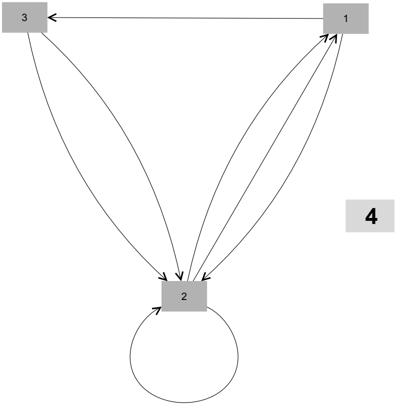
- **Example of a Directed Graph:** $V(G) = \{1, 2, 3, 4\}$ and

Graph Theory Basics, Example

- **Example of a Directed Graph:** $V(G) = \{1, 2, 3, 4\}$ and
- $E(G) = \{(2, 1), (1, 3), (1, 2), (2, 1), (2, 2), (3, 2), (3, 2)\}$.

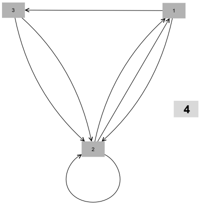
Graph Theory Basics, Example

- **Example of a Directed Graph:** $V(G) = \{1, 2, 3, 4\}$ and
- $E(G) = \{(2, 1), (1, 3), (1, 2), (2, 1), (2, 2), (3, 2), (3, 2)\}$.

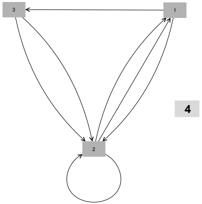


-

Example, continued

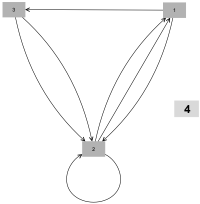


Example, continued

- 

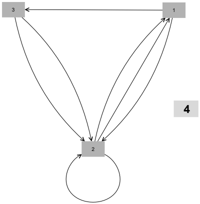
- **Facts.**

Example, continued



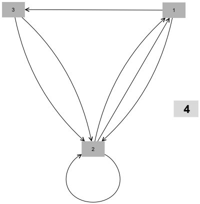
- **Facts.**
- The above graph has two **connected components**.

Example, continued



- **Facts.**
- The above graph has two **connected components**.
- And edge of the form (v_i, v_i) (or $v_i v_i$) is called a **loop** and thus $(2, 2)$ is a loop above.

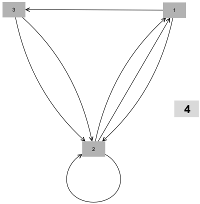
Example, continued



-

- **Facts.**
- The above graph has two **connected components**.
- And edge of the form (v_i, v_i) (or $v_i v_i$) is called a **loop** and thus $(2, 2)$ is a loop above.
- If edge $v_i v_j$ appears more than once in the set $E(G)$ then $v_i v_j$ is called a **multiple edge** (or multi-edge or parallel edge).

Example, continued



- **Facts.**
- The above graph has two **connected components**.
- And edge of the form (v_i, v_i) (or $v_i v_i$) is called a **loop** and thus $(2, 2)$ is a loop above.
- If edge $v_i v_j$ appears more than once in the set $E(G)$ then $v_i v_j$ is called a **multiple edge** (or multi-edge or parallel edge).
- Thus, for example, the edge (3,2) above is a multiple edge.

Graph Theory Basics, continued

- A graph is called **simple** if it has neither loops nor multiple edges.

Graph Theory Basics, continued

- A graph is called **simple** if it has neither loops nor multiple edges.
- In a simple, undirected graph, the use of arrows is unnecessary.

Graph Theory Basics, continued

- A graph is called **simple** if it has neither loops nor multiple edges.
- In a simple, undirected graph, the use of arrows is unnecessary.
- In particular, we consider an undirected edge to “allow” both directions.

Graph Theory Basics, continued

- A graph is called **simple** if it has neither loops nor multiple edges.
- In a simple, undirected graph, the use of arrows is unnecessary.
- In particular, we consider an undirected edge to “allow” both directions.
- Our discussion so far has relied on pictures to understand the meaning of a graph.

Graph Theory Basics, continued

- A graph is called **simple** if it has neither loops nor multiple edges.
- In a simple, undirected graph, the use of arrows is unnecessary.
- In particular, we consider an undirected edge to “allow” both directions.
- Our discussion so far has relied on pictures to understand the meaning of a graph.
- How do we represent a graph on a computer?

Representing a graph on a computer

- **Method 1.** The **adjacency matrix** $A = A(G)$ for a graph G in which $|V(G)| = n$ is an $n \times n$ matrix, which is defined as follows:

Representing a graph on a computer

- **Method 1.** The **adjacency matrix** $A = A(G)$ for a graph G in which $|V(G)| = n$ is an $n \times n$ matrix, which is defined as follows:

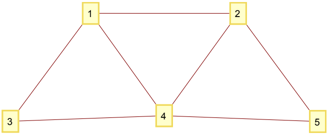
- First, denote the ijth entry of A by a_{ij}.

- For a *simple* graph G, define

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E(G) \\ 0 & \text{if } v_i v_j \notin E(G) \end{cases}$$

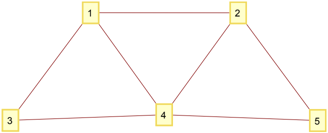
Adjacency Matrix example

- Here's a simple undirected graph on five vertices.



