南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

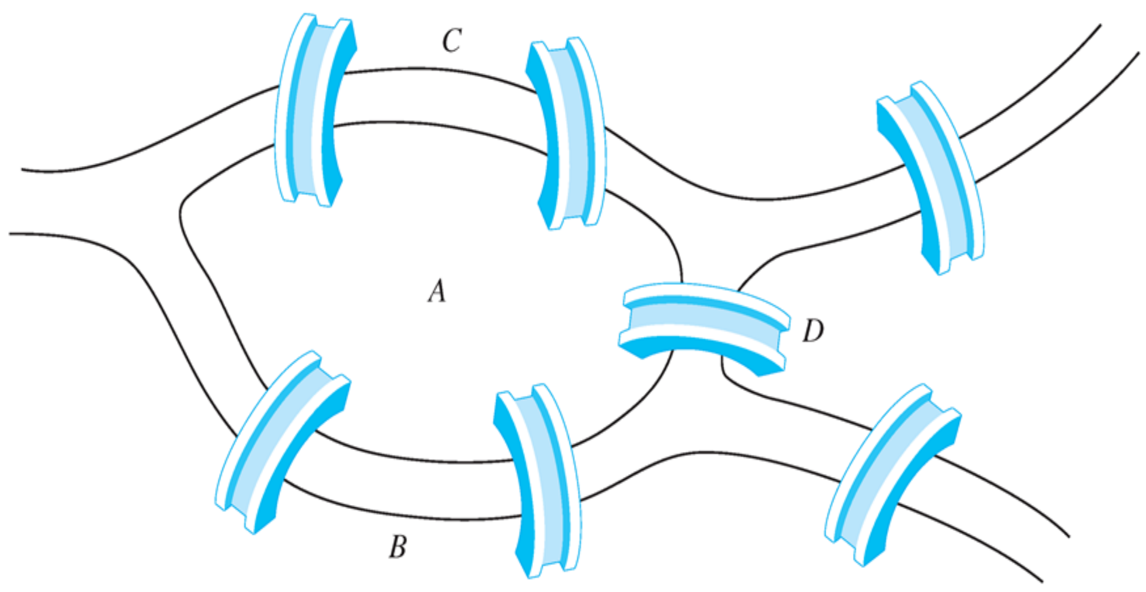
DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room903, Nanshan iPark A7 Building
Email: wangqi@sustc.edu.cn

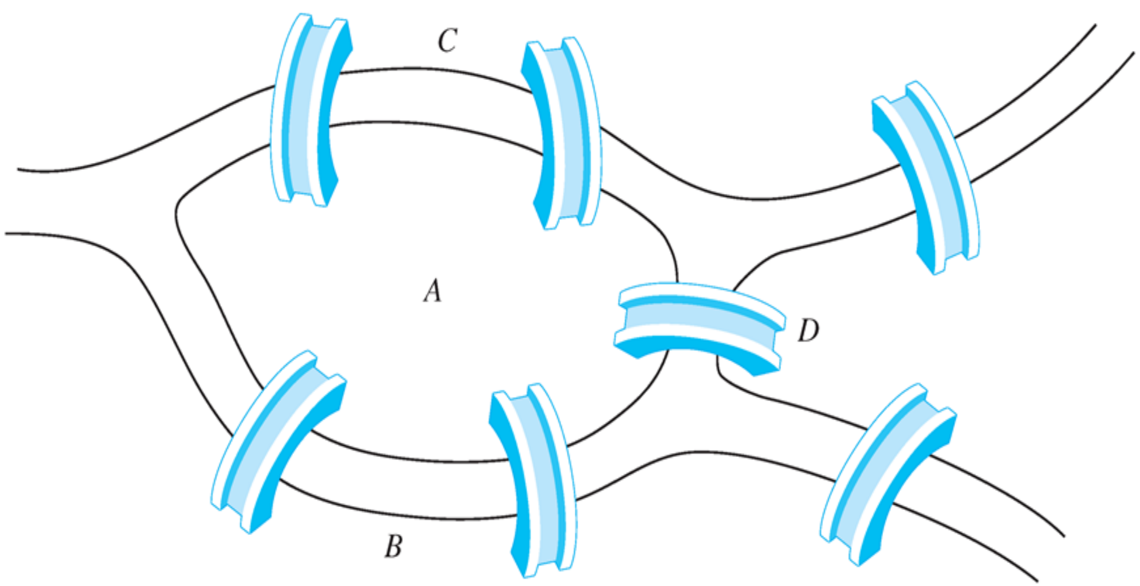
- **Königsberg seven-bridge problem**

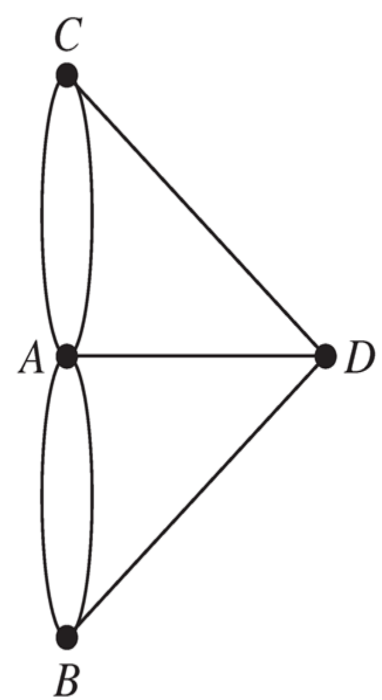
  People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.



- **Königsberg seven-bridge problem**

  People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.





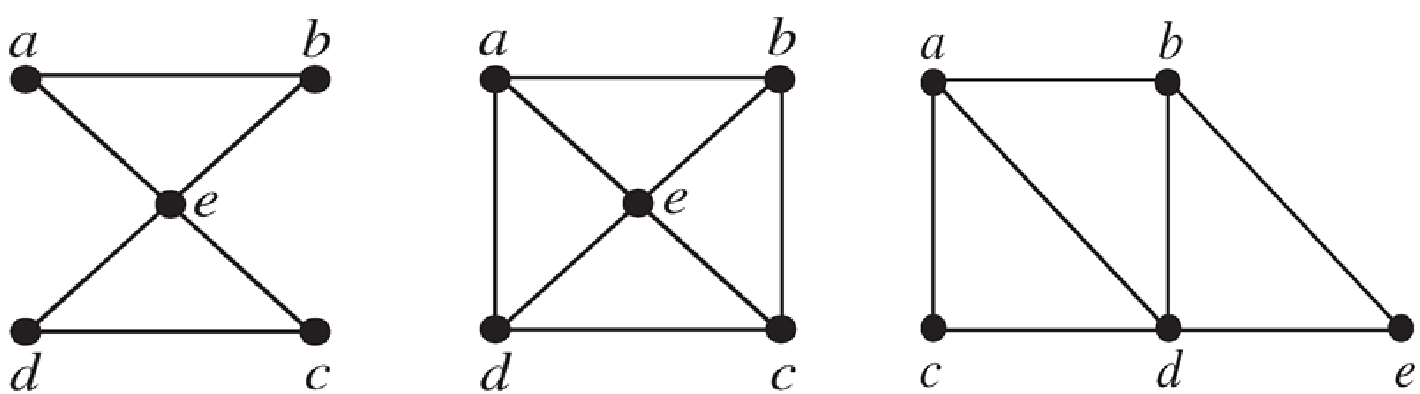
Euler Paths and Circuits

- **Definition** An *Euler circuit* in a graph G is a simple circuit containing every edge of G. An *Euler path* in G is a simple path containing every edge of G.

Euler Paths and Circuits

- **Definition** An *Euler circuit* in a graph G is a simple circuit containing every edge of G. An *Euler path* in G is a simple path containing every edge of G.

Example Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?

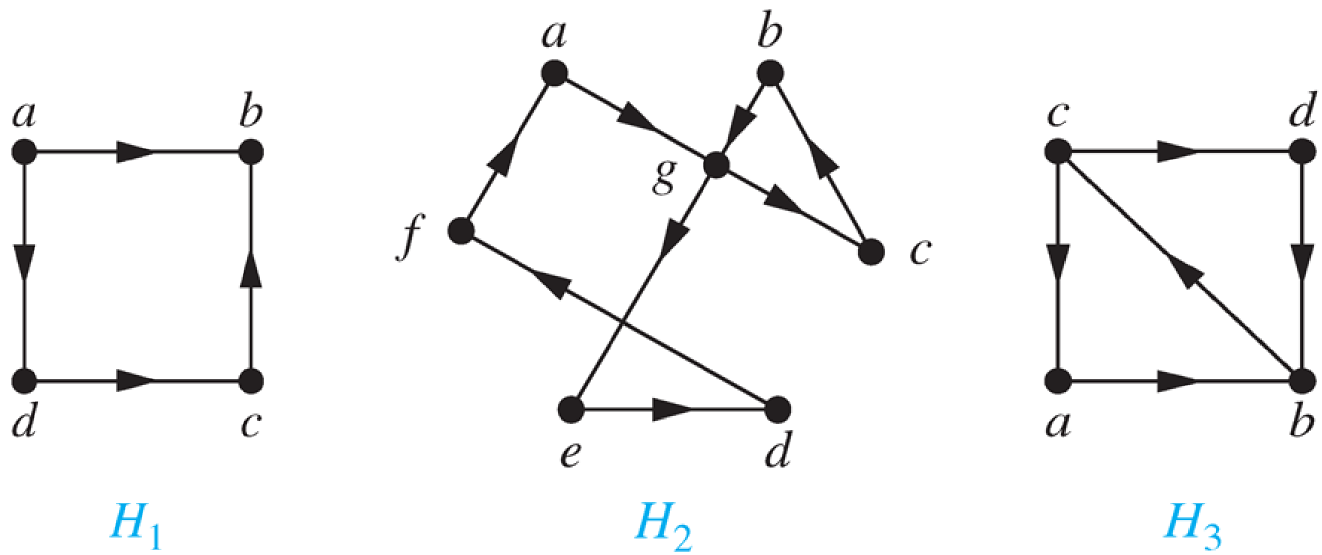


G_1 G_2 G_3

Euler Paths and Circuits

- **Definition** An *Euler circuit* in a graph G is a simple circuit containing every edge of G. An *Euler path* in G is a simple path containing every edge of G.

Example Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - ◇ Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - ◇ The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

Euler Path $\Rightarrow$ The graph has exactly two vertices of odd degree

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

Euler Path $\Rightarrow$ The graph has exactly two vertices of odd degree

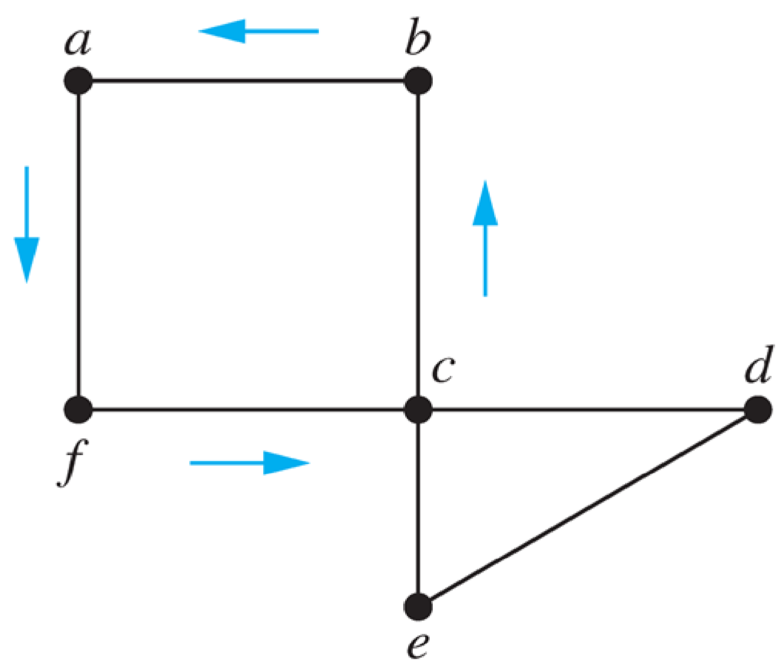
  - The initial vertex and the final vertex of an Euler path have odd degree.

Sufficient Conditons for Euler Circuits and Paths

- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.

Sufficient Conditons for Euler Circuits and Paths

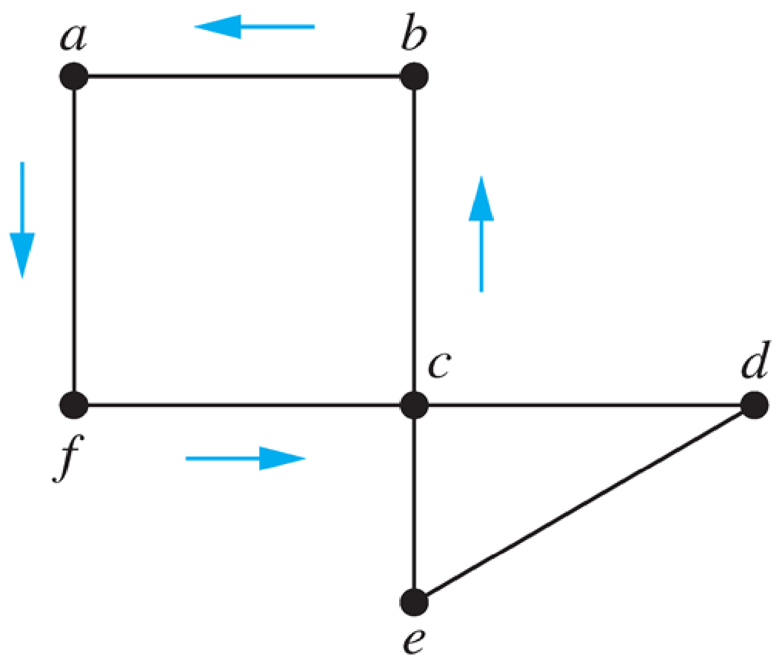
- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.



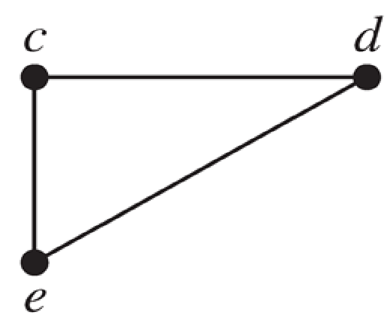
G

Sufficient Conditons for Euler Circuits and Paths

- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.



G



H

Algorithm for Constructing an Euler Circuits

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex.
H := G with the edges of this circuit removed
while H has edges
  subciruit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge in circuit.
  H := H with edges of *subciruit* and all isolated vertices removed
  circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
return *circuit*{*circuit* is an Euler circuit}

Algorithm for Constructing an Euler Circuits

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex.
H := G with the edges of this circuit removed
while H has edges
  subciruit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge in circuit.
  H := H with edges of *subciruit* and all isolated vertices removed
  circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
return *circuit*{*circuit* is an Euler circuit}

Algorithm for Constructing an Euler Circuits

- **procedure** *Euler*(G: connected multigraph with all vertices of even degree)
  circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges
  successively added to form a path that returns to this vertex.
  $\boxed{H := G \text{ with the edges of this circuit removed}}$
  while H has edges
  subciruit := a circuit in H beginning at a vertex in H that also is
  an endpoint of an edge in circuit.
  $H := H$ with edges of *subciruit* and all isolated vertices removed
  $\boxed{circuit := circuit \text{ with } subcircuit \text{ inserted at the appropriate vertex.}}$
  return *circuit*{*circuit* is an Euler circuit}

Necessary and Sufficient Conditions

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

Necessary and Sufficient Conditions

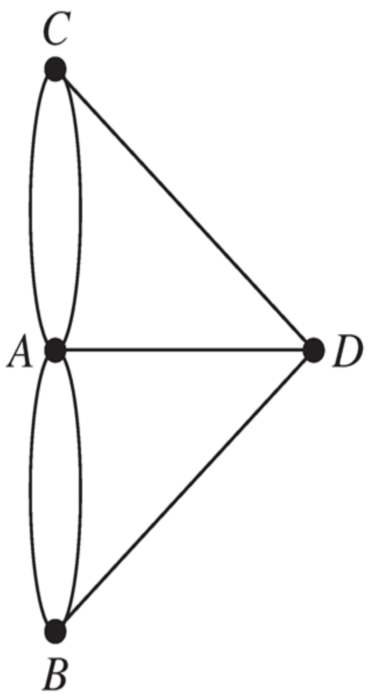
- **Theorem** A connected multigraph with at least two vertices has an *Euler circuit* if and only if each of its vertices has **even** degree.

  Theorem A connected multigraph has an *Euler path* but not an *Euler circuit* if and only if it has exactly two vertices of odd degree.

Necessary and Sufficient Conditions

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

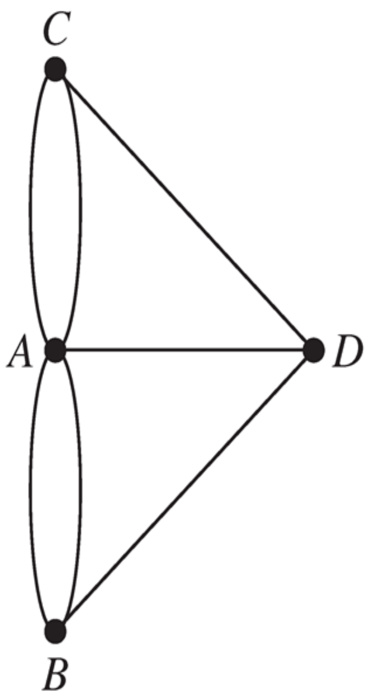
  Theorem A connected multigraph has an *Euler path* but not an *Euler circuit* if and only if it has exactly two vertices of odd degree.



