TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Discrete Mathematics

Nguyễn Khánh Phương

Department of Computer Science
School of Information and Communication Technology
E-mail: phuongnk@soict.hust.edu.vn

PART 1
COMBINATORIAL THEORY
(Lý thuyết tổ hợp)

PART 2
GRAPH THEORY
(Lý thuyết đồ thị)

Content of Part 2

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chapter 1

Fundamental concepts

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Content

1. Graph in practice

2. Graph types

3. Degree of vertex

4. Subgraph

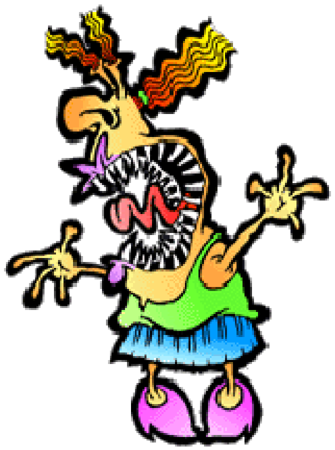
5. Isomorphism of Graphs

6. Path and cycle

7. Connectedness

8. Special graphs

9. Graph Coloring problem

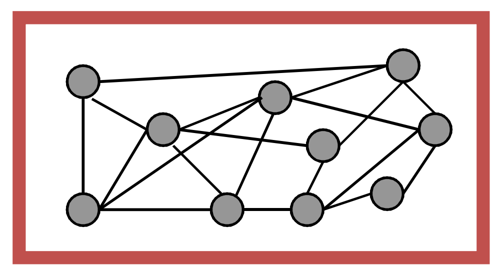


What is graph?

- In Maths:
  Drawing or diagram represent data by using coordinate system

Not what we want to mention

- In discrete mathematics:
  This is discrete structure with highly intuitive, very useful for expressing relationships.

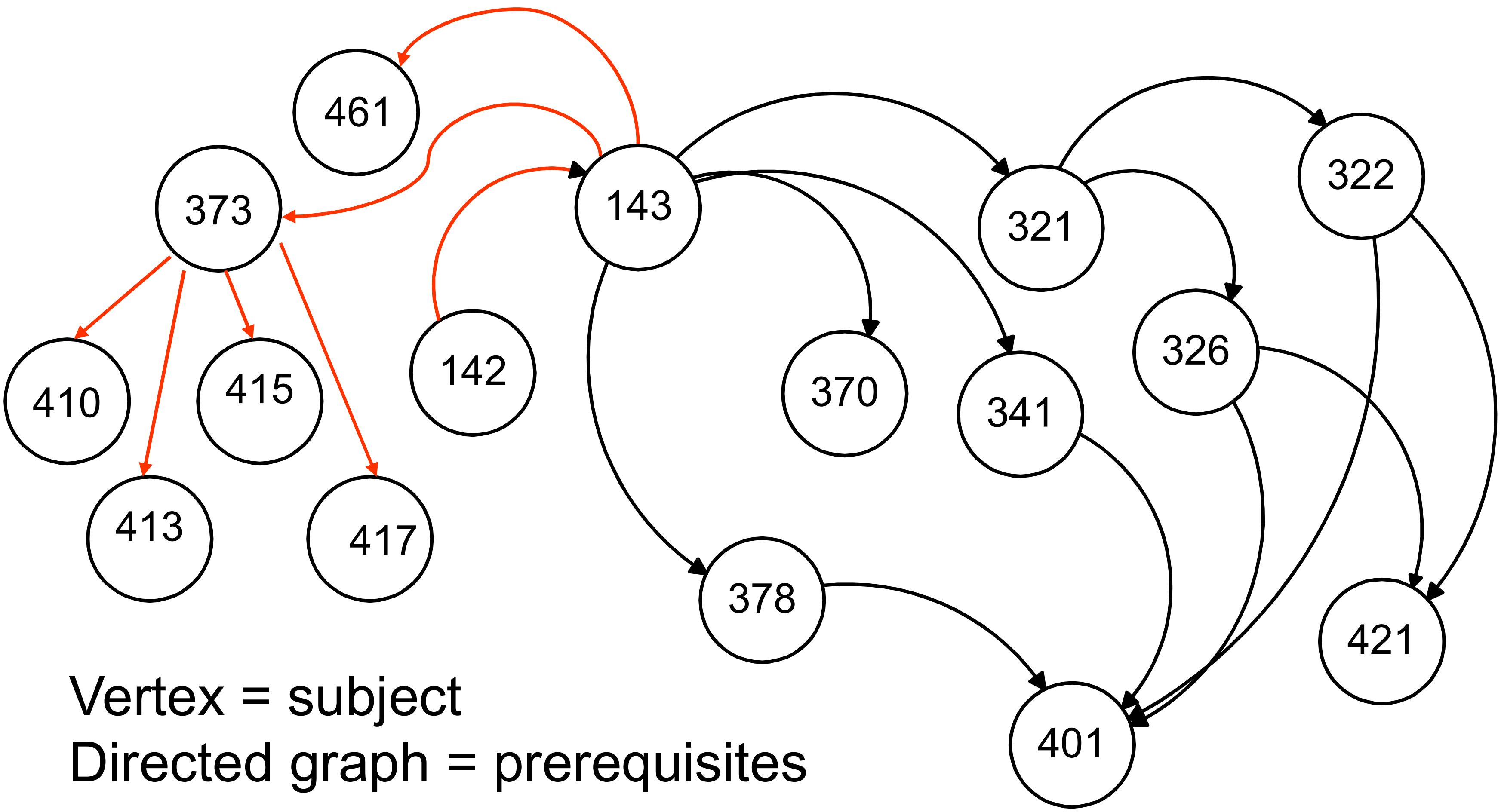


Applications of graph in practice

Has the potential to be applied in many fields:

- Internet
- Traffic network
- Electrical network
- Water supply network
- Scheduling
- Flow optimization, circuit design
- DNA gene analysis
- Computer games
- Object oriented design
- ….

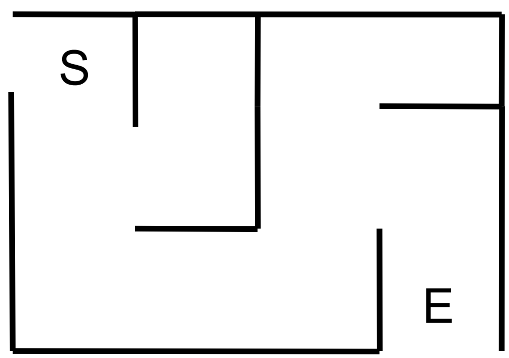
Relationship between subjects

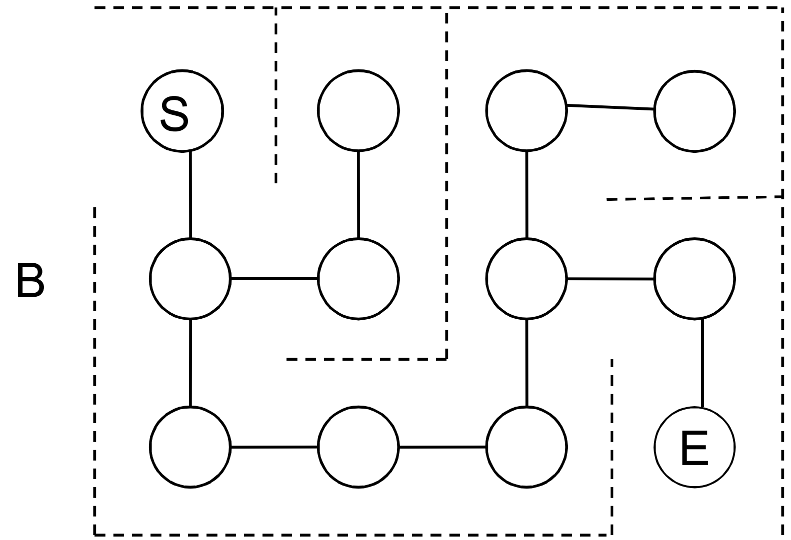


Vertex = subject
Directed graph = prerequisites

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Represent maze

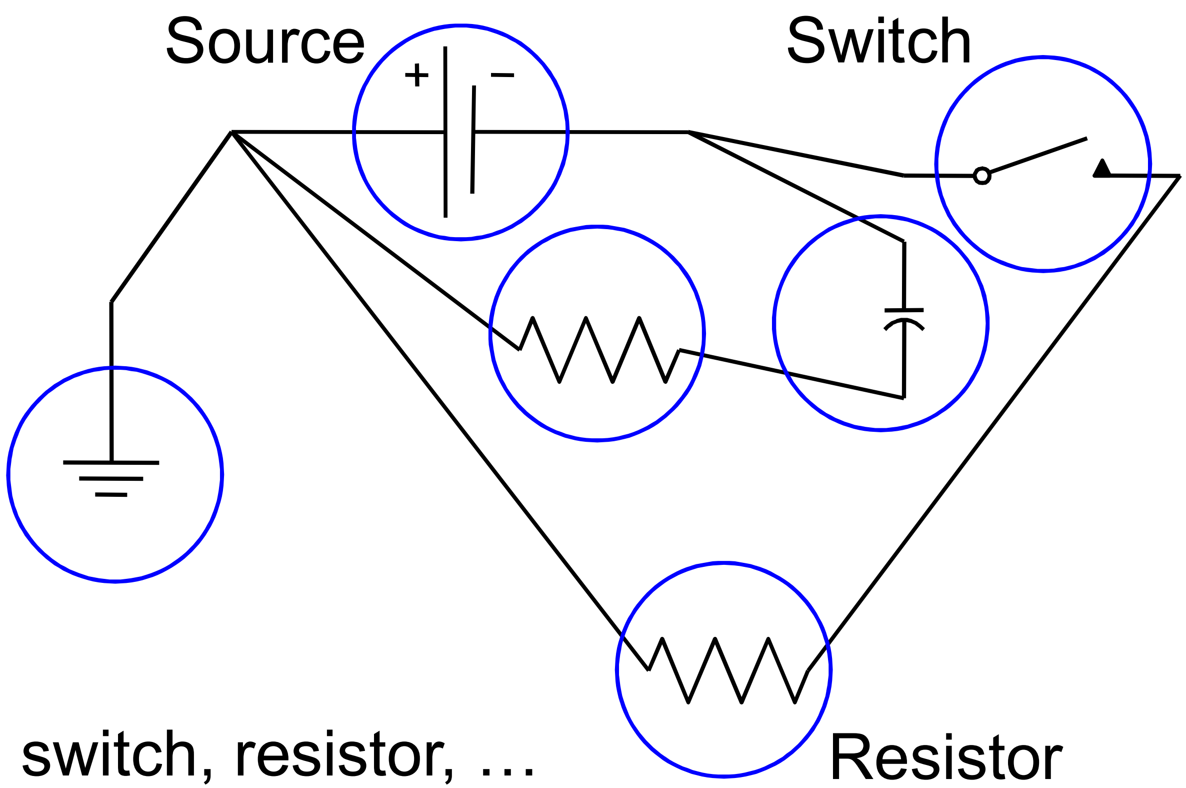




Vertex = room
Edge = doorways or hallways

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

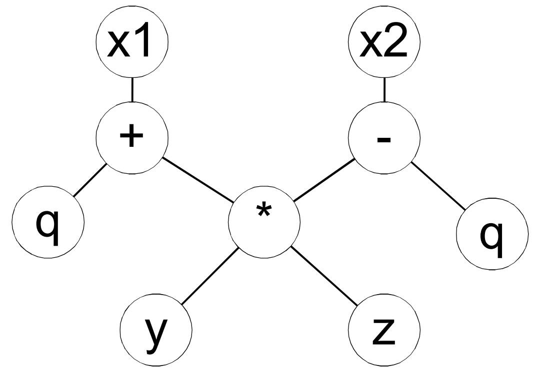
Electrical Circuits



Vertex = source, switch, resistor, …
Edge = connecting wire

Program statements

```
x1=q+y*z
x2=y*z-q
```

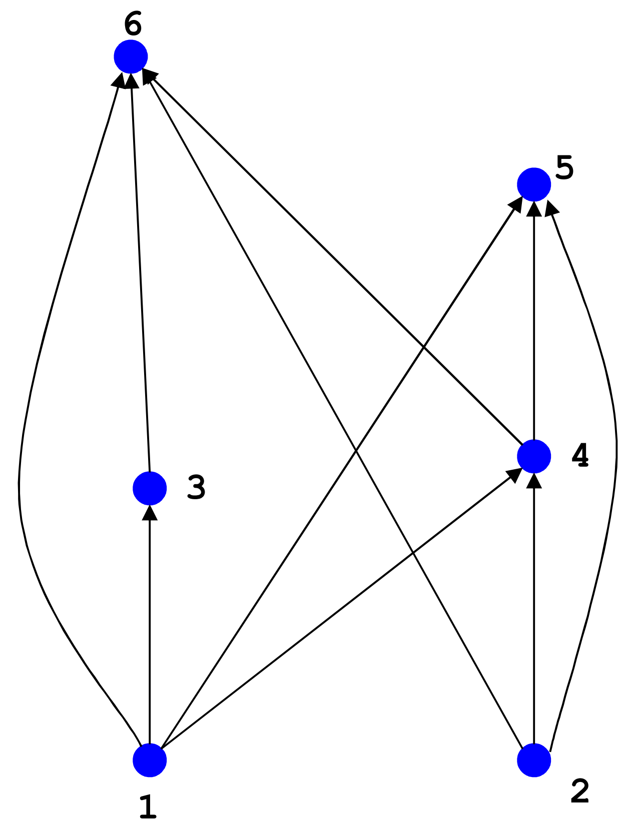


Vertex = notation/operator
Edge = relationship

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Precedence

S_1	`a=0;`
S_2	`b=1;`
S_3	`c=a+1`
S_4	`d=b+a;`
S_5	`e=d+1;`
S_6	`e=c+d;`



Which statements need to execute before S_6?

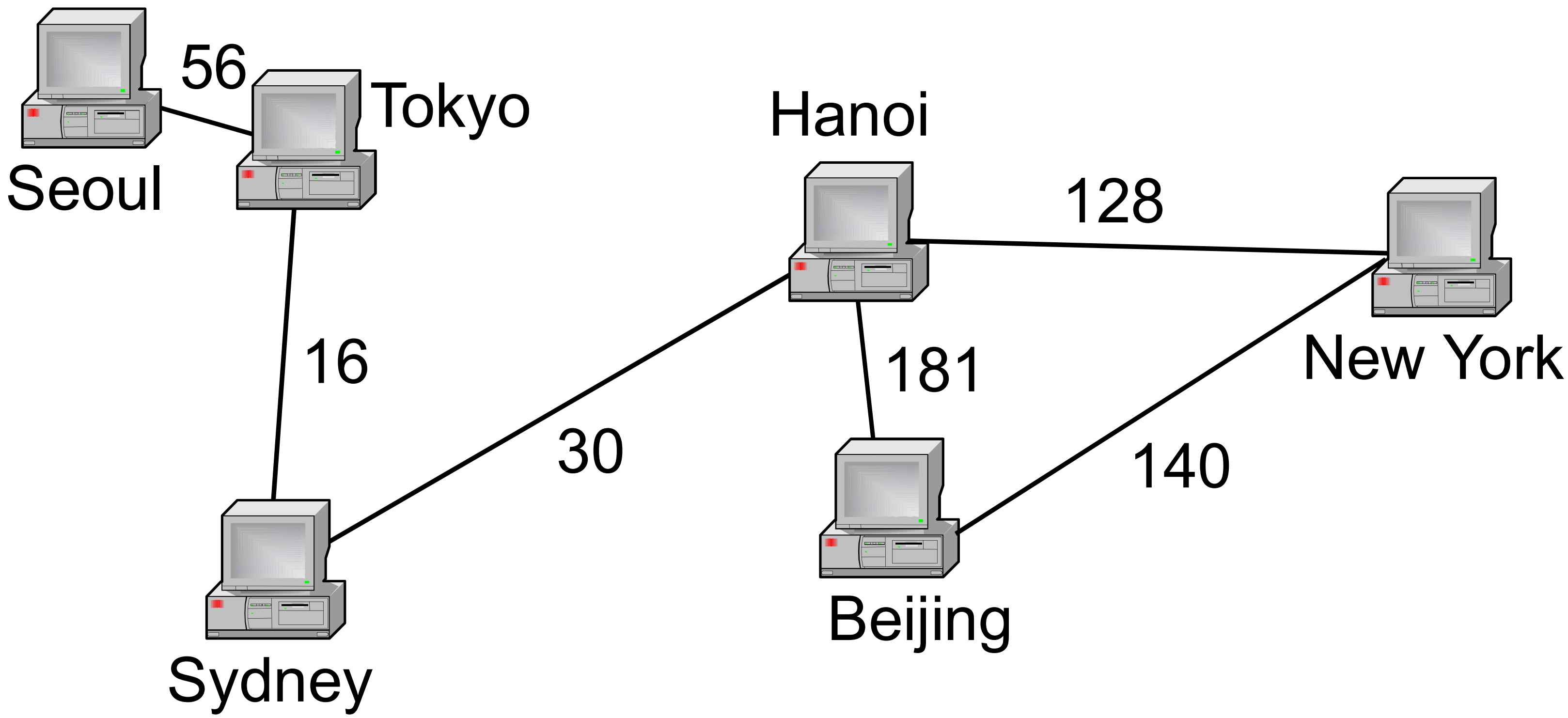
S_1, S_2, S_3, S_4

Vertex = statement

Edge = precedence

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Information Transmission in a Computer Network



Vertex = computer
Edge = communication speed

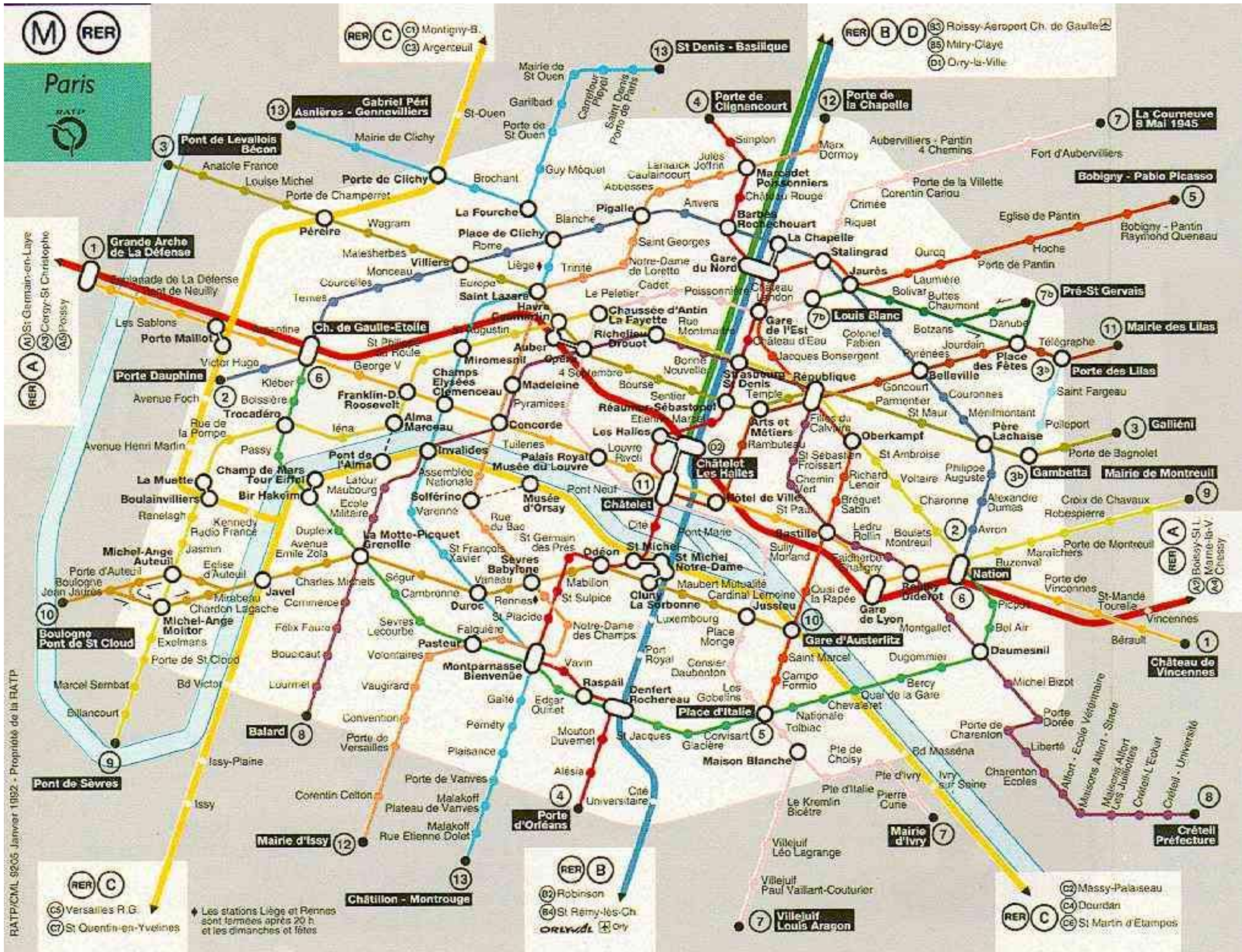
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Traffic Flow on Highways

Vertex = city
Edge = the amount of traffic on highways connecting cities

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

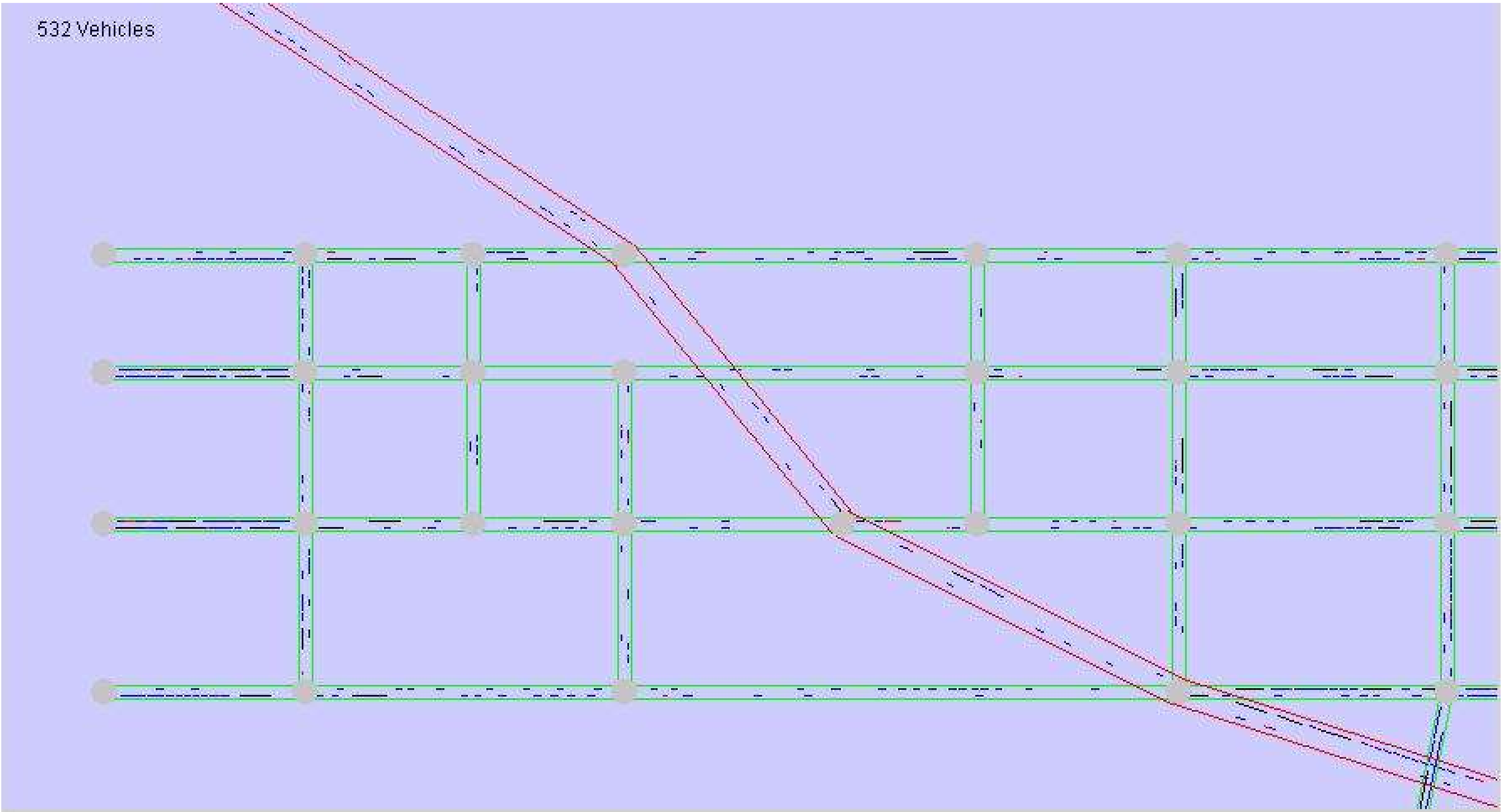
Bus system



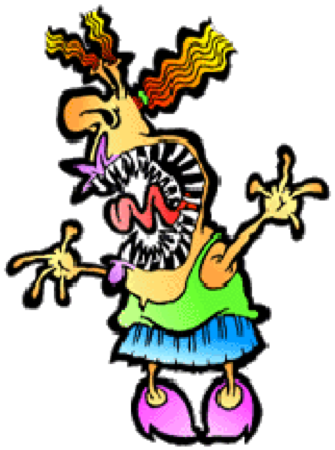
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Subway system

Street map



Content

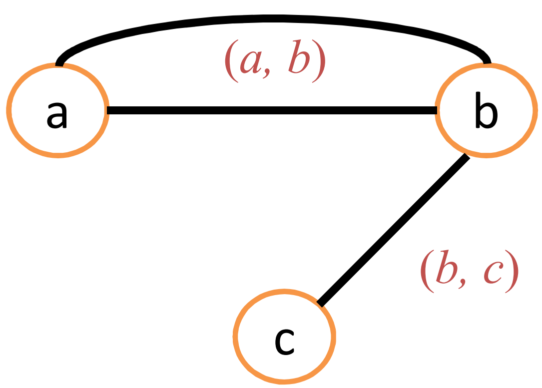


Undirected Graphs (Đồ thị vô hướng)

Definition. Undirected simple (multi) graph $G = (V,E)$ consists of 2 sets:

- Vertex set V is a finite set, each element is called vertex
- Edge set E is the set (family) of unordered pairs

$$(u, v), \text{ where } u, v \in V, u \neq v$$



Undirected Multigraph

Undirected (simple) graph

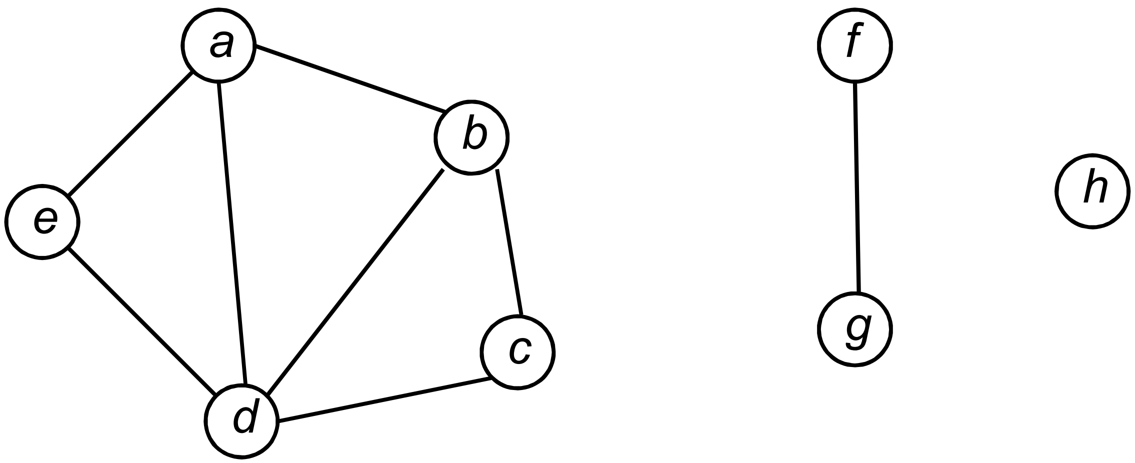
(a, b): multiple edges (parallel edges)

Undirected Simple Graph

- **Example:** Simple graph $G_1 = (V_1, E_1)$, where

$V_1=\{a, b, c, d, e, f, g, h\}$,

$E_1=\{(a,b), (b,c), (c,d), (a,d), (d,e), (a,e), (d,b), (f,g)\}$.