Adjacency Matrix example

- Here's a simple undirected graph on five vertices.



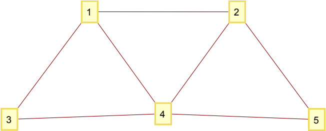
- And here is the associated adjacency matrix.

Out[13]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Adjacency Matrix example

- Here's a simple undirected graph on five vertices.



- And here is the associated adjacency matrix.

Out[13]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

- Note that a change in the numbering of the vertices will change the adjacency matrix.

Efficiency thoughts

- If the representation of a graph G on n vertices is an $n \times n$ adjacency matrix A, then

Efficiency thoughts

- If the representation of a graph G on n vertices is an $n \times n$ adjacency matrix A, then
- the run time of any algorithm that must read the contents of the matrix will be $\Omega(n^2)$.

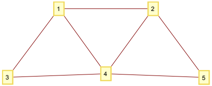
Efficiency thoughts

- If the representation of a graph G on n vertices is an $n \times n$ adjacency matrix A, then
- the run time of any algorithm that must read the contents of the matrix will be $\Omega(n^2)$.
- Thus, when G is **sparse** (which means $|E(G)|$ is relatively small) we should look for another way of storing G.

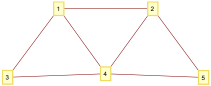
Another way of representing a graph on a computer

- **Method 2.** An **adjacency list** gives, for each vertex, its list of neighbors.

Another way of representing a graph on a computer

- **Method 2.** An **adjacency list** gives, for each vertex, its list of neighbors.
- 

Another way of representing a graph on a computer

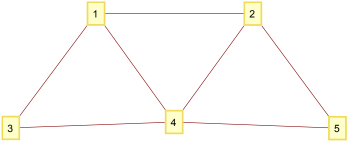
- **Method 2.** An **adjacency list** gives, for each vertex, its list of neighbors.
- 

- 1. 2,3,4
  2. 1,4,5
  3. 1,4
  4. 1,2,3,5
  5. 2,4

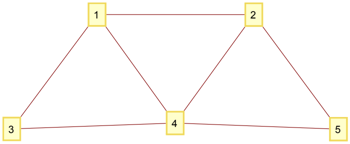
Basics, continued

- The **degree** of a vertex v_i, written $deg(v_i)$ (or sometimes d_i) is the number of vertices to which v_i is adjacent.

Basics, continued

- The **degree** of a vertex v_i, written $deg(v_i)$ (or sometimes d_i) is the number of vertices to which v_i is adjacent.
- 

Basics, continued

- The **degree** of a vertex v_i, written $deg(v_i)$ (or sometimes d_i) is the number of vertices to which v_i is adjacent.
- 

- In this example, $deg(1){=}3$, $deg(2){=}3$, $deg(3){=}2$, $deg(4){=}4$, and $deg(5){=}2$.

A Theorem

- **Theorem.** For any graph G, we have $\sum_{v \in V(G)} deg(v) = 2|E(G)|$.

A Theorem

- **Theorem.** For any graph G, we have $\sum_{v \in V(G)} deg(v) = 2|E(G)|$.
- Why??

A Theorem

- **Theorem.** For any graph G, we have $\sum_{v \in V(G)} deg(v) = 2|E(G)|$.
- Why??
- Hence the size of the adjacency list is $2|E(G)|$.

List versus Matrix

- Although the adjacency list may often be more convenient and efficient,

List versus Matrix

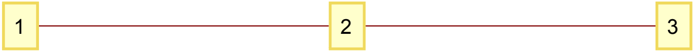
- Although the adjacency list may often be more convenient and efficient,
- the adjacency matrix has special, sometimes magical, properties that make it preferable to use.

List versus Matrix

- Although the adjacency list may often be more convenient and efficient,
- the adjacency matrix has special, sometimes magical, properties that make it preferable to use.
- Let's illustrate the latter by way of an example.

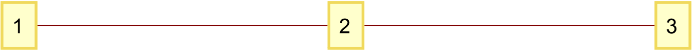
Magical property of adjacency matrix

- Start with graph $G =$



Magical property of adjacency matrix

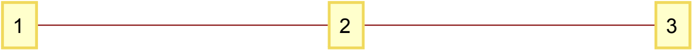
- Start with graph $G =$



- whose adjacency matrix is $A =$

Magical property of adjacency matrix

- Start with graph $G =$



- whose adjacency matrix is $A =$

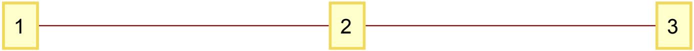
-

Out[16]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Magical property of adjacency matrix

- Start with graph $G =$



- whose adjacency matrix is $A =$

Out[16]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Consider the matrix
$$A^2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

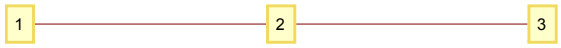
Magical property of adjacency matrix A

- **So far we have** $A^2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

Magical property of adjacency matrix A

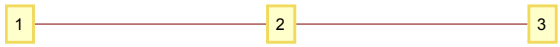
- **So far we have** $A^2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

- for graph $G =$



Magical property of adjacency matrix A

- **So far we have** $A^2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$
- for graph $G =$



- **Observations.**
  1. $A^2(1,1) = 1$ and there is exactly one path of length 2 from vertex 1 to vertex 1, namely $1 \to 2 \to 1$.
  2. $A^2(2,2) = 2$ and there are exactly two paths of length 2 from vertex 2 to vertex 2, namely $2 \to 3 \to 2$ and $2 \to 1 \to 2$.
  3. **ETC**

Magical property of adjacency matrix A

- Theorem
  Let G be any simple graph on n vertices with $(n \times n)$ adjacency matrix A. Then $A^k(i,j)$ is the number of paths of length k from vertex i to vertex j for all $1 \leq i, j \leq n$.