Necessary and Sufficient Conditions

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

  Theorem A connected multigraph has an *Euler path* but not an *Euler circuit* if and only if it has exactly two vertices of odd degree.



No Euler circuit

Example

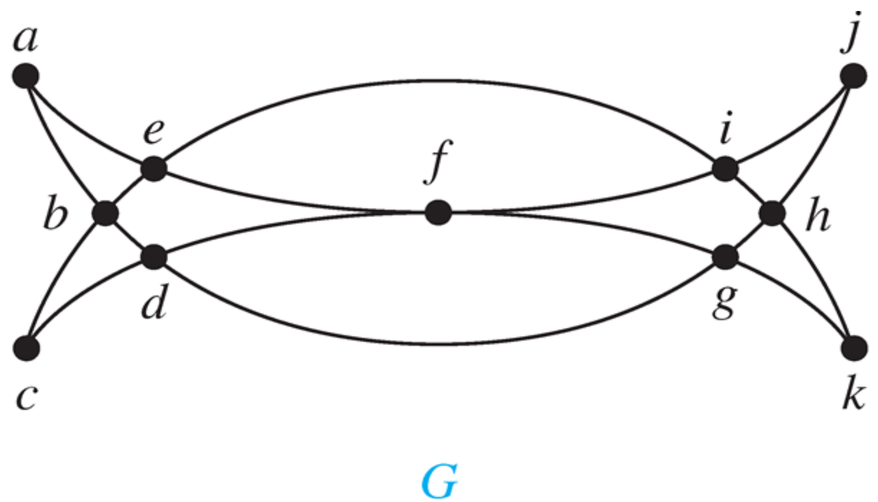
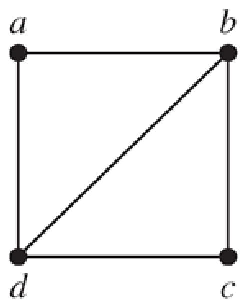


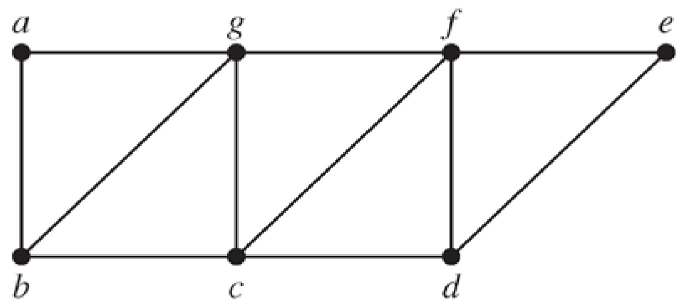
FIGURE 6 **Mohammed's Scimitars.**

Euler Circuits and Paths

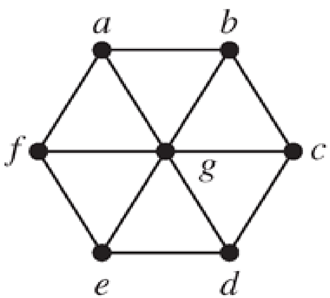
- **Example**



G_1



G_2



G_3

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Chinese Postman Problem Meigu Guan [60']

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - ◇ street in a neighborhood
  - ◇ road in a transportation network
  - ◇ link in a communication network
  - ◇ ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

k-Postman Chinese Postman Problem (k-PCPP)

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

k-Postman Chinese Postman Problem (k-PCPP)

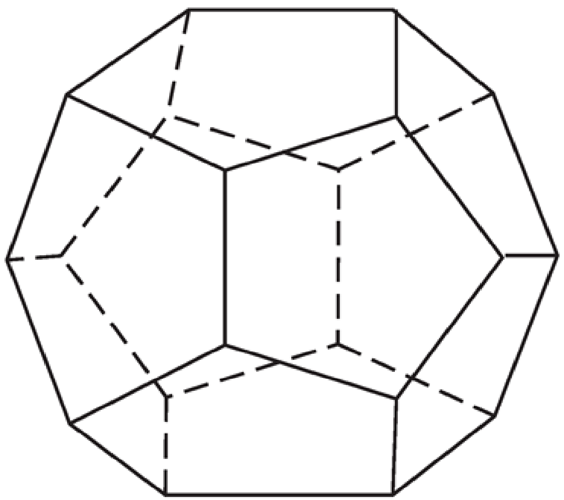
$\in$ NPC

Hamilton Paths and Circuits

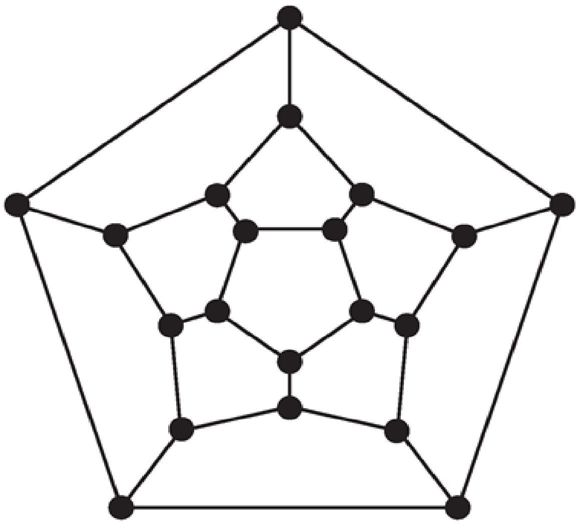
- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?

Hamilton Paths and Circuits

- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?



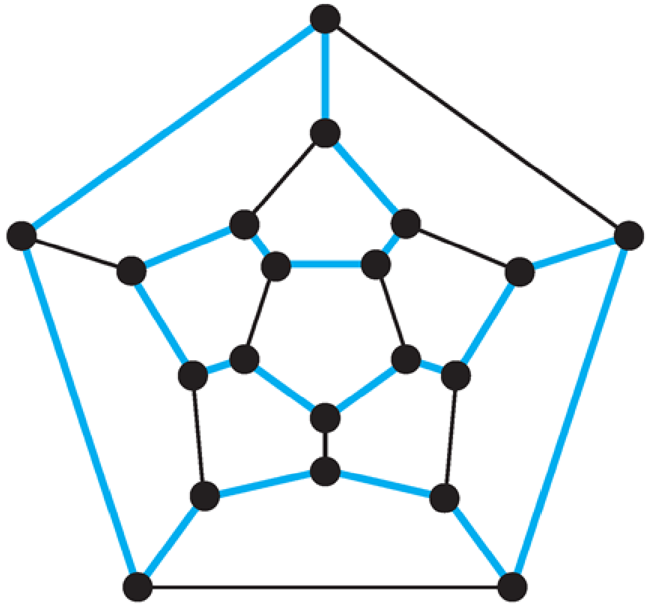
(a)



(b)

Hamilton Paths and Circuits

- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?

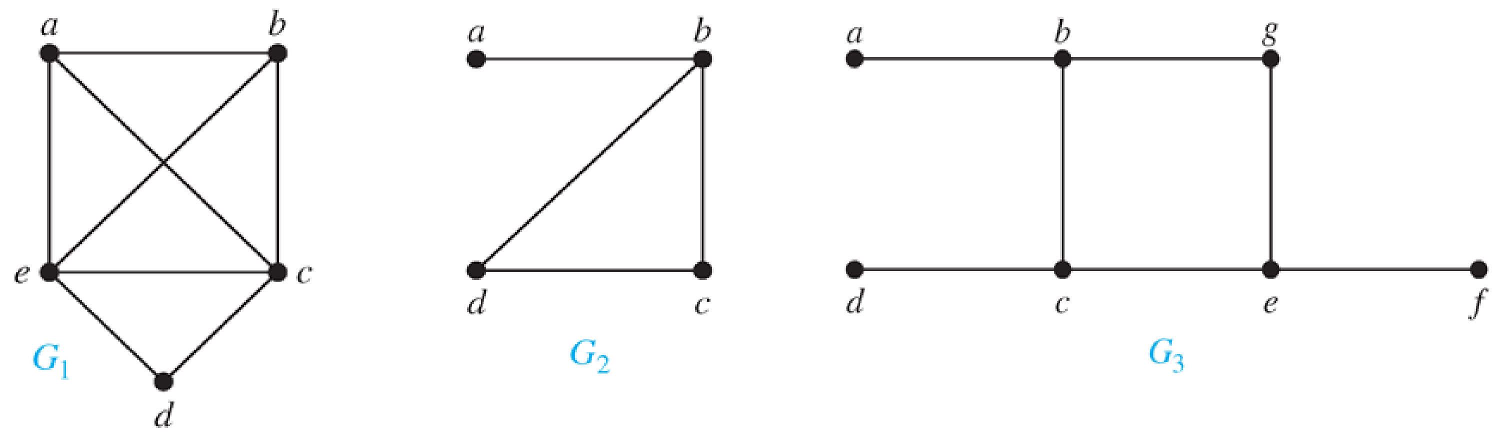


Hamilton Paths and Circuits

- **Definition**: A simple path in a graph G that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph G that passes through every vertex exactly once is called a *Hamilton circuit*.

- **Definition**: A simple path in a graph G that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph G that passes through every vertex exactly once is called a *Hamilton circuit*.

  Example Which of these simple graphs has a Hamilton circuit or, if not, a Hamilton path?



Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

  Ore's Theorem If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

  Ore's Theorem If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

  Hamilton path problem $\in$ NPC

Applications of Hamilton Paths and Circuits

- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

Applications of Hamilton Paths and Circuits

- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

  Traveling Salesperson Problem (TSP) asks for the shortest route a traveling salesperson should take to visit a set of cities.

Applications of Hamilton Paths and Circuits

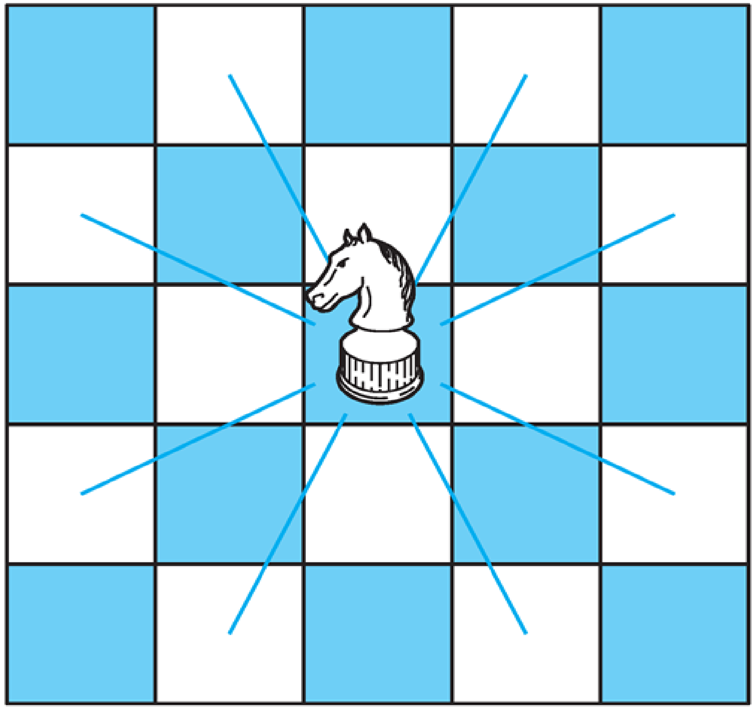
- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

  Traveling Salesperson Problem (TSP) asks for the shortest route a traveling salesperson should take to visit a set of cities.

  the decision version of the TSP $\in$ NPC

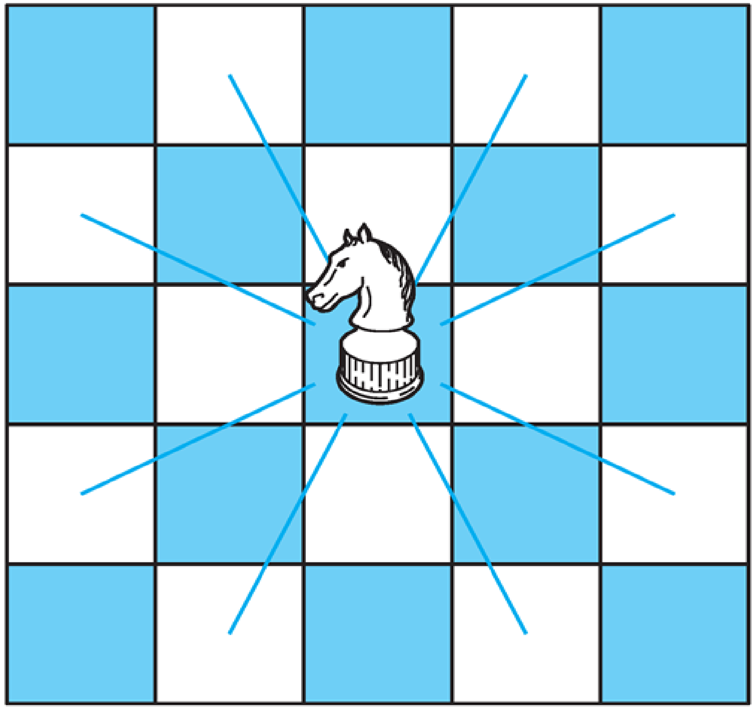
Applications of Hamilton Paths and Circuits

- Can we traverse every space (and come back) in the 5×5 chessboard?



Applications of Hamilton Paths and Circuits

- Can we traverse every space (and come back) in the 5×5 chessboard?



What about in 6×6 chessboard?

Shortest Path Problems

- Using graphs with weights assigned to their edges

Shortest Path Problems

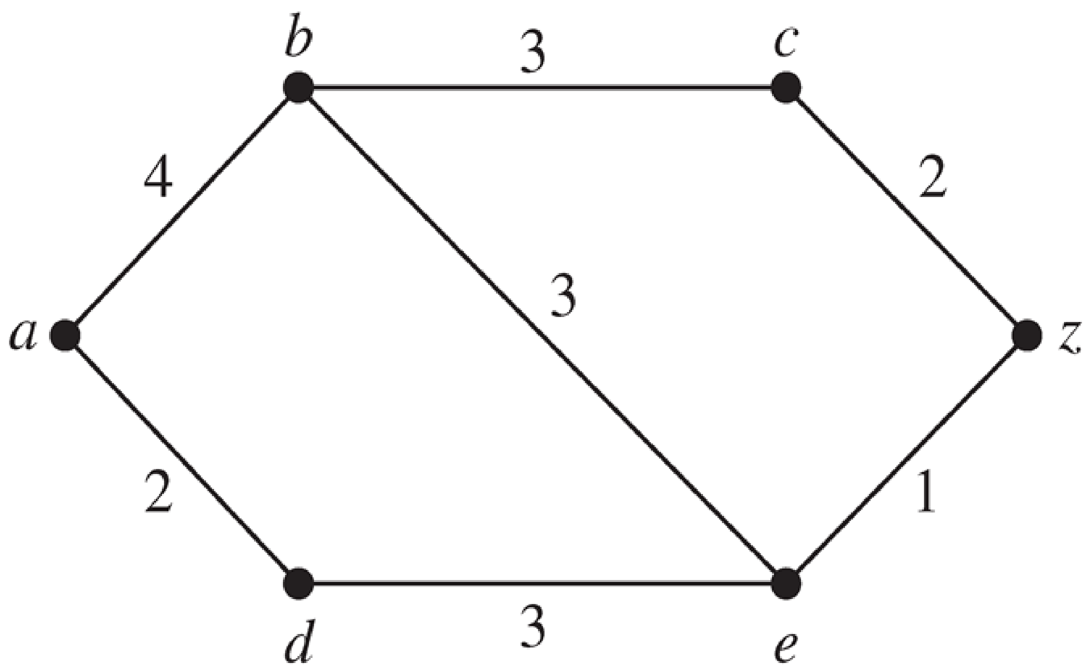
- Using graphs with weights assigned to their edges

  Such graphs are called *weighted graphs* and can model lots of questions involving distances, time consuming, fares, etc.

Shortest Path Problems

- Using graphs with weights assigned to their edges

  Such graphs are called *weighted graphs* and can model lots of questions involving distances, time consuming, fares, etc.