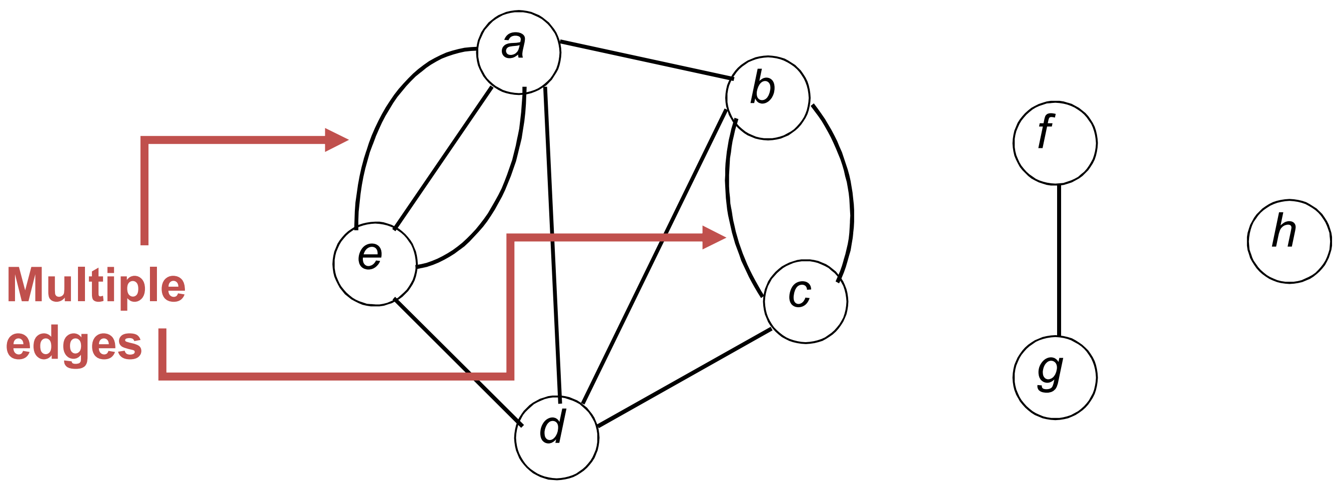


Graph G_1

Undirected MultiGraph

- **Example:** Multigraph $G_2 = (V_2, E_2)$, where $V_2=\{a, b, c, d, e, f, g, h\}$,

  $E_2=\{(a,b), (b,c), (b,c), (c,d), (a,d), (d,e), (a,e), (a,e), (a, e), (d,b), (f,g)\}$.



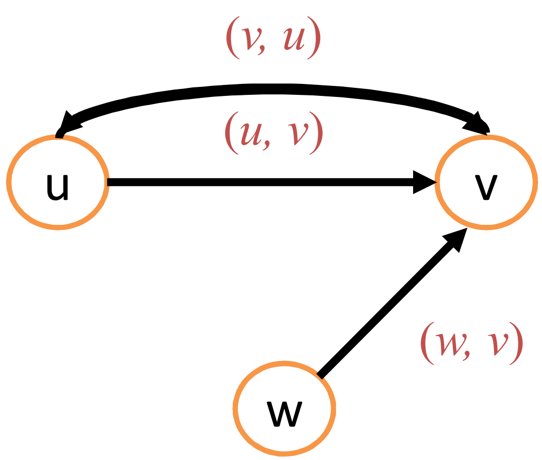
Graph G_2

Directed Graph

Definition. Directed simple (multi) graph $G = (V,E)$ consists of 2 sets:

- Vertex set V is finite element, each element is called vertex
- Edge set E is set (family) of ordered pairs

$$(u,\ v),\ \text{where } u,\ v \in V,\ u \neq v$$



Directed multigraph

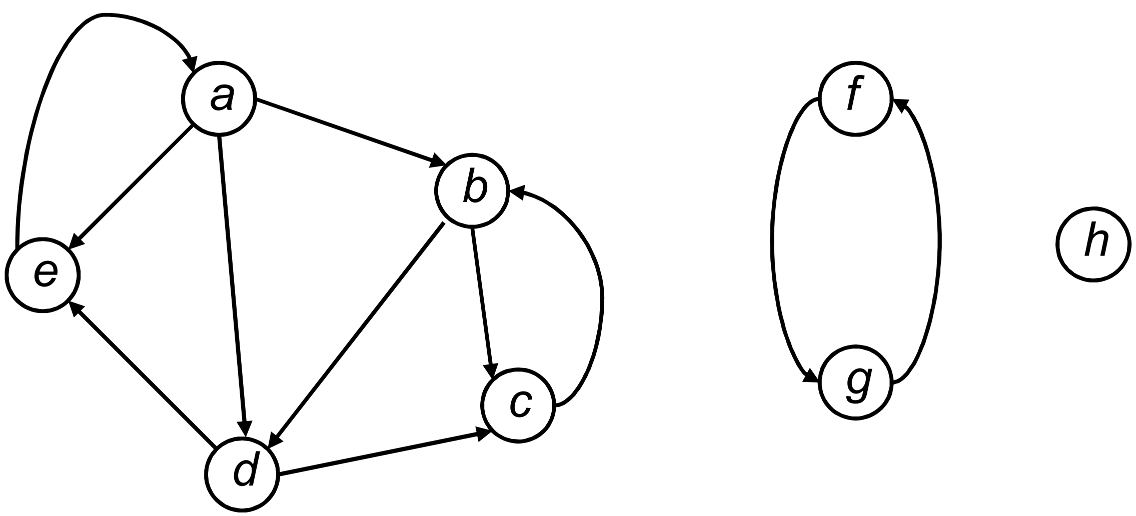
(Simple) Directed graph
~ Digraph

Simple digraph

Example: Simple digraph $G_3 = (V_3, E_3)$, where

$V_3 = \{a, b, c, d, e, f, g, h\}$,

$E_3 = \{(a,b), (b,c), (c,b), (d,c), (a,d), (b, d), (a,e), (d,e), (e,a), (f,g), (g,f)\}$



Graph G_3

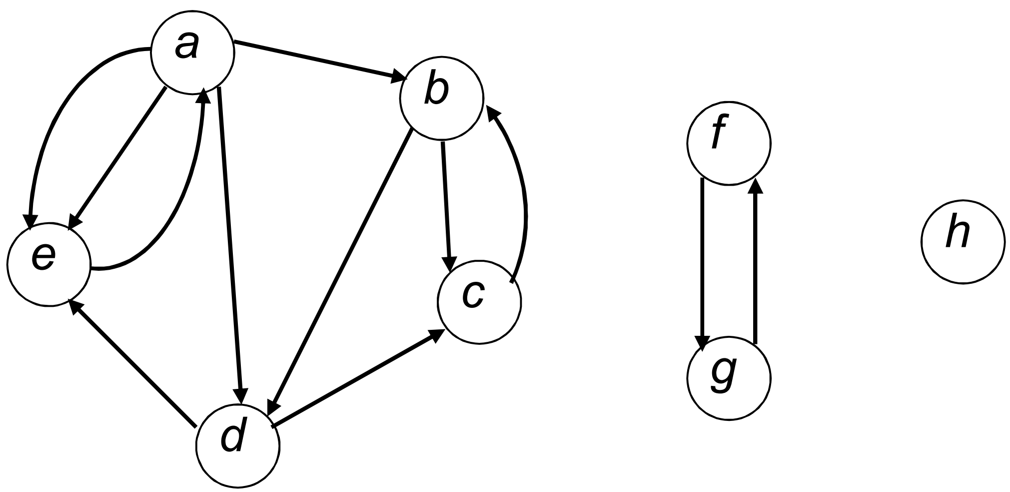
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Directed MultiGraph

Example: Directed multigraph $G_4 = (V_4, E_4)$, where

$V_4 = \{a, b, c, d, e, f, g, h\}$,

$E_4 = \{(a,b), (b,c), (c,b), (d,c), (a,d), (b, d), (a,e), (a,e),$
$(d,e), (e,a), (f,g), (g,f)\}$



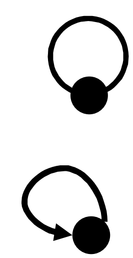
Graph G_4

Graph types: Summary

Type	Edge type	Has multiple edges?
Undirected graph	Undirected	No
Undirected multigraph	Undirected	Yes
Directed graph	Directed	No
Directed multigraph	Directed	Yes

- Note:
  - Pseudo graph (**Giả đồ thị)** is multigraph that contains loops as well as multiple edges between vertices

Loop (Khuyên)

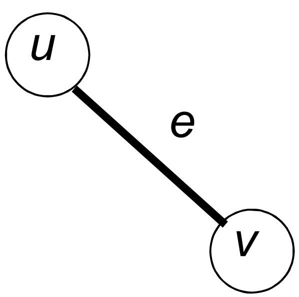


NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

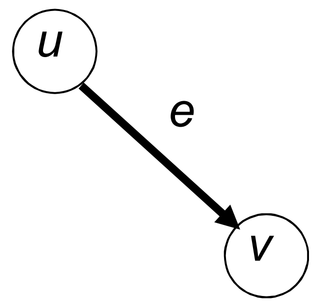
Graph Terminology

We have graph terminology related to relationship between vertices and edges:

- *Adjacency (Kề nhau), connect (nối), degree (bậc), start (bắt đầu), end (kết thúc), indegree (bán bậc vào), outdegree (bán bậc ra),…*



Undirected edge *e*=(*u*,*v*)



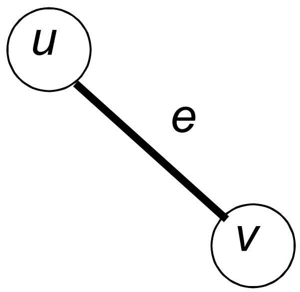
Directed edge *e*=(*u*,*v*)

Adjacency (Kề)

Given G the undirected graph with edge set E. Assume $e \in E$ is pair (u,v).

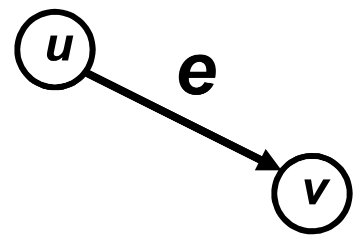
Then we say:

- u, v are *adjacent* / *neighbors* / *connected* (*kề nhau/lân cận/nối với nhau*)
- Edge e *connects* u and v.
- Vertices u and v are *endpoints* (*đầu mút*) of e.

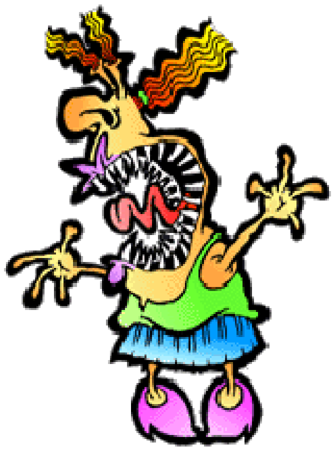


Adjacency in directed graph

- G is directed graph (either simple or multiple) and assume $e = (u,v)$ is an edge of G. We say:
  - *u and v are adjacent, u is adjacent to v, v is adjacent from u*
  - *e goes out of u, e goes into v.*
  - *e connects u with v, e goes from u to v*
  - *initial vertex* (Đỉnh đầu) of e is u
  - *terminal vertex* (Đỉnh cuối) of e is v



Content



Degree of a vertex (Bậc của đỉnh) in undirected graph

Assume G is undirected graph, $v \in V$ is a vertex.

- *Degree* of vertex v, $\deg(v)$, the number of edges incident on a vertex.
- Vertex with degree 0 is called isolated (*đỉnh cô lập*).
- Vertex with degree 1 is called pendant (*đỉnh treo*).
- Symbol often used:

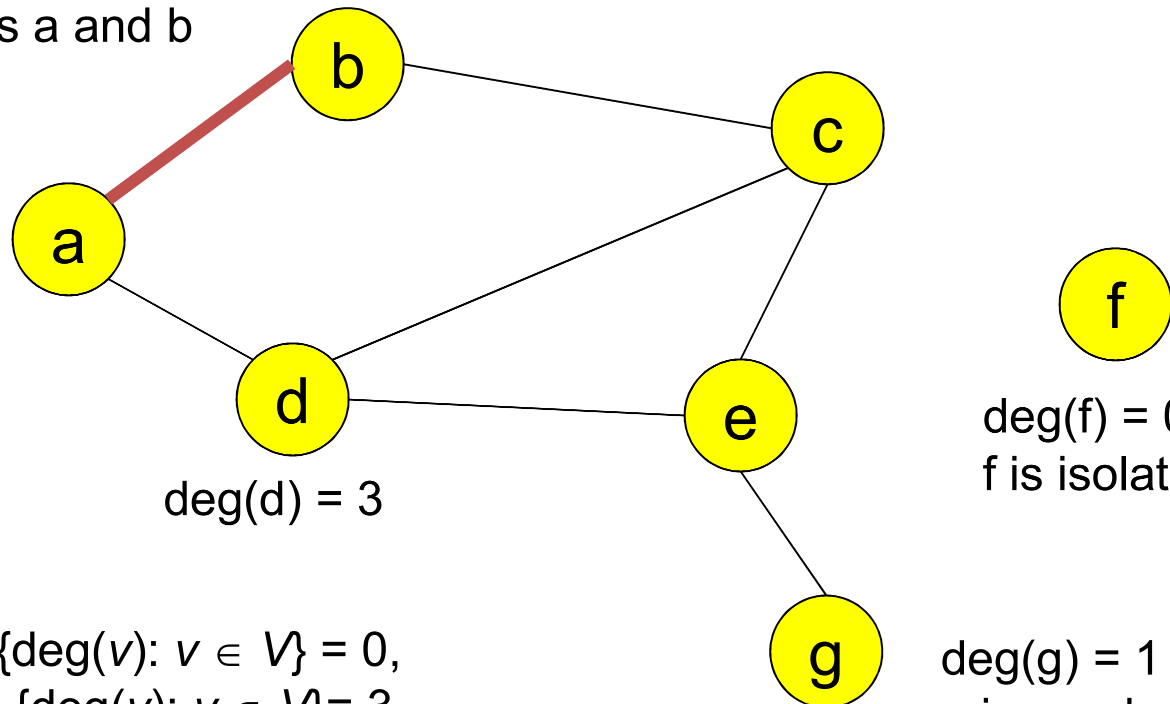
$$\delta(G) = \min \{\deg(v) : v \in V\},$$

$$\Delta(G) = \max \{\deg(v) : v \in V\}.$$

Example

Edge(a, b) is **incident** with two vertices a and b

b is adjacent to c and c is adjacent to b



deg(d) = 3

deg(f) = 0
f is isolated (đỉnh cô lập)

deg(g) = 1
g is pendant (đỉnh treo)

$\delta(G) = \min\{\deg(v): v \in V\} = 0$,
$\Delta(G) = \max\{\deg(v): v \in V\} = 3$.

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Theorem: Undirected graph

Theorem 1 (The Handshaking theorem):

The undirected Graph G = (V, E):

$$\sum_{v \in V} \deg(v) = 2|E|$$

(why?) Every edge connects 2 vertices

Theorem 2:

An undirected graph has even number of vertices with odd degree

Proof: V_1 is the set of even degree vertices and V_2 refers to odd degree vertices

$$\underbrace{2|E| = \sum_{v \in V} \deg(v)}_{\text{even}} = \underbrace{\sum_{u \in V_1} \deg(u)}_{\text{even}} + \underbrace{\sum_{u \in V_2} \deg(u)}_{\text{even}}$$

➔ $\deg(u)$ is even for $u \in V_1$ ➔ The first term in the right hand side is even

The sum of the two terms on the right hand side is even since sum is 2|E|

➔ Second term is also even: $\sum_{u \in V_2} \deg(u) = even$

$\deg(u)$ is odd for $u \in V_2$

➔ number of vertices with odd degree is even

Example

An undirected graph $G=(V,E)$ with 14 vertices and 25 edges, each vertex has degree either 3 or 5. How many vertices with degree 3 are there in this G?

Solution:

Assume G has x vertices of degree 3.

Then, there are $(14-x)$ vertices of degree 5.

$$\sum_{v \in V} \deg(v) = 2|E|$$

As $|E| = 25$,

sum of all degrees $= 2|E| = 50$.

Therefore, $3x + 5(14-x) = 50$

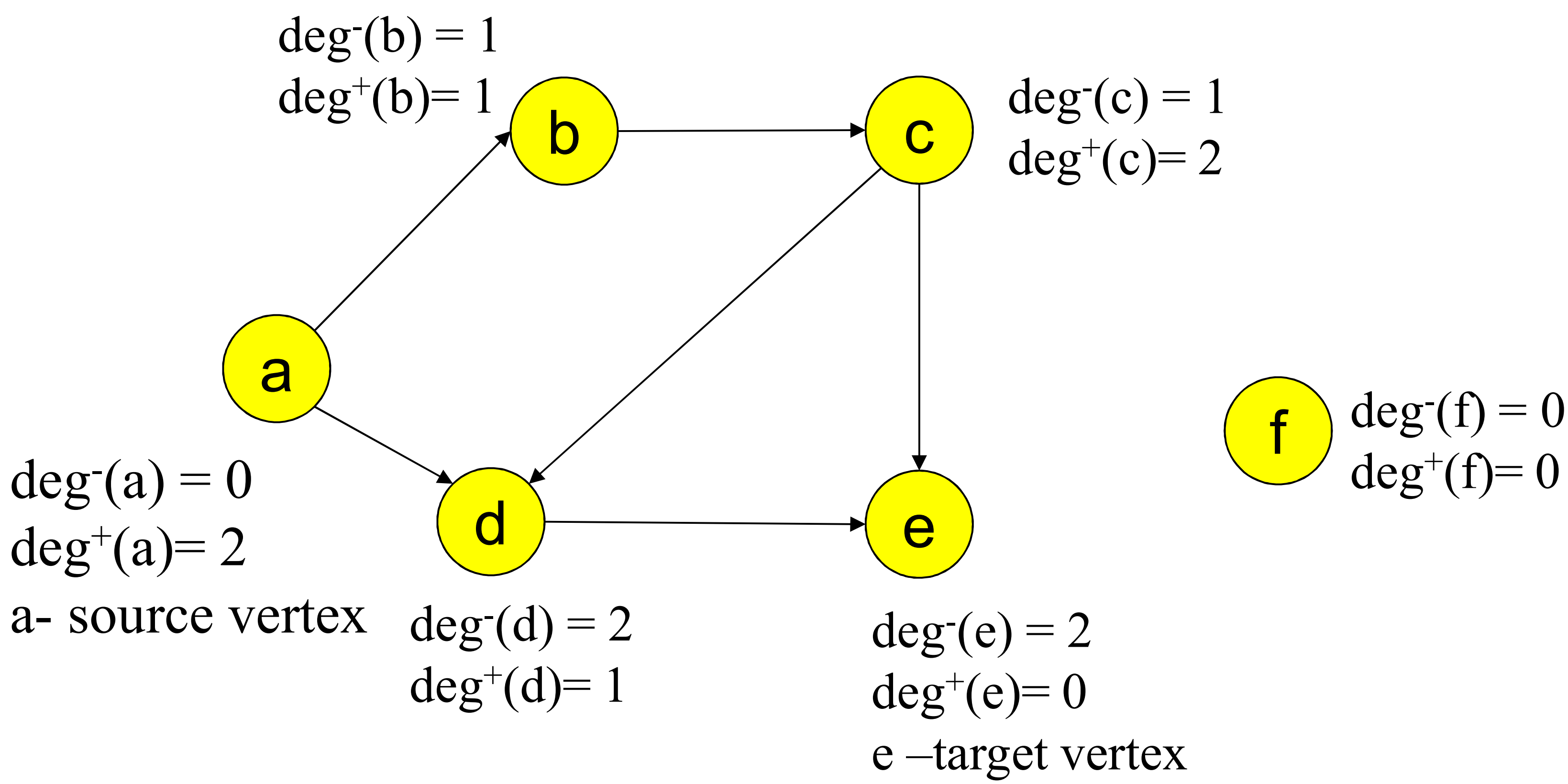
So $x = 10$.

Degree of a vertex in directed graph

Given G directed graph, v is a vertex of G:

- *In-degree* (bán bậc vào) of v, $\deg^-(v)$, number of edges for which v is terminal vertex (goes into v).
- *Out-degree* (bán bậc ra) of v, $\deg^+(v)$, number of edges for which v is initial vertex (goes out of v).
- *Degree* of v, $\deg(v)=\deg^-(v)+\deg^+(v)$

Example



NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

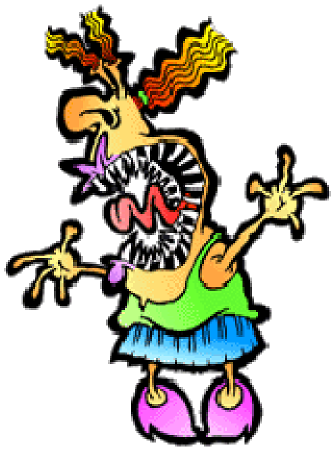
Theorem: Directed graph

Theorem 3 The directed Graph (either simple or multiple) G = (V, E):

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = \frac{1}{2}\sum_{v \in V} \deg(v) = |E|$$

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Content



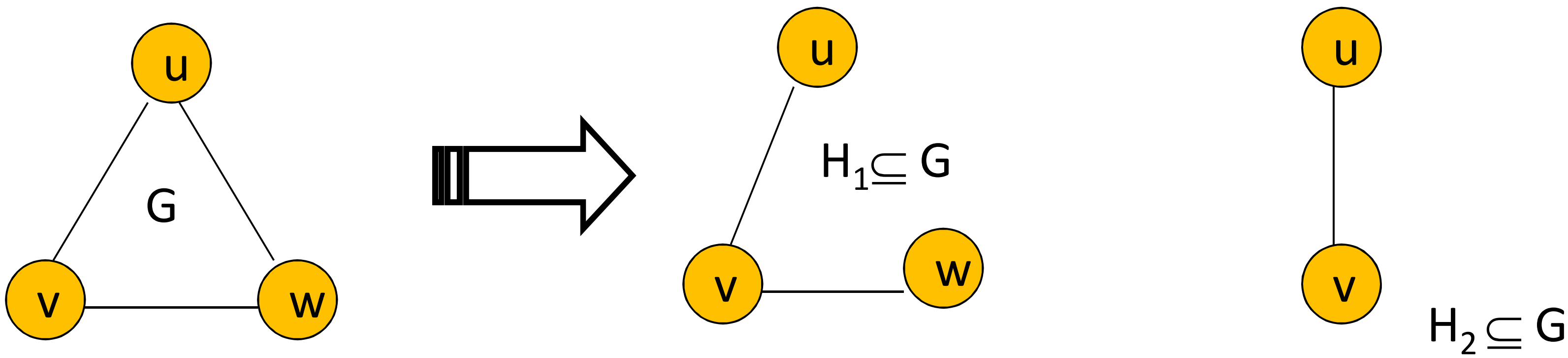
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

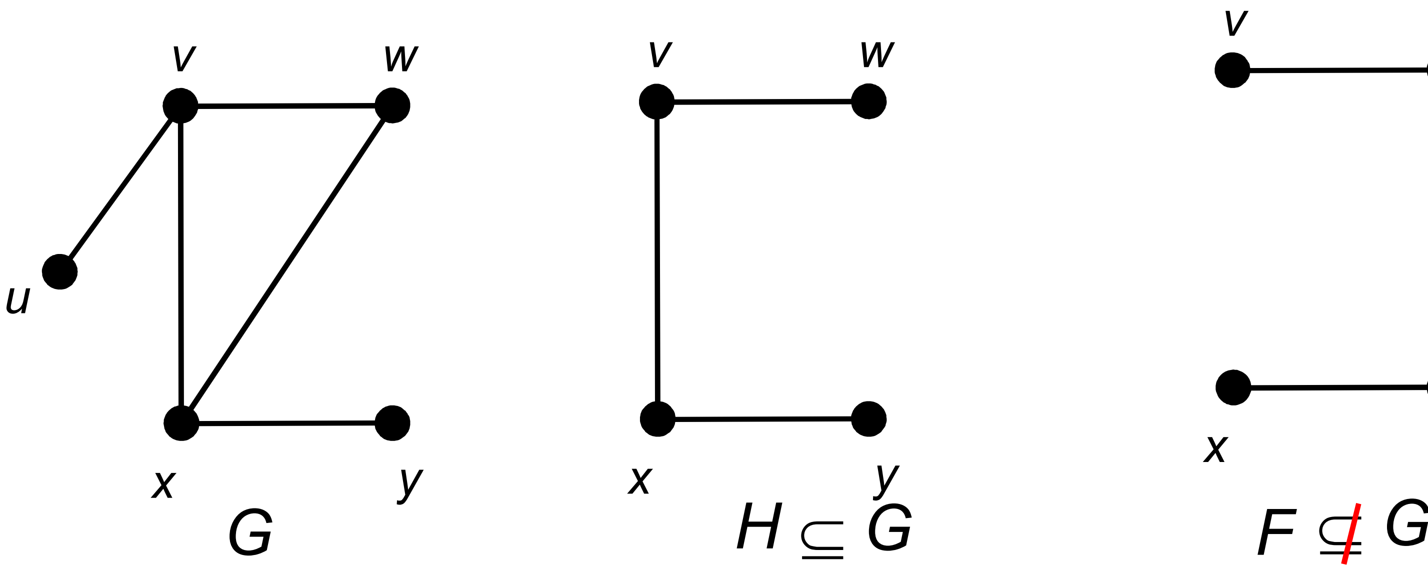
4. Subgraph (Đồ thị con)

A subgraph of a graph G = (V, E) is a graph H =(V’, E’) where V’ is a subset of V and E’ is a subset of E.

Denoted by: $H \subseteq G$

Example: V = {u, v, w}, E = ({u, v}, {v, w}, {w, u}}

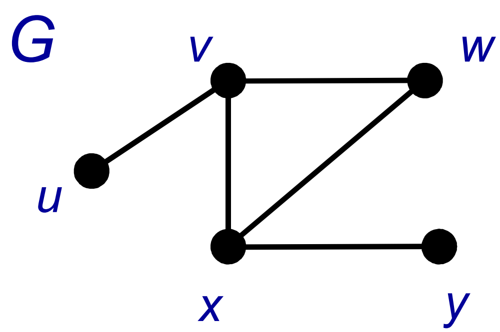




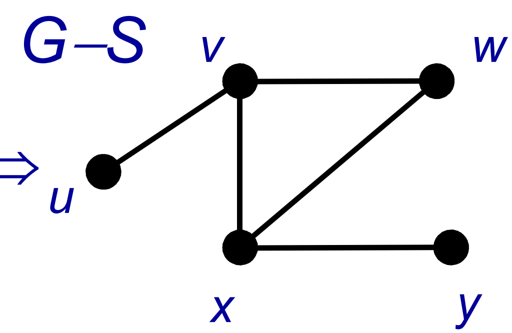
The deletion of vertices

Definition. $G = (V, E)$ is undirected graph. Assume $S \subseteq V$. We call the removal of vertices set S from the graph is to remove all vertices in S and their adjacent edges.

We denote the obtained graph by G-S



Assume $S=\{x,u\} \Rightarrow$



NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Spanning Subgraph (Đồ thị con bao trùm)

Definition.

Subgraph $H \subseteq G$ is called spanning subgraph of G if vertex set of H is vertex set of G: $V(H) = V(G)$.

Definition.

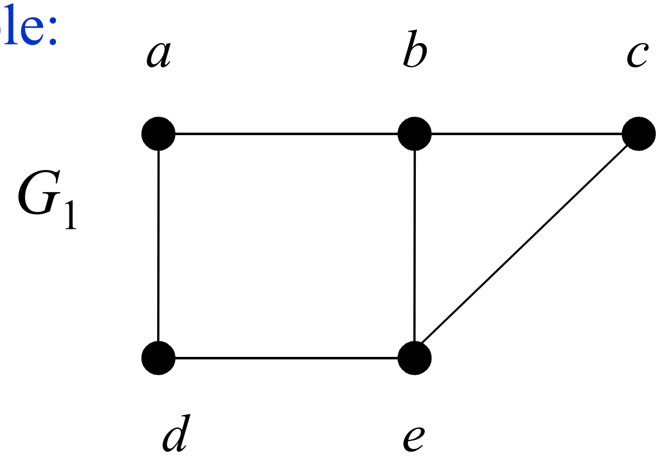
We write $H = G + \{(u,v), (u,w)\}$ to mean

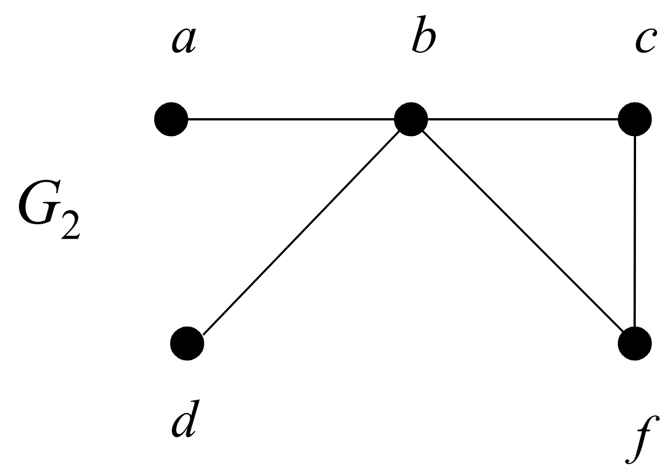
$E(H) = E(G) \cup \{(u,v), (u,w)\}$, where $(u,v), (u,w) \notin E(G)$.

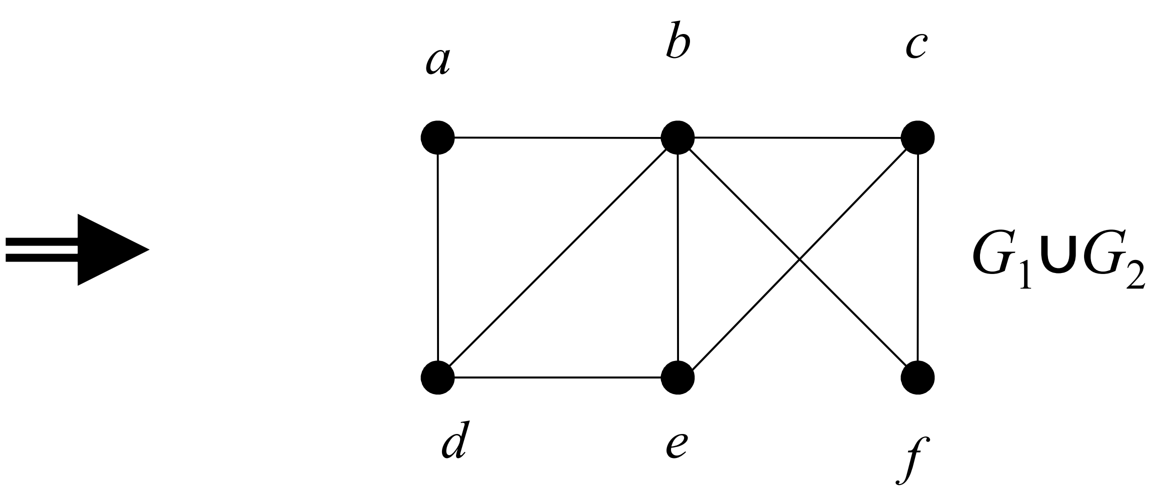
The union of graphs

The union of two simple graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ is the simple graph $G_1 \cup G_2=(V_1 \cup V_2, E_1 \cup E_2)$.