Magical property of adjacency matrix A

- Theorem
  Let G be any simple graph on n vertices with $(n \times n)$ adjacency matrix A. Then $A^k(i, j)$ is the number of paths of length k from vertex i to vertex j for all $1 \leq i, j \leq n$.

  - **Proof.** Take CSCI 5408 in the spring...

One more thing about the idea of the adjacency matrix

Planar Graphs

- We now enter the realm of structural graph theory, where we care about physical properties of graphs.

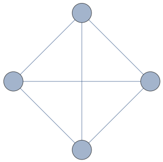
Planar Graphs

- We now enter the realm of structural graph theory, where we care about physical properties of graphs.

- **Definition.** A simple graph G is said to be **planar** if G can be drawn in the plane with no edge crossings except, possibly, at endpoints.

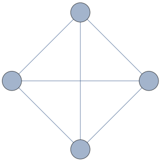
Planar Graphs

- We now enter the realm of structural graph theory, where we care about physical properties of graphs.
- **Definition.** A simple graph G is said to be **planar** if G can be drawn in the plane with no edge crossings except, possibly, at endpoints.
- **Example 1.** $G =$



Planar Graphs

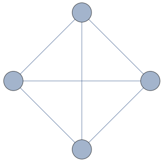
- We now enter the realm of structural graph theory, where we care about physical properties of graphs.
- **Definition.** A simple graph G is said to be **planar** if G can be drawn in the plane with no edge crossings except, possibly, at endpoints.
- **Example 1.** $G =$



- Is G planar??

Examples

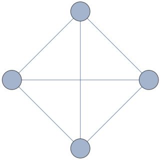
- **Example 1.** $G =$



- Is G planar??

Examples

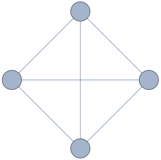
- **Example 1.** $G =$



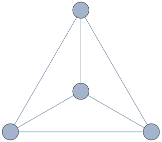
- Is G planar??
- Yes, because $G =$

Examples

- **Example 1.** $G =$



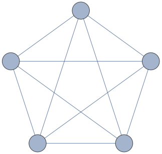
- Is G planar??
- Yes, because $G =$



-

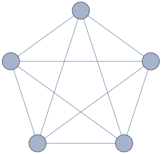
Planarity Examples, continued

- $G =$



Planarity Examples, continued

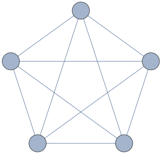
- $G =$



- Is G planar??

Planarity Examples, continued

- $G =$



- Is G planar??
- **Exercise.**

A Tool for Planar Graphs

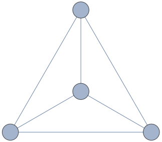
- **Euler's Theorem.** Let G be a connected planar graph that has been embedded in the plane with no edge crossings (except, possibly, at endpoints). Suppose $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$, where $F(G)$ is the number of faces of G in the embedding. Then $V - E + F = 2$.

Planarity and Euler's Formula

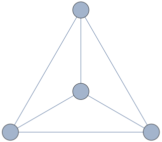
- **Example.** G is planar (and drawn without edge crossings below).

Planarity and Euler's Formula

- **Example.** G is planar (and drawn without edge crossings below).

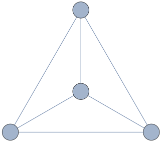
- 

Planarity and Euler's Formula

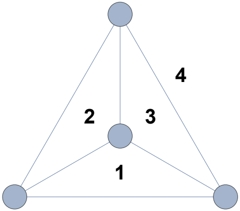
- **Example.** G is planar (and drawn without edge crossings below).
- 
- Note that $V = 4, E = 6$, and $F = 4$ (not 3 !!!!).

Planarity and Euler's Formula

- **Example.** G is planar (and drawn without edge crossings below).

- 

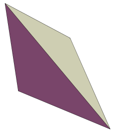
- Note that $V = 4$, $E = 6$, and $F = 4$ (not 3 !!!!).

- 

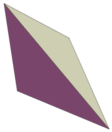
Euler's Formula Example

- Note that the outer infinite region is considered to be a face of any planar embedding of a planar graph.

Euler's Formula Example

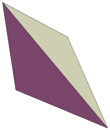
- Note that the outer infinite region is considered to be a face of any planar embedding of a planar graph.
- 

Euler's Formula Example

- Note that the outer infinite region is considered to be a face of any planar embedding of a planar graph.
- 
- Our current example is that of a flattened tetrahedron, which has four faces.

Euler's Formula Example

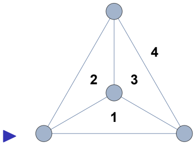
- Note that the outer infinite region is considered to be a face of any planar embedding of a planar graph.

- 

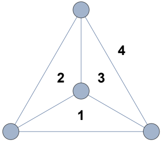
- Our current example is that of a flattened tetrahedron, which has four faces.