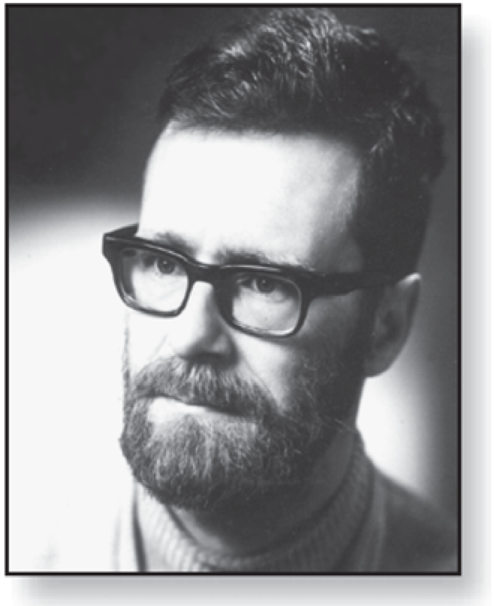
Shortest Path Problems

- **Definition** Let G^α be an weighted graph, with a weight function $\alpha : E \to \mathbf{R}$ on its edges. If $P = e_1 e_2 \cdots e_k$ is a path, then its weight is $\alpha(P) = \sum_{i=1}^{k} \alpha(e_i)$. The *minimum weighted distance* between two vertices is

$$d(u, v) = \min\{\alpha(P) | P : u \to v\}$$

- **Definition** Let G^α be an weighted graph, with a weight function $\alpha : E \to \mathbf{R}$ on its edges. If $P = e_1 e_2 \cdots e_k$ is a path, then its weight is $\alpha(P) = \sum_{i=1}^{k} \alpha(e_i)$. The *minimum weighted distance* between two vertices is

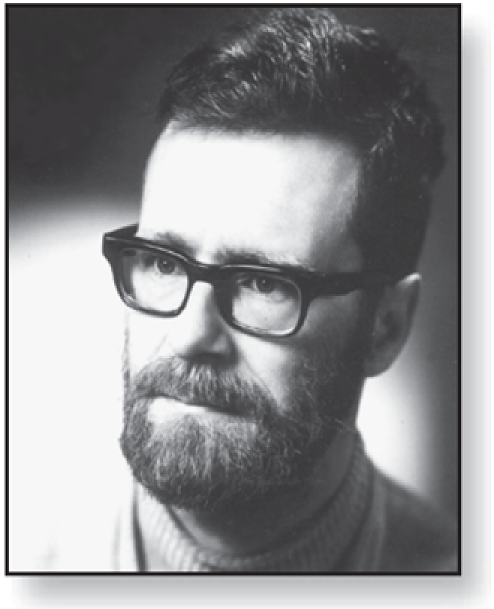
$$d(u, v) = \min\{\alpha(P) | P : u \to v\}$$



Edsger Wybe Dijkstra

Dijkstra's Algorithm

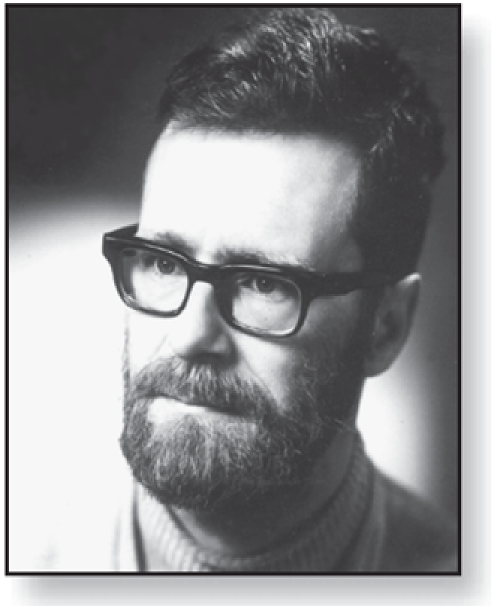
- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$



Edsger Wybe Dijkstra

Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$



Edsger Wybe Dijkstra

Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$

  (ii) while $S \neq V$

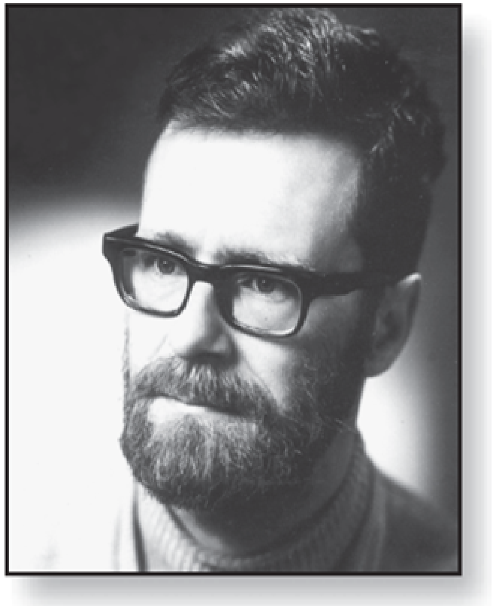
  let $v \notin S$ be the vertex with the

  least value $d(v)$, $S = S \cup \{v\}$

  for each $u \notin S$, replace $d(u)$ by

  $\min\{d(u), d(v) + \alpha(u, v)\}$

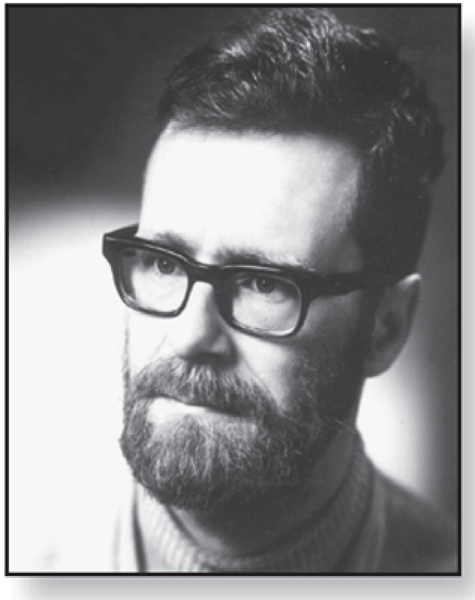
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

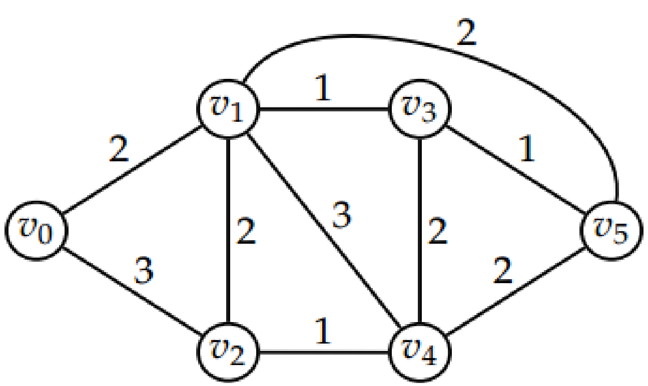
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$

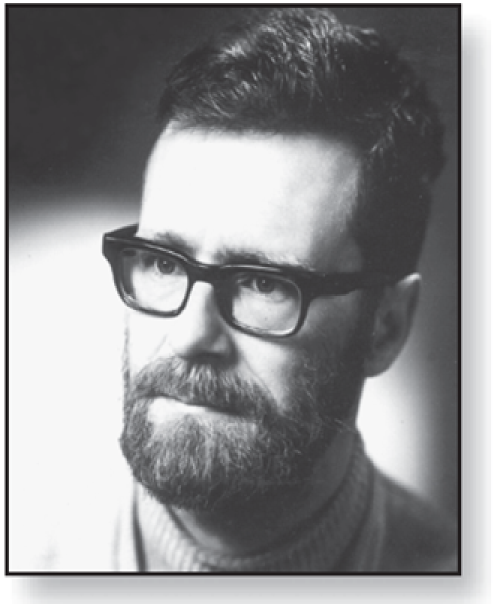
  (ii) while $S \neq V$

  let $v \notin S$ be the vertex with the least value $d(v)$, $S = S \cup \{v\}$

  for each $u \notin S$, replace $d(u)$ by

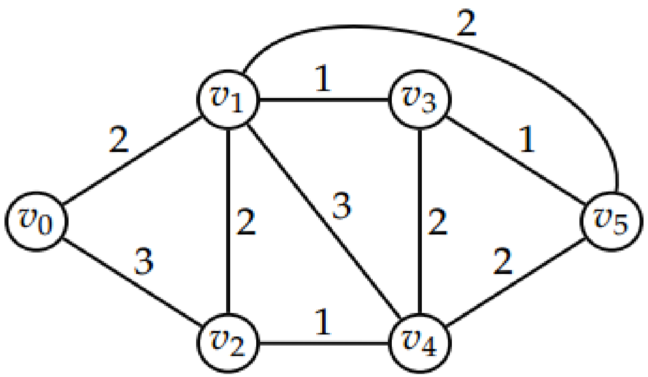
  $\min\{d(u), d(v) + \alpha(u, v)\}$

  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

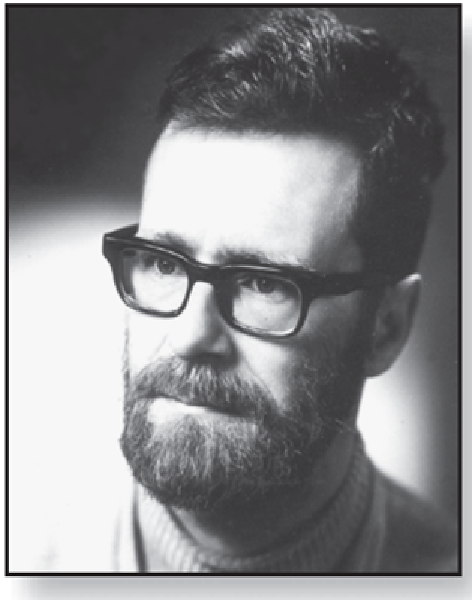
Example



$d(v_0) = 0$, all other $d(v) = \infty$

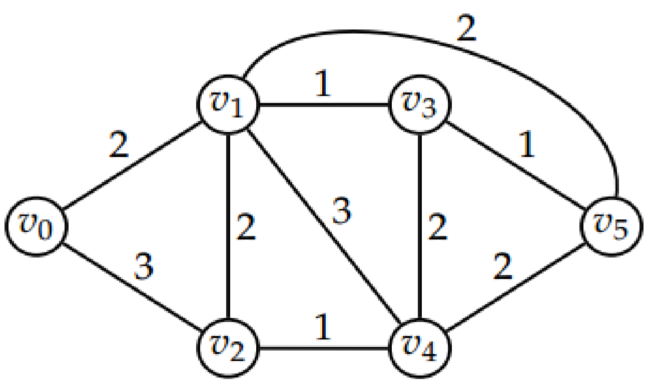
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example

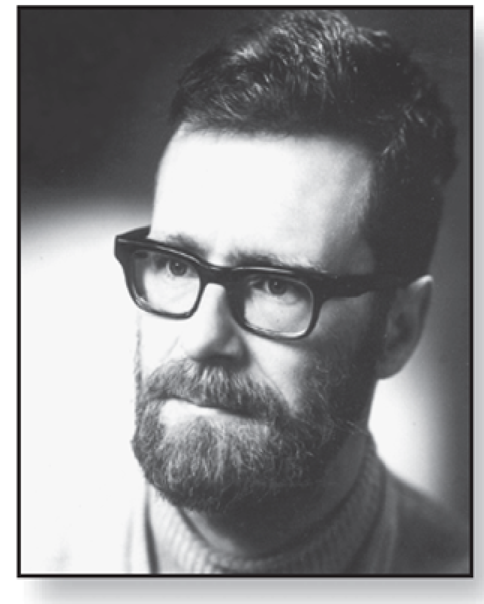


v_0	v_1	v_2	v_3	v_4	v_5
0	∞	∞	∞	∞	∞

$d(v_0) = 0$, all other $d(v) = \infty$

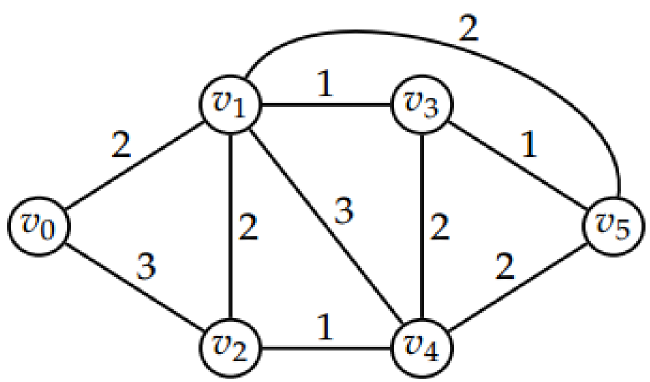
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



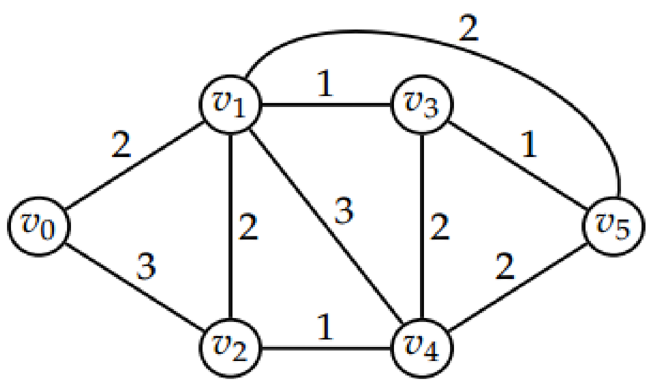
v_0	v_1	v_2	v_3	v_4	v_5
0	∞	∞	∞	∞	∞

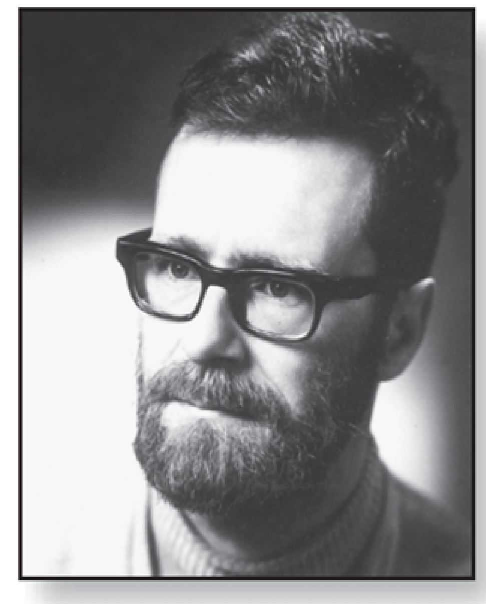
$i = 0$

$d(v_1) = \min\{\infty, 2\} = 2$, $d(v_2) = \min\{\infty, 3\} = 3$

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s

Example





Edsger Wybe Dijkstra

v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	∞	∞	∞

$i = 0$

$d(v_1) = \min\{\infty, 2\} = 2$, $d(v_2) = \min\{\infty, 3\} = 3$

Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$

  (ii) while $S \neq V$

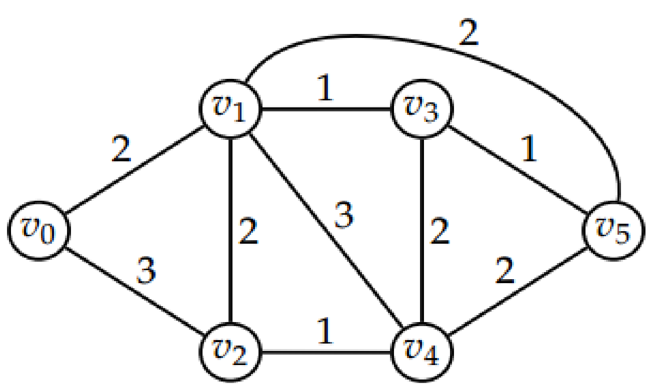
  let $v \notin S$ be the vertex with the least value $d(v)$, $S = S \cup \{v\}$

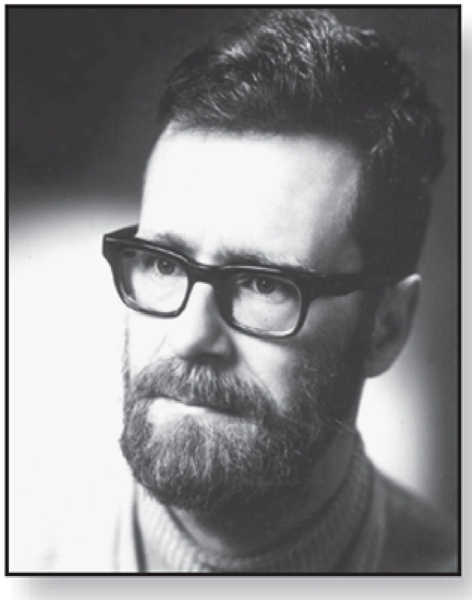
  for each $u \notin S$, replace $d(u)$ by

  $$\min\{d(u), d(v) + \alpha(u, v)\}$$

  (iii) return all $d(v)$'s

  Example





Edsger Wybe Dijkstra

v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	∞	∞	∞

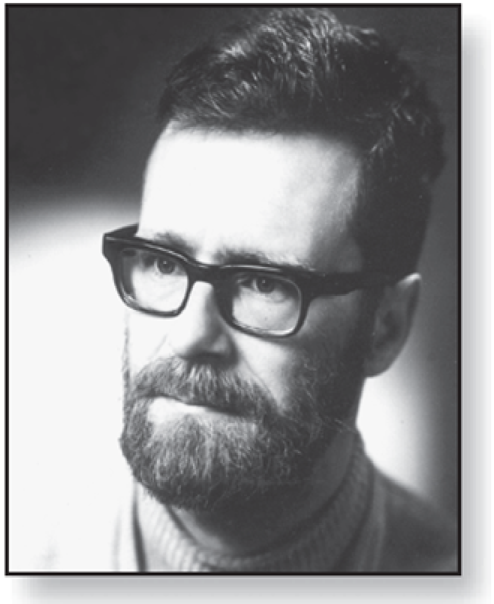
$i = 1$

$d(v_2) = \min\{3, d(v_1) + \alpha(v_1 v_2)\} = \min\{3, 4\} = 3,$

$d(v_3) = 2 + 1 = 3$, $d(v_4) = 2 + 3 = 5,$

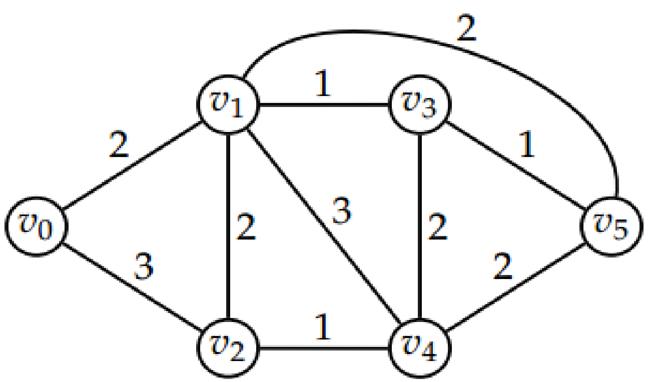
$d(v_5) = 2 + 2 = 4$

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	3	5	4

$i = 1$

$d(v_2) = \min\{3, d(v_1) + \alpha(v_1 v_2)\} = \min\{3, 4\} = 3,$
$d(v_3) = 2 + 1 = 3,\ d(v_4) = 2 + 3 = 5,$
$d(v_5) = 2 + 2 = 4$

