Eulerian Cycle (2A)

Young Won Lim
5/25/18

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

Path and Trail

A **path** is a **trail** in which all **vertices** are distinct.
(except possibly the first and last)

A **trail** is a **walk** in which all **edges** are distinct.

	Vertices	Edges	
Walk	may repeat	may repeat	(Closed/Open)
Trail	may repeat	cannot repeat	(Open)
Path	cannot repeat	cannot repeat	(Open)
Circuit	may repeat	cannot repeat	(Closed)
Cycle	cannot repeat	cannot repeat	(Closed)

https://en.wikipedia.org/wiki/Eulerian_path

Simple Paths and Cycles

Most literatures require that all of the **edges** and **vertices** of a **path** be distinct from one another.

But, some do not require this and instead use the term **simple path** to refer to a **path** which contains no repeated **vertices**.

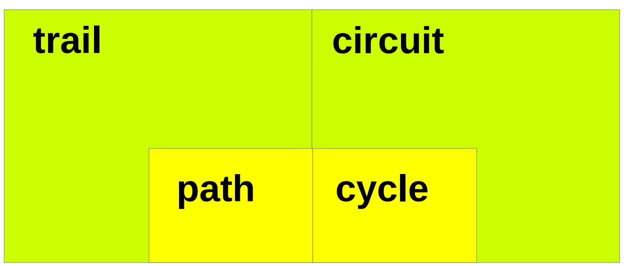
A **simple cycle** may be defined as a **closed walk** with no repetitions of **vertices** and **edges** allowed, other than the repetition of the **starting** and **ending vertex**

There is considerable variation of terminology!!!
Make sure which set of definitions are used...

https://en.wikipedia.org/wiki/Eulerian_path

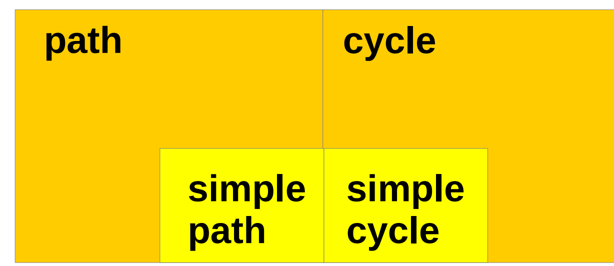
Simple Paths and Cycles

Most literatures



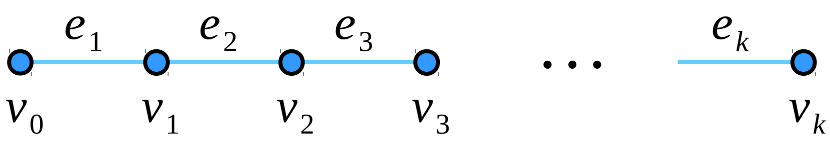
narrow sense path & cycle

some other literatures



wide sense path & cycle

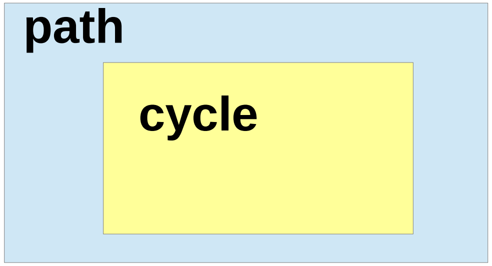
Paths and Cycles



One of a kind

path $v_{0,}\ e_{1,}\ v_{1,}\ e_{2,}\ \cdots,\ e_k,\ v_k$

cycle $v_{0,}\ e_{1,}\ v_{1,}\ e_{2,}\ \cdots,\ e_k,\ v_k \quad (v_0 = v_k)$



path $v_{0,}\ e_{1,}\ v_{1,}\ e_{2,}\ \cdots,\ e_k,\ v_k \quad (v_0 \neq v_k)$

cycle $v_{0,}\ e_{1,}\ v_{1,}\ e_{2,}\ \cdots,\ e_k,\ v_k \quad (v_0 = v_k)$

Two different kinds

Euler Cycle

Some people reserve the terms **path** and **cycle** to mean non-self-intersecting path and cycle. — no repeating vertices

A (potentially) self-intersecting path is known as a **trail** or an **open walk**; — repeating vertices

and a (potentially) self-intersecting cycle, a **circuit** or a **closed walk**. — repeating vertices