Example:



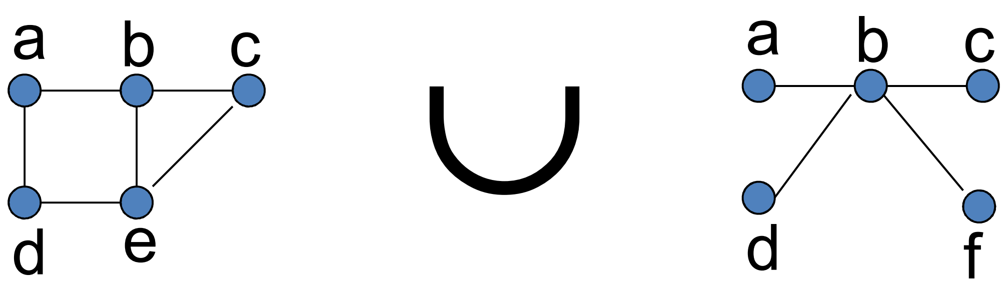




The union of graphs (Hợp của hai đồ thị)

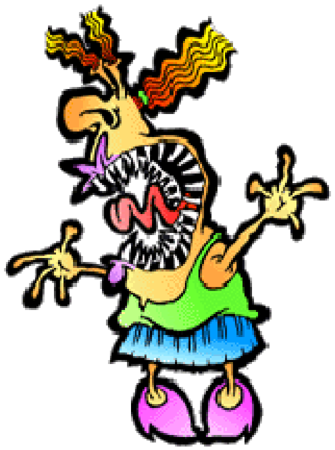
The union of two simple graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ is the simple graph $G_1 \cup G_2=(V_1 \cup V_2, E_1 \cup E_2)$.

Example:



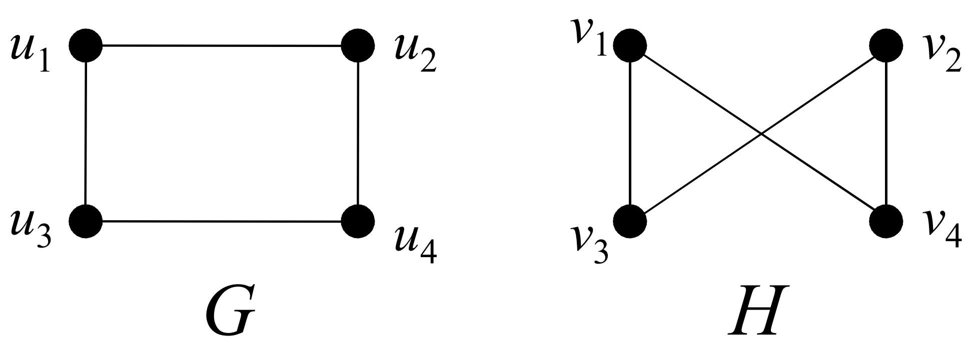
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Content



NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

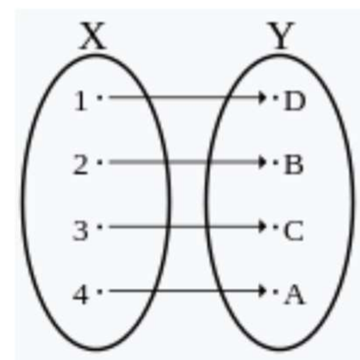
Isomorphism of Graphs



G is isomorphic to H

Definition. The simple graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ are isomorphic if there is an one-to-one and onto function f from V_1 to V_2 with the property that a and b is adjacent in G_1 iff $f(a)$ and $f(b)$ is adjacent in G_2, $\forall a,b \in V_1$

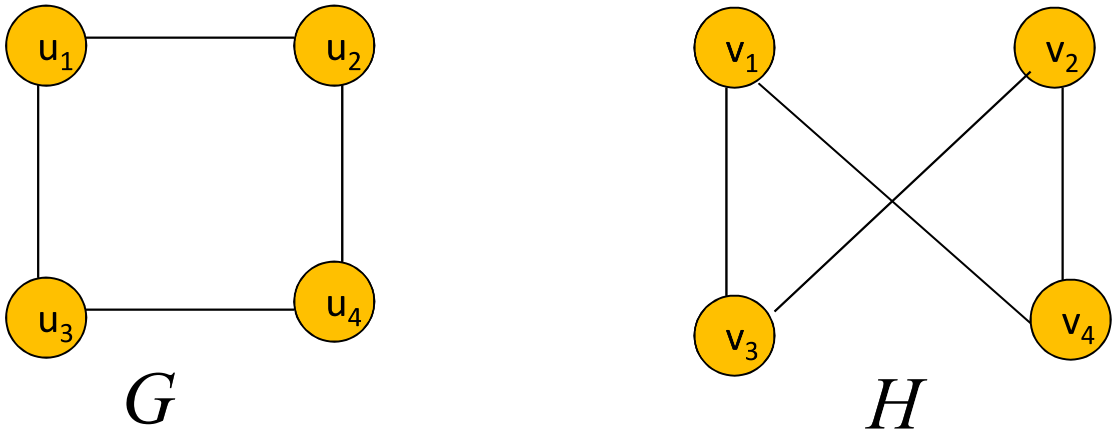
f is called an isomorphism.



Application Example:

In chemistry, to find if two compounds have the same structure

Example. Show that G and H are isomorphic.



Solution.

The function f with $f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence between $V(G)$ and $V(H)$.

Isomorphism graphs there will be:

(1) The same number of vertices

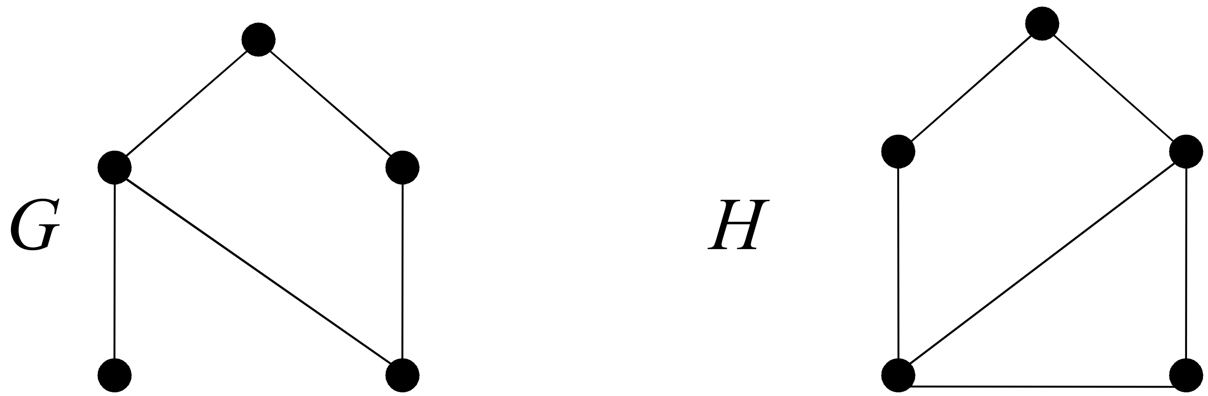
(2) The same number of edges

(3) The same number of degree

Isomorphism of Graphs

Given figures, judging whether they are isomorphic in general is not an easy task.

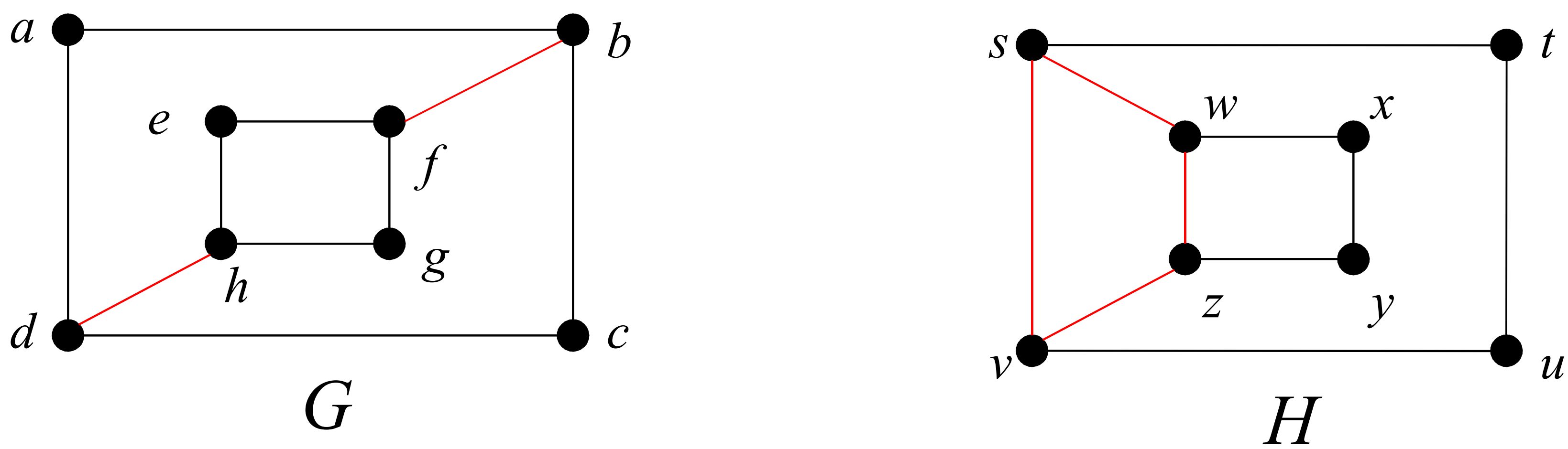
Example. Show that G and H are not isomorphic.



Solultion :

G has a vertex of degree $= 1$, H doesn't

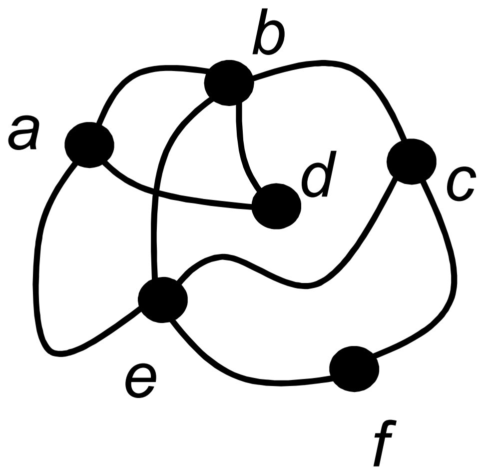
Example. Determine whether *G* and *H* are not isomorphic.

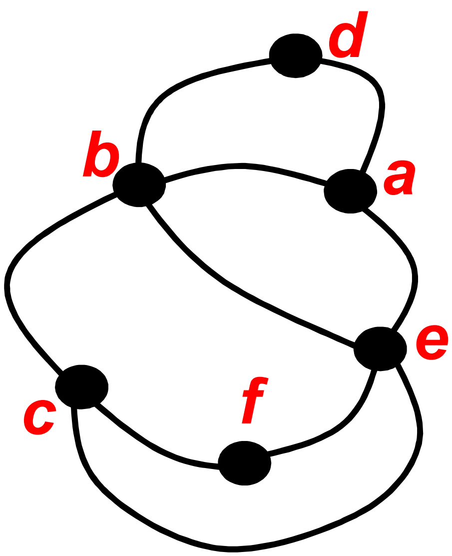


Solultion : In *G,* deg(*a*)=2, which must correspond to either *t*, *u*, *x*, or *y* in H. Degree each of these four vertices in H is adjacent to another vertex of degree two in H, which is not true for a in G

➔ G and H are not isomorphic.

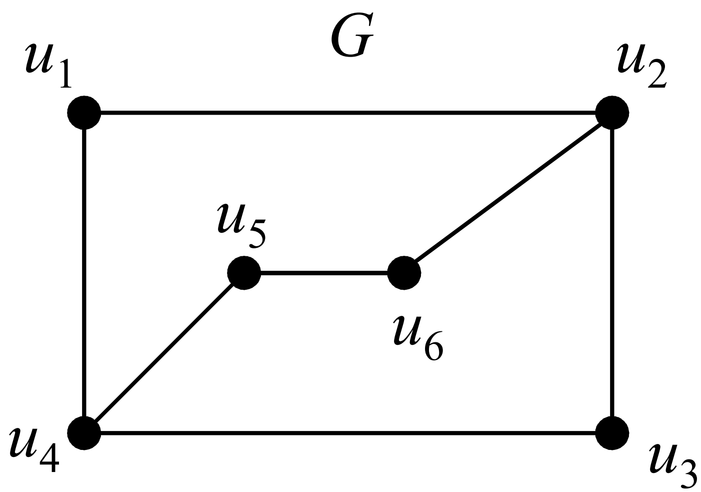
Example. Determine whether the graphs G and H are isomorphic.

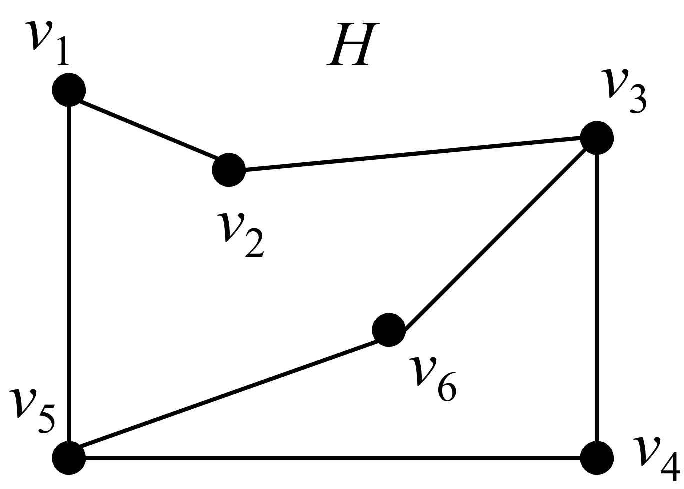




NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Example. Determine whether the graphs G and H are isomorphic.



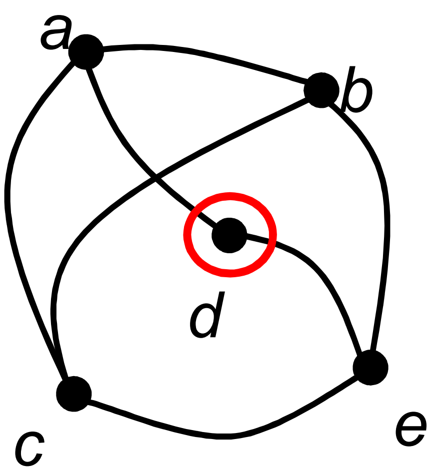


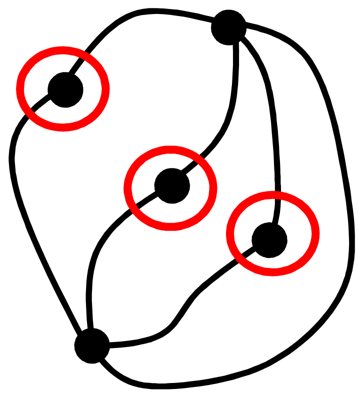
Solution:

$f(u_1)=v_6, f(u_2)=v_3, f(u_3)=v_4, \ f(u_4)=v_5, f(u_5)=v_1, f(u_6)=v_2$

$\Rightarrow$ Yes

Example. Determine whether the graphs G and H are isomorphic.





- *Same number of vertices*
- *Same number of edges*
- *Different number of vertices with degree 2* $(1 \neq 3)$

Content

1. Graph in practice
2. Graph types
3. Degree of vertex
4. Subgraph
5. Isomorphism of Graphs
6. **Path and cycle**
7. Connectedness
8. Special graphs
9. Graph Coloring problem

Path

Definition for Directed Graphs

A **Path** of length $n > 0$ from u to v in G is a sequence of n edges e_1, e_2 , e_3, …, e_n of G such that $f(e_1) = (x_o, x_1)$, $f(e_2) = (x_1, x_2)$, …, $f(e_n) = (x_{n-1}, x_n)$, where $x_0 = u$ and $x_n = v$.

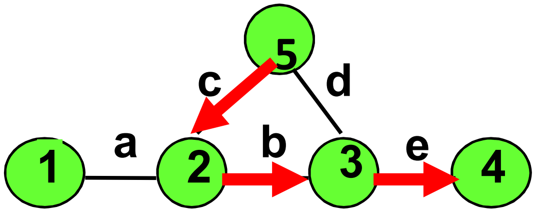
A path is said to pass through $x_0, x_1, \ldots, x_n$ or traverse e_1, e_2 , e_3, …, e_n

- A path is called *elementary* **(đường đi cơ bản)** if all the edges are distinct.
- A path is called *simple* **(đường đi đơn)** if all the vertices are distinct.
- A path is *closed* if $v_0 = v_n$.
- A closed elementary path is called a *cycle*. A cycle is called simple if all the vertices are distinct (except $v_0 = v_n$).

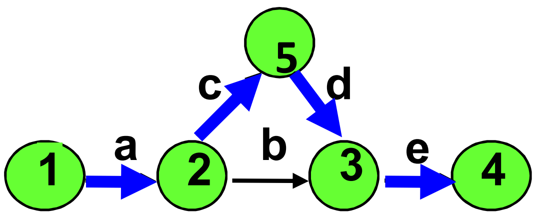
Path

A path is called *elementary* if all the edges are distinct.
A path is called *simple* if all the vertices are distinct.

Path: **5, 2, 3, 4 [OR: {5, 2}, {2, 3}, {3, 4}**
OR: c,b,e]
all the vertices are distinct→ simple path

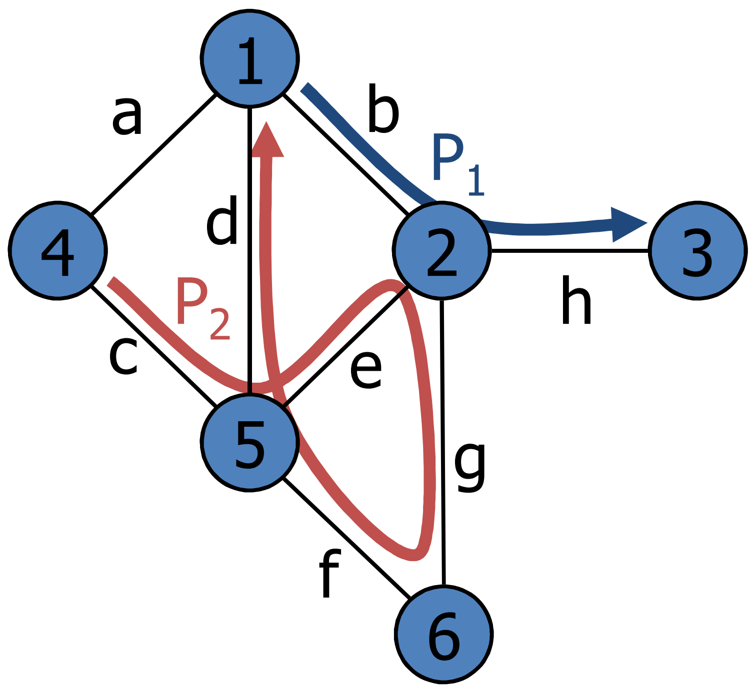


***Path* (directed) 1, 2, 5, 3, 4**
[OR: (1, 2), (2, 5), (5, 3), (3, 4)
OR: a, c, d, e]
- **is the simple path**



Path

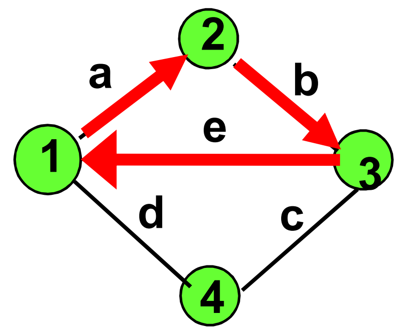
- P_1= (1, 2, 3) is the simple path
- P_2=(4,5,2,6,5,1) is the path but not the simple path



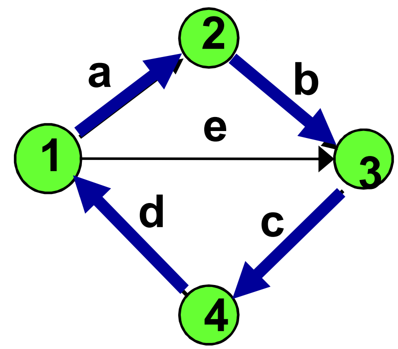
Cycle

- A closed elementary path is called a *cycle*.
  - A path is called *elementary* if all the edges are distinct.
  - A path is *closed* if $v_0 = v_n$.
- A cycle is called simple if all the vertices are distinct (except $v_0 = v_n$).

Simple cycle: (1, 2, 3, 1)

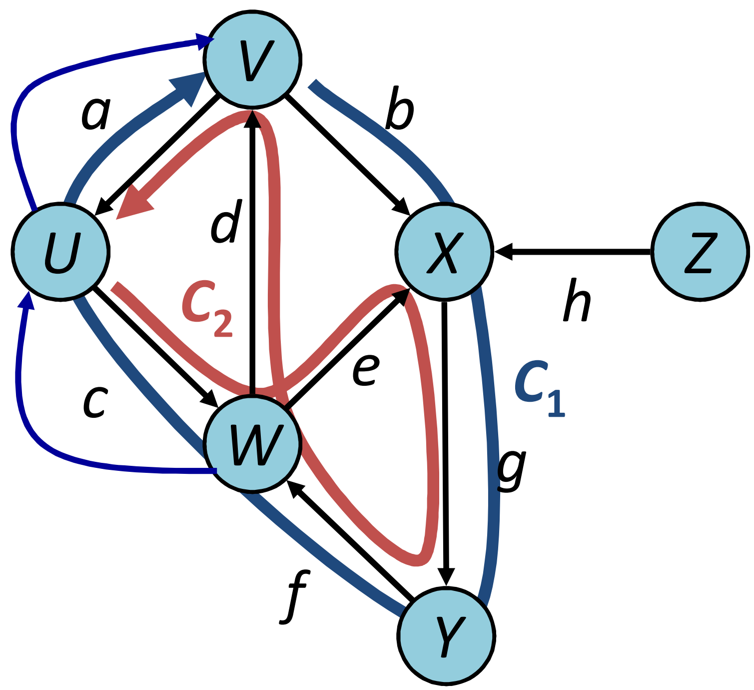


Simple cycle: (1, 2, 3, 4, 1)



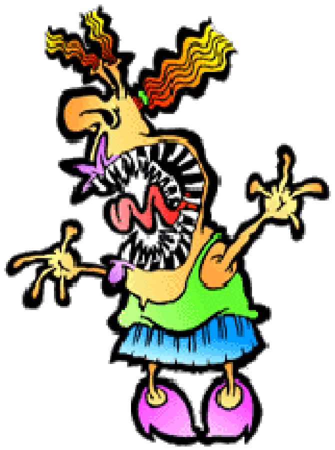
Cycle on directed graph

- $C_1=(V,X,Y,W,U,V)$ is simple cycle
- $C_2=(U,W,X,Y,W,V,U)$ is cycle but not simple cycle



Content

1. Graph in practice
2. Graph types
3. Degree of vertex
4. Subgraph
5. Isomorphism of Graphs
6. Path and cycle
7. **Connectedness**
8. Special graphs
9. Graph Coloring problem



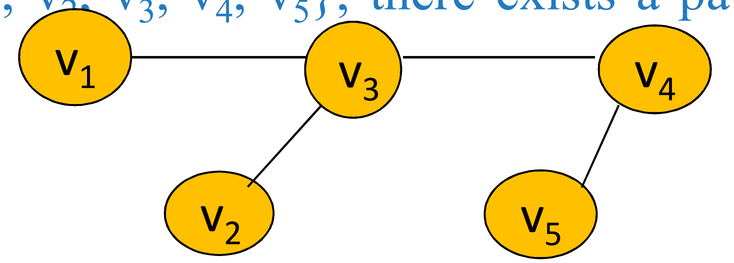
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Connectedness

Undirected Graph

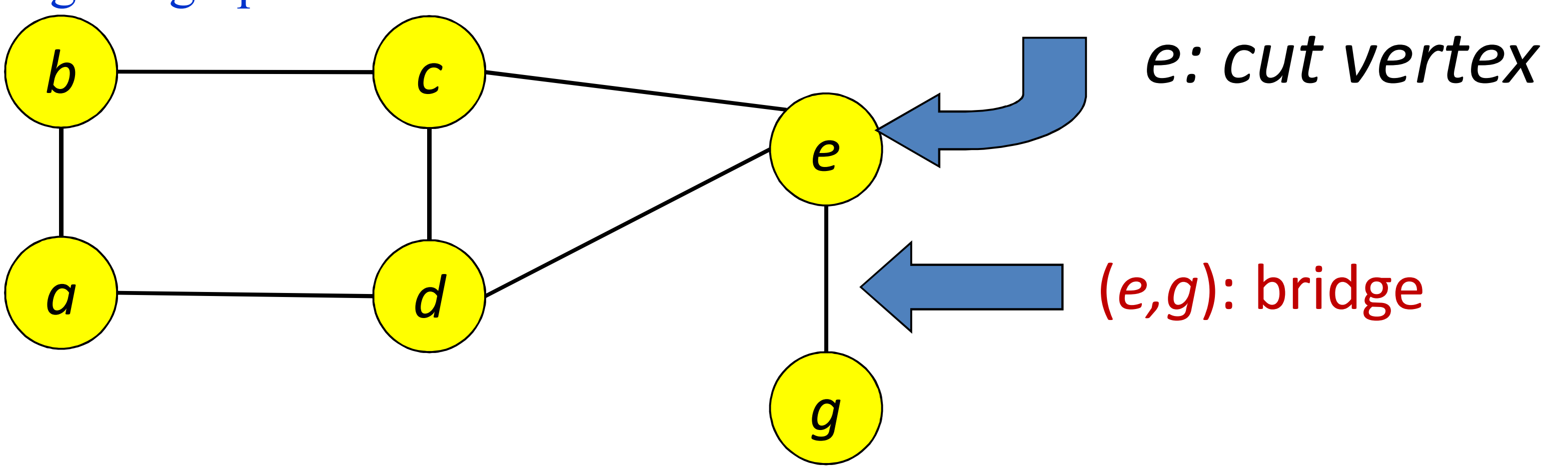
An undirected graph is connected if there exists is a simple path between every pair of vertices

Example: G (V, E) is connected since for V = {v_1, v_2, v_3, v_4, v_5}, there exists a path between {v_i, v_j}, $1 \leq i, j \leq 5$



Articulation Point (Cut vertex): removal of a vertex produces a subgraph with more connected components than in the original graph. The removal of a cut vertex from a connected graph produces a graph that is not connected

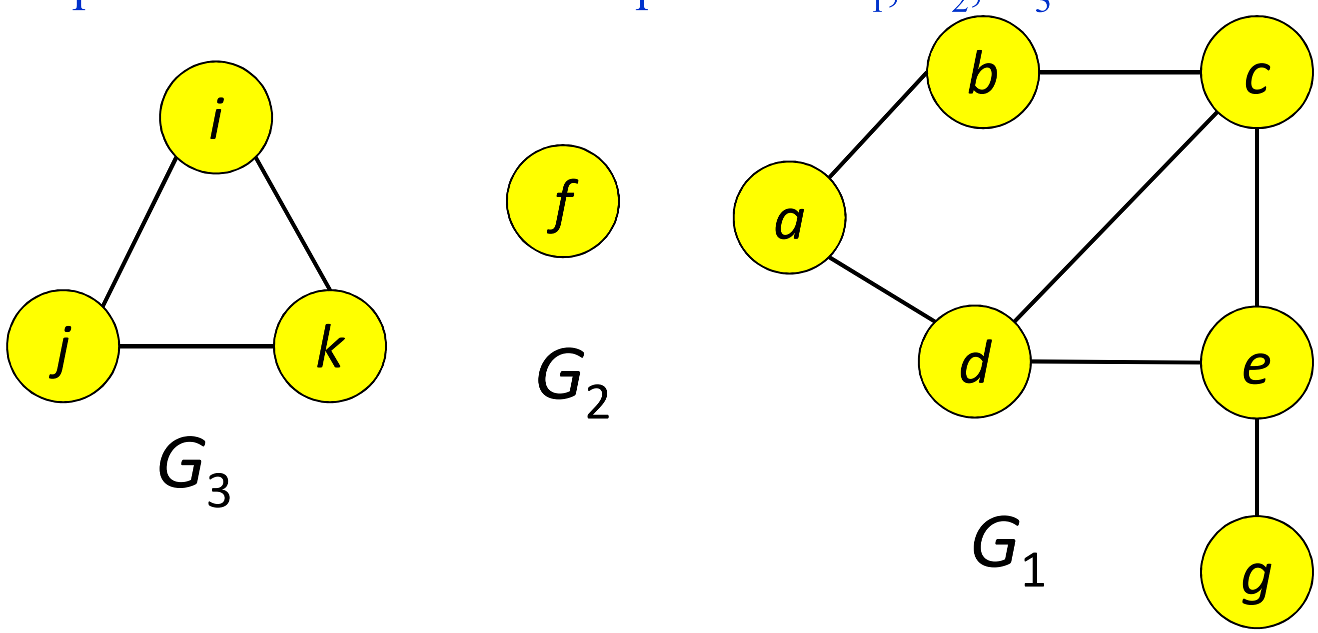
Bridge: An edge whose removal produces a subgraph with more connected components than in the original graph.