- So don't forget to count that infinite outer face when using Euler's formula.

Back to the example

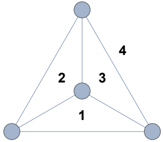


Back to the example

- 

- $V - E + F = 4 - 6 + 4 = 2.$

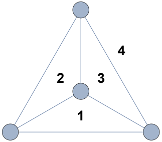
Back to the example

- 

- $V - E + F = 4 - 6 + 4 = 2.$

- Whew, the formula works in our example.

Back to the example

- 

- $V - E + F = 4 - 6 + 4 = 2.$

- Whew, the formula works in our example.

- For a proof of Euler's Theorem, take CSCI 5408 in the spring.

More Examples on Whiteboard

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.
- **Proof.** Suppose G has been embedded in the plane without edge crossings and that $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$.

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.

- **Proof.** Suppose G has been embedded in the plane without edge crossings and that $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$.

- **1.** By Euler's Theorem we have $V - E + F = 2$.

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.

- **Proof.** Suppose G has been embedded in the plane without edge crossings and that $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$.

- **1.** By Euler's Theorem we have $V - E + F = 2$.

- **2.** Suppose $d_1, d_2, \ldots, d_V$ are the degrees of the V vertices of G.

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.
- **Proof.** Suppose G has been embedded in the plane without edge crossings and that $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$.
- **1.** By Euler's Theorem we have $V - E + F = 2$.
- **2.** Suppose $d_1, d_2, \ldots, d_V$ are the degrees of the V vertices of G.
- Then $\sum_{i=1}^{n} d_i = 2E$ (why?)

Another Planarity Tool

- **Lemma A.** In every simple planar graph G there is a vertex of degree 5 or less.
- **Proof.** Suppose G has been embedded in the plane without edge crossings and that $V = |V(G)|$, $E = |E(G)|$, and $F = |F(G)|$.
- **1.** By Euler's Theorem we have $V - E + F = 2$.
- **2.** Suppose $d_1, d_2, \ldots, d_V$ are the degrees of the V vertices of G.
- Then $\sum_{i=1}^{n} d_i = 2E$ (why?)
- **3.** Now $3F \leq 2E$ because each face of G has three or more edges.

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.
- Then since G is simple, $f_i \geq 3$ for $i = 1, 2, \ldots, F$.

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.

- Then since G is simple, $f_i \geq 3$ for $i = 1, 2, \ldots, F$.

- That is, a face of size 1 or 2 would come from a loop or a multiple edge, which can't happen since G is simple.

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.

- Then since G is simple, $f_i \geq 3$ for $i = 1, 2, \ldots, F$.

- That is, a face of size 1 or 2 would come from a loop or a multiple edge, which can't happen since G is simple.

- Thus $f_1 + f_2 + \cdots + f_F = 2E$ because each edge borders exactly two faces.

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.
- Then since G is simple, $f_i \geq 3$ for $i = 1, 2, \ldots, F$.
- That is, a face of size 1 or 2 would come from a loop or a multiple edge, which can't happen since G is simple.
- Thus $f_1 + f_2 + \cdots + f_F = 2E$ because each edge borders exactly two faces.
- In that case $3 + 3 + \cdots + 3(F \ times) \leq 2E$

More explanation about $3F \leq 2E$

- Suppose $f_1, f_2, \ldots, f_F$ are the sizes of the F faces of embedded graph G.
- Then since G is simple, $f_i \geq 3$ for $i = 1, 2, \ldots, F$.
- That is, a face of size 1 or 2 would come from a loop or a multiple edge, which can't happen since G is simple.
- Thus $f_1 + f_2 + \cdots + f_F = 2E$ because each edge borders exactly two faces.
- In that case $3 + 3 + \cdots + 3(F \ \textit{times}) \leq 2E$
- $\Rightarrow 3F \leq 2E$, as desired.

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:
- $V - E + F = 2$

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:
- $V - E + F = 2$
- $\Rightarrow V - E + \frac{2}{3}E \geq 2$

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:
- $V - E + F = 2$
- $\Rightarrow V - E + \frac{2}{3}E \geq 2$
- $\Rightarrow V - \frac{1}{3}E \geq 2$

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:
- $V - E + F = 2$
- $\Rightarrow V - E + \frac{2}{3}E \geq 2$
- $\Rightarrow V - \frac{1}{3}E \geq 2$
- $\Rightarrow 6V - 2E \geq 12$

Proof of Lemma A, continued

- But $\Rightarrow 3F \leq 2E \Rightarrow F \leq \frac{2}{3}E \qquad (*)$
- Subsitutue $(*)$ into Euler's Formula:
- $V - E + F = 2$
- $\Rightarrow V - E + \frac{2}{3}E \geq 2$
- $\Rightarrow V - \frac{1}{3}E \geq 2$
- $\Rightarrow 6V - 2E \geq 12$
- $\Rightarrow 2E \leq 6V - 12.$

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is
- $\frac{\sum_{i=1}^{V} d_i}{V} = \frac{2E}{V}$.

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is
- $\frac{\sum_{i=1}^{V} d_i}{V} = \frac{2E}{V}$.
- Combining the above with the fact that $2E \leq 6V - 12$ yields

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is
- $\frac{\sum_{i=1}^{V} d_i}{V} = \frac{2E}{V}$.
- Combining the above with the fact that $2E \leq 6V - 12$ yields
- $\frac{2E}{V} \leq \frac{6V-12}{V} = 6 - \frac{12}{V}$.