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$

  (ii) while $S \neq V$

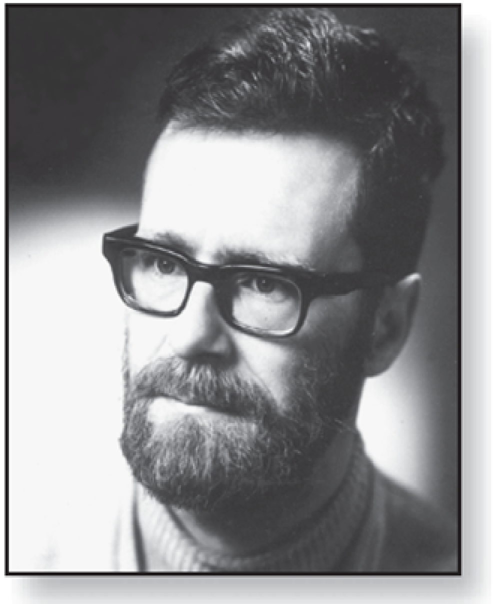
  let $v \notin S$ be the vertex with the

  least value $d(v)$, $S = S \cup \{v\}$

  for each $u \notin S$, replace $d(u)$ by

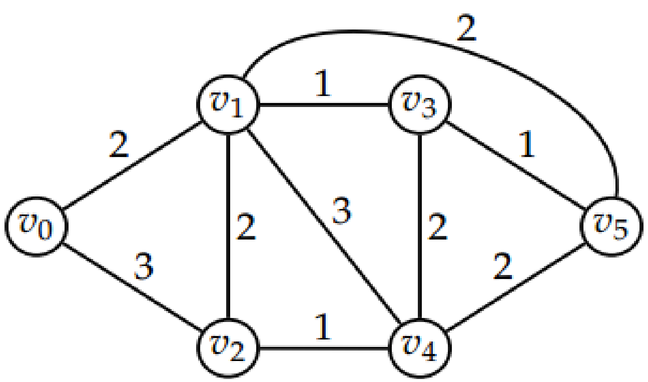
  $\min\{d(u), d(v) + \alpha(u, v)\}$

  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	3	5	4

$i = 2$

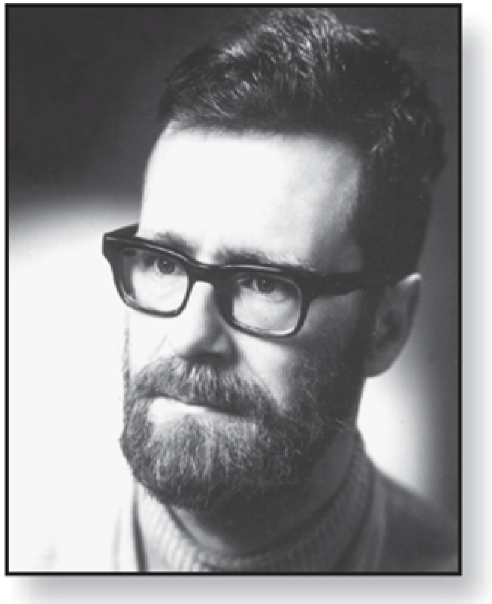
$d(v_3) = \min\{3, \infty\} = 3,$

$d(v_4) = \min\{5, 3 + 1\} = 4,$

$d(v_5) = \min\{4, \infty\} = 4$

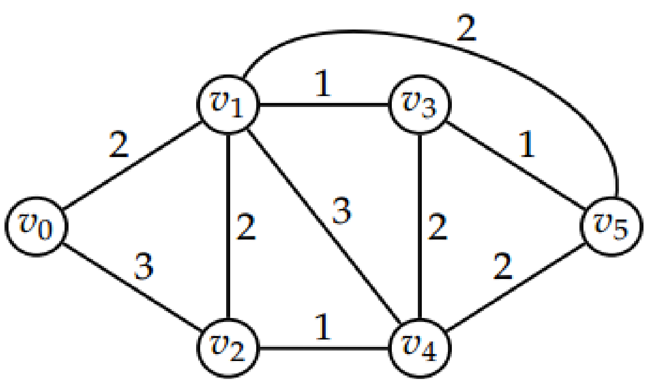
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	3	4	4

$i = 2$

$d(v_3) = \min\{3, \infty\} = 3,$

$d(v_4) = \min\{5, 3 + 1\} = 4,$

$d(v_5) = \min\{4, \infty\} = 4$

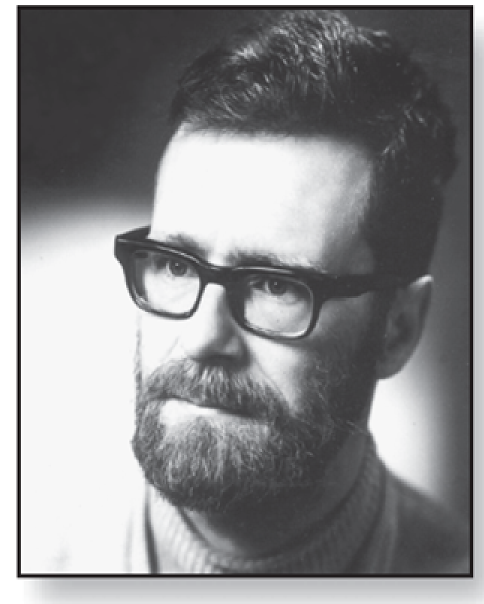
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$

  (ii) while $S \neq V$

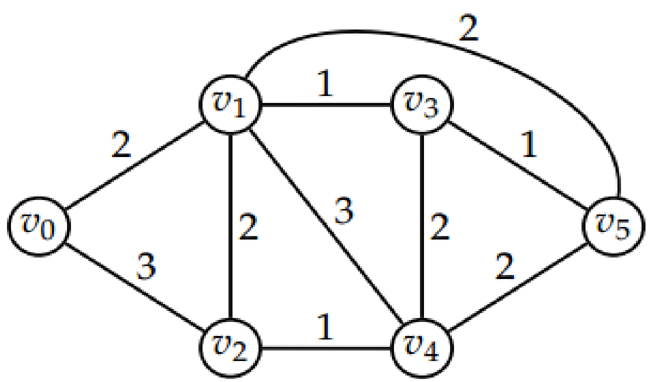
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$

  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	3	4	4

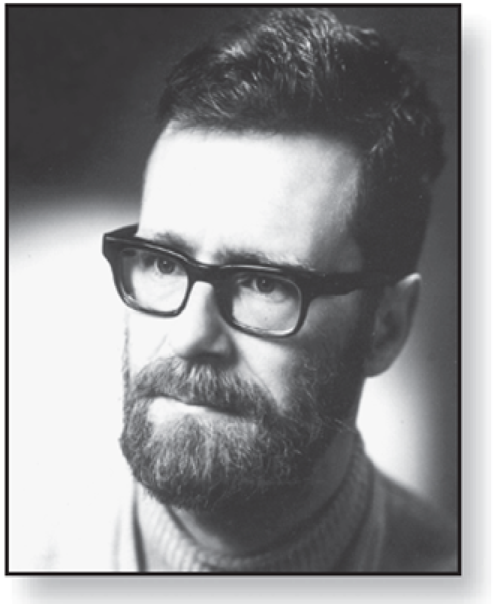
$i = 3$

$d(v_4) = \min\{4, 3 + 2\} = 4,$

$d(v_5) = \min\{4, 3 + 1\} = 4$

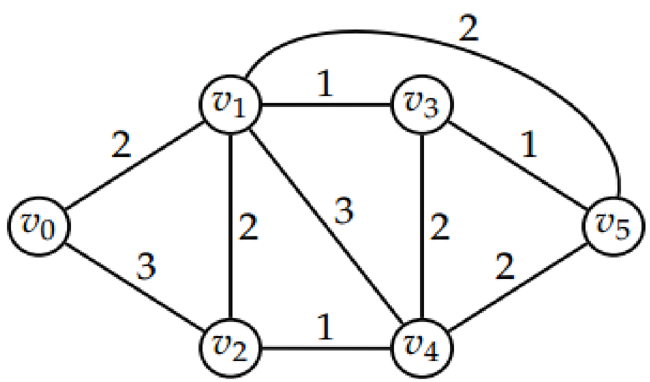
Dijkstra's Algorithm

- (i) Set $d(v_0) = 0$ and $d(v) = \infty$ for all $v \neq v_0$, $S = \emptyset$
  (ii) while $S \neq V$
  let $v \notin S$ be the vertex with the
  least value $d(v)$, $S = S \cup \{v\}$
  for each $u \notin S$, replace $d(u)$ by
  $\min\{d(u), d(v) + \alpha(u, v)\}$
  (iii) return all $d(v)$'s



Edsger Wybe Dijkstra

Example



v_0	v_1	v_2	v_3	v_4	v_5
0	2	3	3	4	4

$i = 4$

$d(v_5) = \min\{4, 4 + 2\} = 4$

Dijkstra's Algorithm

- **Theorem** Dijkstra's algorithm finds the length of a shortest path between two vertices in a connnected simple undirected weighted graph.

Dijkstra's Algorithm

- **Theorem** Dijkstra's algorithm finds the length of a shortest path between two vertices in a connnected simple undirected weighted graph.

  Correctness

Dijkstra's Algorithm

- **Theorem** Dijkstra's algorithm finds the length of a shortest path between two vertices in a connnected simple undirected weighted graph.

  Correctness

  Theorem Dijkstra's algorithm uses $O(n^2)$ operations (additions and comparisons) in a connected simple undirected weighted graph with n vertices.

Dijkstra's Algorithm

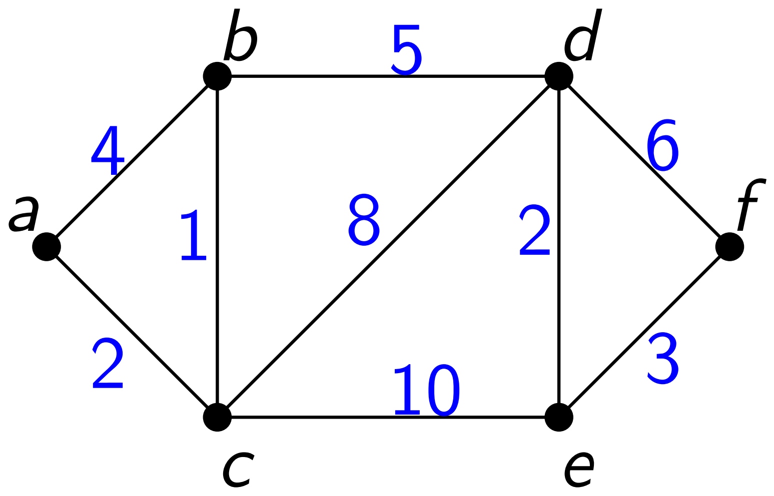
- **Theorem** Dijkstra's algorithm finds the length of a shortest path between two vertices in a connnected simple undirected weighted graph.

  Correctness

  Theorem Dijkstra's algorithm uses $O(n^2)$ operations (additions and comparisons) in a connected simple undirected weighted graph with n vertices.

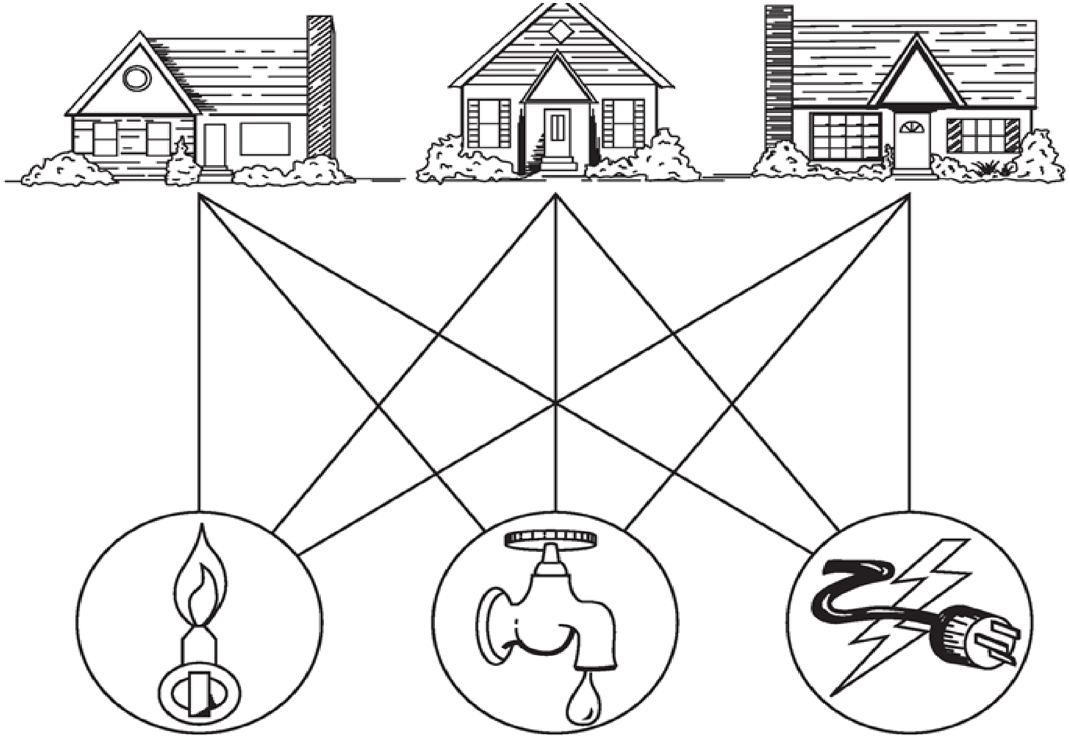
  Complexity

Another Example



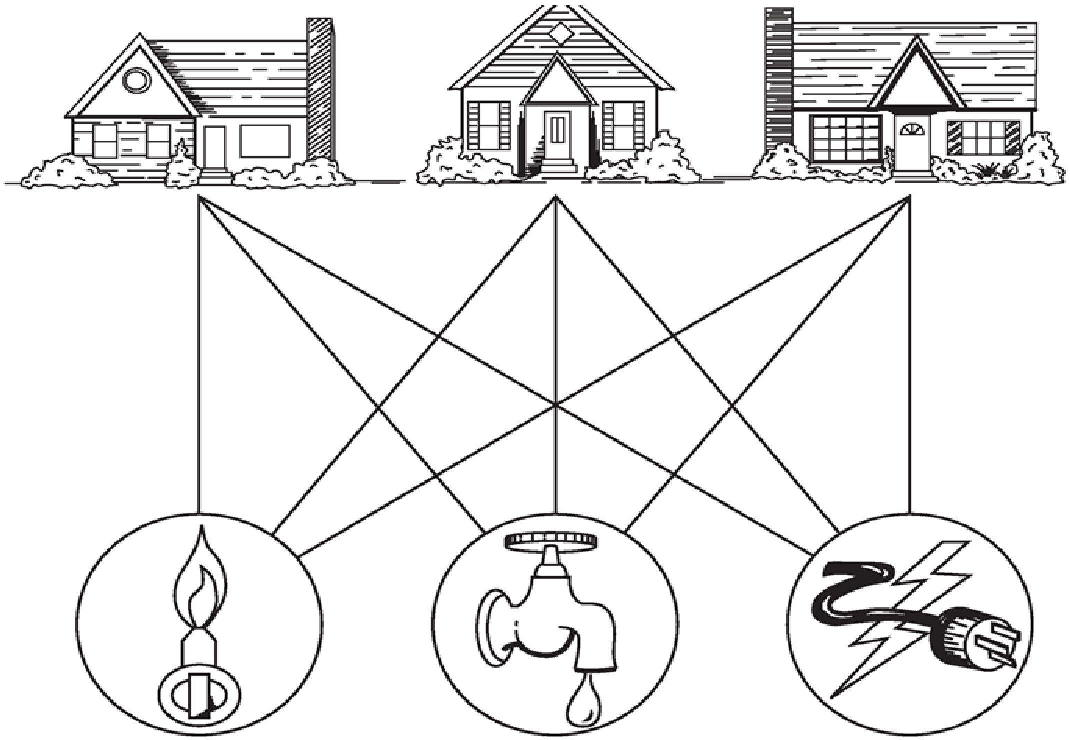
Planar Graphs

- Join three houses to each of three seperate utilities.