Connectedness

- If a graph is not connected then it splits up into a number of connected subgraphs, called its *connected components*.
- The connected components of G can be defined as its maximal connected subgraphs. This means that G_1 is a connected component of G if:
  - G_1 is a connected subgraph of G
  - G_1 is not itself a proper subgraph of any other *connected* subgraph of G. This second condition is what we mean by the term maximal; it says that if H is a connected subgraph such that $G_1 \subseteq H$, then $G_1 = H$.

Example: Graph G has 3 connected components: G_1, G_2, G_3

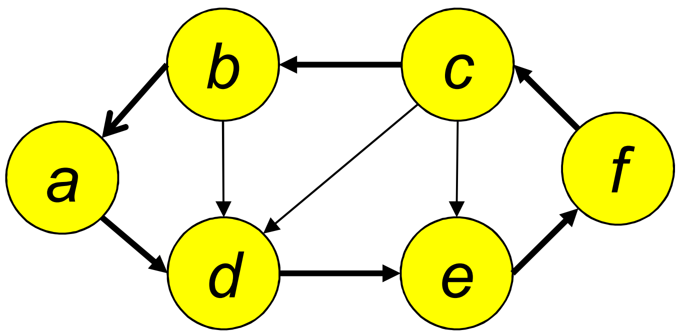


Connectedness

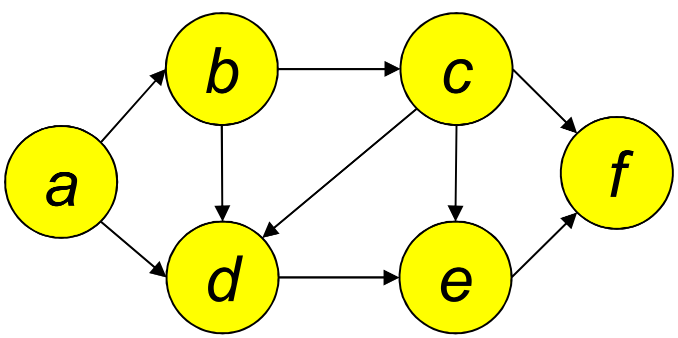
Directed Graph

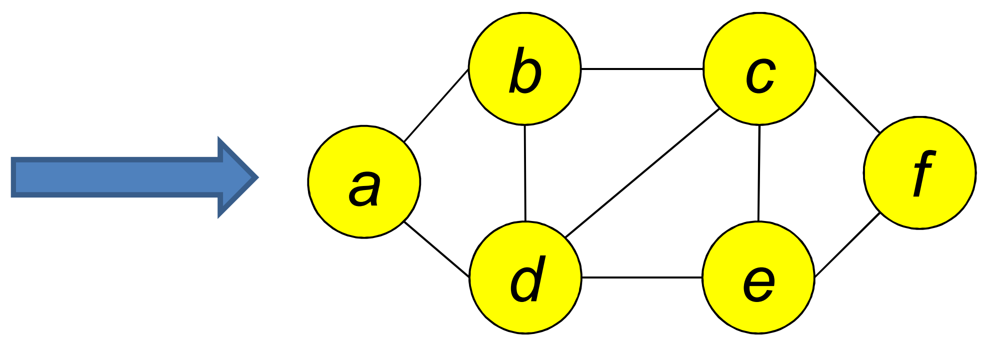
- A directed graph is **strongly connected** if there is a path from *u* to *v* and from *v* to *u* whenever *u* and *v* are vertices in the graph
- A directed graph is **weakly connected** if its corresponding undirected graph is connected.

A strongly connected Graph can be weakly connected but the vice-versa is not true (why?)



Strongly connected graph





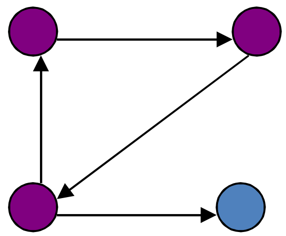
Weakly connected graph

Connectedness

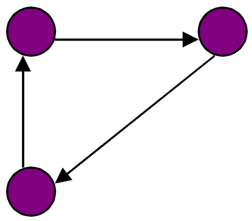
Directed Graph

- **Strongly connected Components:** subgraphs of a Graph G that are strongly connected

Example: G1 is the strongly connected component in G



G



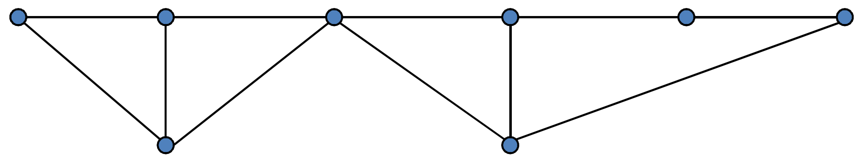
G1

k-Connectivity(k-liên thông)

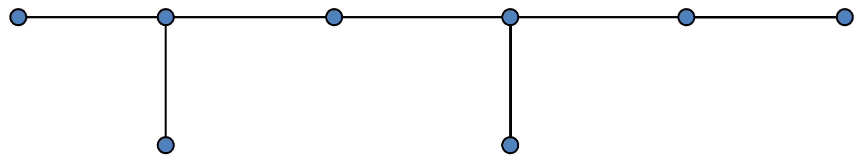
Not all connected graphs have the same value!

Question: Which following graphs are more valuable computer network

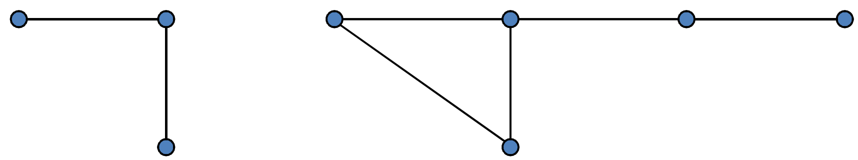
1) G_1



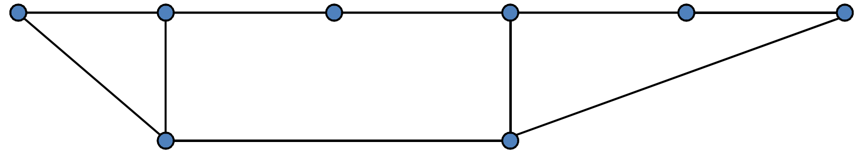
2) G_2



3) G_3



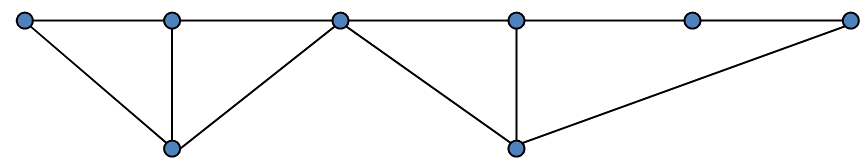
4) G_4



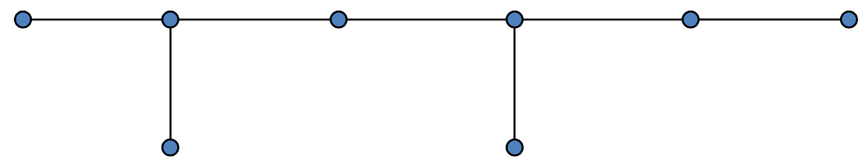
k-Connectivity(k-liên thông)

Answer: We want a computer network still working even when one computer fails

G1) 2^{nd} best. Connected, but there is a weak point - "cut vertex"

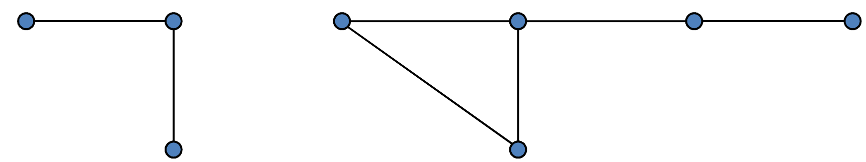


G2) 3^{rd} best. Connected, but each computer is a weak point

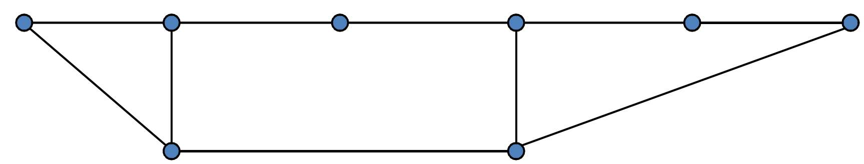


G3) The worst!

Not connected

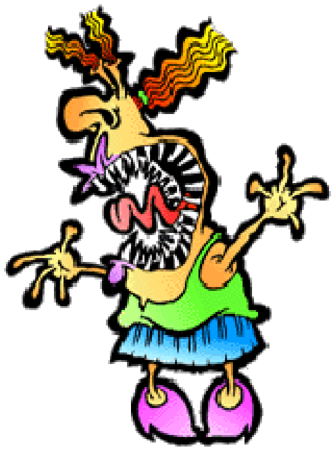


G4) The best! Not connected only when there are 2 computers fail



Content

1. Graph in practice
2. Graph types
3. Degree of vertex
4. Subgraph
5. Isomorphism of Graphs
6. Path and cycle
7. Connectedness
8. **Special graphs**
9. Graph Coloring problem

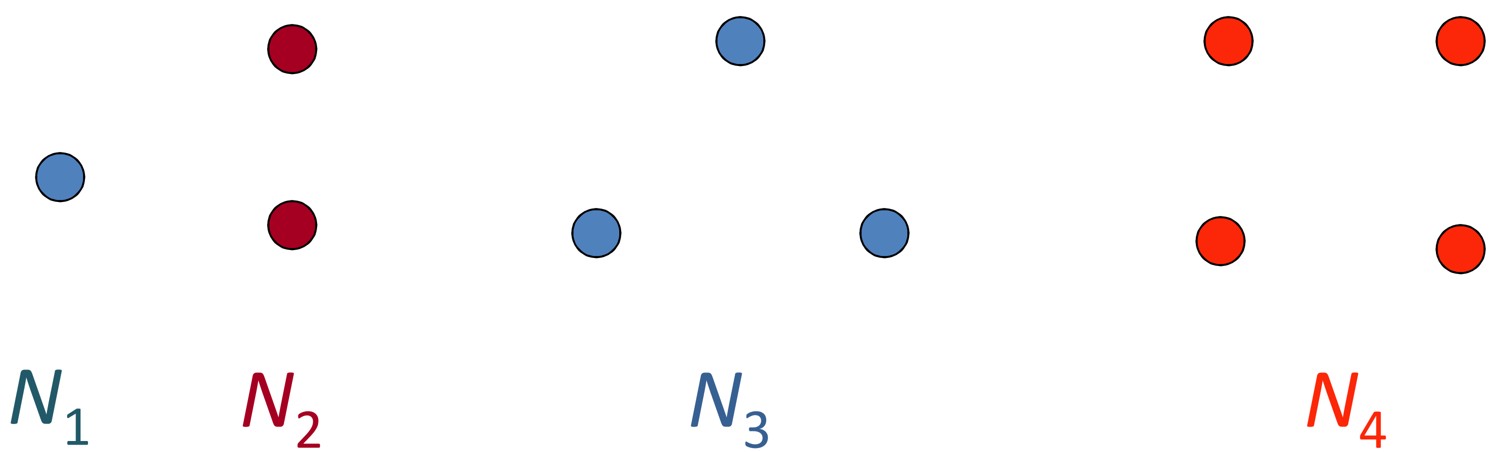


Some special graphs

1. Null graph
2. Complete graphs K_n
3. Cycles C_n
4. Wheels W_n
5. n-Cubes Q_n
6. Bipartite graphs
7. Complete bipartite graphs $K_{m,n}$
8. r-regular graph
9. Planar graph
10. Euler graph and Hamilton graph

Null graph

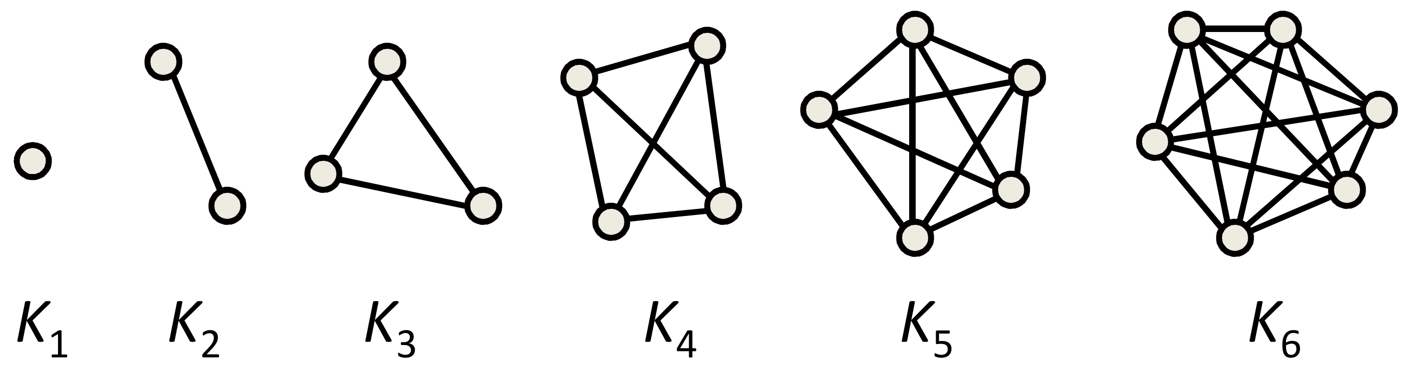
Definition: Null graph (N_n) is an undirected graph G = (V, E) where E = ∅



Complete graph

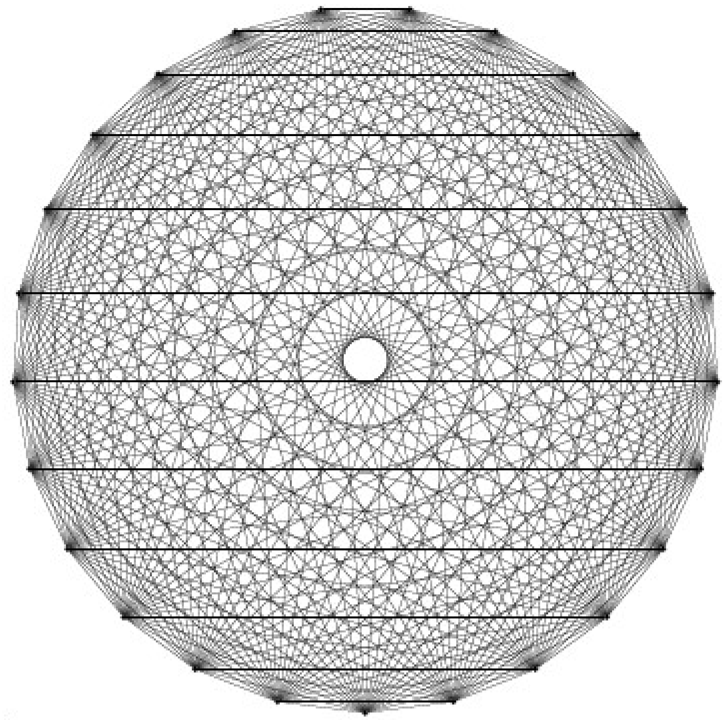
Complete graph (K_n): is the simple graph that contains exactly one edge between each pair of distinct vertices.

Example:



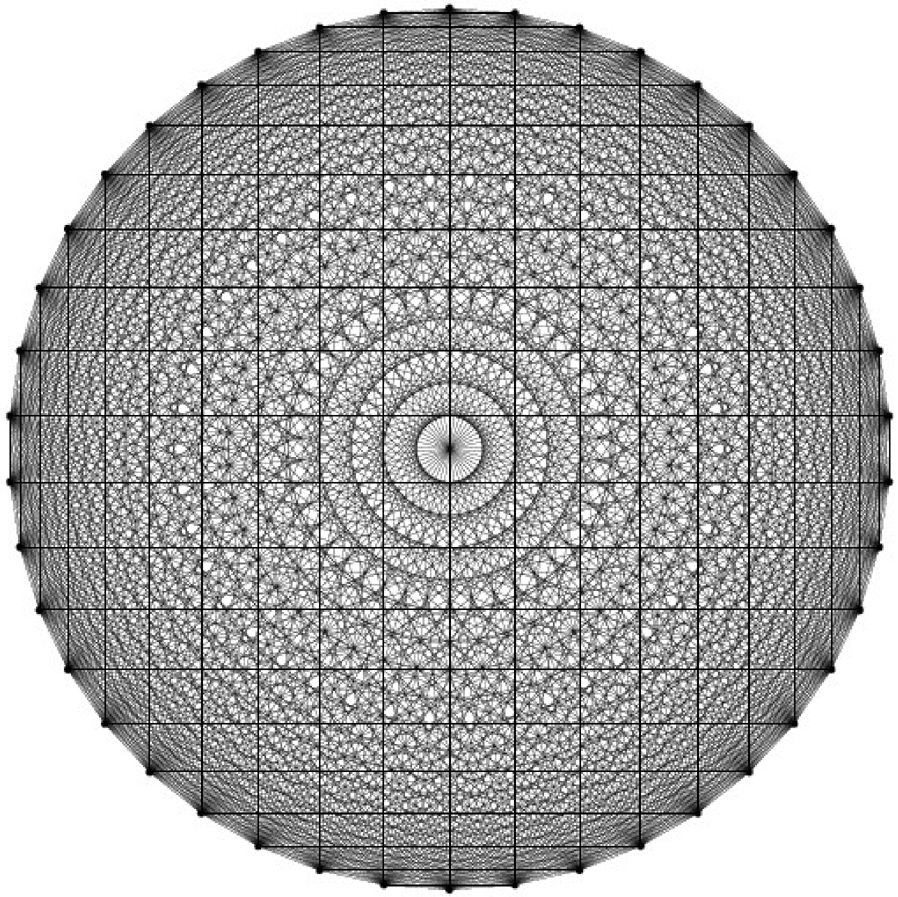
Note: Graph K_n has $\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$ *edges*

Complete Graphs (Đồ thị đầy đủ)



K_{25}

Complete Graphs (Đồ thị đầy đủ)

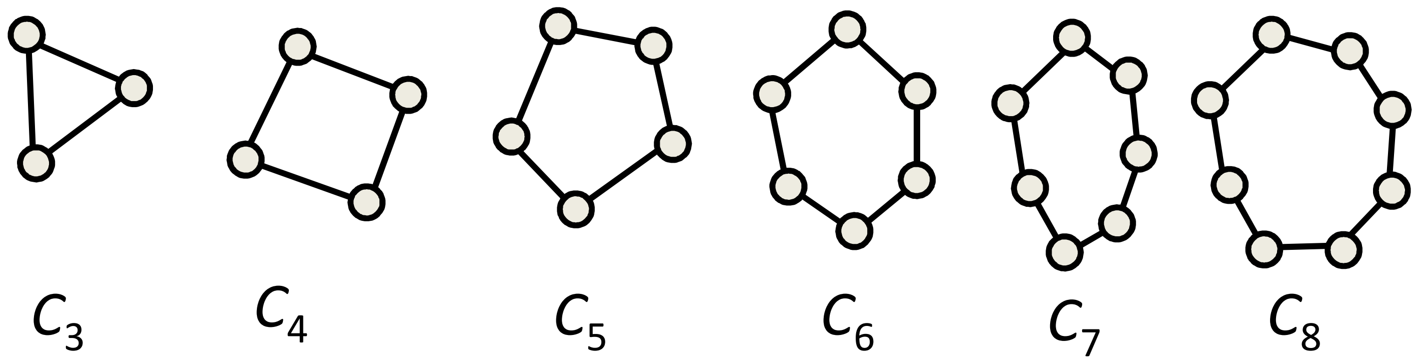


K_{42}

Cycles

Cycle (C_n , $n \geq 3$): is an undirected graph consisting of n vertices v_1, v_2, v_3 … v_n and edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_3, v_4\}$ … $\{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

Example:

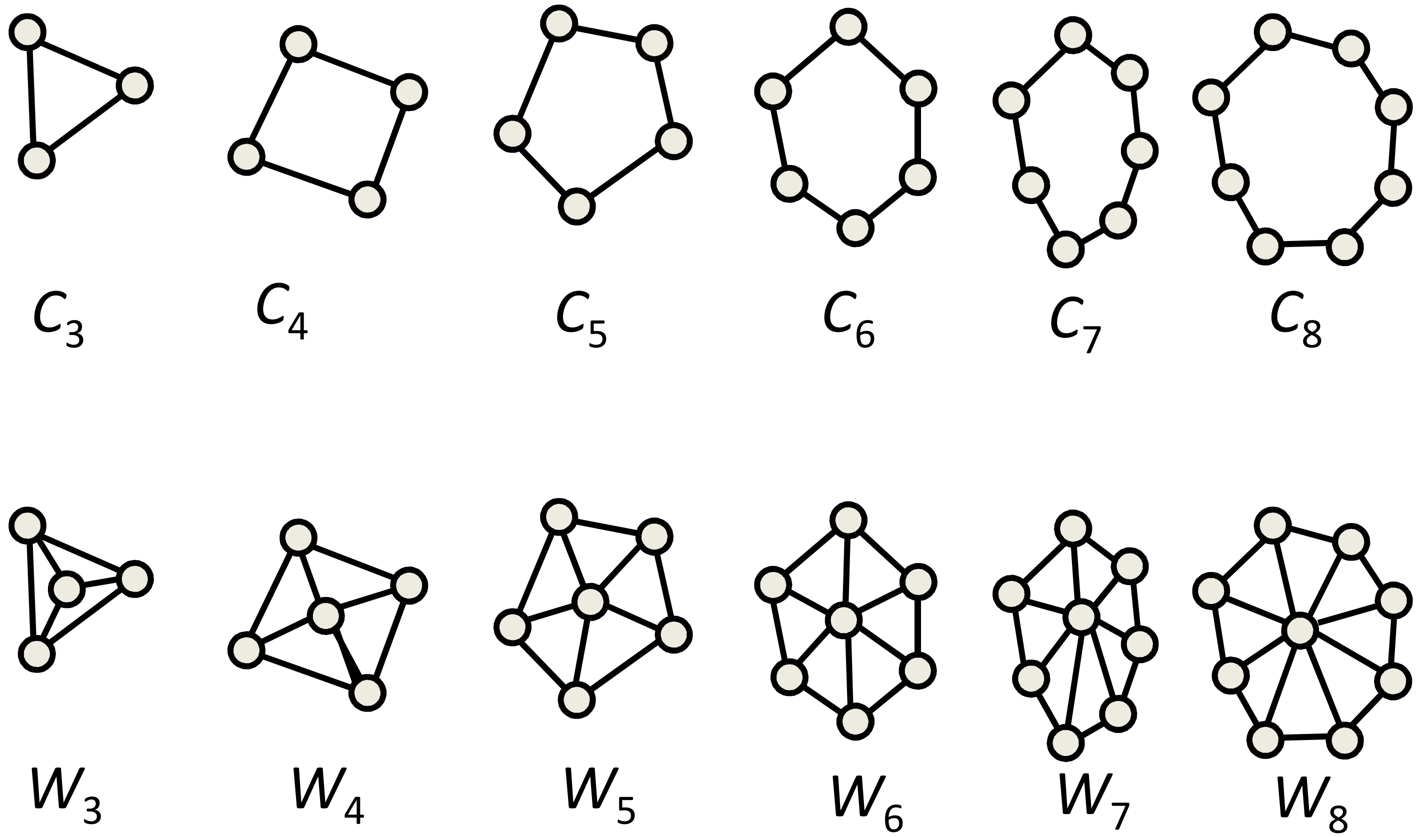


Note: Graph C_n has ? edges

Wheels

Wheel (W_n , $n \geq 3$): is an undirected graph obtained by adding additional vertex to C_n and connecting all vertices to this new vertex by new edges.

Example:



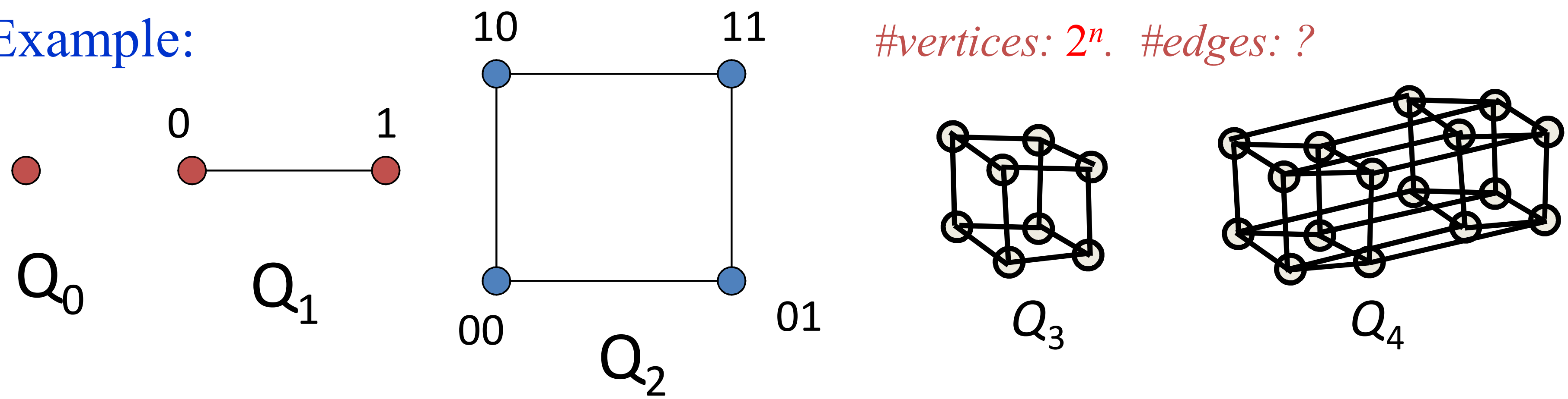
Wheel W_n has ? edges

N-cubes

N-cubes (Q_n): vertices represented by 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ by exactly one bit positions

Example:

#vertices: 2^n. #edges: ?



- $Q_0=\{\{v_0\},\varnothing\}$ (1 vertex, not edge)
- $n\in\mathbf{N}$: if $Q_n=(V,E)$ where $V=\{v_1,\ldots,v_a\}$ and $E=\{e_1,\ldots,e_b\}$, then $Q_{n+1}=(V\cup\{v_1{}',\ldots,v_a{}'\},$
  $E\cup\{e_1{}',\ldots,e_b{}'\}\cup\{\{v_1,v_1{}'\},\{v_2,v_2{}'\},\ldots,\{v_a,v_a{}'\}\})$

It means Q_{n+1} obtained by connecting pairs of verticies of Q_n và Q'_n

Some special graphs

1. Null graph
2. Complete graphs K_n
3. Cycles C_n
4. Wheels W_n
5. n-Cubes Q_n
6. Bipartite graphs
7. Complete bipartite graphs $K_{m,n}$
8. r-regular graph
9. Planar graph
10. Euler graph and Hamilton graph

Bipartite graph

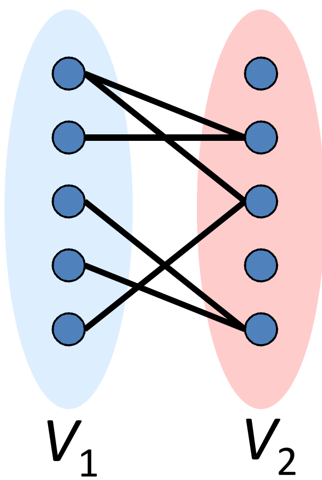
Graph $G=(V,E)$ is a bipartite graph if and only if

$V = V_1 \cup V_2$, $V_1 \cap V_2 = \varnothing$

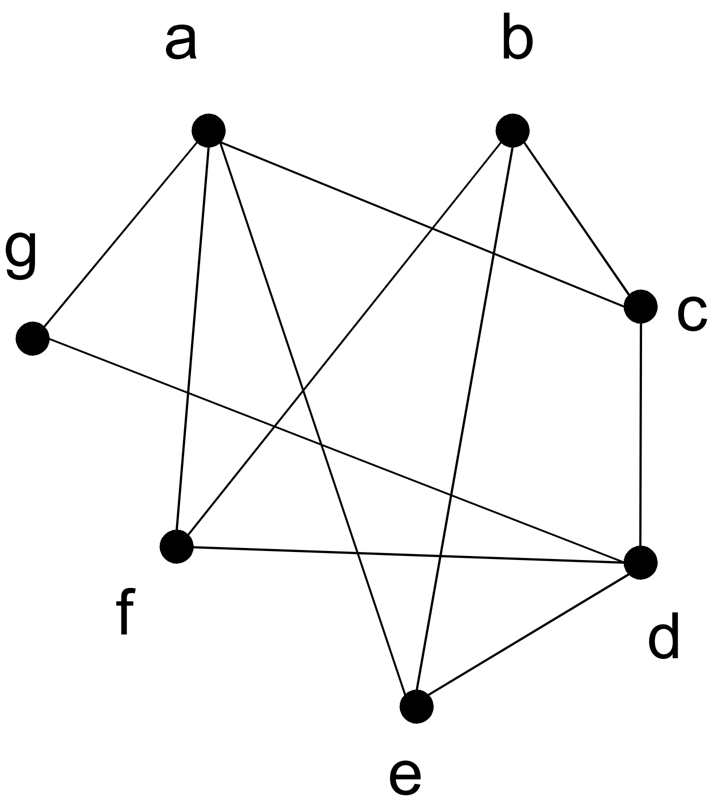
$\forall e \in E$: $\exists v_1 \in V_1, v_2 \in V_2$: $e = \{v_1, v_2\}$

(In a simple graph G, if V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2)

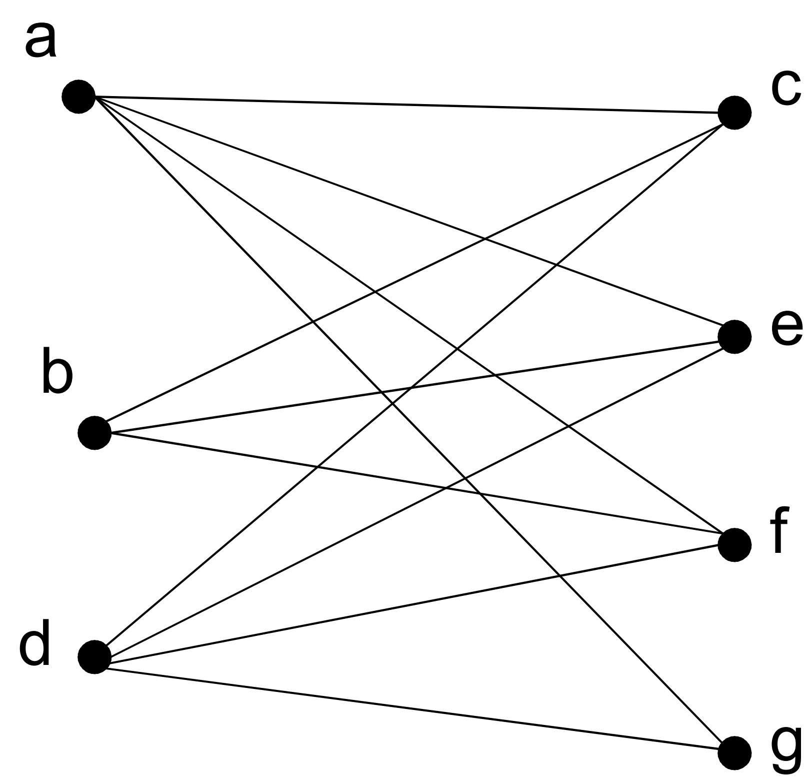
Example:



Example. Is the graph G bipartite ?