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is
- $\frac{\sum_{i=1}^{V} d_i}{V} = \frac{2E}{V}$.
- Combining the above with the fact that $2E \leq 6V - 12$ yields
- $\frac{2E}{V} \leq \frac{6V-12}{V} = 6 - \frac{12}{V}$.
- But since $V > 0$ we have $6 - \frac{12}{V} < 6$, which means

Proof of Lemma A, continued

- So far we have $2E \leq 6V - 12$
- Note that the average degree of a vertex in G is
- $\frac{\sum_{i=1}^{V} d_i}{V} = \frac{2E}{V}$.
- Combining the above with the fact that $2E \leq 6V - 12$ yields
- $\frac{2E}{V} \leq \frac{6V-12}{V} = 6 - \frac{12}{V}$.
- But since $V > 0$ we have $6 - \frac{12}{V} < 6$, which means
- graph G must have a vertex of degree 5 or less. **QED**

Summary of Lemma A

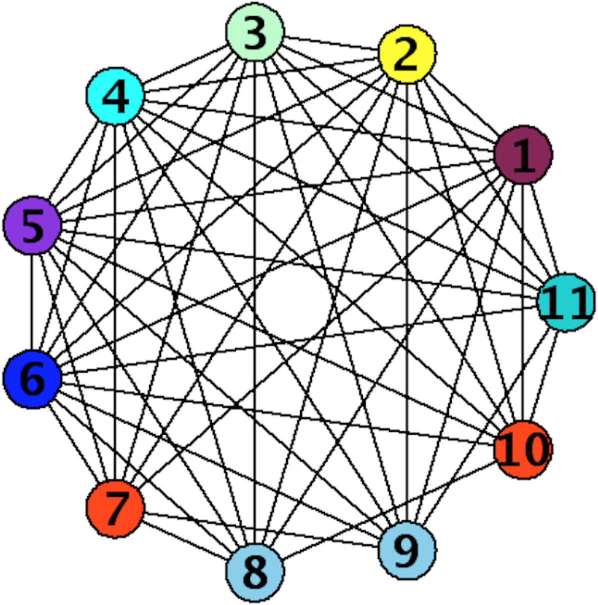
- **Every simple planar graph has a vertex of degree 5 or less.**

Summary of Lemma A

- **Every simple planar graph has a vertex of degree 5 or less.**

- **Exercise:** why must we assume that G is simple in the hypotheses of Lemma A?

Next Topic: Graph Coloring



Graph Coloring Basics

- A k-**coloring** of a simple graph G is an assignment of all colors $1, 2, \ldots, k$ to the vertices of G.

Graph Coloring Basics

- A k-**coloring** of a simple graph G is an assignment of all colors $1, 2, \ldots, k$ to the vertices of G.
- A **proper** k-coloring of a simple graph G is an assignment of all colors $1, 2, \ldots, k$ to the vertices of G such that no two adjacent vertices receive the same color.

Graph Coloring Basics

- A k-**coloring** of a simple graph G is an assignment of all colors $1, 2, \ldots, k$ to the vertices of G.
- A **proper** k-coloring of a simple graph G is an assignment of all colors $1, 2, \ldots, k$ to the vertices of G such that no two adjacent vertices receive the same color.
- The **chromatic number** of G, written $\chi(G)$, is the smallest value of k for which G has a proper k-coloring.

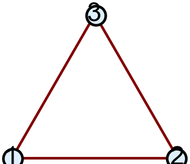
Graph Coloring Basics

- A k-**coloring** of a simple graph G is an assignment of all colors $1, 2, \dots, k$ to the vertices of G.

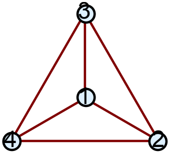
- A **proper** k-coloring of a simple graph G is an assignment of all colors $1, 2, \dots, k$ to the vertices of G such that no two adjacent vertices receive the same color.

- The **chromatic number** of G, written $\chi(G)$, is the smallest value of k for which G has a proper k-coloring.

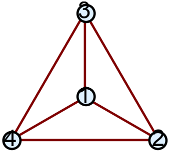
- **Examples.**

χ [] =

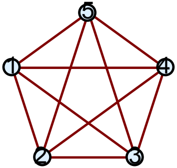
Chromatic Number $\chi(G)$ Examples, continued

χ [] = ?

Chromatic Number $\chi(G)$ Examples, continued

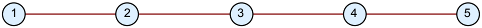
- $\chi\Big[$ 

$\Big] =$?

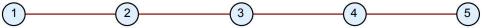
- $\chi\Big[$ 

$\Big] =$?

Chromatic Number $\chi(G)$ Examples, continued

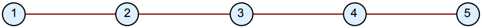
- x[] = ?

Chromatic Number $\chi(G)$ Examples, continued

- x[] = ?

- Let P be any path of length ≥ 2. What is $\chi(P)$?

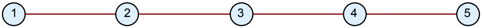
Chromatic Number $\chi(G)$ Examples, continued

- χ[] = ?

- Let P be any path of length ≥ 2. What is $\chi(P)$?