Planar Graphs

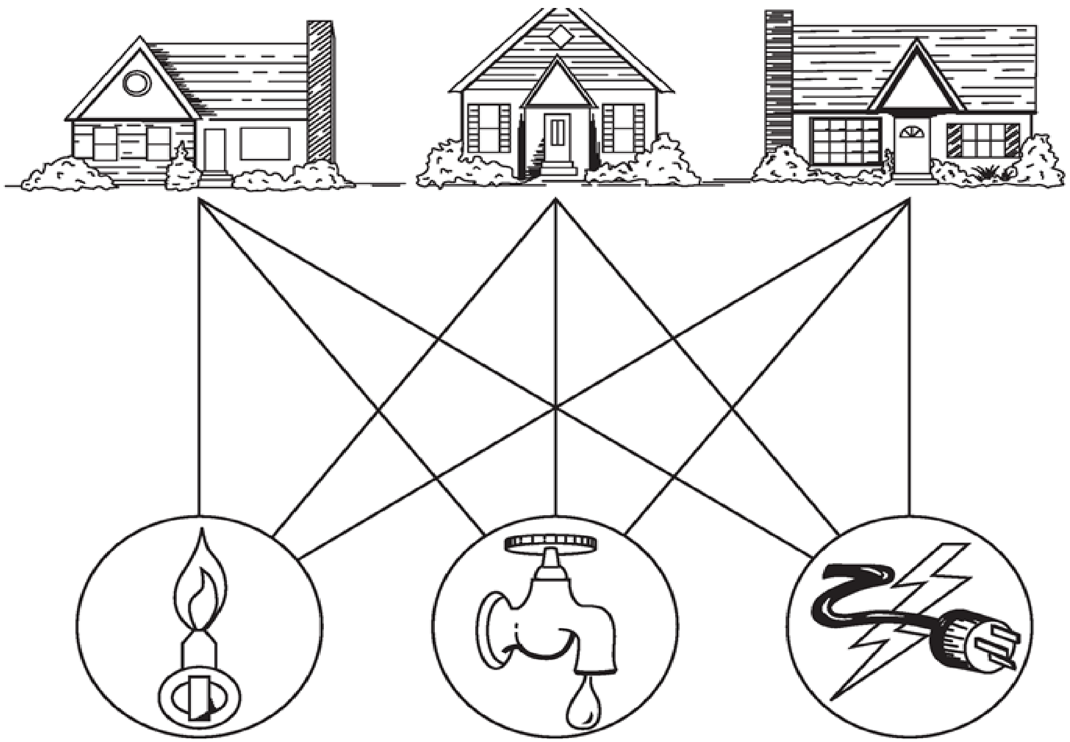
- Join three houses to each of three seperate utilities.



Can this graph be drawn in the plane s.t. no two of its edges cross?

Planar Graphs

- Join three houses to each of three seperate utilities.



Can this graph be drawn in the plane s.t. no two of its edges cross? $K_{3,3}$

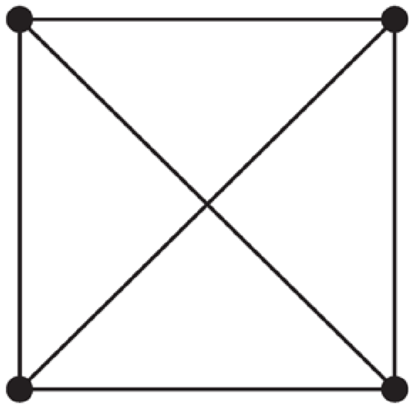
Planar Graphs

- **Definition** A graph is called *planar* if it can be drawn in the plane without any edges crossing. Such a drawing is called a *planar representation* of the graph.

Planar Graphs

- **Definition** A graph is called *planar* if it can be drawn in the plane without any edges crossing. Such a drawing is called a *planar representation* of the graph.

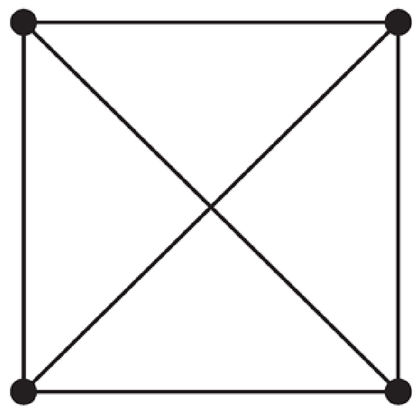
Example Is K_4 planar?

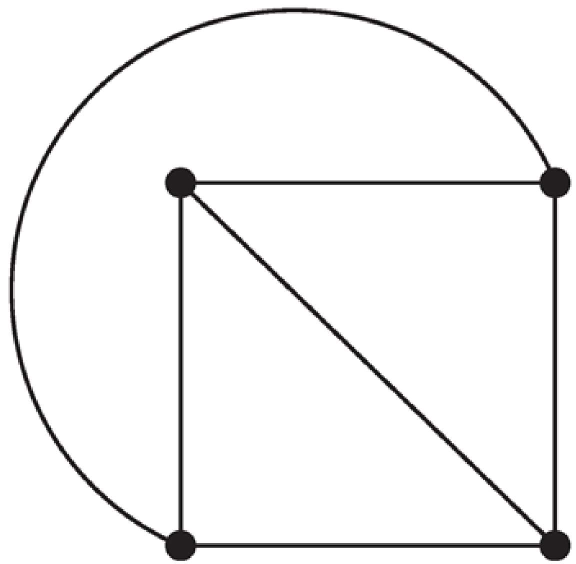


Planar Graphs

- **Definition** A graph is called *planar* if it can be drawn in the plane without any edges crossing. Such a drawing is called a *planar representation* of the graph.

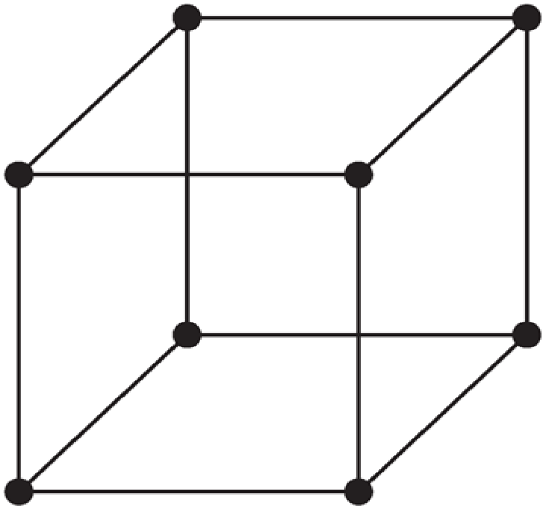
Example Is K_4 planar?





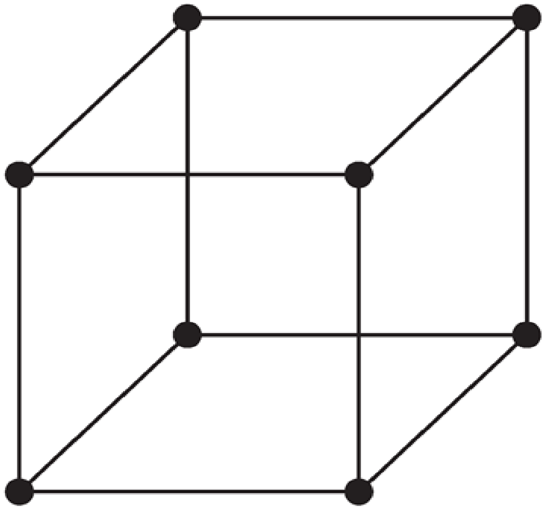
Planar Graphs

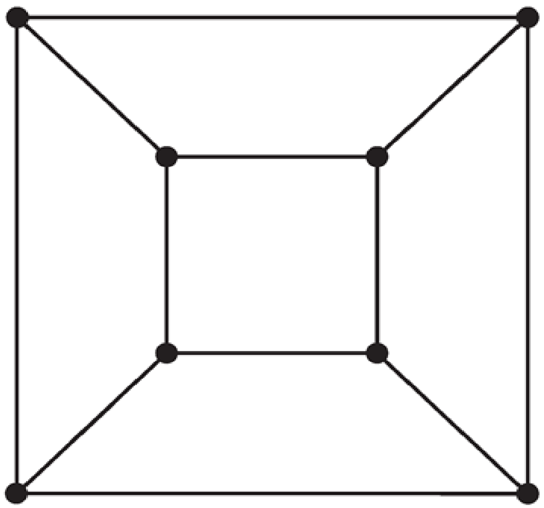
- **Example**



Planar Graphs

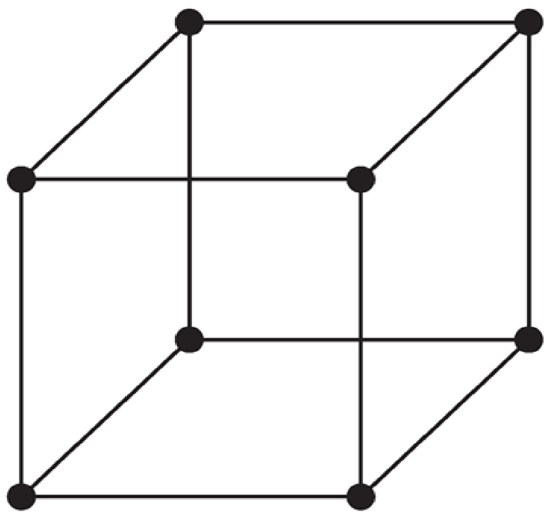
- **Example**

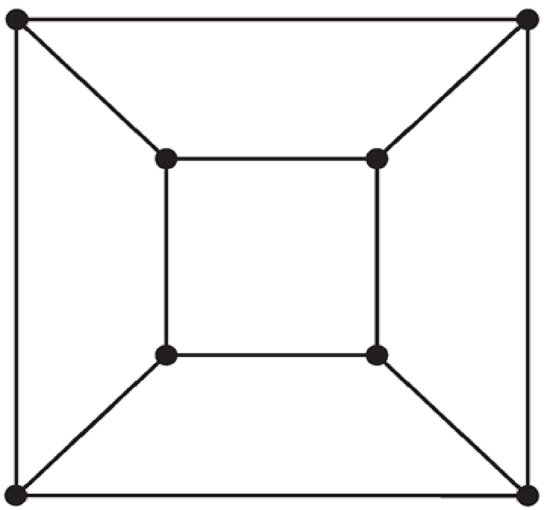


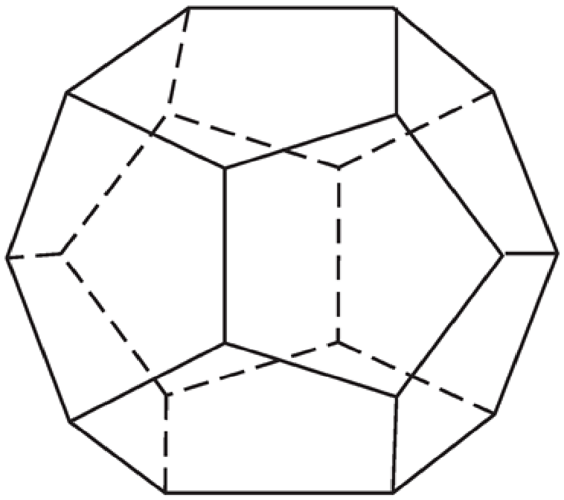


Planar Graphs

- **Example**



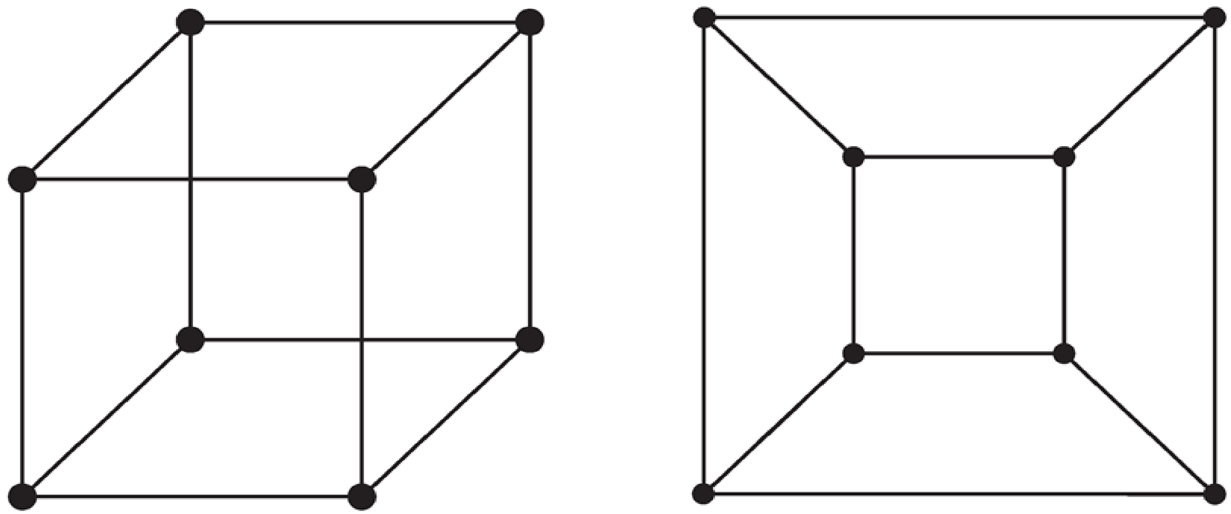


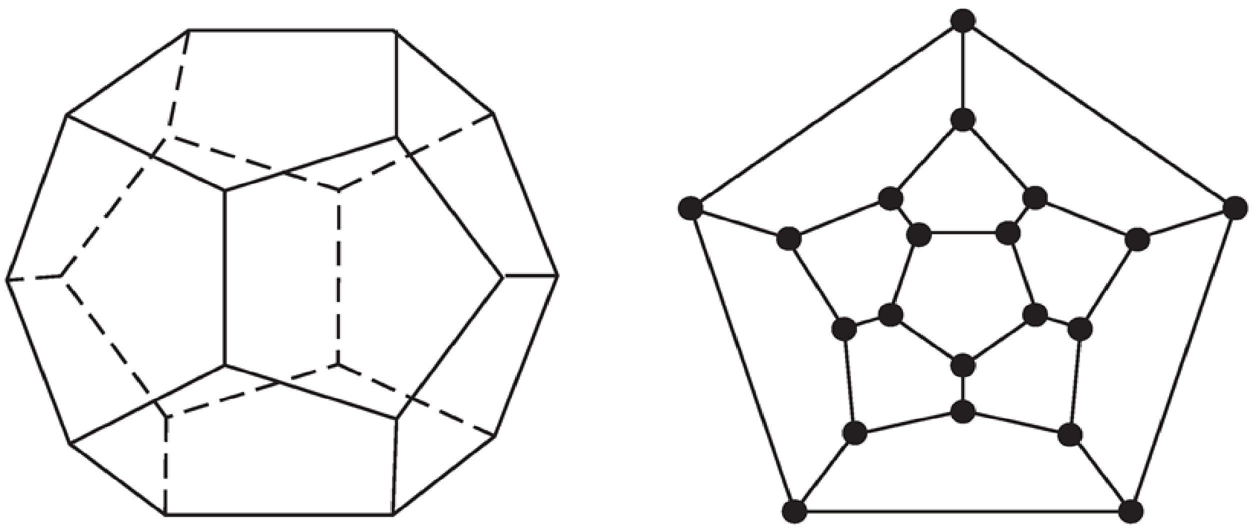


(a)

Planar Graphs

- **Example**



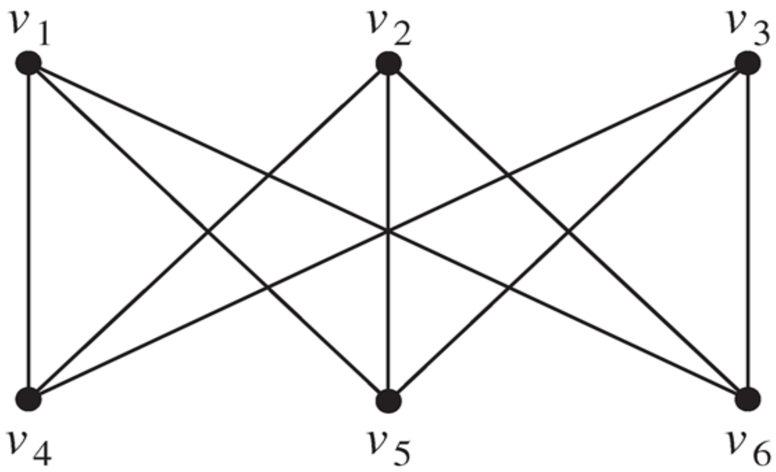


(a)

(b)

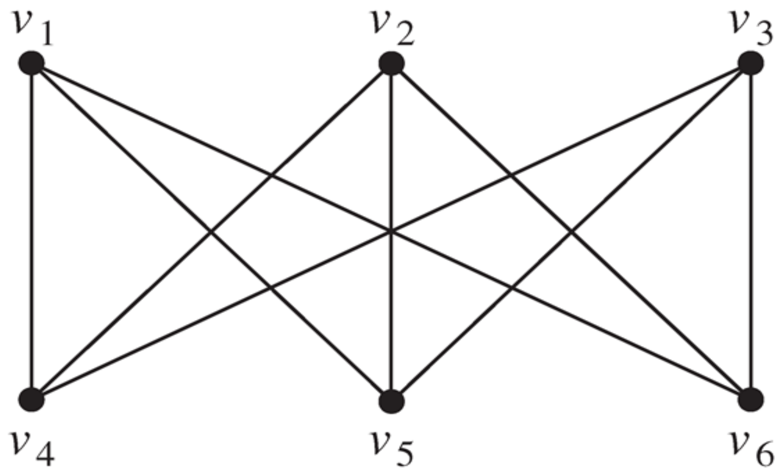
Planar Graphs

- **Example**



Planar Graphs

- **Example**



Applications

◇ IC design
◇ degin of road networks

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (by induction)

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (**by induction**)

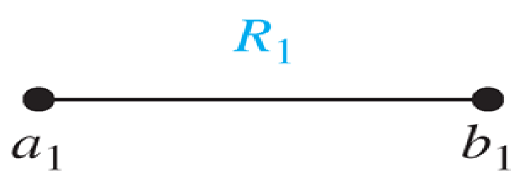
  Basic step:

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (**by induction**)

  Basic step:



Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (by induction)

  Basic step:

  Inductive Hypothesis:

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (by induction)

  Basic step:

  Inductive Hypothesis:

$$r_k = e_k - v_k + 2$$

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (by induction)

  Basic step:

  Inductive Hypothesis:

  Inductive step:

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

  Proof (by induction)

  Basic step:

  Inductive Hypothesis:

  Inductive step:
  Let $\{a_{k+1}, b_{k+1}\}$ be the edge that is added to G_k to obtain G_{k+1}.