G



Yes !

Bipartite graph: Examples of application

- **Document/Term Graphs**: Here set V_1 are documents and set V_2 are terms or words, and there is an edge (v_1, v_2) if the word v_2 is in the document v_1. Such graphs are use often to analyze text, for example to cluster the documents.
- **Movies preferences**: In 2009, Netflix gave a $1Million prize to the group that was best able to predict how much someone would enjoy a movie based on their preferences. This can be viewed as a bipartite graph problem. The viewers are the vertices V_1 and the movies the vertices V_2 and there is an edge from v_1 to v_2 if v_1 viewed v_2. In this case, the edges are weighted by the rating the viewer gave. The winner was algorithm called "BellKor's Pragmatic Chaos".
- **Students and classes:** We might create a graph that maps every student to the classes they are taking. Such a graph can be used to determine conflicts, e.g. when classes cannot be scheduled together.
- **Object recognition**
- **Food Recommendation System**
- …

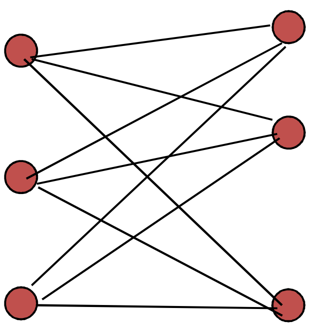
Complete bipartite graph

With $m,n \in \mathbf{N}$, complete bipartite graph $K_{m,n}$ is bipartite graph where

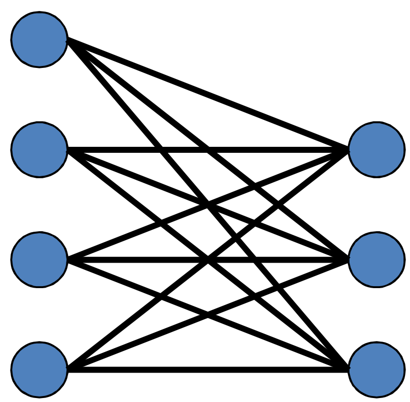
- $|V_1| = m$, $|V_2| = n$
- $E = \{(v_1,v_2) | v_1 \in V_1 \text{ và } v_2 \in V_2\}$.

$K_{m,n}$ is the graph that has its vertex set portioned into two subsets of m and n vertices, respectively. There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.

Example:



$K_{3,3}$



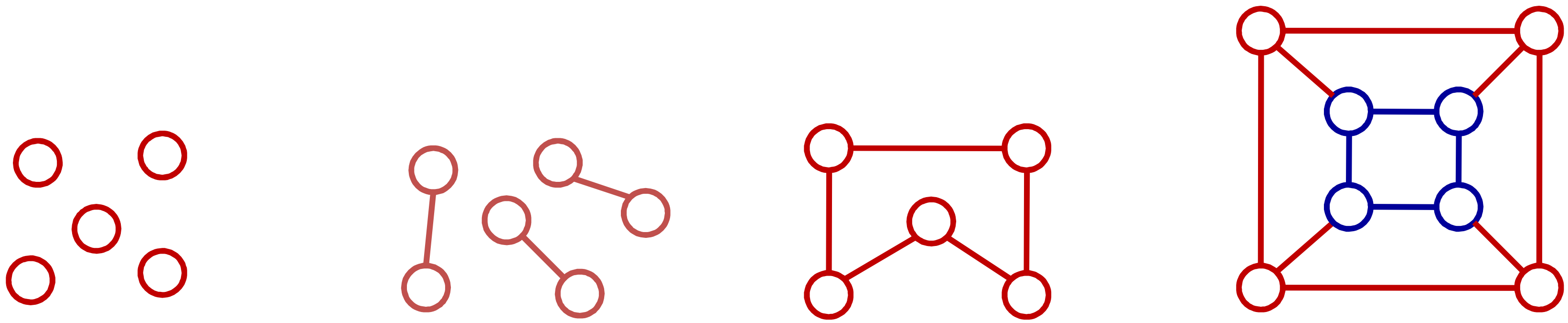
$K_{4,3}$

$K_{m,n}$ has _____ vertices and _____ edges.

r-regular graph

Definition. Graph G = (V, E) is r-regular graph if each vertex $v \in V$ has degree deg(v)= r (has the same number of neighbors)

Example:



0-regular graph | 1-regular graph | 2-regular graph | 3-regular graph

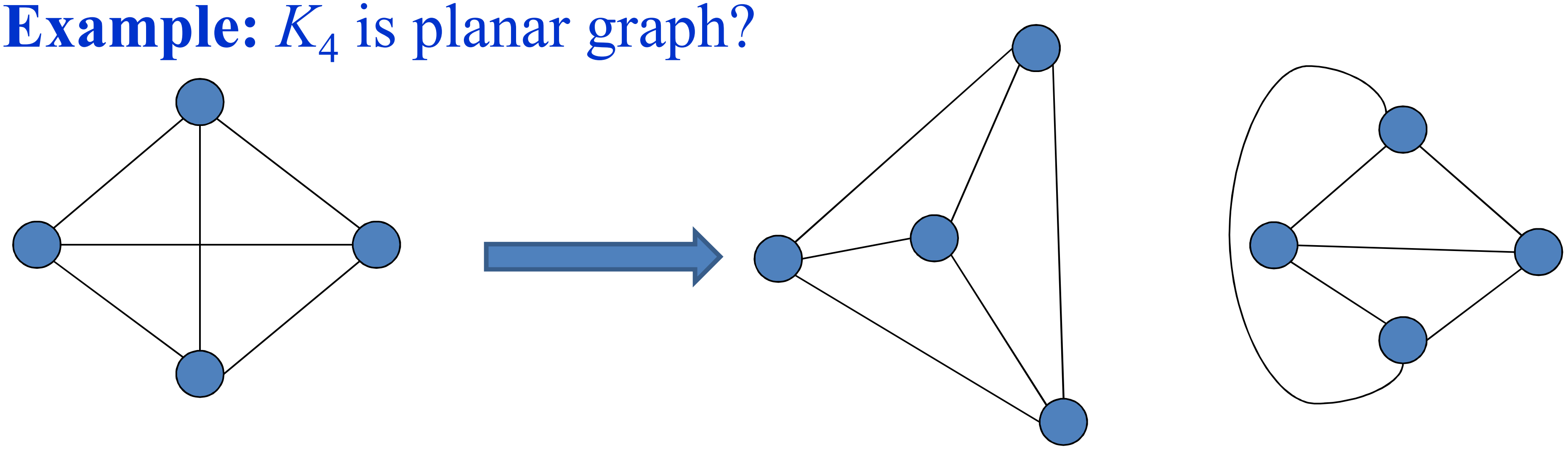
Some special graphs

1. Null graph
2. Complete graphs K_n
3. Cycles C_n
4. Wheels W_n
5. n-Cubes Q_n
6. Bipartite graphs
7. Complete bipartite graphs $K_{m,n}$
8. r-regular graph
9. **Planar graph**
10. Euler graph and Hamilton graph

Planar Graphs (Đồ thị phẳng)

- A graph is called *planar* if it can be drawn in the plane without any edges crossing.
  - A crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint.
  - Such a drawing is called a *planar representation* of the graph.

Example: K_4 is planar graph?



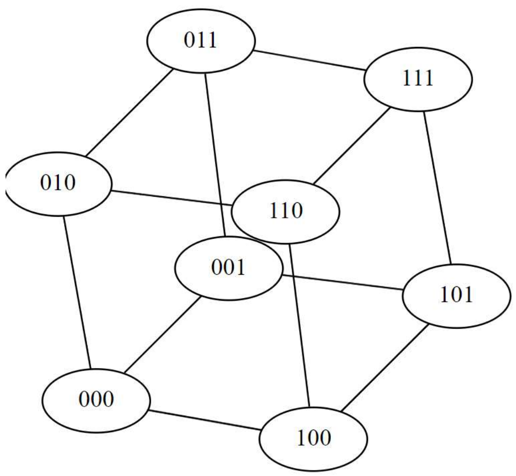
Yes, K_4 is planar graph

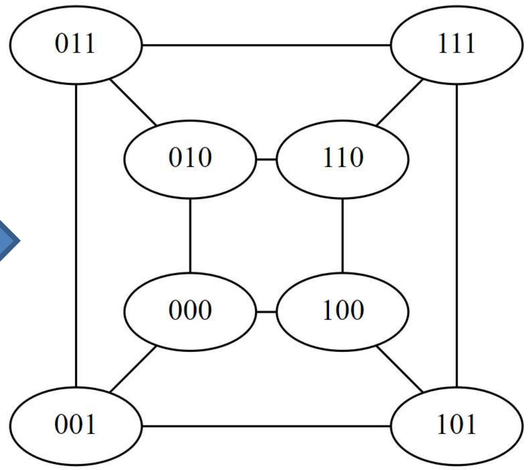
A graph may be planar even if it is usually drawn with crossings, since it may be possible to draw it in another way without crossings.

Planar Graphs (Đồ thị phẳng)

- A graph is called *planar* if it can be drawn in the plane without any edges crossing.

Example: Q_3 is planar graph?



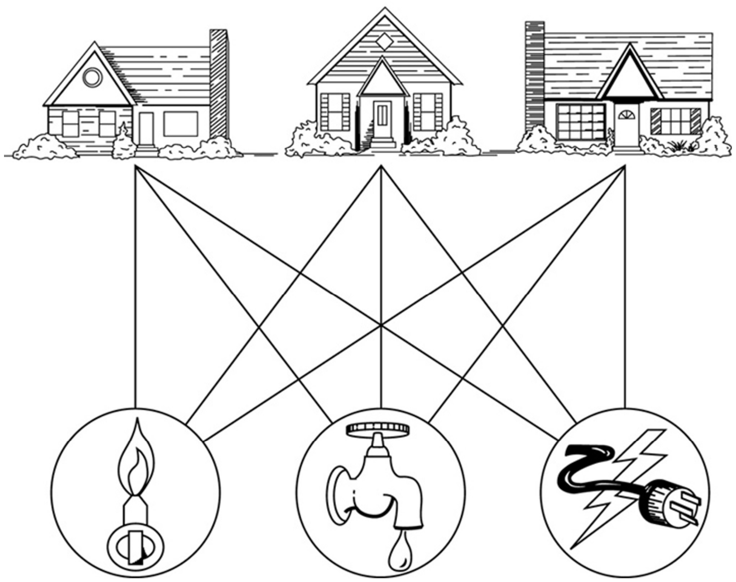


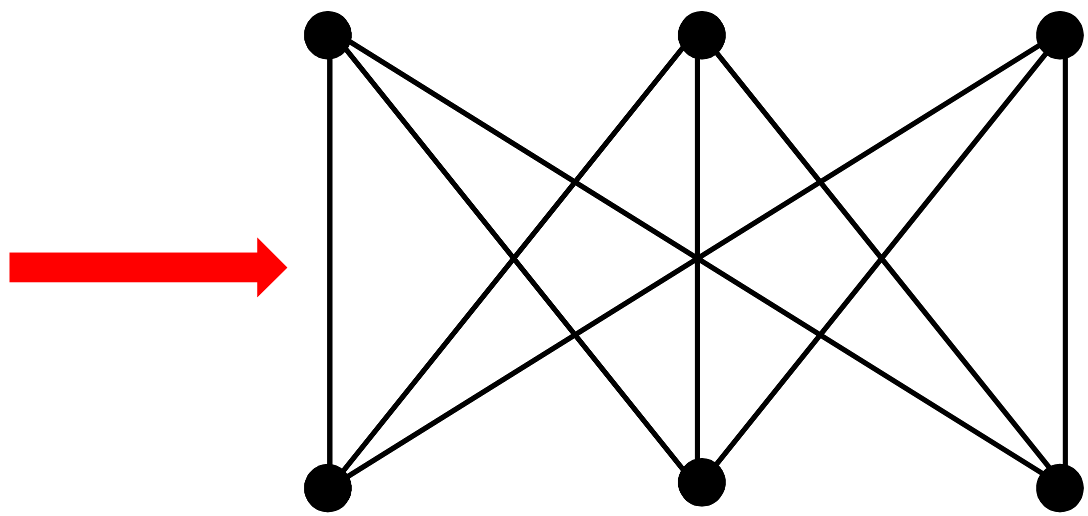
Yes, Q_3 is planar graph!

A graph may be planar even if it represents a 3-dimensional object.

Example: Three house and three utilities problem

- Is it possible to join the three houses to the three utilities in such a way that none of the connections cross?



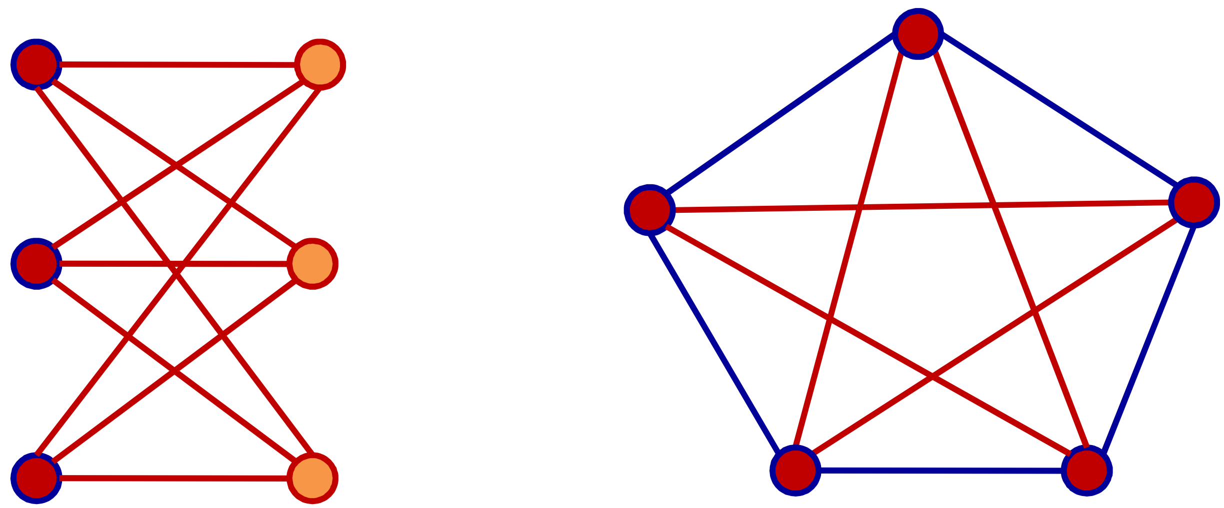


$K_{3,3}$

→ Phrased another way, this question is equivalent to: Given the complete bipartite graph $K_{3,3}$, can $K_{3,3}$ be drawn in the plane so that no two of its edges cross? ($K_{3,3}$ is planar graph?)

$K_{3,3}$ and K_5 are not planar graphs

- $K_{3,3}$ and K_5 are not planar graphs.

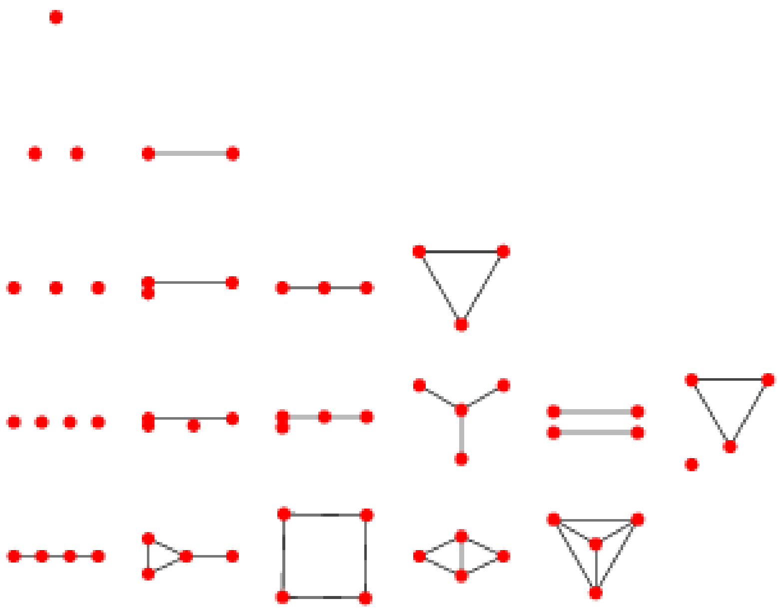


- We can prove that a particular graph is planar by showing how it can be drawn without any crossings.
- However, not all graphs are planar.
- It may be difficult to show that a graph is nonplanar. We would have to show that there is *no way* to draw the graph without any edges crossing.

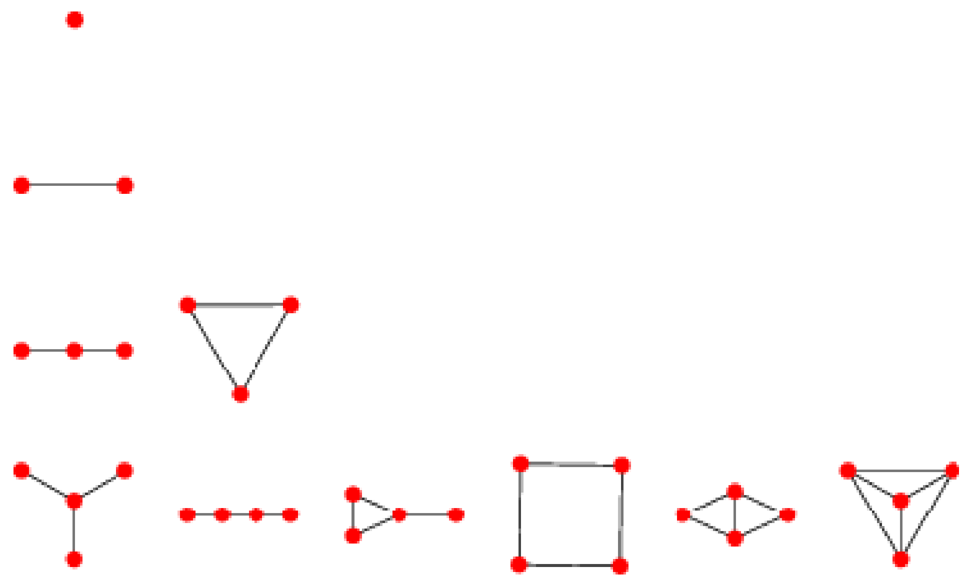
$n=1$

Planar Graphs (Đồ thị phẳng)

The number of planar graphs when number of vertices *n*=1, 2, 3, … are respectively 1, 2, 4, 11, 33, 142, 822, 6966, 79853,….



Planar graph



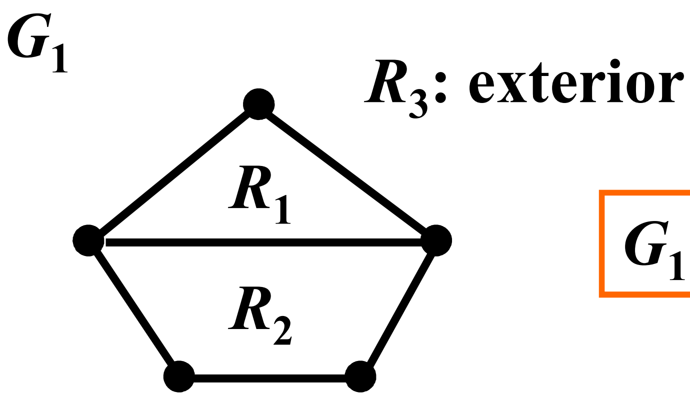
Planar connected graph

And the number of corresponding planar connected graphs are 1, 1, 2, 6, 20, 99, 646, 5974, 71885,…

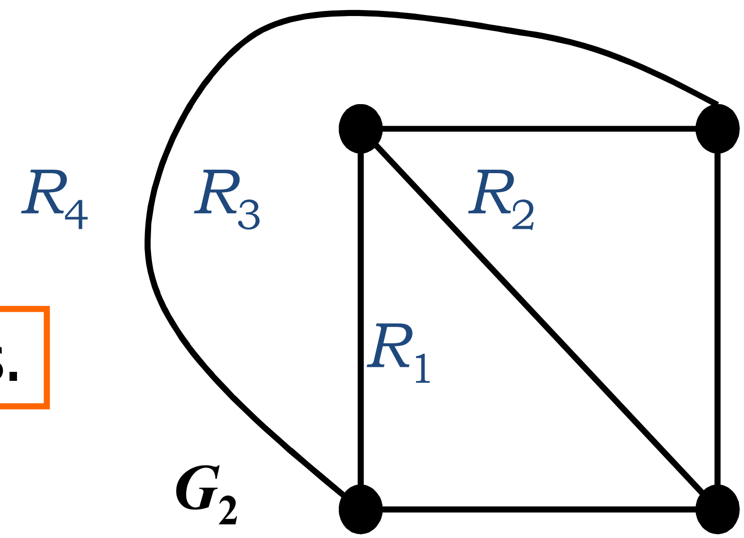
Regions

- Euler showed that all planar representations of a graph split the plane into the same number of *regions*, including an unbounded region.

Example:



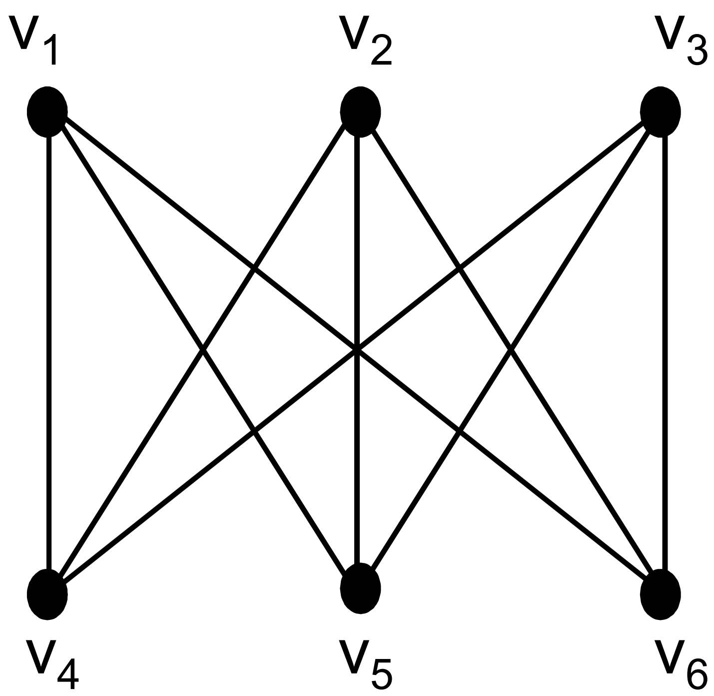
G_1 has 3 regions.



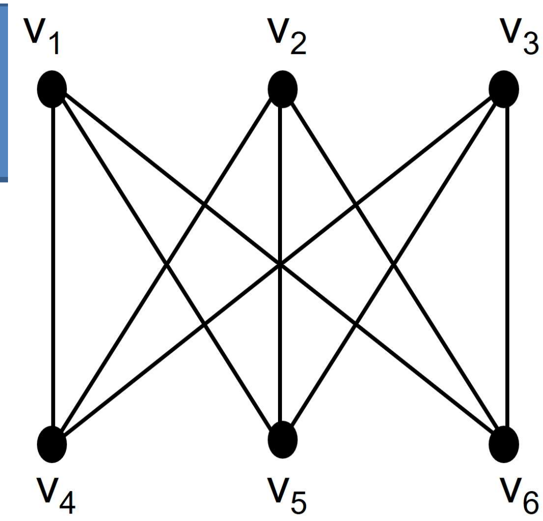
G_2 has 4 regions.

Regions

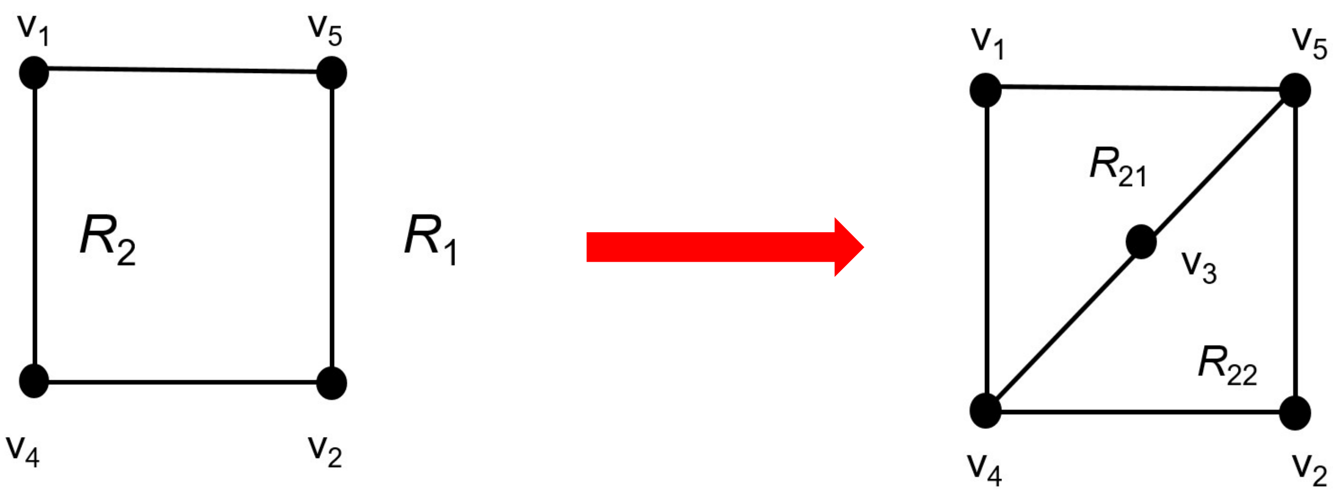
- In any planar representation of $K_{3,3}$, vertex v_1 must be connected to both v_4 and v_5, and v_2 also must be connected to both v_4 and v_5.



Regions



- The four edges $\{v_1, v_4\}$, $\{v_4, v_2\}$, $\{v_2, v_5\}$, $\{v_5, v_1\}$ form a closed curve that splits the plane into two regions, R_1 and R_2.

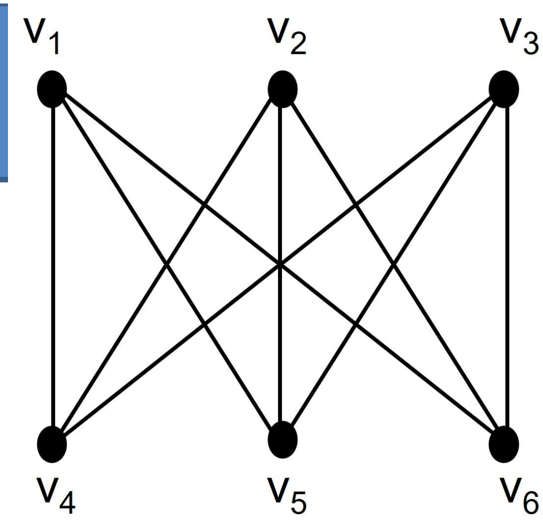


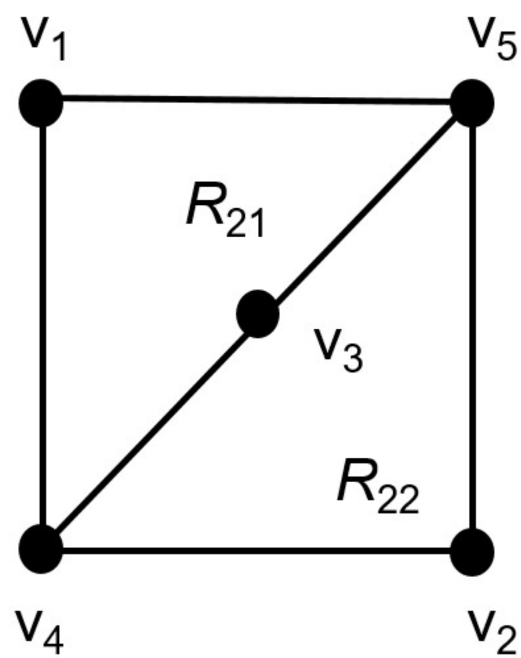
- Next, we note that v_3 must be in either R_1 or R_2.