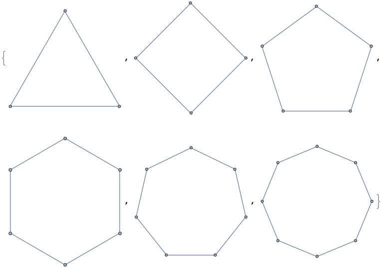
- For $n \geq 3$, a cycle of length n, denoted by C_n, is an undirected path of vertices $v_1, v_2, \ldots, v_n$ together with edge $v_1 v_n$.

Chromatic Number $\chi(G)$ Examples, continued

- $x[$  $] = ?$

- Let P be any path of length ≥ 2. What is $\chi(P)$?

- For $n \geq 3$, a cycle of length n, denoted by C_n, is an undirected path of vertices $v_1, v_2, \ldots, v_n$ together with edge $v_1 v_n$.

Out[35]= 

Chromatic Number of C_n

- What is $\chi(C_3)$?

Chromatic Number of C_n

- What is $\chi(C_3)$?

- What is $\chi(C_4)$?

Chromatic Number of C_n

- What is $\chi(C_3)$?
- What is $\chi(C_4)$?
- What is $\chi(C_5)$?

Chromatic Number of C_n

- What is $\chi(C_3)$?
- What is $\chi(C_4)$?
- What is $\chi(C_5)$?
- $\vdots$

Chromatic Number of C_n

- What is $\chi(C_3)$?
- What is $\chi(C_4)$?
- What is $\chi(C_5)$?
- $\vdots$
- What is $\chi(C_n)$?

Chromatic Number of C_n

- What is $\chi(C_3)$?
- What is $\chi(C_4)$?
- What is $\chi(C_5)$?
- $\vdots$
- What is $\chi(C_n)$?
- Prove that your answer is correct.

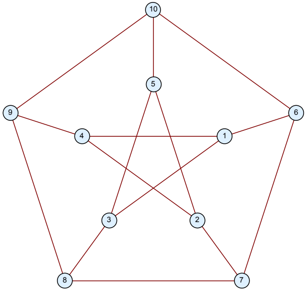
Last chromatic number example for now

- The *Petersen Graph* P is shown below.

Last chromatic number example for now

- The *Petersen Graph P* is shown below.

Out[29]=

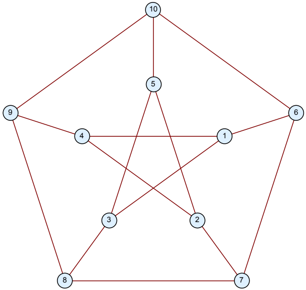


-

Last chromatic number example for now

- The *Petersen Graph* P is shown below.

Out[29]=



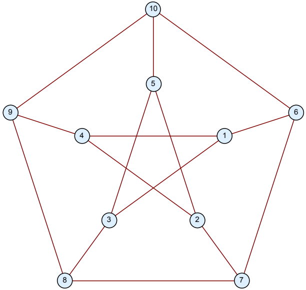
-

- What is $\chi(P)$?

Last chromatic number example for now

- The *Petersen Graph* P is shown below.

Out[29]=



-

- What is $\chi(P)$?
- Prove that your answer is correct.

Scheduling and Graph Coloring

- Graph coloring is a euphemism for *scheduling* or separating sets of vertices into *independent sets*, namely vertices that are mutually non-adjacent.

Scheduling and Graph Coloring

- Graph coloring is a euphemism for *scheduling* or separating sets of vertices into *independent sets*, namely vertices that are mutually non-adjacent.

- What follows is an example of how to avoid scheduling conflicts, and finding an optimal schedule.

Scheduling and Graph Coloring

- Graph coloring is a euphemism for *scheduling* or separating sets of vertices into *independent sets*, namely vertices that are mutually non-adjacent.

- What follows is an example of how to avoid scheduling conflicts, and finding an optimal schedule.

- **Problem.** How many days must be scheduled for committee meetings of Pariliament if every committee meets all day, and some members serve on more than one committee?

Scheduling and Graph Coloring

- Graph coloring is a euphemism for *scheduling* or separating sets of vertices into *independent sets*, namely vertices that are mutually non-adjacent.

- What follows is an example of how to avoid scheduling conflicts, and finding an optimal schedule.

- **Problem.** How many days must be scheduled for committee meetings of Pariliament if every committee meets all day, and some members serve on more than one committee?

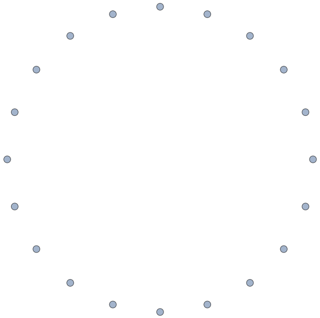
- **Set-up.** Each committee is represented by a vertex; two vertices (committees) are adjacent iff the vertices have (at least) one committee member in common.

Parliamentary Scheduling, best case

- The best possible case, in terms of scheduling, would be a graph with no edges.

Parliamentary Scheduling, best case

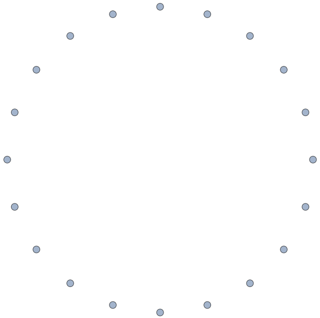
- The best possible case, in terms of scheduling, would be a graph with no edges.