Euler's Formula

- **Theorem** (Euler's Formula) Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

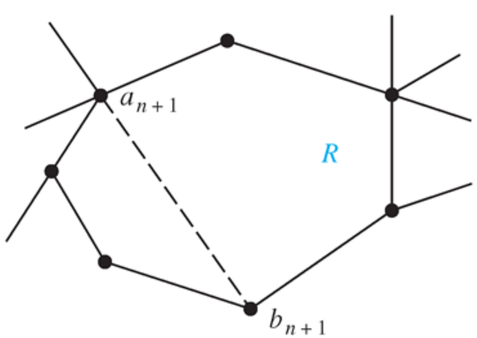
  Proof (by induction)

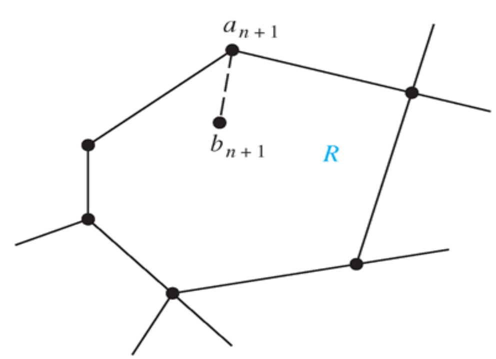
  Basic step:

  Inductive Hypothesis:

  Inductive step:

  Let $\{a_{k+1}, b_{k+1}\}$ be the edge that is added to G_k to obtain G_{k+1}.



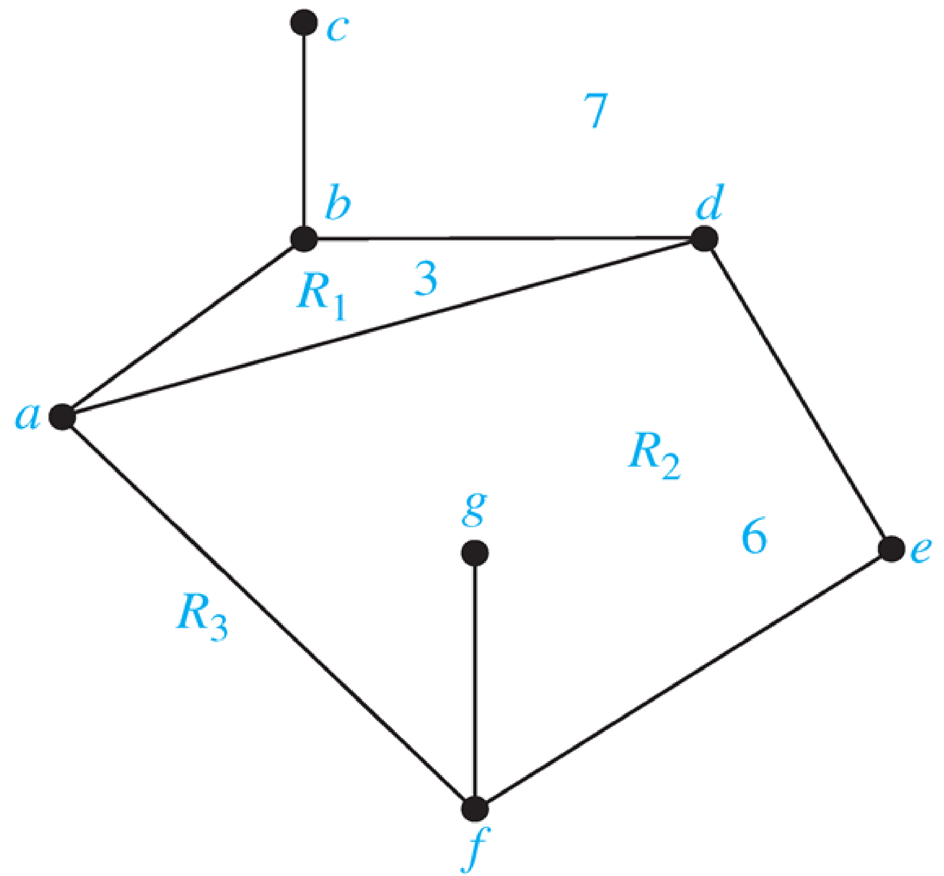


The Degree of Regions

- **Definition** The *degree* of a region is defined to be the number of edges on the boundary of this region. When an edge occurs twice on the boundary, it contributes two to the degree.

The Degree of Regions

- **Definition** The *degree* of a region is defined to be the number of edges on the boundary of this region. When an edge occurs twice on the boundary, it contributes two to the degree.



Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

  Proof The degree of every region is at least 3.

Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

  Proof The degree of every region is at least 3.

  - ◇ G is simple
  - ◇ $v \geq 3$

Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

  Proof The degree of every region is at least 3.

  - G is simple
  - $v \geq 3$

  The sum of the degrees of the regions is exactly twice the number of edges in the graph.

Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

  Proof The degree of every region is at least 3.

  - G is simple
  - $v \geq 3$

  The sum of the degrees of the regions is exactly twice the number of edges in the graph.

$$2e = \sum_{\text{all regions } R} \deg(R) \geq 3r$$

Corollaries

- **Corollary 1** If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

  Proof The degree of every region is at least 3.

  - G is simple
  - $v \geq 3$

  The sum of the degrees of the regions is exactly twice the number of edges in the graph.

  $$2e = \sum_{\text{all regions } R} \deg(R) \geq 3r$$

  By Euler's formula, the proof is completed.

Corollaries

- **Corollary 2** If G is a connected planar simple graph, then G has a vertex of degree not exceeding 5.

Corollaries

- **Corollary 2** If G is a connected planar simple graph, then G has a vertex of degree not exceeding 5.

  Proof

Corollaries

- **Corollary 2** If G is a connected planar simple graph, then G has a vertex of degree not exceeding 5.

  Proof

  (By contradiction)

  By Corollary 1 and the Handshaking Theorem.

Corollaries

- **Corollary 2** If G is a connected planar simple graph, then G has a vertex of degree not exceeding 5.

  Proof

  (By contradiction)

  By Corollary 1 and the Handshaking Theorem.

  Corollary 3 In a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

Corollaries

- **Corollary 2** If G is a connected planar simple graph, then G has a vertex of degree not exceeding 5.

  Proof

  (By contradiction)

  By Corollary 1 and the Handshaking Theorem.

  Corollary 3 In a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

  Proof similar to that of Corollary 1.

Examples

- Show that K_5 is nonplanar.

Examples

- Show that K_5 is nonplanar.

  Using Corollary 1

Examples

- Show that K_5 is nonplanar.

  Using Corollary 1

  Show that $K_{3,3}$ is nonplanar.

Examples

- Show that K_5 is nonplanar.

  Using Corollary 1

  Show that $K_{3,3}$ is nonplanar.

  Using Corollary 3

Examples

- Show that K_5 is nonplanar.

  Using Corollary 1

  Show that $K_{3,3}$ is nonplanar.

  Using Corollary 3

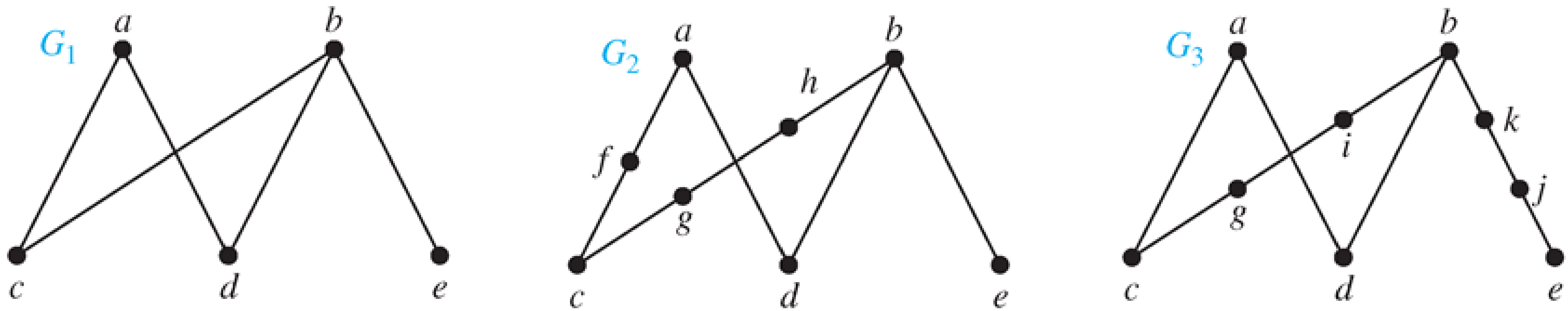
  Corollary 2 is used in the proof of Five Color Theorem.

Kuratowski's Theorem

- **Definition** If a graph is planar, so will be any graph obtained by removing an edge $\{u, v\}$ and adding a new vertex w together with edges $\{u, w\}$ and $\{w, v\}$. Such an operation is called an *elementary subdivision*. The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *homomorphic* if they can be obtained from the same graph by a sequence of elementary subdivisions.

Kuratowski's Theorem

- **Definition** If a graph is planar, so will be any graph obtained by removing an edge $\{u, v\}$ and adding a new vertex w together with edges $\{u, w\}$ and $\{w, v\}$. Such an operation is called an *elementary subdivision*. The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *homomorphic* if they can be obtained from the same graph by a sequence of elementary subdivisions.

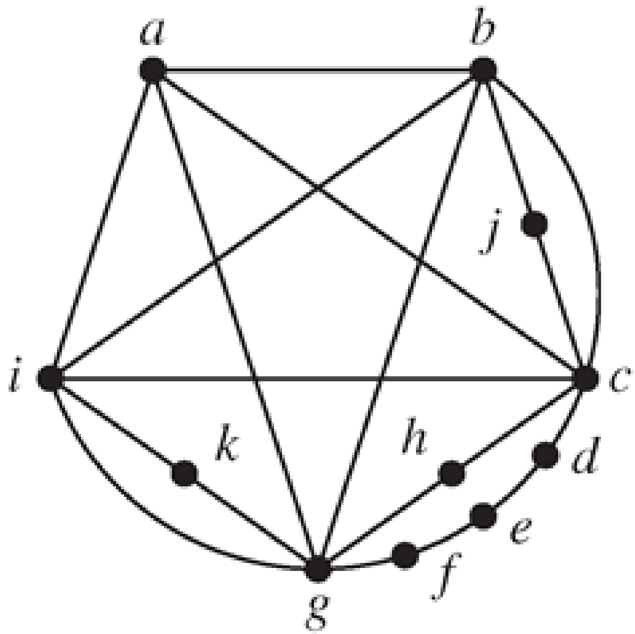


Kuratowski's Theorem

- **Definition** If a graph is planar, so will be any graph obtained by removing an edge $\{u, v\}$ and adding a new vertex w together with edges $\{u, w\}$ and $\{w, v\}$. Such an operation is called an *elementary subdivision*. The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *homomorphic* if they can be obtained from the same graph by a sequence of elementary subdivisions.

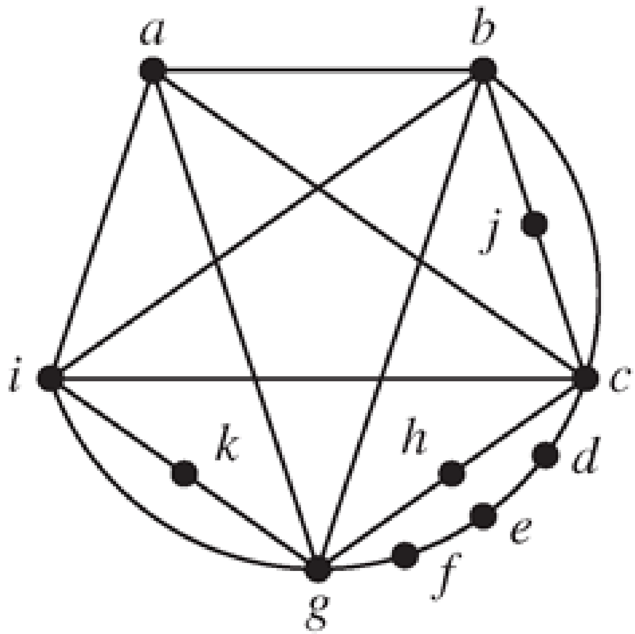
  Theorem A graph is nonplanar if and only if it contains a subgraph homomorphic to $K_{3,3}$ or K_5.