Case 1: Assume v_3 is in R_2. Then the edges $\{v_3, v_4\}$ and $\{v_4, v_5\}$ separate R_2 into two subregions, R_{21} and R_{22}.

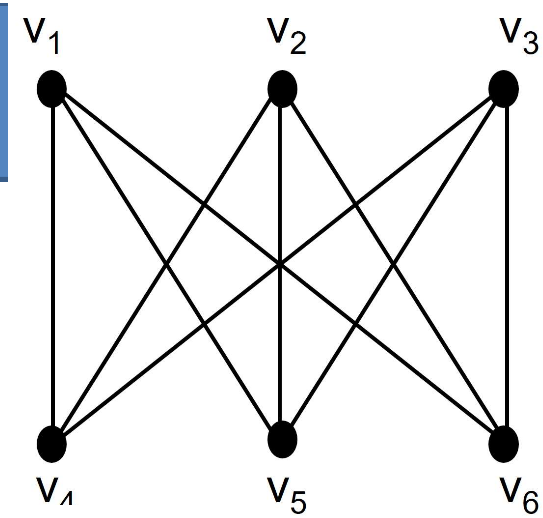
Regions

- Now there is no way to place vertex v_6 without forcing a crossing:
  - If v_6 is in R_1 then $\{v_6, v_3\}$ must cross an edge
  - If v_6 is in R_{21} then $\{v_6, v_2\}$ must cross an edge
  - If v_6 is in R_{22} then $\{v_6, v_1\}$ must cross an edge

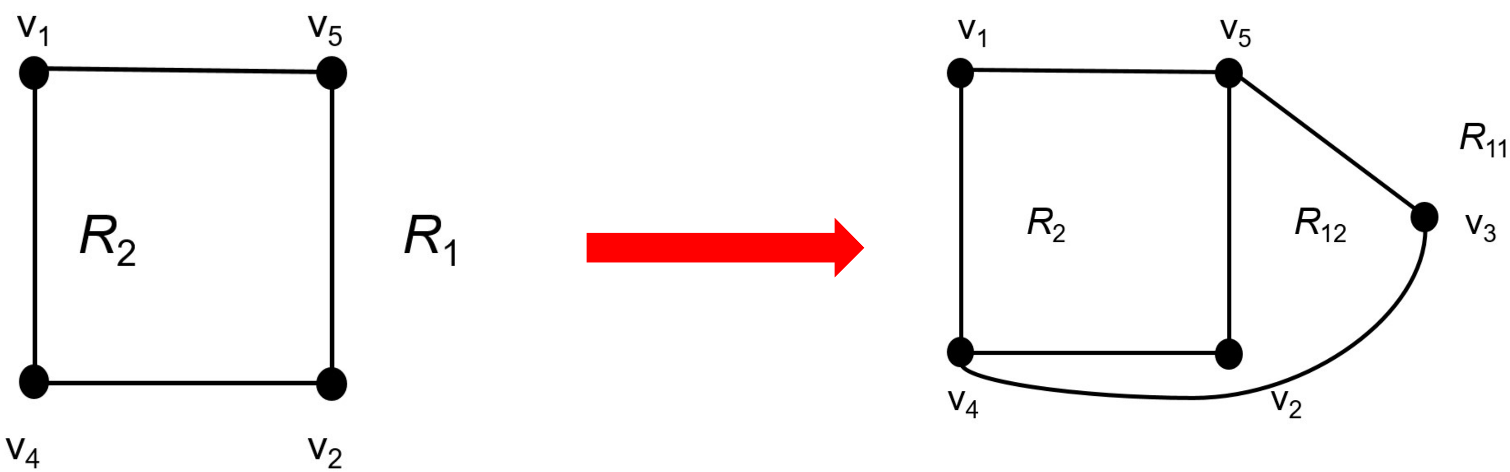




Regions



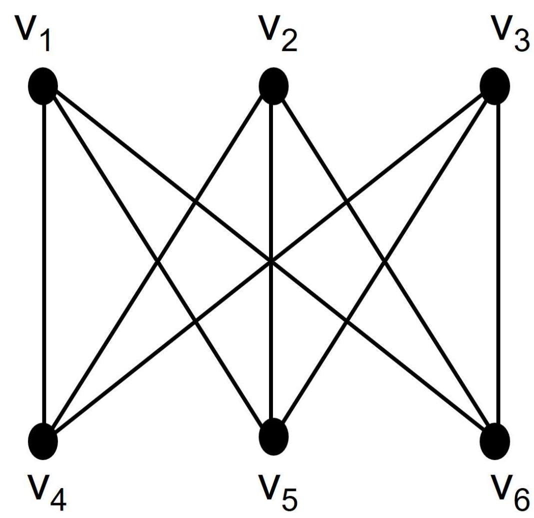
- The four edges $\{v_1, v_4\}$, $\{v_4, v_2\}$, $\{v_2, v_5\}$, $\{v_5, v_1\}$ form a closed curve that splits the plane into two regions, R_1 and R_2.



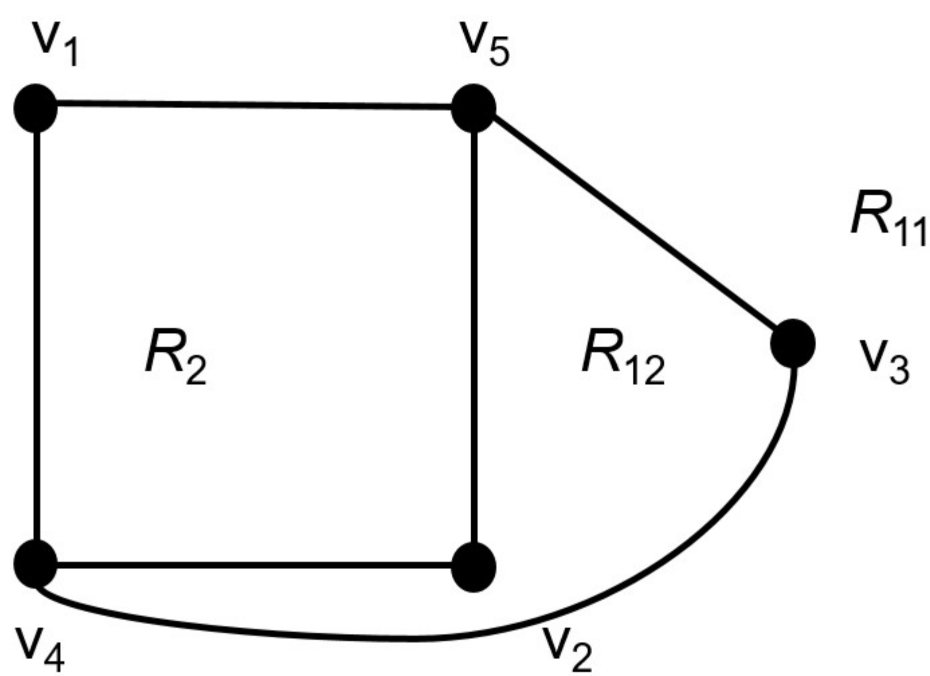
- Next, we note that v_3 must be in either R_1 or R_2.

Case 2: Assume v_3 is in R_1. Then the edges $\{v_3, v_4\}$ and $\{v_4, v_5\}$ separate R_1 into two subregions, R_{11} and R_{12}.

Regions

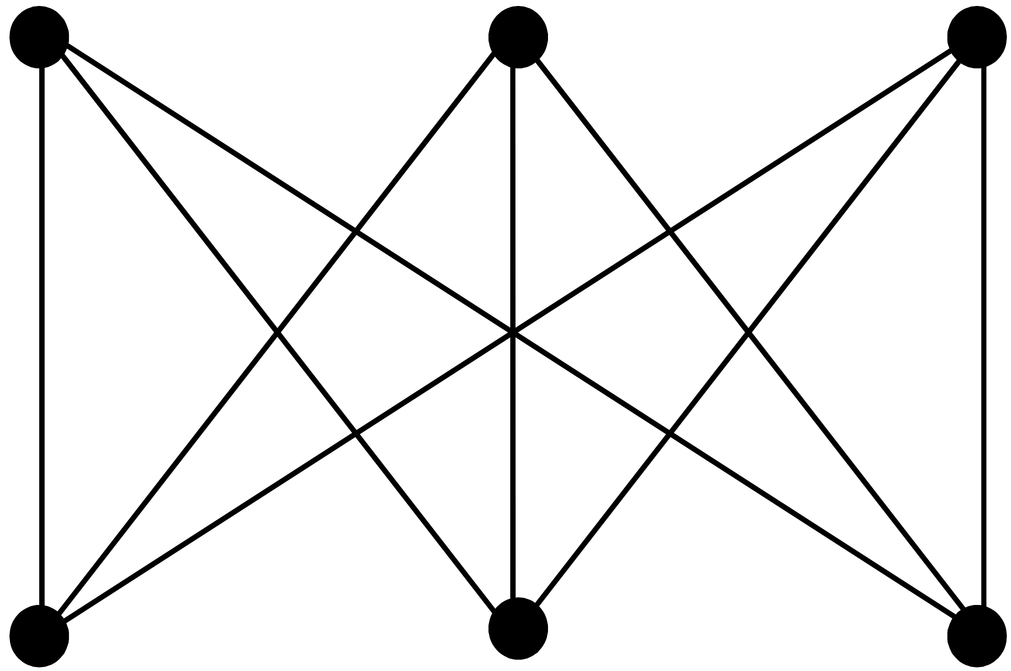


- Now there is no way to place vertex v_6 without forcing a crossing:
  - If v_6 is in R_2 then $\{v_6, v_3\}$ must cross an edge
  - If v_6 is in R_{11} then $\{v_6, v_2\}$ must cross an edge
  - If v_6 is in R_{12} then $\{v_6, v_1\}$ must cross an edge



Planar Graphs

- Consequently, the graph $K_{3,3}$ must be nonplanar.



$K_{3,3}$

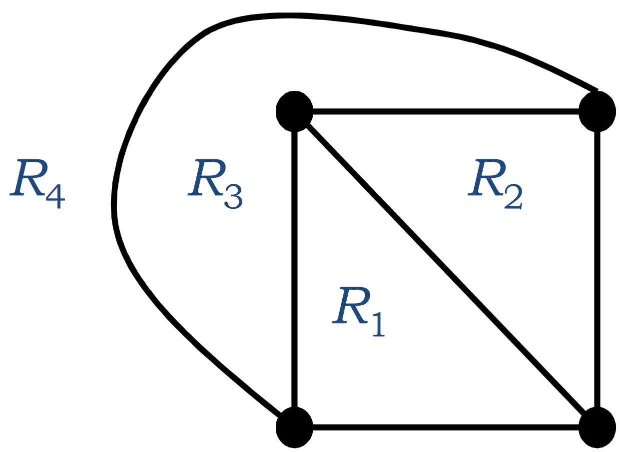
Regions

- Euler devised a formula for expressing the relationship between the number of vertices, edges, and regions of a planar graph.
- These *may* help us determine if a graph can be planar or not.

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

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 $r = e - v + 2$.



\# of edges, $e = 6$
\# of vertices, $v = 4$
\# of regions, $r = e - v + 2 = 4$

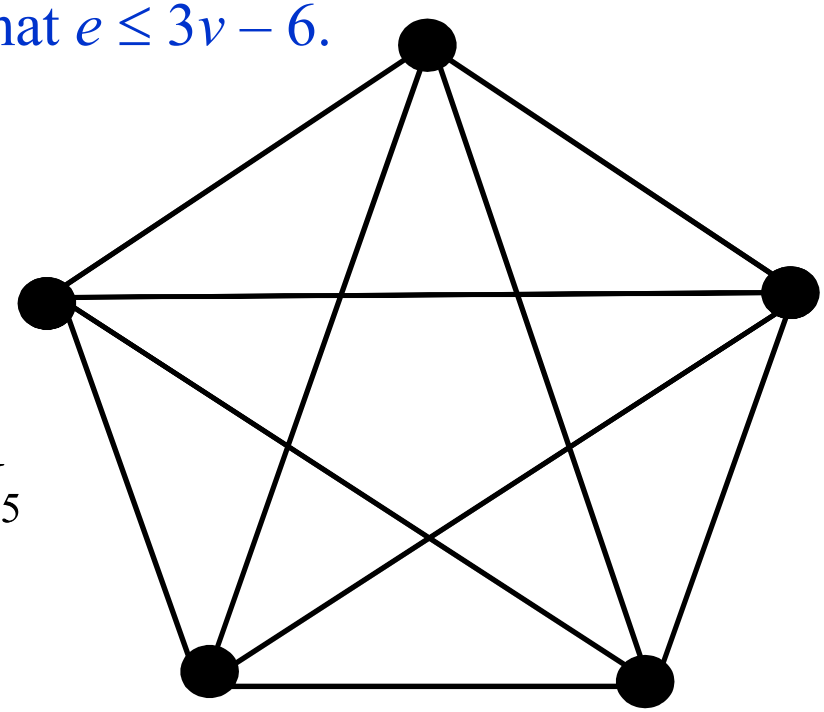
Euler's Formula (Cont.)

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

Is K_5 planar?

- K_5 has 5 vertices and 10 edges.
- We see that $v \geq 3$.
- So, if K_5 is planar, it must be true that $e \leq 3v - 6$.
- $3v - 6 = 3*5 - 6 = 15 - 6 = 9$.
- So e must be ≤ 9.
- But $e = 10$.
- So, K_5 is nonplanar.

K_5



Euler's Formula (Cont.)

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

If G has one or two vertices, it is true;
thus, we assume that G has at least three vertices.

If the degree of each vertex were at least 6, then by Handshaking Theorem,
2e ≥ 6v, i.e., e ≥ 3v,

$$2e = \sum_{v \in V} \deg(v)$$

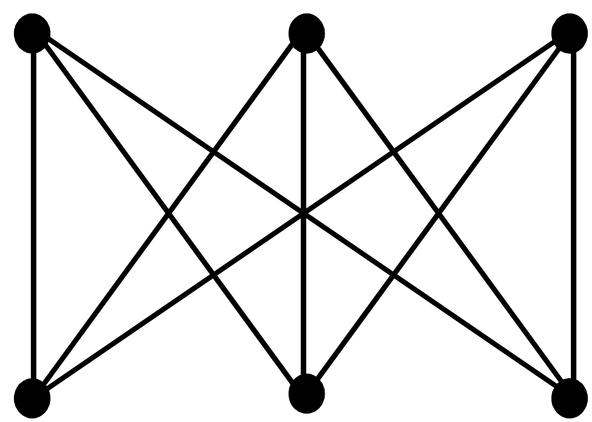
but this contradicts the inequality from
Corollary 1: e ≤ 3v – 6.

Euler's Formula (Cont.)

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

Is $K_{3,3}$ planar?

- $K_{3,3}$ has 6 vertices and 9 edges.
- Obviously, $v \geq 3$ and there are no circuits of length 3.
- If $K_{3,3}$ were planar, then $e \leq 2v - 4$ would have to be true.
- $2v - 4 = 2*6 - 4 = 8$
- So e must be ≤ 8.
- But $e = 9$.
- So $K_{3,3}$ is nonplanar.

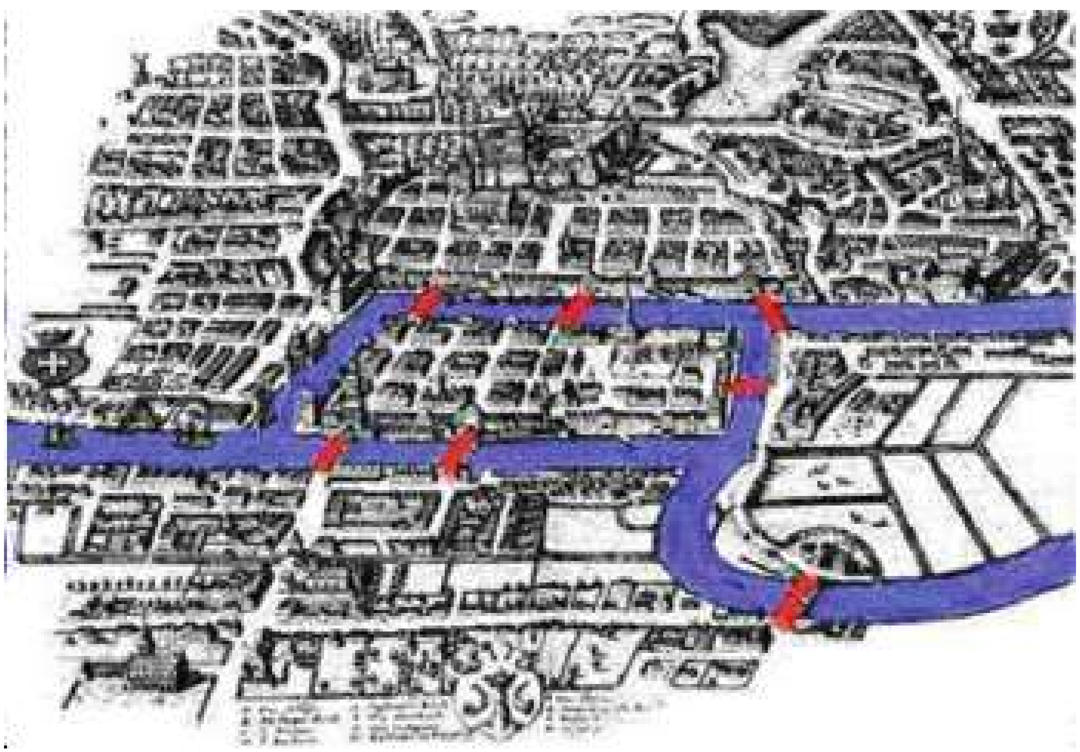


Some special graphs

1. Null graph
2. Complete graphs K_n
3. Cycles C_n
4. Wheels W_n
5. n-Cubes Q_n
6. Bipartite graphs
7. Complete bipartite graphs $K_{m,n}$
8. r-regular graph
9. Planar graph
10. **Euler graph and Hamilton graph**

Seven Bridges of Königsberg

- The city of Königsberg, Russia was set on both sides of the Pregel River, and included two large islands (Kneiphof and Lomse) which were connected to each other, or to the two mainland portions of the city, by seven bridges. The problem was to devise a walk through the city that would cross each of those bridges exactly once.
- In 1736, Euler proved that the problem has no solution.

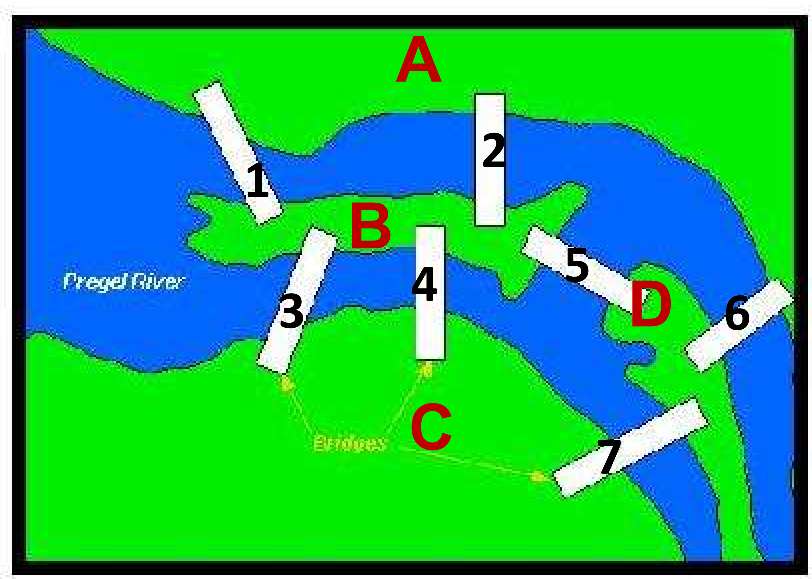


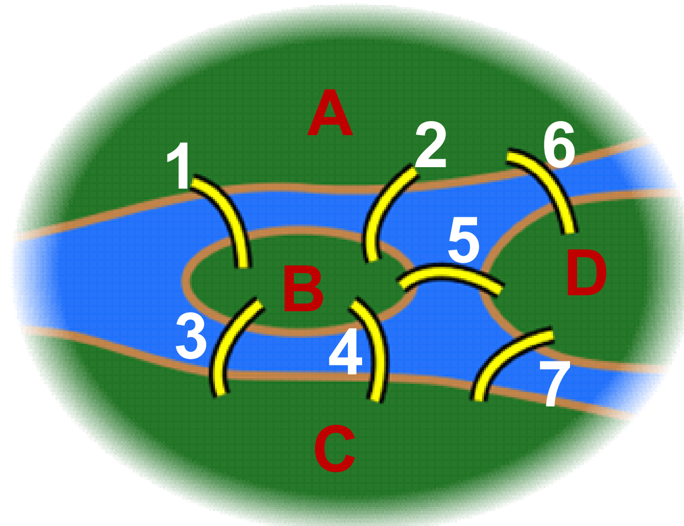
Leonhard Euler

1707-1783

Seven Bridges of Königsberg

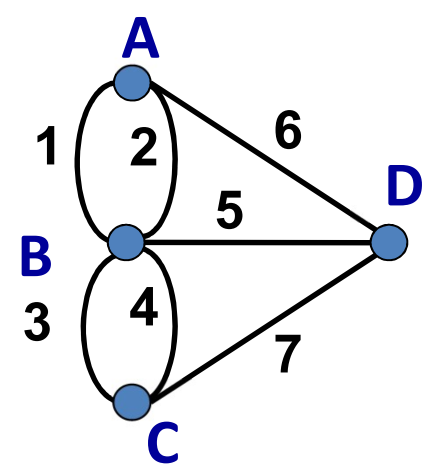
- To prove it, Euler reformulated the problem in graph terms:
  - Each land mass ~ a vertex
  - Each bridge ~ an edge





Is there a way to go through all 7 bridges, each exactly once, and then return to the starting position?

Whether or not there exists a cycle on a graph G that traverses through every edge of G exactly once.



Euler graph

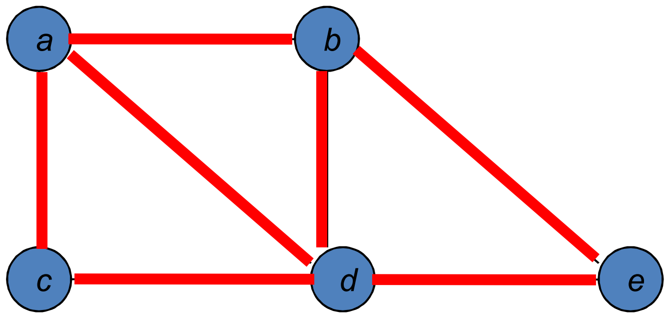
- Definition
- Recognize Euler graph

Euler graph

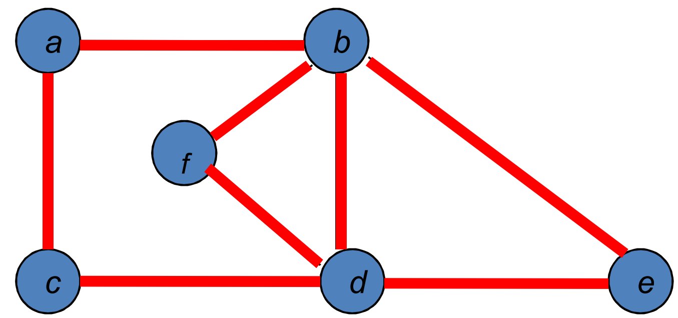
- ***Euler path*** (***Eulerian trail***, ***Euler walk***) in graph is a path that traverses through every edge exactly once.
- ***Euler cycle*** (***Eulerian circuit***, ***Euler tour***) in graph is a Euler path begins and ends at the same vertex (is a cycle that traverses through every edge exactly once).

- Graph consisting of Euler cycle is called as ***Euler graph***.
- Graph consisting of Euler path is called as ***Half Euler graph.***
- Apparently, all Euler graphs are also half-Euler graphs.

Example



Half Euler graph



Euler graph

Euler path: a, c, d, b, e, d, a, b

Euler cycle: a, c, d, e, b, d, f, b, a

Euler path in graph is a path that traverses through every edge exactly once.
Euler cycle in graph is a cycle that traverses through every edge exactly once.

Half Euler graph: graph consists of Euler path

Euler graph: graph consists of Euler cycle

Euler's 1st theorem

- If a graph has any vertices of odd degree, then it can not have any Euler cycle.
- If a graph is connected and every vertex has an even degree, then it has at least one Euler cycle.

Proof:

- If a node has an odd degree, and the cycle starts at this node, then it must end elsewhere. This is because after we leave the node the first time the node has even degree, and every time we return to the node we must leave it. (On the paired arc.)
- If a node has an odd degree, and the cycle begins else where, then it must end at the node. This is a contradiction, since a cycle must end where it began.

If a graph has all even degree nodes, then an Euler Circuit exists.

Algorithm:

- Step One: Randomly move from node to node, until stuck. Since all nodes had even degree, the circuit must have stopped at its starting point. (It is a circuit.)
- Step Two: If any of the arcs have not been included in our circuit, find an arc that touches our partial circuit, and add in a new circuit.
  - Each time we add a new circuit, we have included more nodes.
  - Since there are only a finite number of nodes, eventually the whole graph is included.

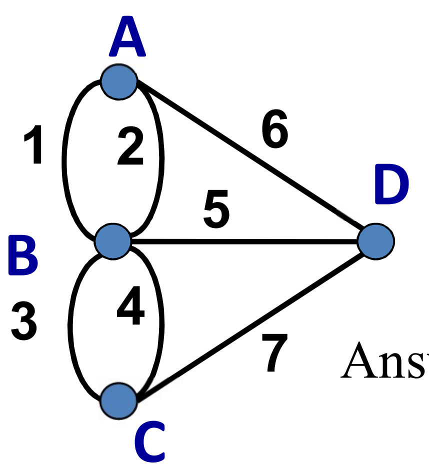
Euler's 2nd theorem

- If a graph has more than two vertices of odd degree, then it cannot have an Euler path.
- If a graph is connected and has exactly two vertices of odd degree, then is has at least one Euler path. Any such path must start at one of the odd degree vertices and must end at the other odd degree vertex.

Euler's theorems

If a graph is connected and if the number of odd degree vertices

- = 0, then Euler cycle (Theorem 1)
- = 2, then Euler path (Theorem 2)
  - This Euler path must start at one of the odd degree vertices and must end at the other odd degree vertex.



Is there a way to go through all 7 bridges, each exactly once, and then return to the starting position?

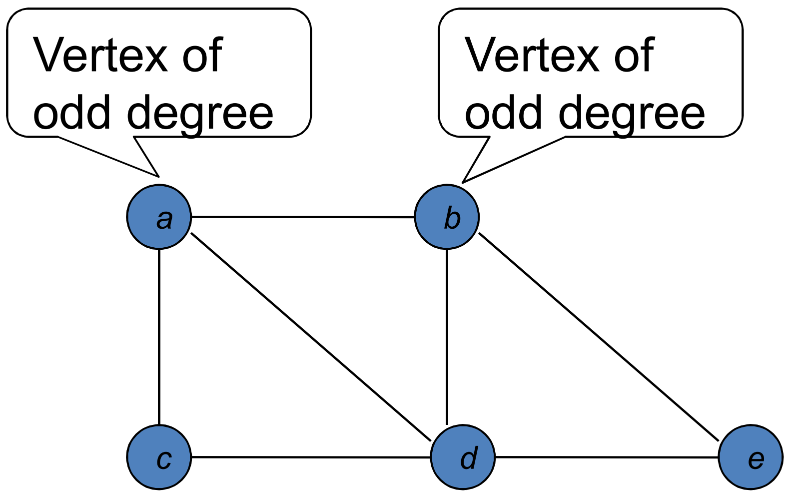
Whether or not there exists a cycle on a graph G that traverses through every edge of G exactly once.