This ambiguity can be avoided by using the terms **Eulerian trail** and **Eulerian circuit** when self-intersection is allowed — repeating vertices

https://en.wikipedia.org/wiki/Eulerian_path

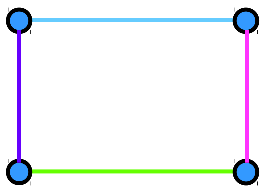
Euler Cycle

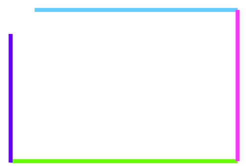
visits every **edge** exactly once

the existence of **Eulerian cycles**

all **vertices** in the graph have an **even** degree

connected graphs with **all vertices** of **even** degree h
ave an **Eulerian cycles**





non-repeating edges
repeatable vertices
} **circuit**

Eulerian circuit : more suitable terminology

https://en.wikipedia.org/wiki/Eulerian_path

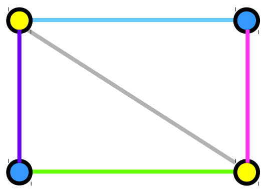
Euler Path

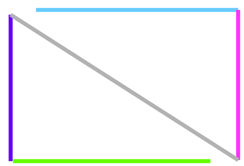
visits <u>every</u> <u>**edge**</u> exactly <u>once</u>

the existence of **Eulerian paths**

all the **vertices** in the graph have an **even** degree

<u>except</u> only **two** vertices with an **odd** degree





An **Eulerian path** starts and ends at <u>different</u> vertices
An **Eulerian cycle** starts and ends at the <u>same</u> vertex.

non-repeating edges
repeatable vertices } **trail**

Eulerian trail : more suitable terminology

https://en.wikipedia.org/wiki/Eulerian_path

Conditions for Eulerian Cycles and Paths

An odd vertex = a vertex with an odd degree
An even vertex = a vertex with an even degree

# of **odd** vertices	Eulerian **Path**	Eulerian **Cycle**
0	No	**Yes**
2	**Yes**	No
4,6,8, …	No	No
1,3,5,7, …	No such graph	No such graph