-

Parliamentary Scheduling, best case

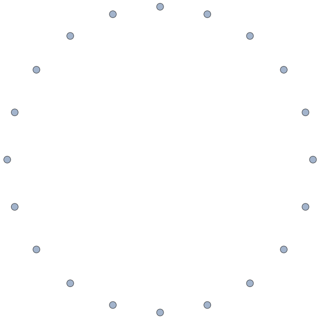
- The best possible case, in terms of scheduling, would be a graph with no edges.

- 

- What is the chromatic number of the above graph?

Parliamentary Scheduling, best case

- The best possible case, in terms of scheduling, would be a graph with no edges.

- 

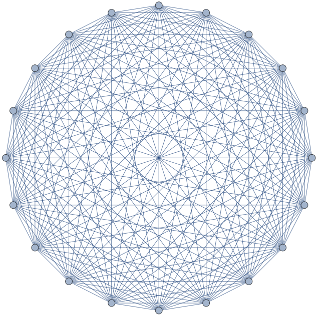
- What is the chromatic number of the above graph?
- How many days of committee meetings must be scheduled?

Parliamentary Scheduling, worst case

- The worst possible case, in terms of scheduling, would be a graph with all possible pairs of edges.

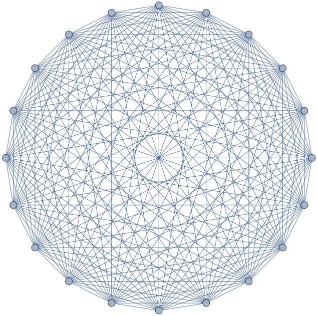
Parliamentary Scheduling, worst case

- The worst possible case, in terms of scheduling, would be a graph with all possible pairs of edges.

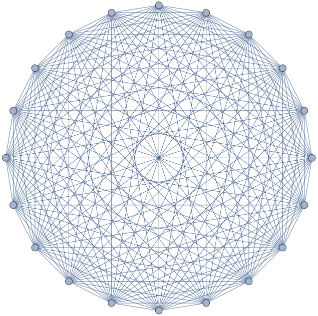


-

Parliamentary Scheduling, worst case

- The worst possible case, in terms of scheduling, would be a graph with all possible pairs of edges.
- 
- What is the chromatic number of the above graph (20 vertices)?

Parliamentary Scheduling, worst case

- The worst possible case, in terms of scheduling, would be a graph with all possible pairs of edges.
- 
- What is the chromatic number of the above graph (20 vertices)?
- How many days of committee meetings must be scheduled?

Parliamentary Scheduling, random case

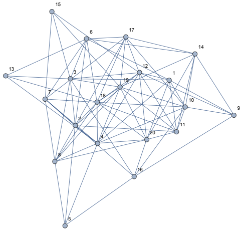
- What about the random graph (on 20 vertices) below?

Parliamentary Scheduling, random case

- What about the random graph (on 20 vertices) below?

```
RandomGraph[{20, 79}, VertexLabels → "Name", ImageSize → 500]
```

Out[50]=



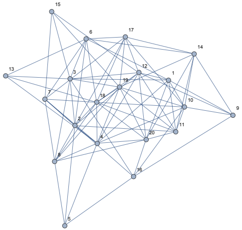

```
In[51]:= EdgeList[%50]

Out[51]= {1 ⟷ 3, 1 ⟷ 4, 1 ⟷ 6, 1 ⟷ 9, 1 ⟷ 10, 1 ⟷ 11, 1 ⟷ 12, 1 ⟷ 20, 2 ⟷ 3, 2 ⟷ 4, 2 ⟷ 5, 2 ⟷ 6, 2 ⟷ 7, 2 ⟷ 8, 2 ⟷ 11, 2 ⟷ 12, 2 ⟷ 17, 2 ⟷ 18,
  3 ⟷ 6, 3 ⟷ 8, 3 ⟷ 12, 3 ⟷ 13, 3 ⟷ 15, 3 ⟷ 16, 3 ⟷ 19, 4 ⟷ 5, 4 ⟷ 7, 4 ⟷ 8, 4 ⟷ 10, 4 ⟷ 12, 4 ⟷ 13, 4 ⟷ 18, 4 ⟷ 19, 4 ⟷ 20, 5 ⟷ 8,
  5 ⟷ 16, 6 ⟷ 7, 6 ⟷ 13, 6 ⟷ 14, 6 ⟷ 17, 6 ⟷ 18, 7 ⟷ 8, 7 ⟷ 15, 7 ⟷ 16, 7 ⟷ 17, 7 ⟷ 18, 8 ⟷ 18, 8 ⟷ 19, 8 ⟷ 20, 9 ⟷ 10, 9 ⟷ 12,
  9 ⟷ 14, 9 ⟷ 16, 10 ⟷ 11, 10 ⟷ 14, 10 ⟷ 16, 10 ⟷ 17, 10 ⟷ 18, 10 ⟷ 19, 11 ⟷ 12, 11 ⟷ 14, 11 ⟷ 18, 11 ⟷ 19, 11 ⟷ 20, 12 ⟷ 15,
  12 ⟷ 17, 12 ⟷ 18, 12 ⟷ 20, 13 ⟷ 18, 14 ⟷ 17, 14 ⟷ 19, 14 ⟷ 20, 15 ⟷ 19, 16 ⟷ 20, 17 ⟷ 18, 17 ⟷ 19, 18 ⟷ 19, 18 ⟷ 20, 19 ⟷ 20}
```