Examples

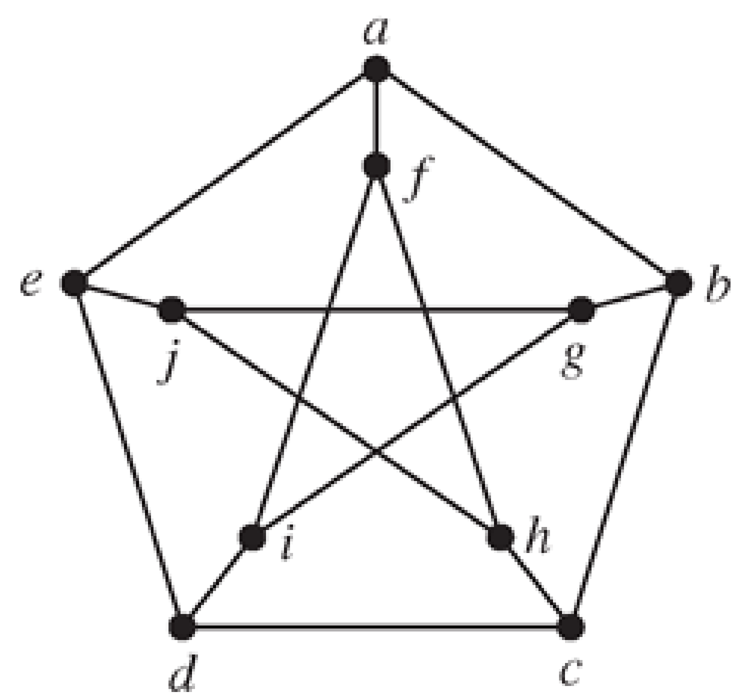


G

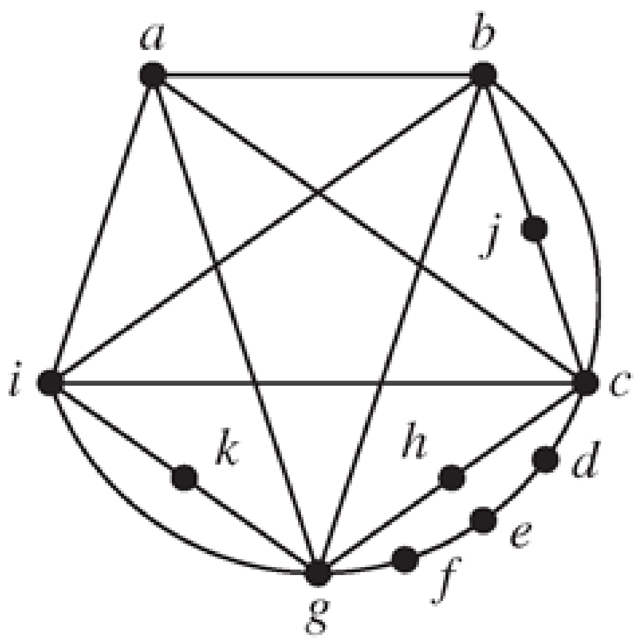
Examples



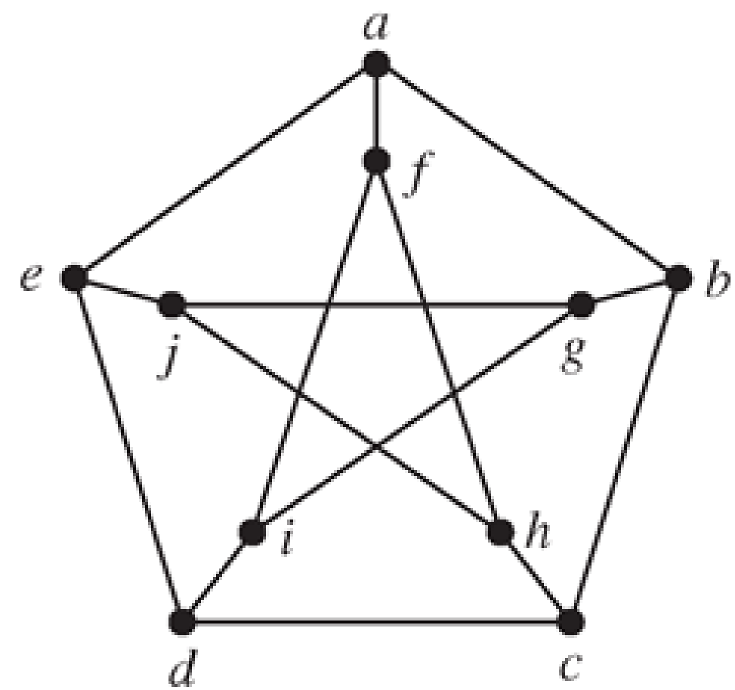
G

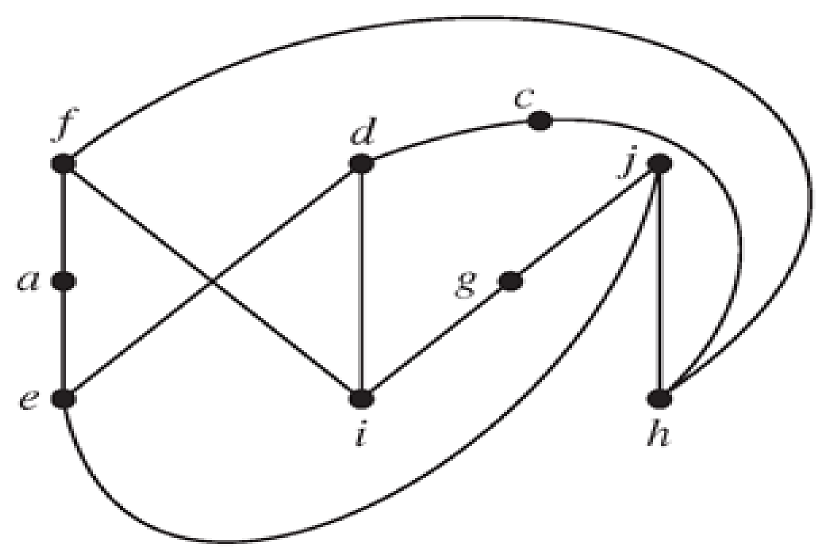


Examples



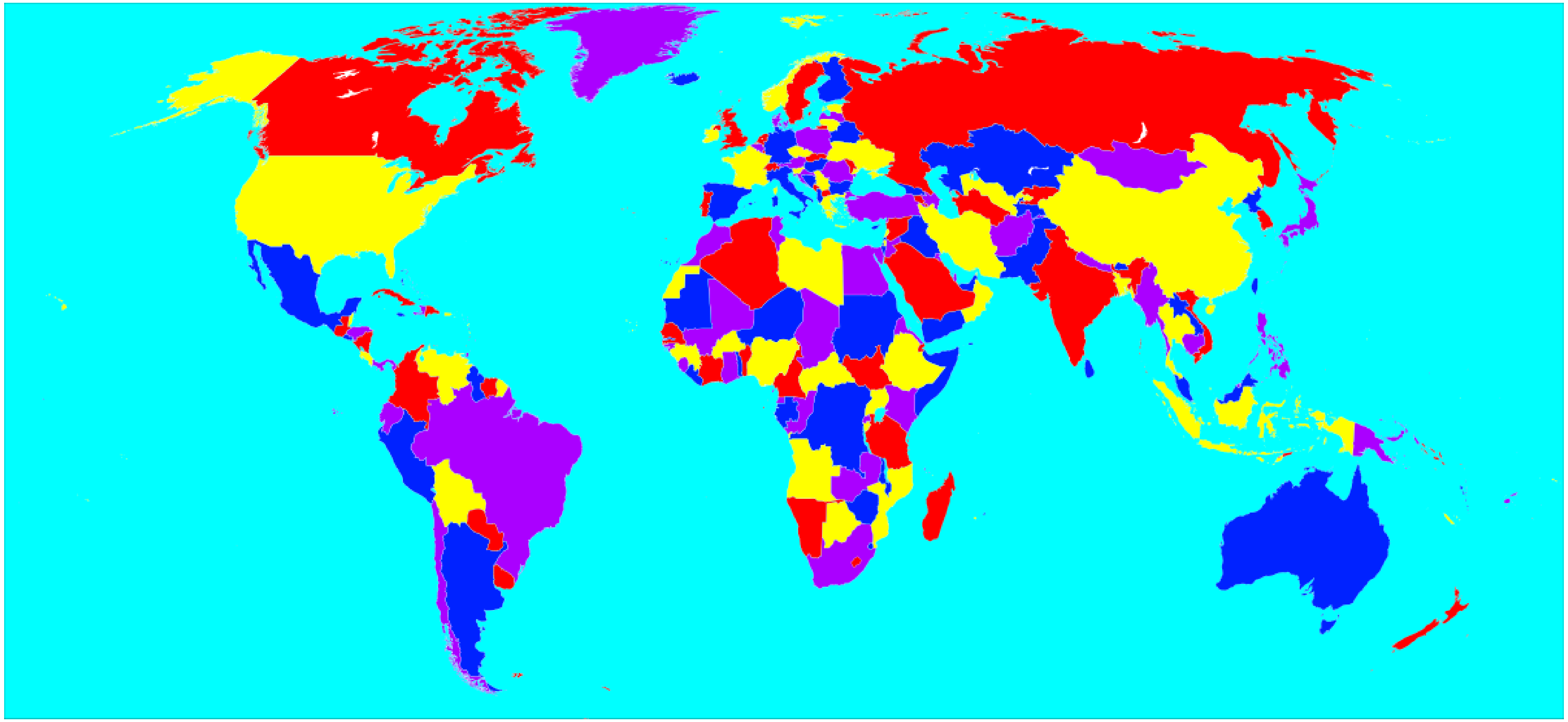
G





Graph Coloring

- **Four-color theorem** Given any separation of a plane into contiguous regions, producing a figure called a *map*, no more that four colors are required to color the regions of the map so that no two adjacent regions have the same color.



Four-color theorem

◇ first proposed by Francis Guthrie in 1852

◇ his brother Frederick Guthrie told Augustus De Morgan

◇ De Morgan wrote to William Hamilton

◇ Alfred Kempe proved it incorrectly in 1879

◇ Percy Heawood found an error in 1890 and proved the *five-color theorem*

◇ Finally, Kenneth Appel and Wolfgang Haken proved it with case by case analysis by computer in 1976 (*the first computer-aided proof*)

◇ Kempe's incorrect proof serves as a basis

Graph Coloring

- A *coloring* of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

Graph Coloring

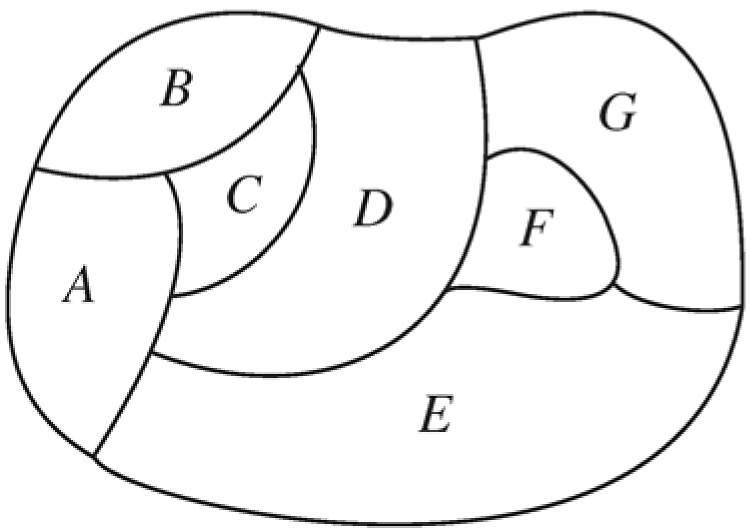
- A *coloring* of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

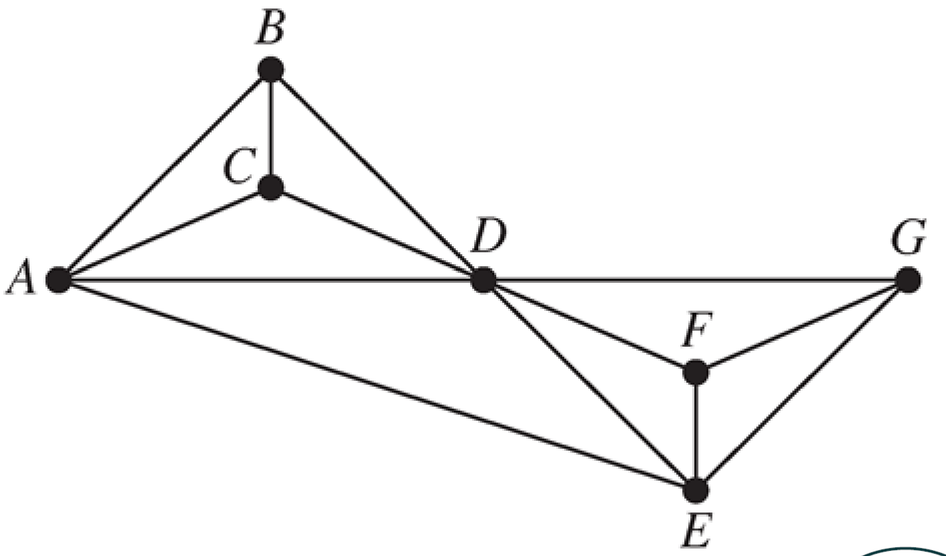
  The *chromatic number* of a graph is the least number of colors needed for a coloring of this graph, denoted by $\chi(G)$.

Graph Coloring

- A *coloring* of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

  The *chromatic number* of a graph is the least number of colors needed for a coloring of this graph, denoted by $\chi(G)$.



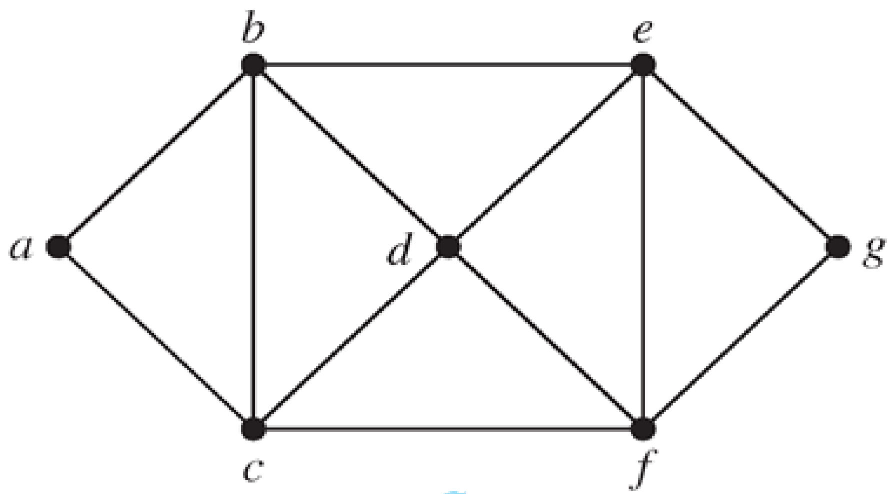


Graph Coloring

- **Theorem** (Four Color Theorem) The chromatic number of a planar graph is no greater than four.

Graph Coloring

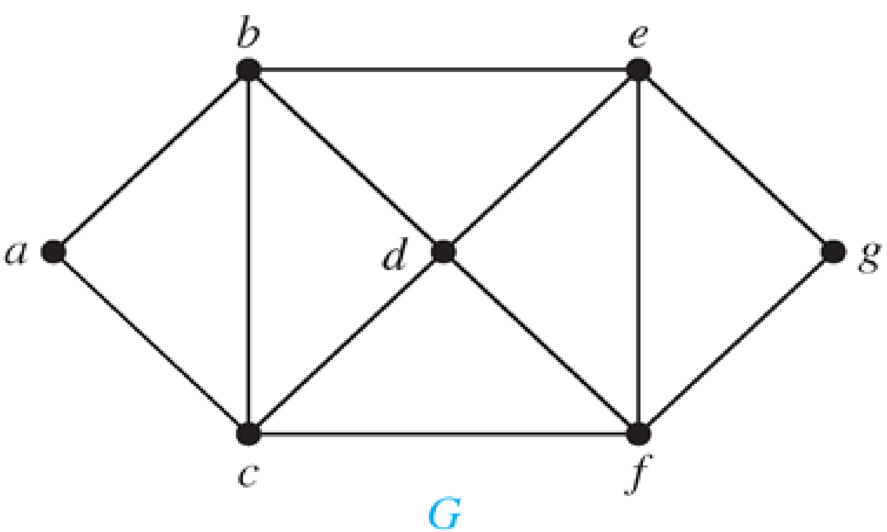
- **Theorem** (Four Color Theorem) The chromatic number of a planar graph is no greater than four.

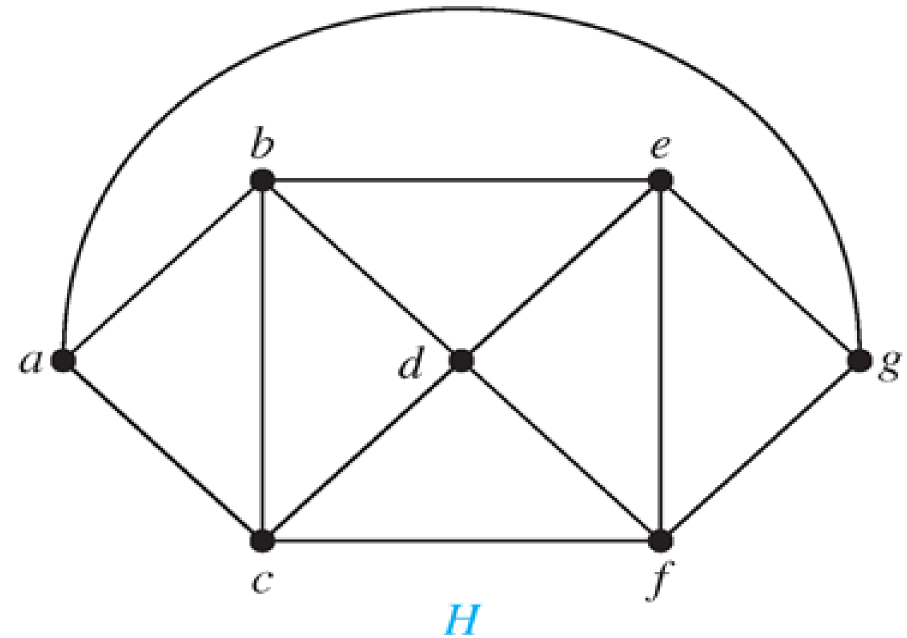


G

Graph Coloring

- **Theorem** (Four Color Theorem) The chromatic number of a planar graph is no greater than four.



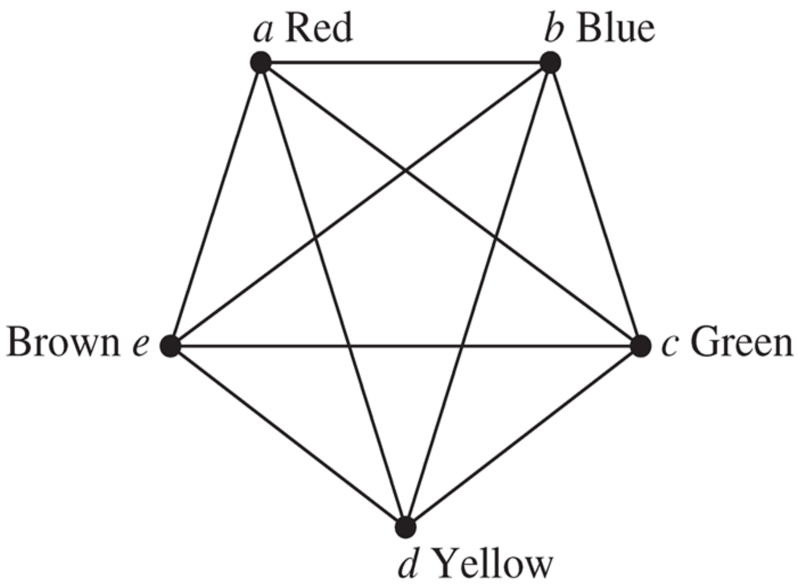


Examples

- What is the chromatic number of K_n, $K_{m,n}$, C_n?