If the graph is connected

http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

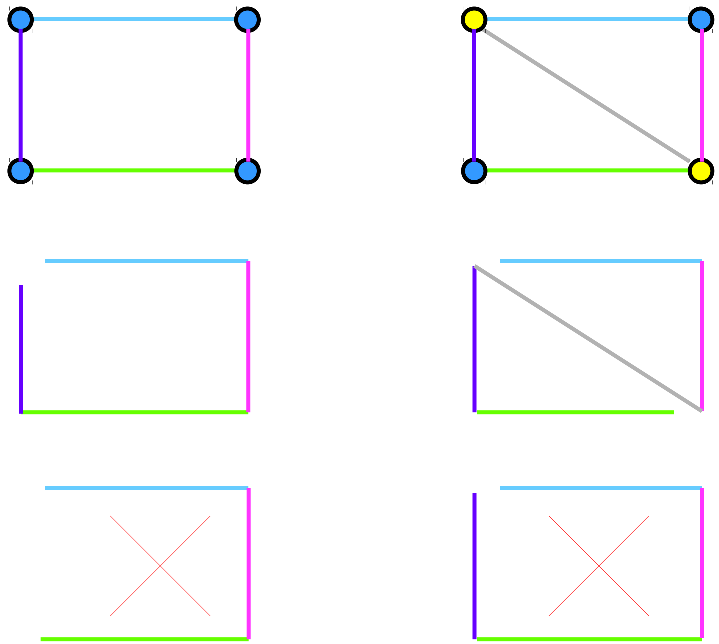
The number of odd vertices

# of **odd** vertices	Eulerian **Path**	Eulerian **Cycle**
0	**No**	**Yes**
2	**Yes**	**No**

of **odd** vertices
= 0

Eulerian Cycle

No Eulerian Path



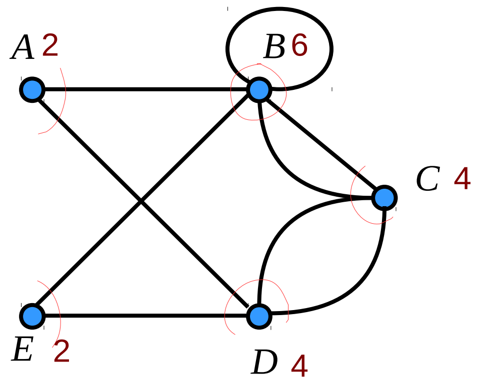
of **odd** vertices
= 2

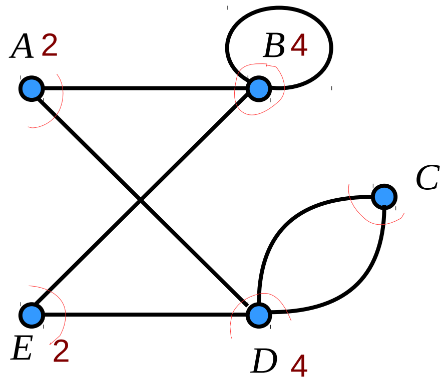
Eulerian Path

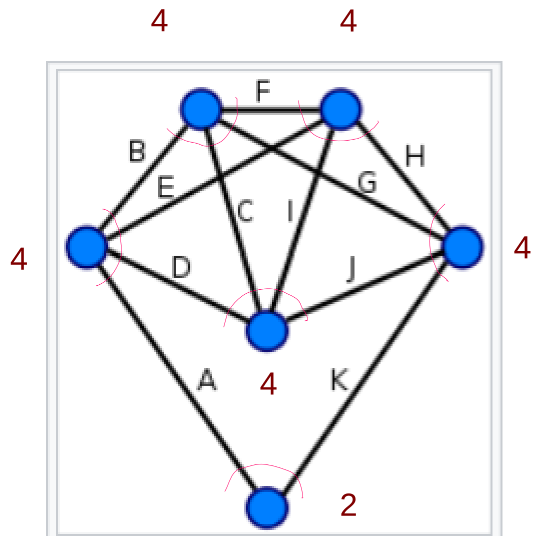
No Eulerian Cycle

Eulerian graph :
a graph with an **Eulerian cycle**
a graph with **every vertex** of **even degree**
(the number of **odd vertices** is 0)

These definitions coincide for connected graphs.

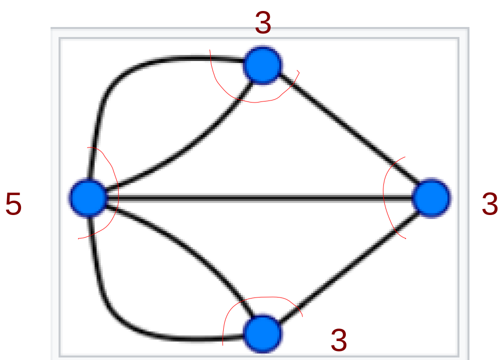






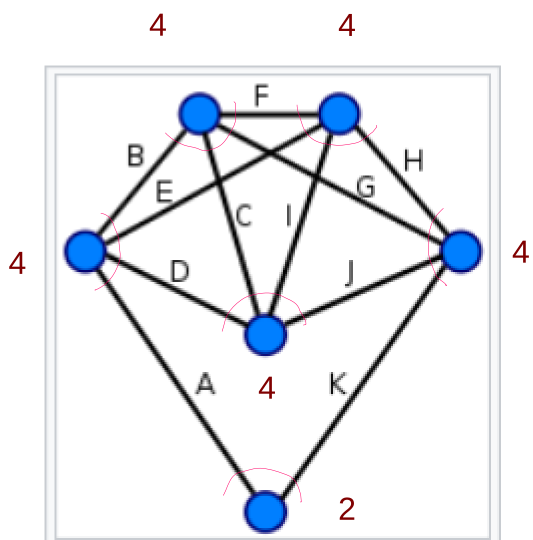
Every vertex of this graph has an even degree. Therefore, this is an Eulerian graph. Following the edges in alphabetical order gives an Eulerian circuit/cycle.

Odd Degree and Even Degree



The Königsberg Bridges multigraph. This multigraph is not Eulerian, therefore, a solution does not exist.

All odd degree vertices

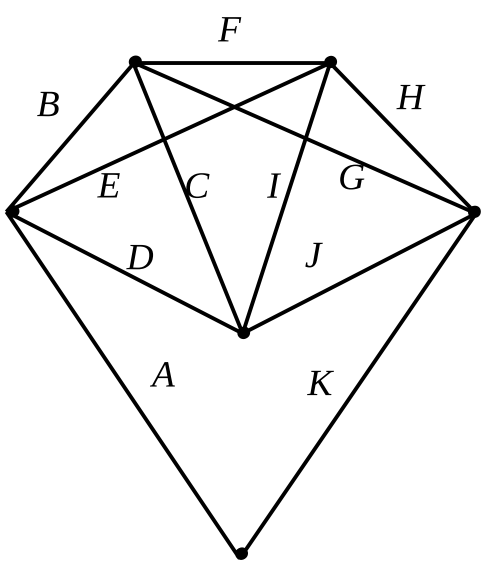


Every vertex of this graph has an even degree. Therefore, this is an Eulerian graph. Following the edges in alphabetical order gives an Eulerian circuit/cycle.