Answer: *There exists vertex of odd degree ➔ don't have Euler cycle*

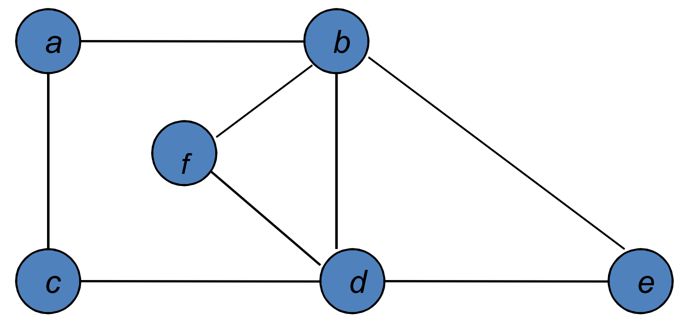
Whether or not there exists Euler path in G*?*

Answer: *There are 3 vertices of degree 3, one vertex of degree 5 ➔ don't have Euler path*

Example



Half Euler graph



Euler graph

Hamilton graph

- Definition
- Recognize Hamilton graph

Euler graph

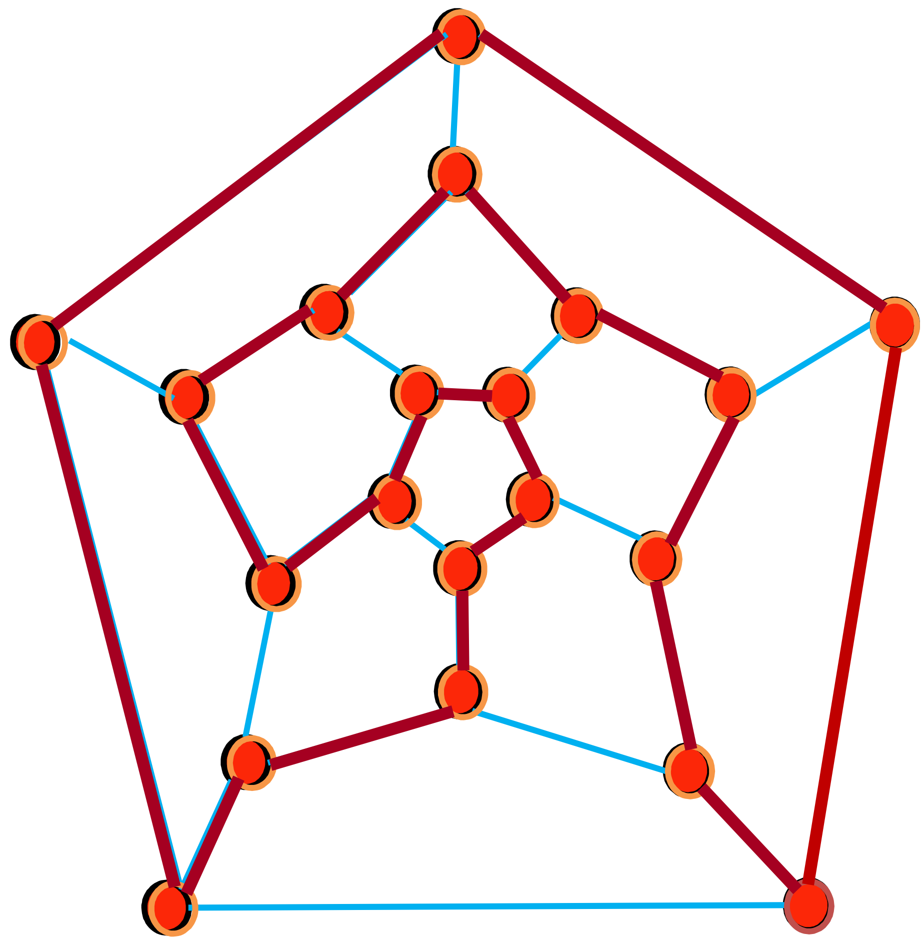
- ***Euler path*** (***Eulerian trail***, ***Euler walk***) in graph is a path that traverses through every edge exactly once.
- ***Euler cycle*** (***Eulerian circuit***, ***Euler tour***) in graph is a Euler path begins and ends at the same vertex (is a cycle that traverses through every edge exactly once).
- Graph consisting of Euler cycle is called as ***Euler graph***.
- Graph consisting of Euler path is called as ***Half Euler graph.***

Hamilton graph

- ***Hamilton path*** in graph is a path that traverses every vertex exactly once.
- ***Hamilton cycle*** in graph is a cycle that traverses every vertex exactly once.
- Graph consisting of Hamilton cycle is called as ***Hamilton graph***.
- Graph consisting of Hamilton path is called as ***Half Hamilton graph.***
- Apparently, all Hamilton graphs are also half-Hamilton graphs.

Example: Hamilton graph

- Is graph consisting of Hamilton cycle: traverse every vertex exactly once.

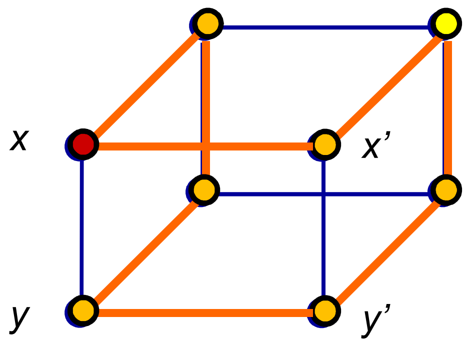


Example: Proof Q_n ($n \geq 3$) is Hamilton graph.

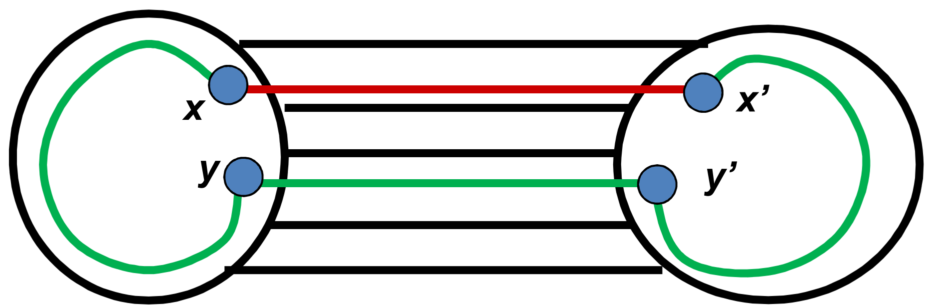
Proof Induction to ***n*.**

- **Basic step: *n*=3 true**
- **Inductive step:** Assume $Q_{n\text{-}1}$ is Hamilton graph.

  Consider Q_n:



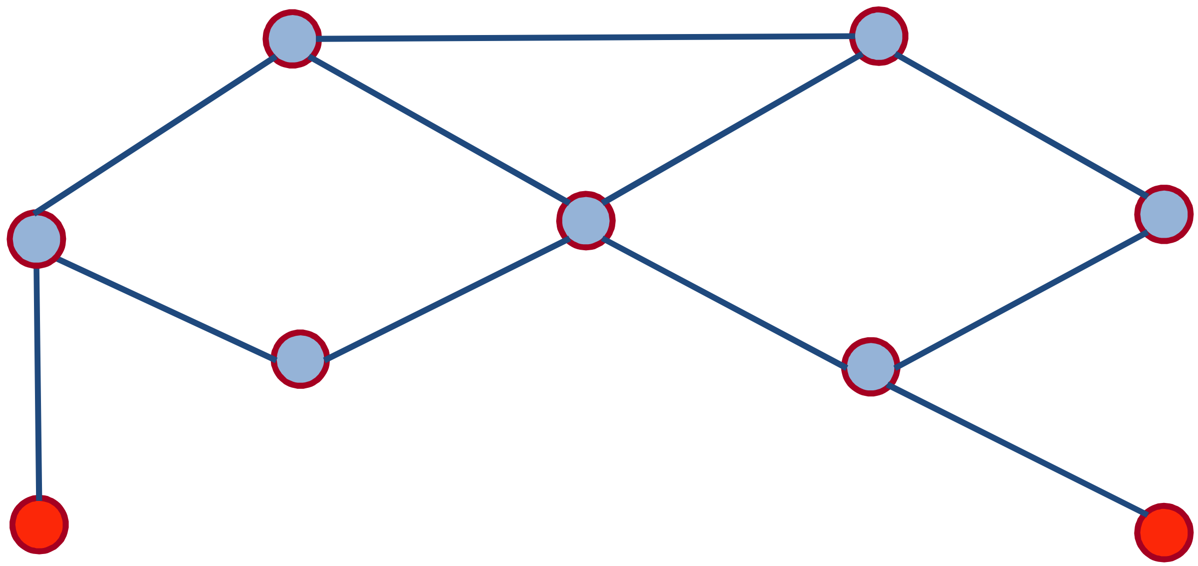
3 - cube



(n –1)-cube (n –1)-cube

Hamilton graph

- Graph has 2 vertices of degree 1$\Rightarrow$ not Hamilton graph

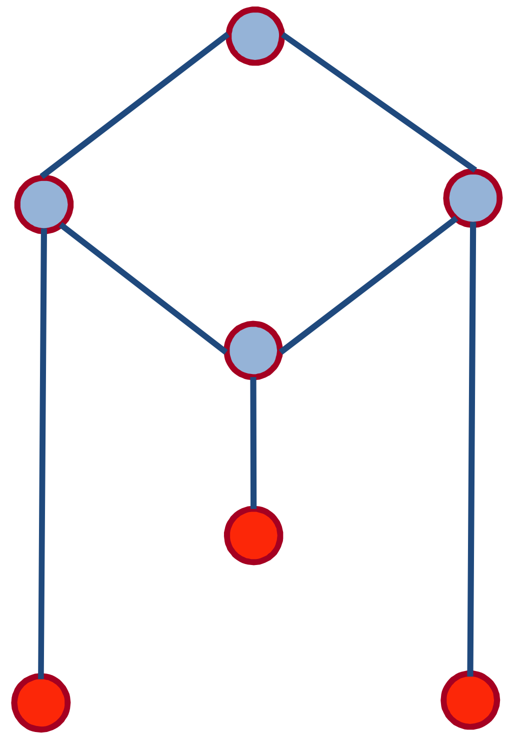


- The above graph is Half Hamilton graph

Half Hamilton graph

- Vertices of degree 1 must be either the starting or ending position of the Hamilton path.

Graph has 3 vertices of degree 1
$\Rightarrow$ Not Halft Hamilton graph

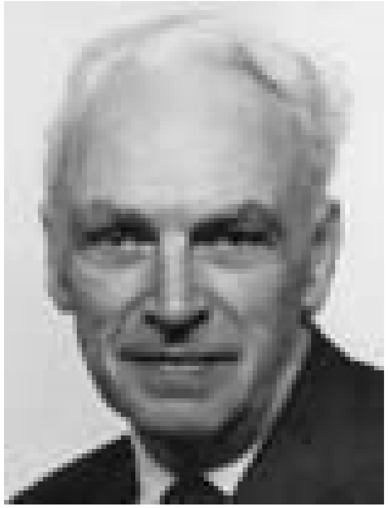


Theorem about existence of Hamilton path

- **Theorem Dirac**: If G is simple connected graph with $n \geq 3$ vertices, and $\forall v$ $\deg(v) \geq n/2$, then G has Hamilton cycle.

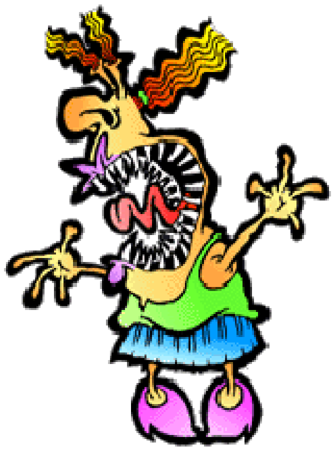
- **Theorem Ore:** If G is simple connected graph with $n \geq 3$ vertices, and $\deg(u) + \deg(v) \geq n$ for all vertices pair u, v not adjacent, then G has Hamilton cycle.

Paul Adrien Maurice Dirac
1902 - 1984
(USA)

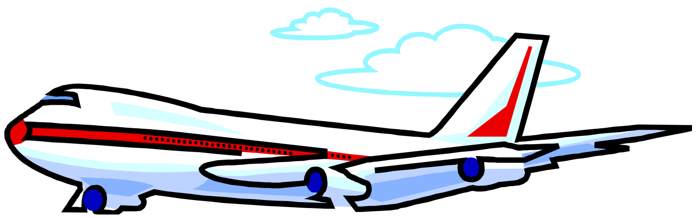


Oystein Ore
1899 - 1968
(Norway)

Content

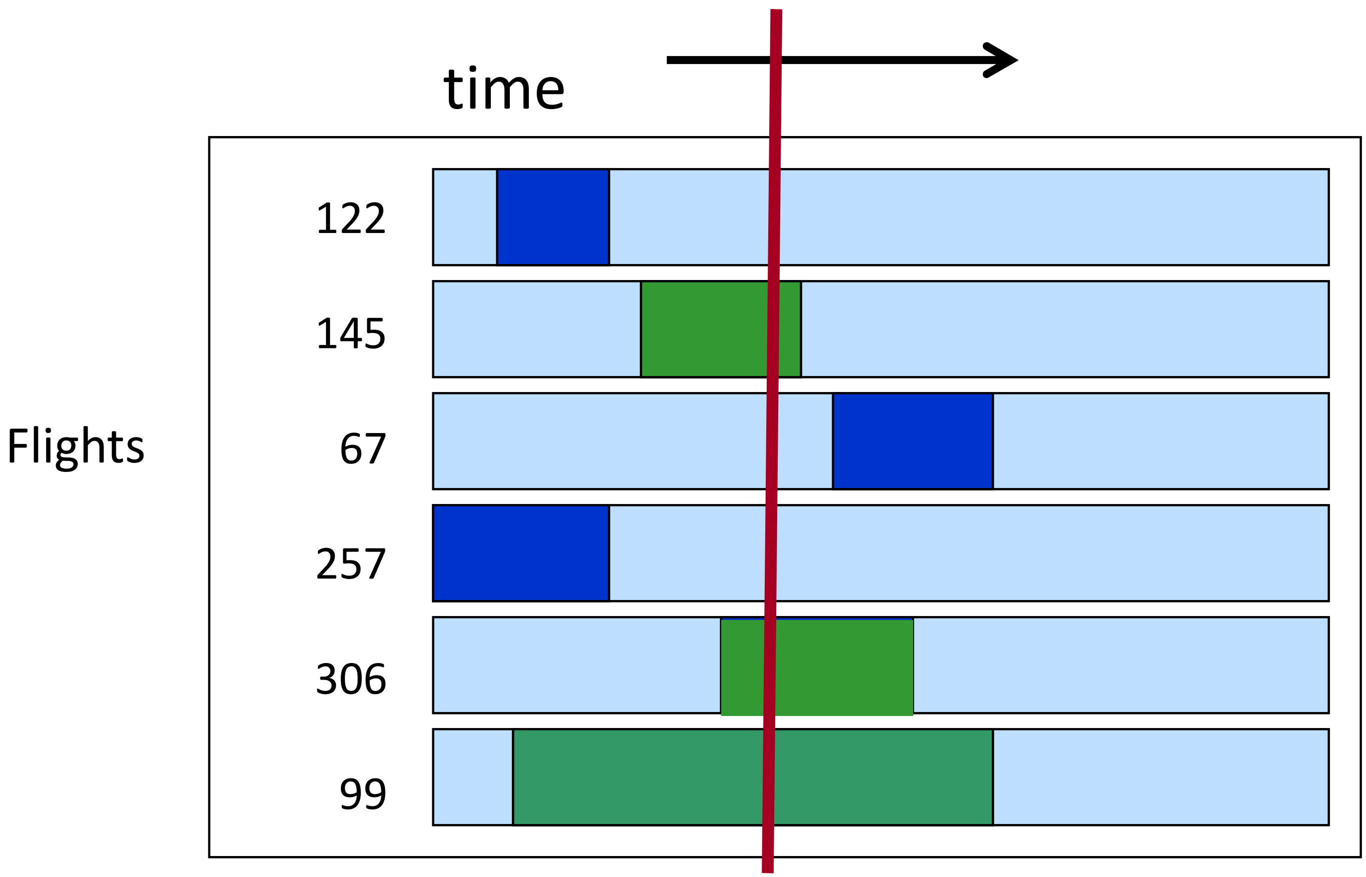


Graph coloring problem: Application example

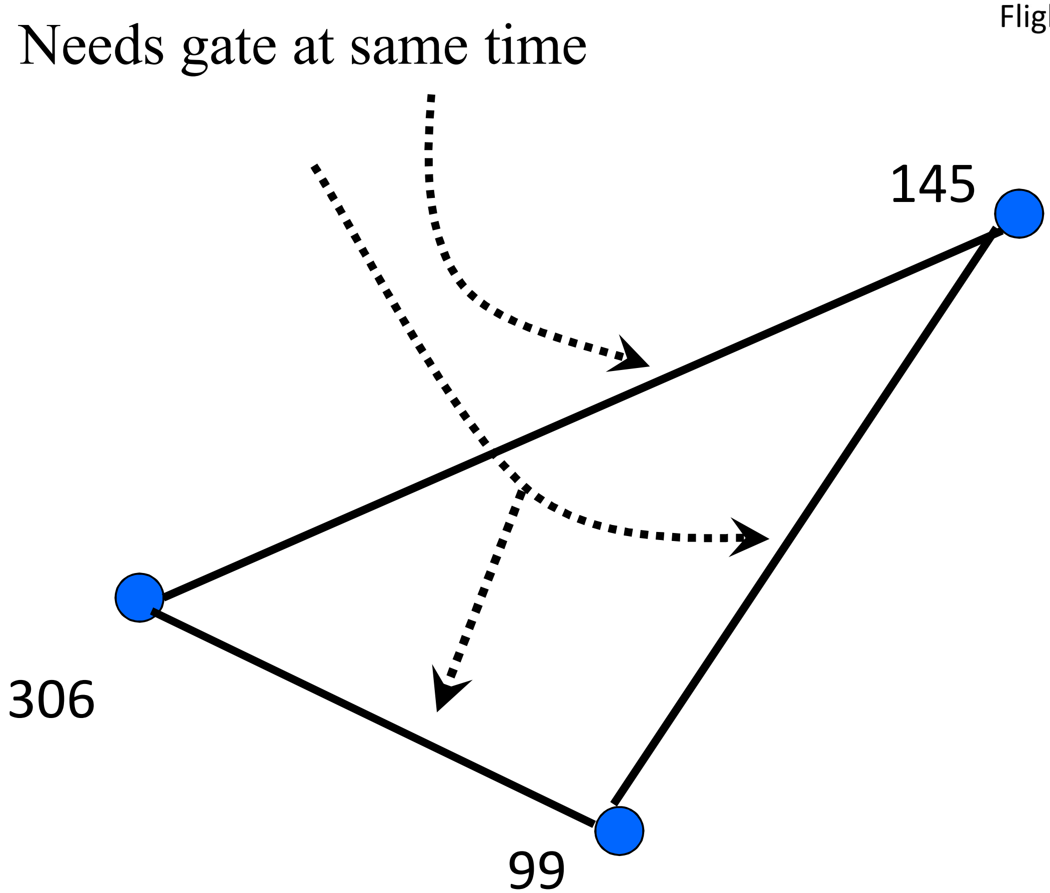


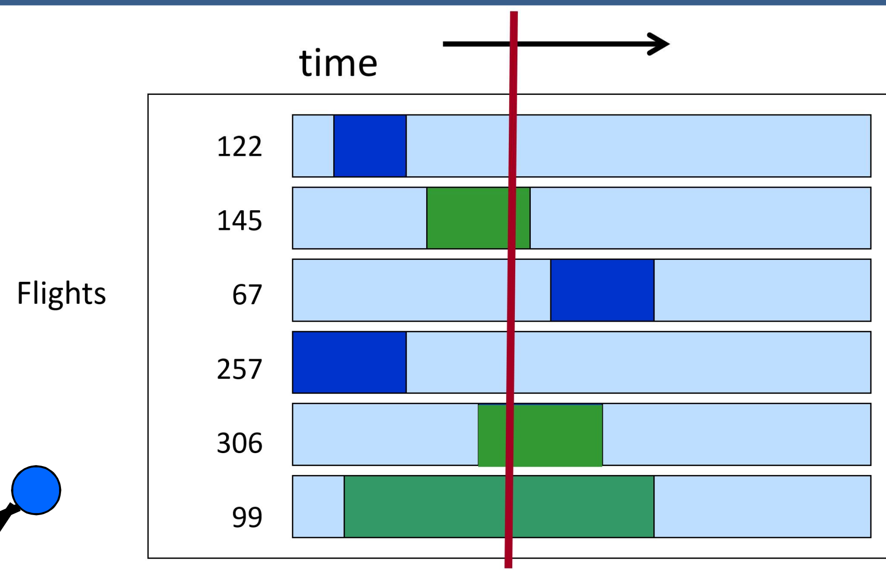
flights need gates, but times overlap.

how many gates needed?



Graph coloring problem: Application example

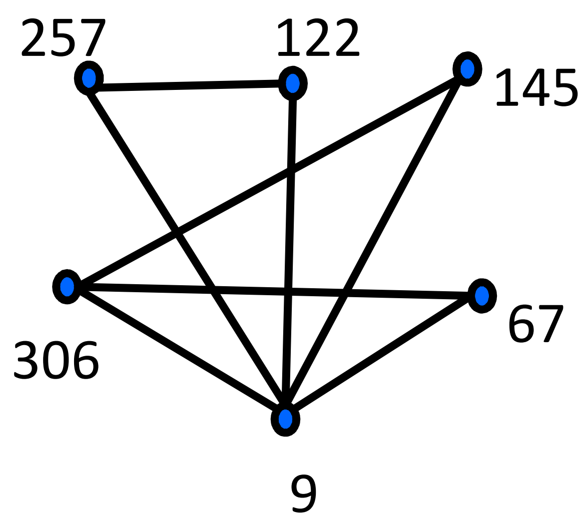


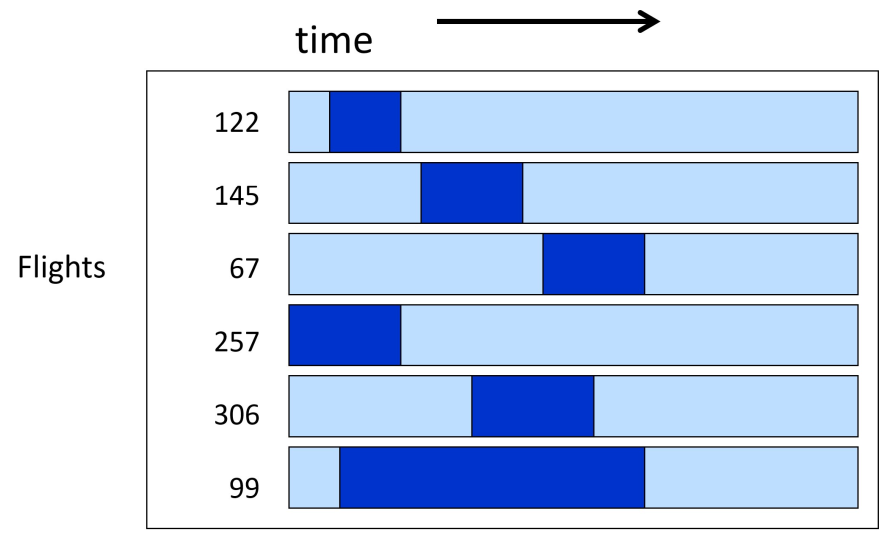


- Each vertex represents a flight
- Each edge represents a conflict

Graph coloring problem: Application example

Conflict Graph





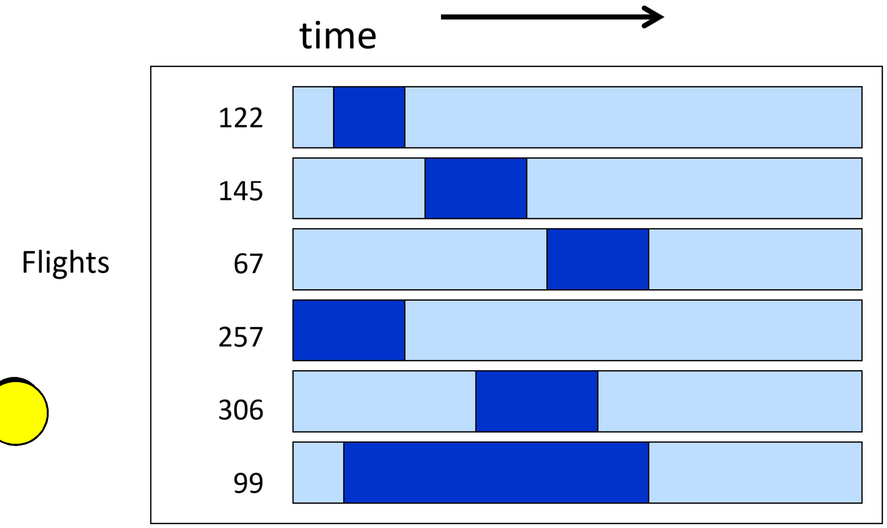
flights need gates, but times overlap.

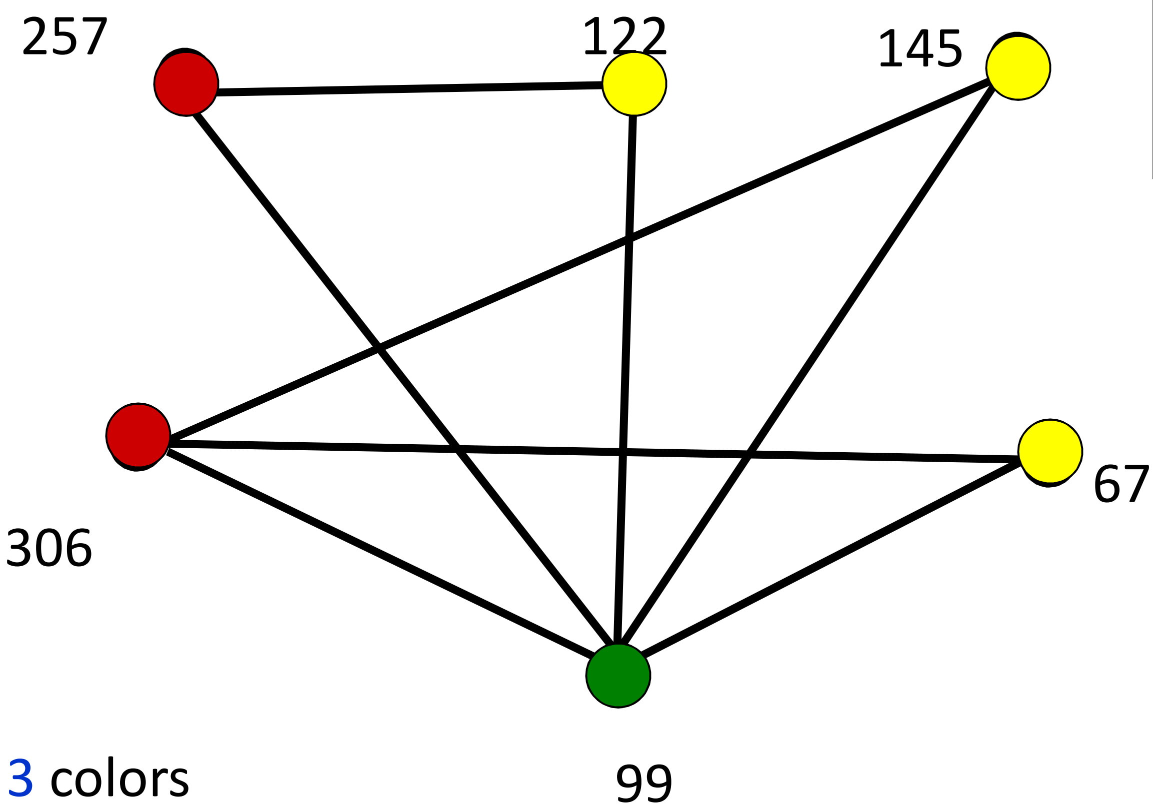
how many gates needed?

assignment of gates to all vertices such that two adjacent vertices get different gates.

- Each vertex represents a flight
- Each edge represents a conflict

Graph coloring problem: Application example





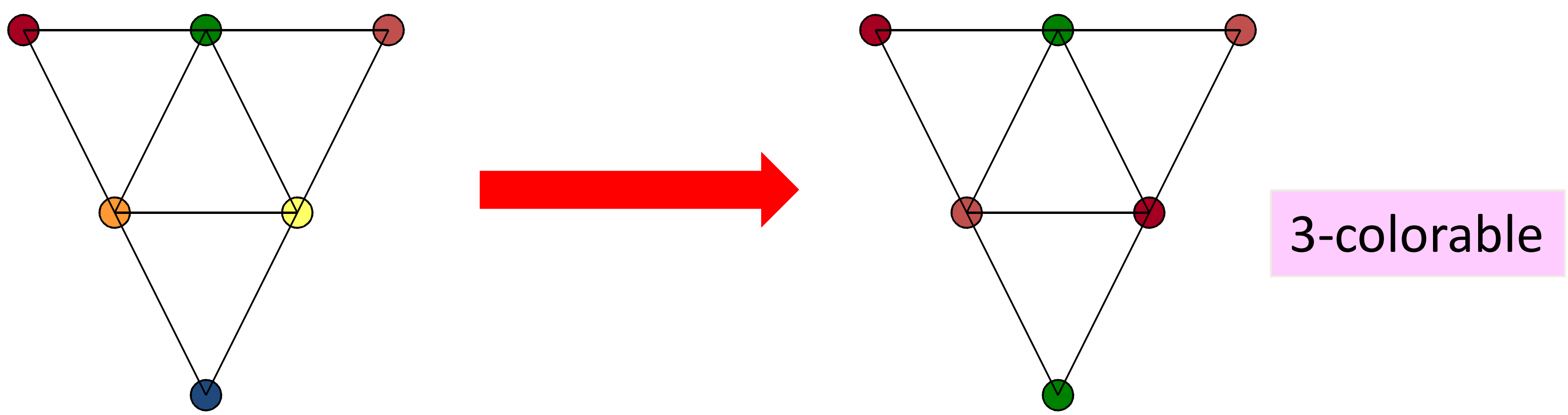
3 colors
3 gates

Assign gates:

257, 306
67, 122,145
99

What is Coloring?

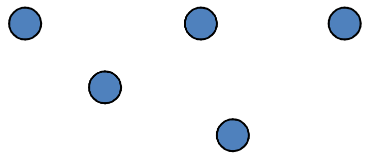
- Graph Coloring Problem is an assignment of colors to all vertices of a given graph such that two adjacent vertices get different colors.
- **Objective:** use minimum number of colors.



Optimum Coloring

Definition. min #colors for G is chromatic number, $\chi(G)$

What graphs have chromatic number one?



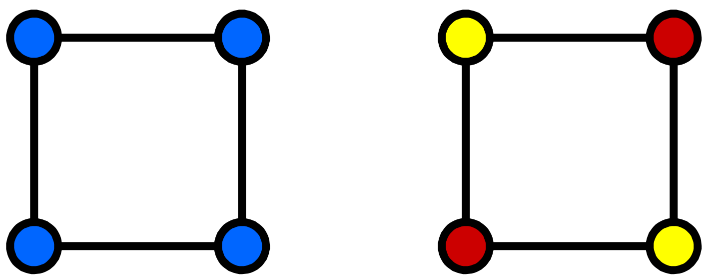
when there are no edges…

What graphs have chromatic number 2?