Parliamentary Scheduling, random case

- What about the random graph (on 20 vertices) below?

```
RandomGraph[{20, 79}, VertexLabels → "Name", ImageSize → 500]
```

Out[50]=




```
In[51]:= EdgeList[%50]
Out[51]= {1 ⟷ 3, 1 ⟷ 4, 1 ⟷ 6, 1 ⟷ 9, 1 ⟷ 10, 1 ⟷ 11, 1 ⟷ 12, 1 ⟷ 20, 2 ⟷ 3, 2 ⟷ 4, 2 ⟷ 5, 2 ⟷ 6, 2 ⟷ 7, 2 ⟷ 8, 2 ⟷ 11, 2 ⟷ 12, 2 ⟷ 17, 2 ⟷ 18,
  3 ⟷ 6, 3 ⟷ 8, 3 ⟷ 12, 3 ⟷ 13, 3 ⟷ 15, 3 ⟷ 16, 3 ⟷ 19, 4 ⟷ 5, 4 ⟷ 7, 4 ⟷ 8, 4 ⟷ 10, 4 ⟷ 12, 4 ⟷ 13, 4 ⟷ 18, 4 ⟷ 19, 4 ⟷ 20, 5 ⟷ 8,
  5 ⟷ 16, 6 ⟷ 7, 6 ⟷ 13, 6 ⟷ 14, 6 ⟷ 17, 6 ⟷ 18, 7 ⟷ 8, 7 ⟷ 15, 7 ⟷ 16, 7 ⟷ 17, 7 ⟷ 18, 8 ⟷ 18, 8 ⟷ 19, 8 ⟷ 20, 9 ⟷ 10, 9 ⟷ 12,
  9 ⟷ 14, 9 ⟷ 16, 10 ⟷ 11, 10 ⟷ 14, 10 ⟷ 16, 10 ⟷ 17, 10 ⟷ 18, 10 ⟷ 19, 11 ⟷ 12, 11 ⟷ 14, 11 ⟷ 18, 11 ⟷ 19, 11 ⟷ 20, 12 ⟷ 15,
  12 ⟷ 17, 12 ⟷ 18, 12 ⟷ 20, 13 ⟷ 18, 14 ⟷ 17, 14 ⟷ 19, 14 ⟷ 20, 15 ⟷ 19, 16 ⟷ 20, 17 ⟷ 18, 17 ⟷ 19, 18 ⟷ 19, 18 ⟷ 20, 19 ⟷ 20}
```

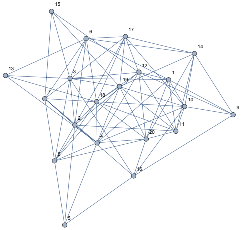
- What is the chromatic number of the above graph (20 vertices)?

Parliamentary Scheduling, random case

- What about the random graph (on 20 vertices) below?

```
RandomGraph[{20, 79}, VertexLabels → "Name", ImageSize → 500]
```

Out[50]=




```
In[51]:= EdgeList[%50]
Out[51]= {1 ↔ 3, 1 ↔ 4, 1 ↔ 6, 1 ↔ 9, 1 ↔ 10, 1 ↔ 11, 1 ↔ 12, 1 ↔ 20, 2 ↔ 3, 2 ↔ 4, 2 ↔ 5, 2 ↔ 6, 2 ↔ 7, 2 ↔ 8, 2 ↔ 11, 2 ↔ 12, 2 ↔ 17, 2 ↔ 18,
  3 ↔ 6, 3 ↔ 8, 3 ↔ 12, 3 ↔ 13, 3 ↔ 15, 3 ↔ 16, 3 ↔ 19, 4 ↔ 5, 4 ↔ 7, 4 ↔ 8, 4 ↔ 10, 4 ↔ 12, 4 ↔ 13, 4 ↔ 18, 4 ↔ 19, 4 ↔ 20, 5 ↔ 8,
  5 ↔ 16, 6 ↔ 7, 6 ↔ 13, 6 ↔ 14, 6 ↔ 17, 6 ↔ 18, 7 ↔ 8, 7 ↔ 15, 7 ↔ 16, 7 ↔ 17, 7 ↔ 18, 8 ↔ 18, 8 ↔ 19, 8 ↔ 20, 9 ↔ 10, 9 ↔ 12,
  9 ↔ 14, 9 ↔ 16, 10 ↔ 11, 10 ↔ 14, 10 ↔ 16, 10 ↔ 17, 10 ↔ 18, 10 ↔ 19, 11 ↔ 12, 11 ↔ 14, 11 ↔ 18, 11 ↔ 19, 11 ↔ 20, 12 ↔ 15,
  12 ↔ 17, 12 ↔ 18, 12 ↔ 20, 13 ↔ 18, 14 ↔ 17, 14 ↔ 19, 14 ↔ 20, 15 ↔ 19, 16 ↔ 20, 17 ↔ 18, 17 ↔ 19, 18 ↔ 19, 18 ↔ 20, 19 ↔ 20}
```

- What is the chromatic number of the above graph (20 vertices)?
- How many days of committee meetings must be scheduled?

Scheduling

In general, if $\chi(G) = k$, then k days of meetings must be scheduled.

Next

- **Map Coloring**

Next

- **Map Coloring**
-