All even degree vertices

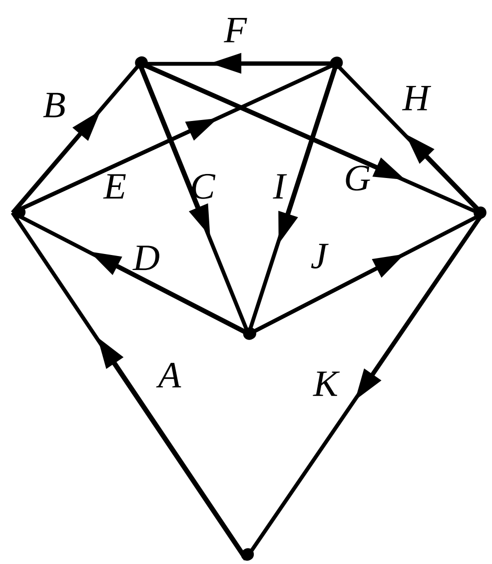
https://en.wikipedia.org/wiki/Eulerian_path

Euler Cycle Example



ABCDEFGHIJK

a path denoted by
the edge names



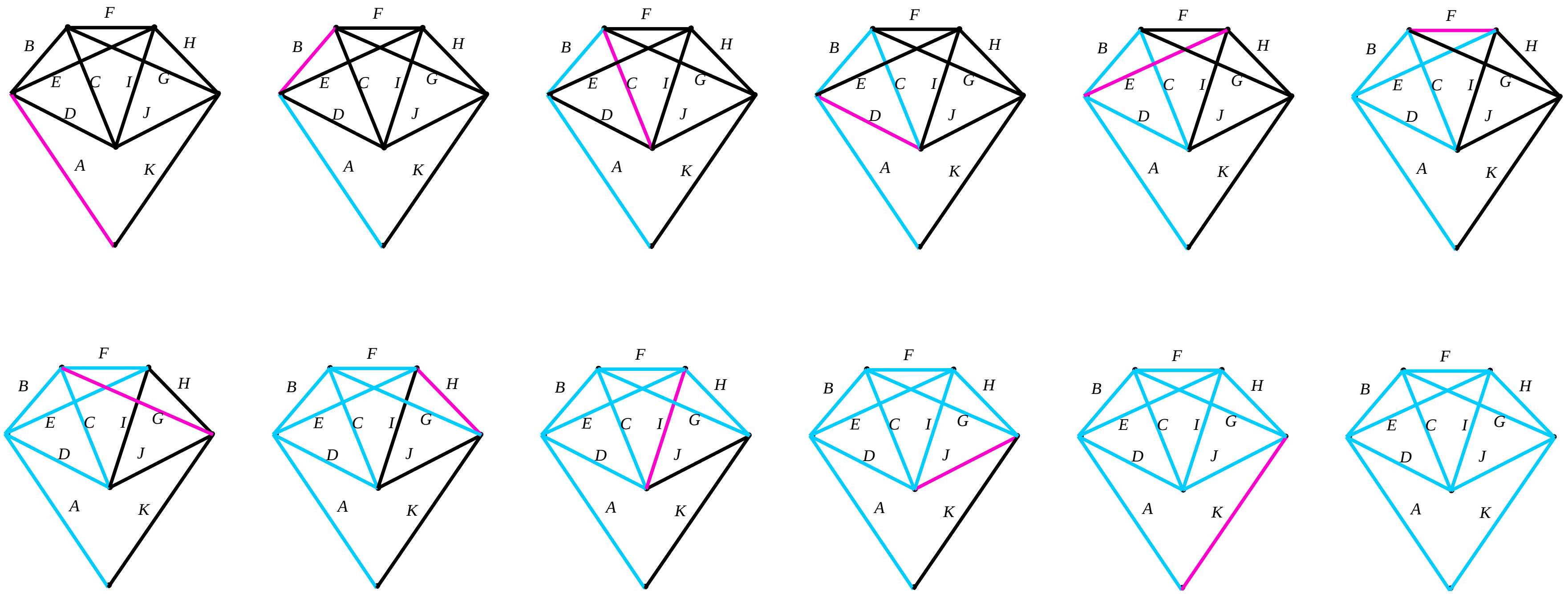
All <u>even</u> degree vertices

Eulerian Cycles

en.wikipedia.org

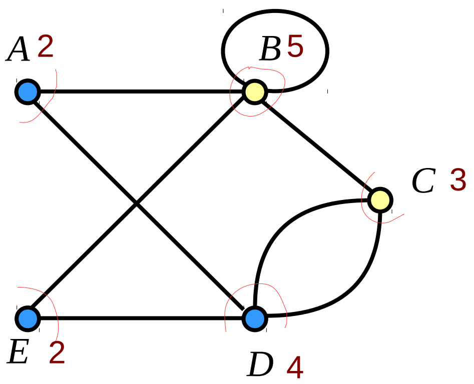
Euler Cycle Example

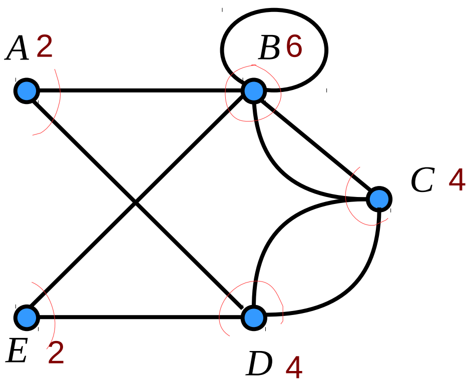
ABCDEFGHIJK

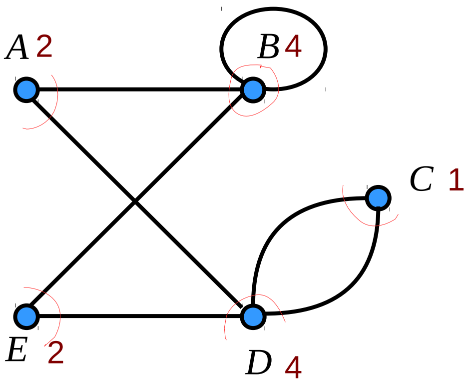


en.wikipedia.org

Euler Path and Cycle Examples







Eulerian Path
1. BBADCDEBC
2. CDCBBADEB

Euerian Cycle
1. CDCBBADEBC

Euerian Cycle
2. CDEBBADC

a path denoted by
the vertex names

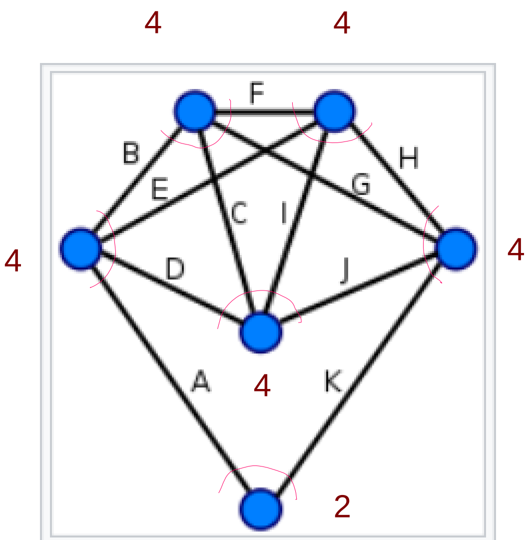