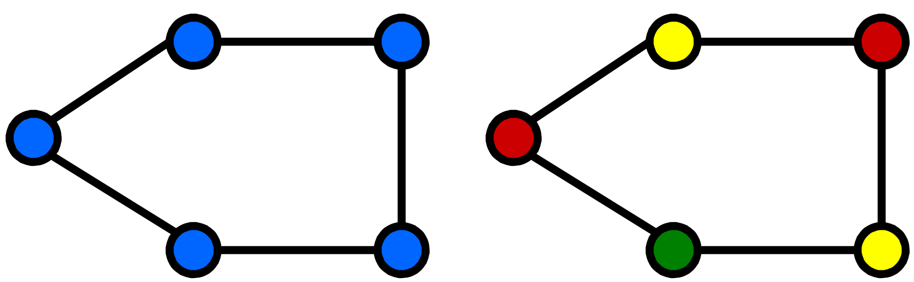
What graphs have chromatic number larger than 2?

A path? A cycle? A triangle?

Simple cycle



$$\chi(C_{even}) = 2$$



$$\chi(C_{odd}) = 3$$

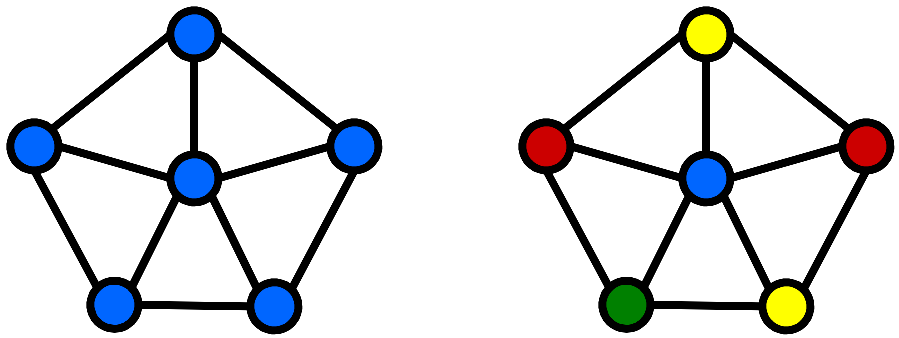
NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Complete graph

$$\chi(K_n) = n$$

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

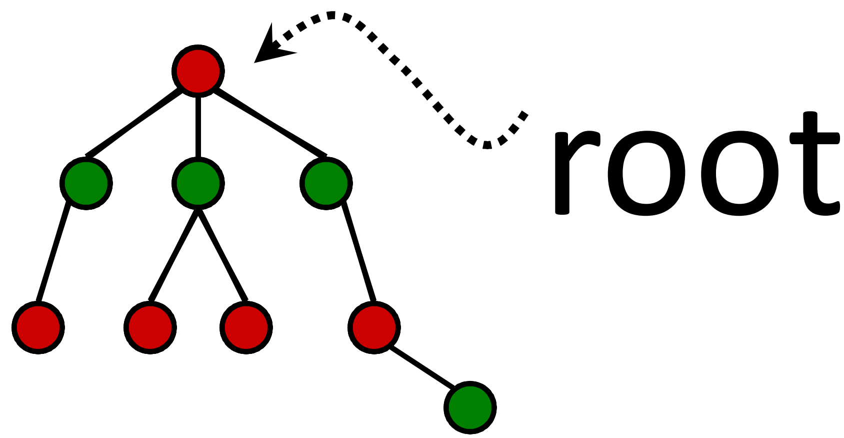
Wheel



W_5

$\chi(W_{odd}) = 4 \quad \chi(W_{even}) = 3$

Tree



Pick any vertex as "root."

if (unique) path from root is

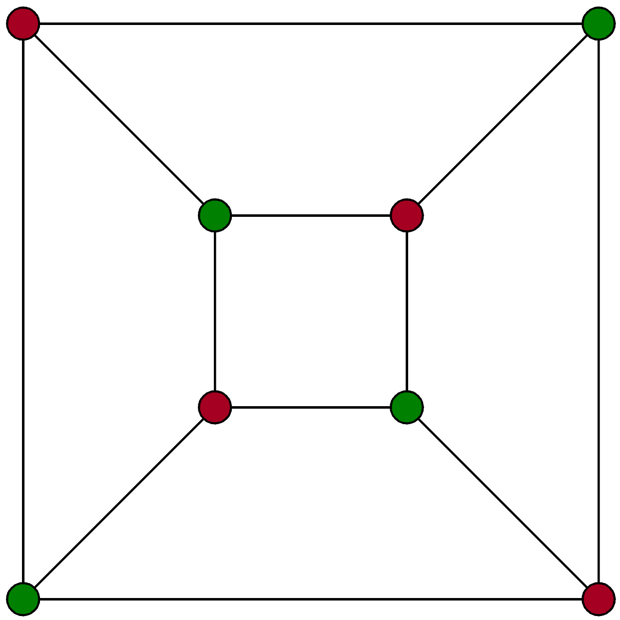
even length:

odd length:

Can prove more formally using induction.

2-colorable graph

When exactly is a graph 2-colorable?



This is 2-colorable.

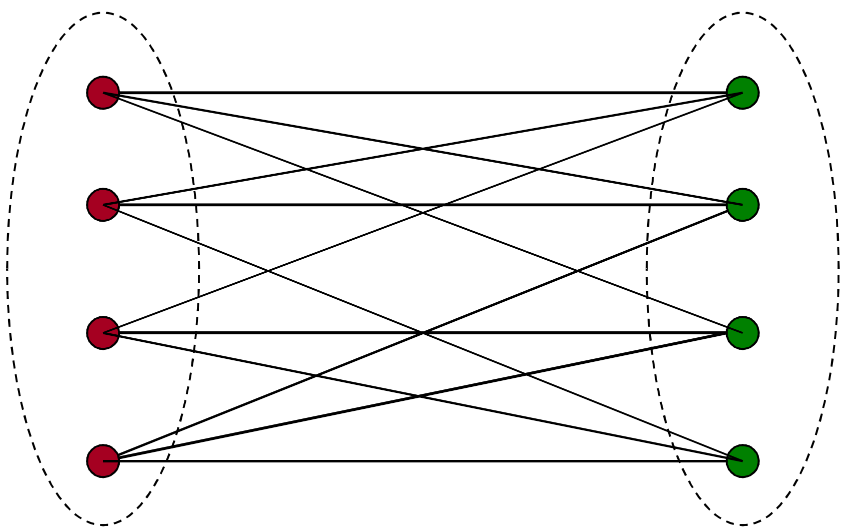
2 colorable: tree, even cycle, etc.

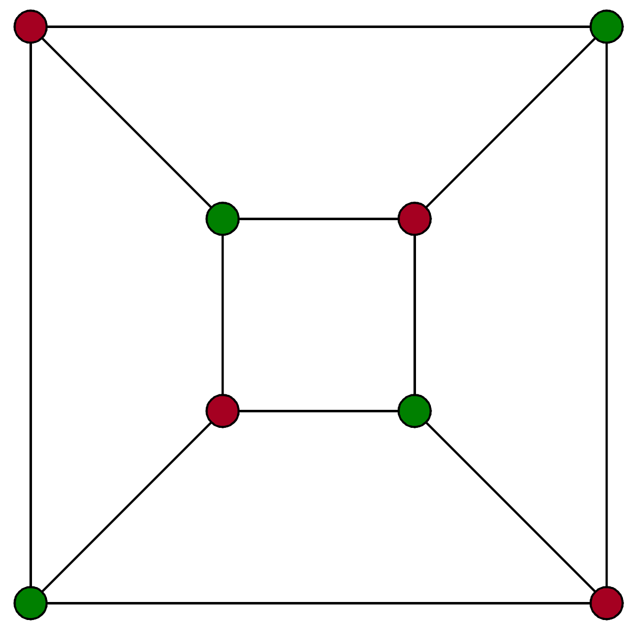
Not 2 colorable: triangle, odd cycle, etc.

Bipartite graphs

When exactly is a graph 2-colorable?

Is a bipartite graph 2-colorable?



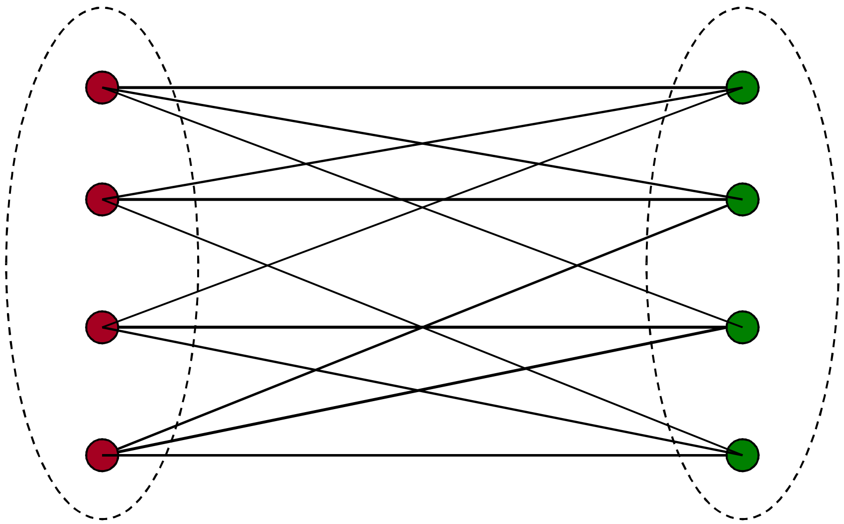


Is a 2-colorable graph bipartite?

Fact. A graph is 2-colorable if and only if it is bipartite.

Bipartite graphs

When exactly is a graph **bipartite**?



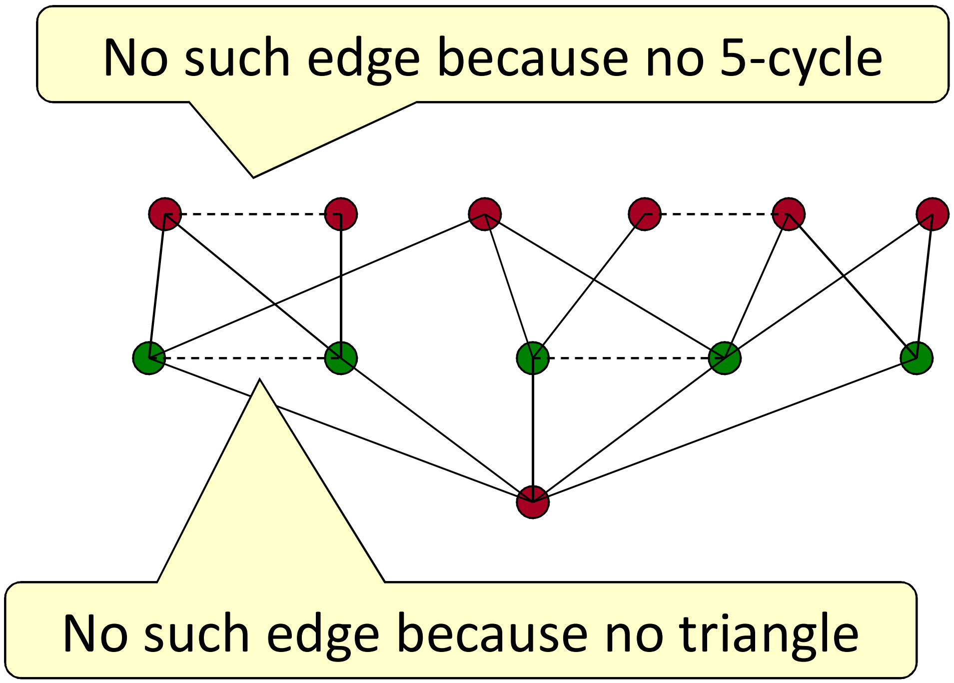
Can a bipartite graph has an odd cycle? NO

If a graph does not have an odd cycle, then it is bipartite?

Theorem. A graph is bipartite if and only if it has no odd cycle.

Bipartite graphs

When exactly is a graph **bipartite**?



1. The idea is like coloring a tree.
2. Pick a vertex v, color it red.
3. Color all its neighbor green.
4. Color all neighbors of green vertices red
5. Repeat until all vertices are colored.

If a graph does not have an odd cycle, then it is bipartite?

Theorem. A graph is bipartite if and only if it has no odd cycle.

Coloring problem: Some examples of Applications

Application 1: Schedule Final Exam

This is a graph coloring problem.

- Each vertex is a course,
- Two courses are connected with an edge if they have a student in common.

Suppose want to schedule some final exams for CS courses with following course numbers:

1007, 3137, 3157, 3203, 3261, 4115, 4118, 4156

Suppose also that there are no students in common taking the following pairs of courses:

1007-3137

1007-3157, 3137-3157

1007-3203

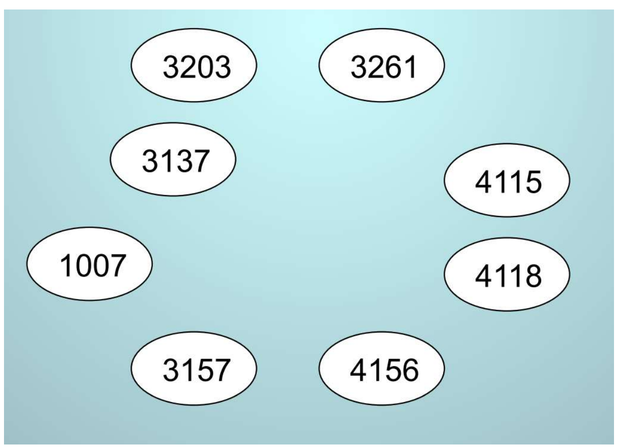
1007-3261, 3137-3261, 3203-3261

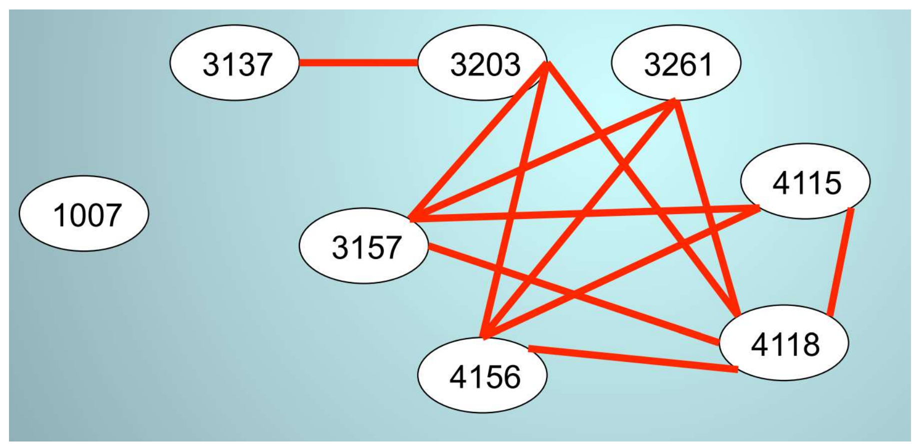
1007-4115, 3137-4115, 3203-4115, 3261-4115

1007-4118, 3137-4118

1007-4156, 3137-4156, 3157-4156

How many exam slots are necessary to schedule exams?





How many colors are necessary ?

Coloring problem: Some examples of Applications

Many problems can be formulated as a graph coloring problem including Time Tabling, Scheduling, Register Allocation, Channel Assignment.

Coloring problem: Some examples of Applications

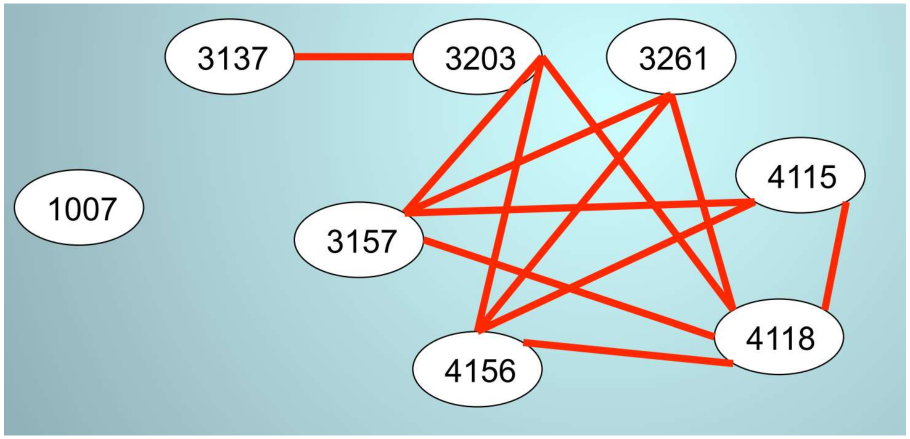
Application 1: Schedule Final Exam

This is a graph coloring problem.

The graph is obviously not 1-colorable because there exist edges

The graph is not 2-colorable because there exist triangles

Is it 3-colorable ? Try to color by **Red**, **Green**, **Blue**

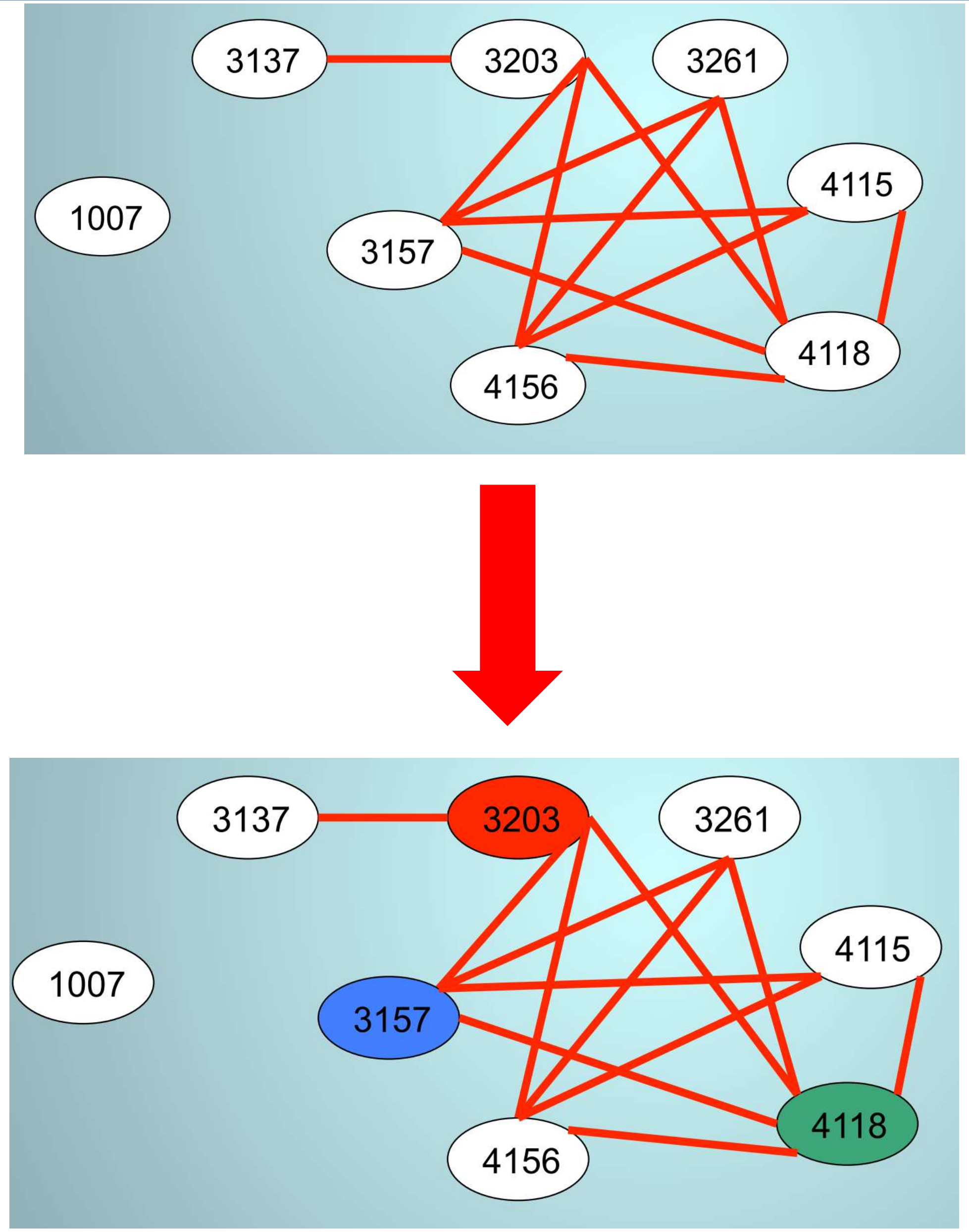


How many colors are necessary ?

Coloring problem: Some examples of Applications

Application 1: Schedule Final Exam

This is a graph coloring problem.

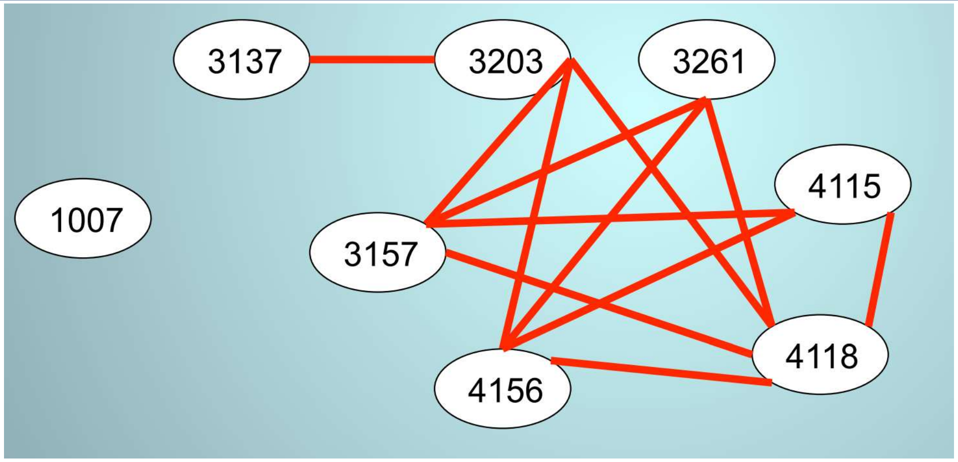


Pick a triangle and color the vertices
3203-Red, **3157-Blue**, **4118-Green**

Coloring problem: Some examples of Applications

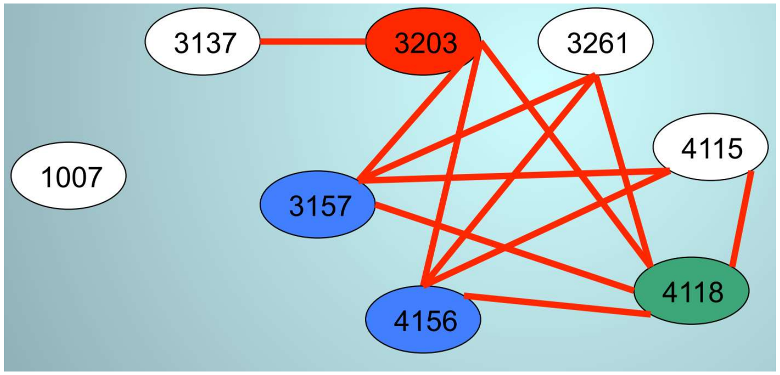
Application 1: Schedule Final Exam

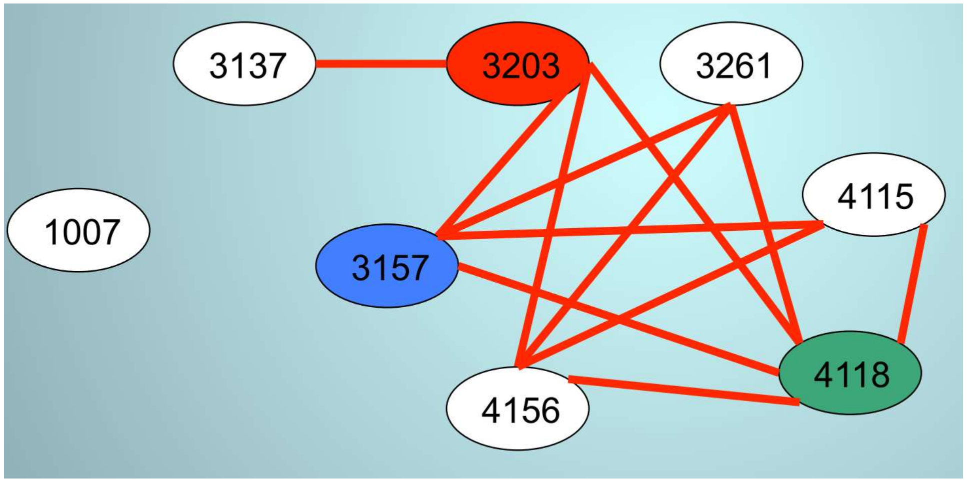
This is a graph coloring problem.



Pick a triangle and color the vertices
3203-Red, **3157-Blue**, **4118-Green**

So 4156 must be **Blue**

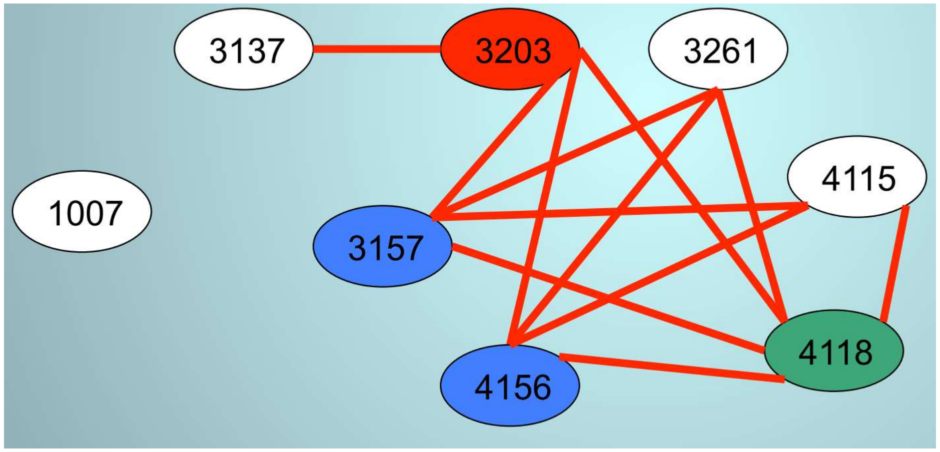


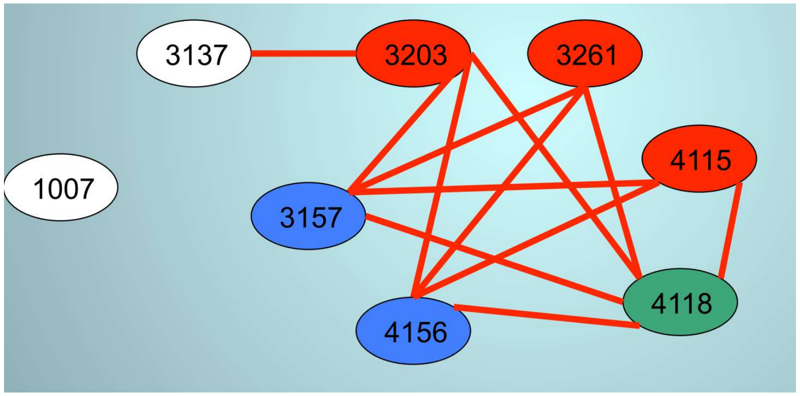


Coloring problem: Some examples of Applications

Application 1: Schedule Final Exam

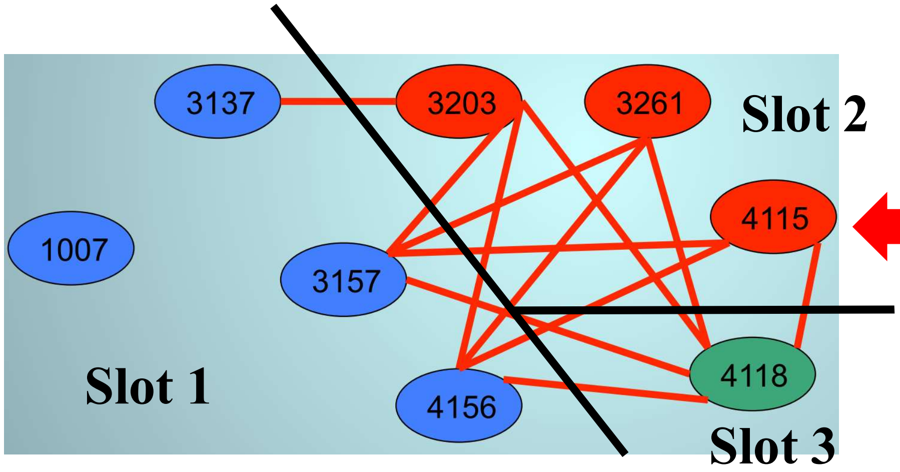
This is a graph coloring problem.





So 3261 and 4115 must be **Red**

3137 and 1007 easy to color – pick **Blue**



Thus, we need 3 time slots

Coloring problem: Some examples of Applications

Application 2: Register allocation

This is a graph coloring problem.

	Inputs:		a, b
Step 1.	c	$=$	$a+b$
2.	d	$=$	$a*c$
3.	e	$=$	$c+3$
4.	f	$=$	$c-e$
5.	g	$=$	$a+f$
6.	h	$=$	$f+1$
	Outputs:		d, g, h

- Given a program, we want to execute it as quick as possible.
- Calculations can be done most quickly if the values are stored in registers.
- But registers are very expensive, and there are only a few in a computer.
- Therefore, we need to use the registers efficiently.

Coloring problem: Some examples of Applications

Application 2: Register allocation

This is a graph coloring problem.

$$
\begin{array}{rrcl}
 & \text{Inputs:} & & a, b \\
\text{Step 1.} & c & = & a+b \\
2. & d & = & a*c \\
3. & e & = & c+3 \\
4. & f & = & c-e \\
5. & g & = & a+f \\
6. & h & = & f+1 \\
 & \text{Outputs:} & & d, g, h
\end{array}
$$

- Each vertex is a variable.
- Two variables are connected with an edge if they cannot be put into the same register.

a and b cannot use the same register, because they store different values.

c and d cannot use the same register otherwise the value of c is overwritten.

Each color corresponds to a register.