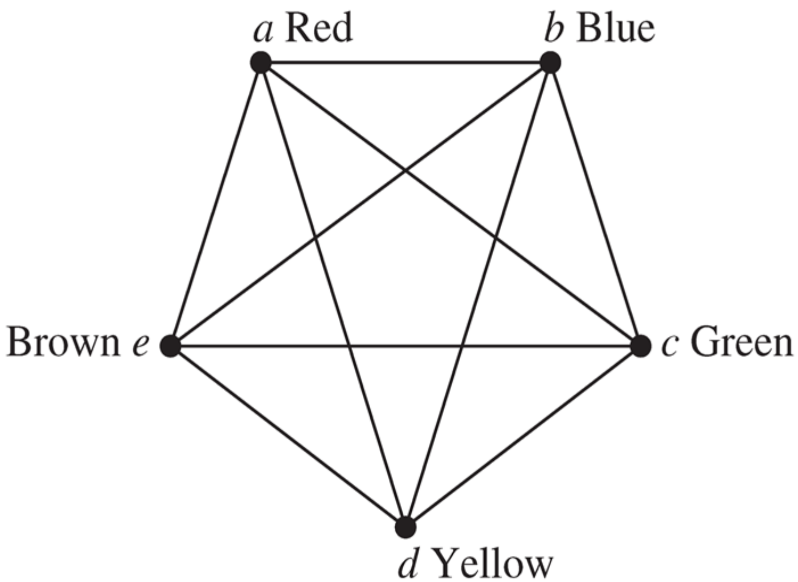
Examples

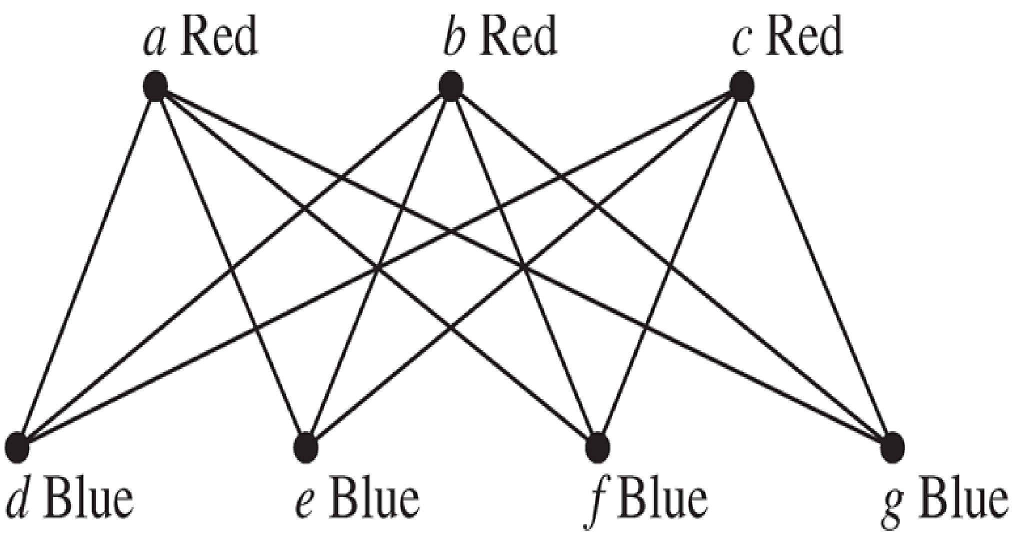
- What is the chromatic number of K_n, $K_{m,n}$, C_n?



Examples

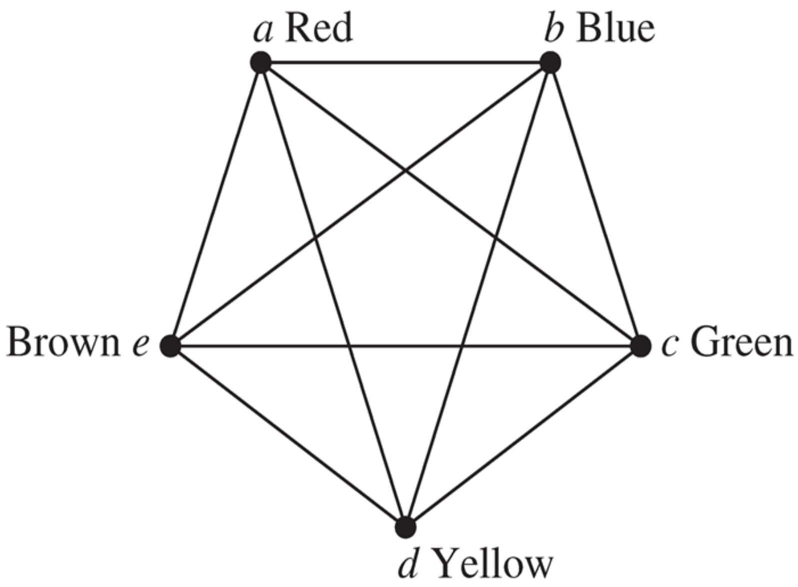
- What is the chromatic number of K_n, $K_{m,n}$, C_n?

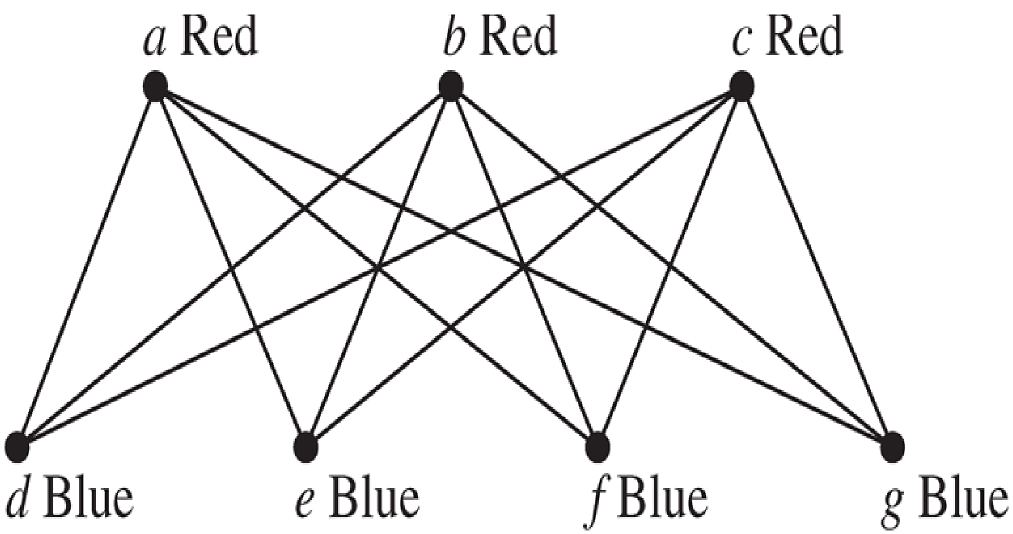


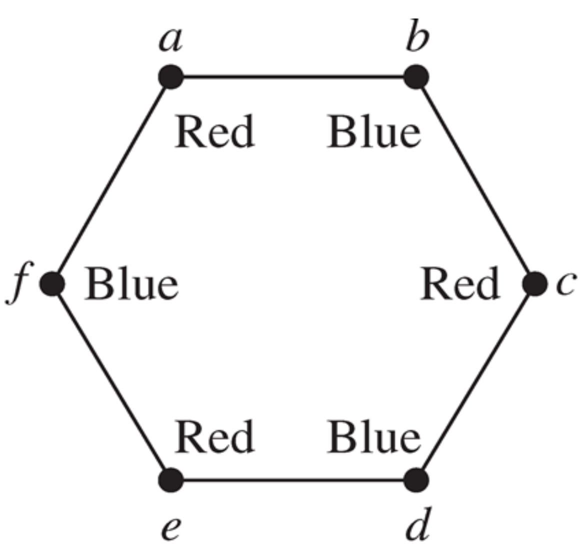


Examples

- What is the chromatic number of K_n, $K_{m,n}$, C_n?

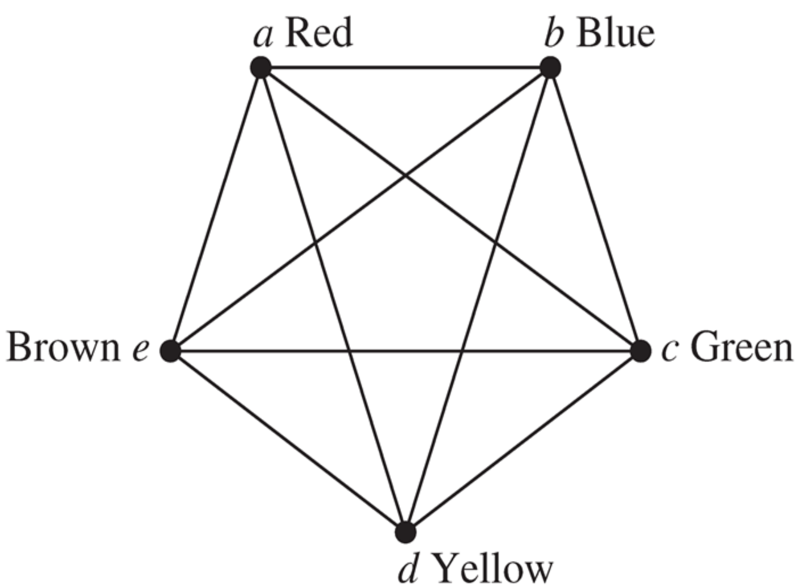


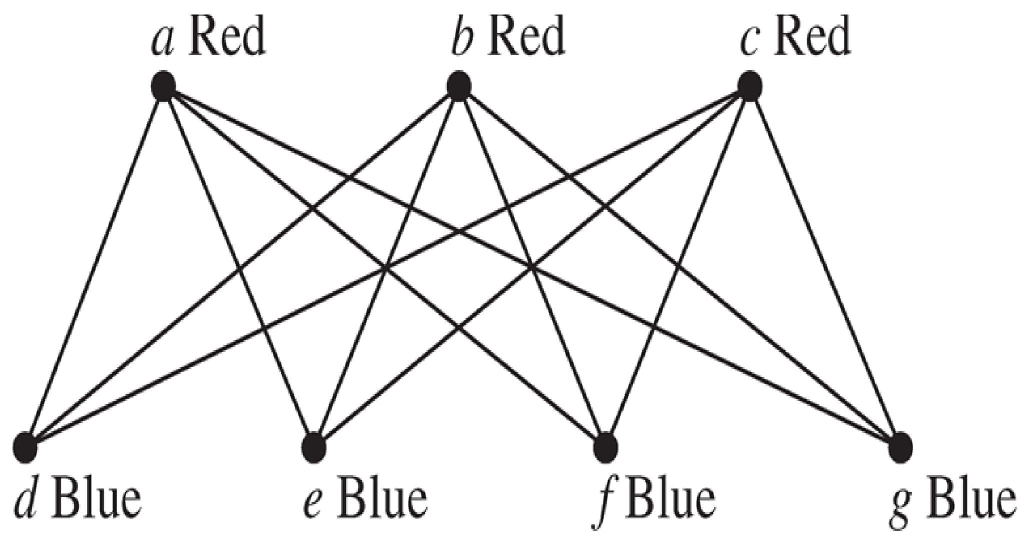


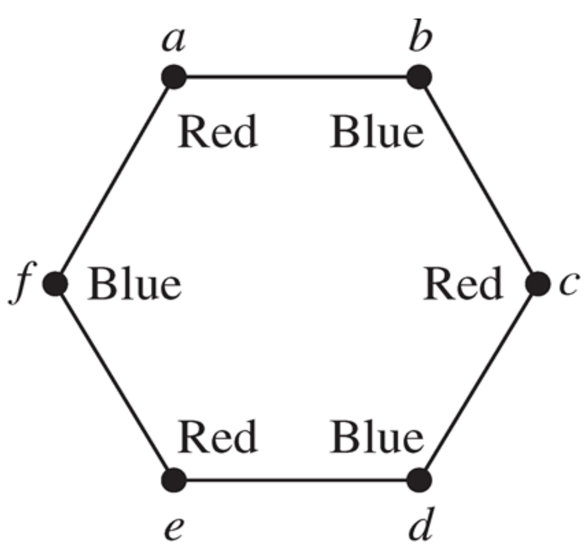


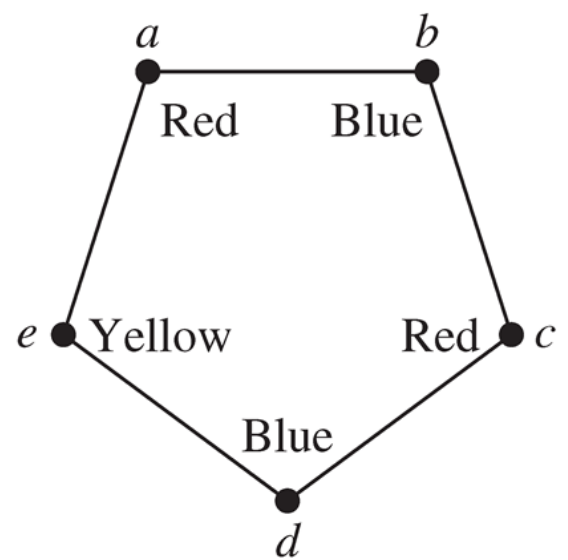
Examples

- What is the chromatic number of K_n, $K_{m,n}$, C_n?





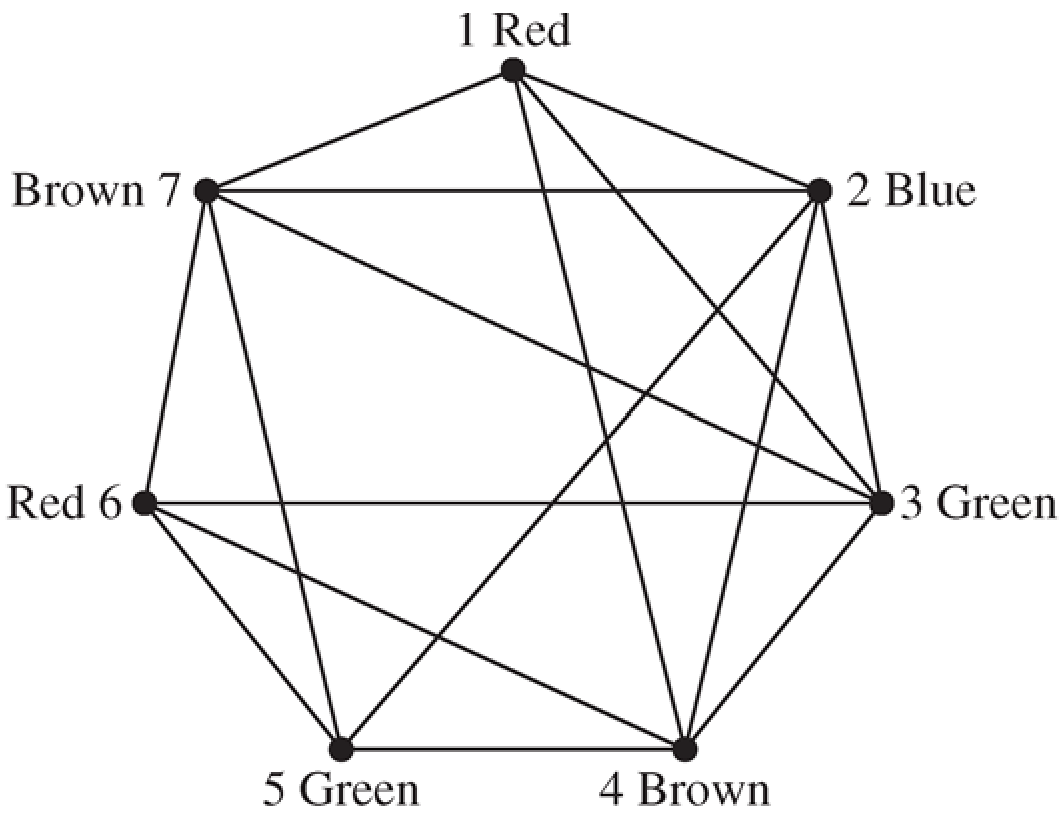




Applications of Graph Coloring

- Scheduling Final Exams

  Vertices represent courses, and there is an edge between two vertices if there is a common student in the courses.



Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

Applications of Graph Coloring

- ## Channel Assignments

  Televesion channels 2 through 13 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel . How can the assignment of channels be modeled by graph coloring?

Applications of Graph Coloring

- Channel Assignments

  Televesion channels 2 through 13 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel . How can the assignment of channels be modeled by graph coloring?

Graph Coloring $\in$ NPC

Announcements

- **Homework assignment** 6

  ◇ P650 Ex. 11, P668 Ex. 64, 66, P676 Ex. 28, 29, 32, P677 Ex. 45, 52, P690 Ex. 20, P691 Ex. 28, P704 Ex. 10, P716 Ex.2, 14, P725 Ex. 6, P726 Ex. 17, P756 Ex. 16, P757 Ex. 46, P784 Ex. 24, P795 Ex. 11, 12, P802 Ex. 2, 6

  ◇ Due on *Jan. 2nd, 2018 at the beginning of class*

  ◇ Please write your homeowrk **neatly**, as a courtesy to graders.

Next Lecture

- Trees ...