http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Eulerian Cycles of Undirected Graphs

An **undirected** graph has an **Eulerian** **cycle**
if and only if every **vertex** has **even degree**,
and all of its **vertices** with **nonzero degree**
belong to a **single** **connected** **component**.

An **undirected** graph can be
decomposed into **edge-disjoint cycles**
if and only if all of its **vertices** have **even degree**.

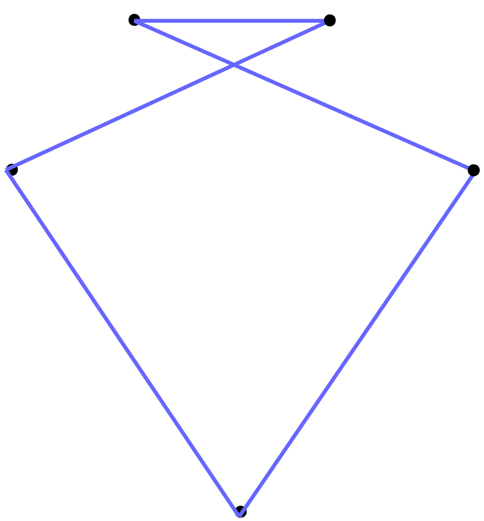
So, a graph has an **Eulerian** **cycle**
if and only if it can be decomposed
into **edge-disjoint cycles**
and its **nonzero-degree** vertices
belong to a **single connected component**.

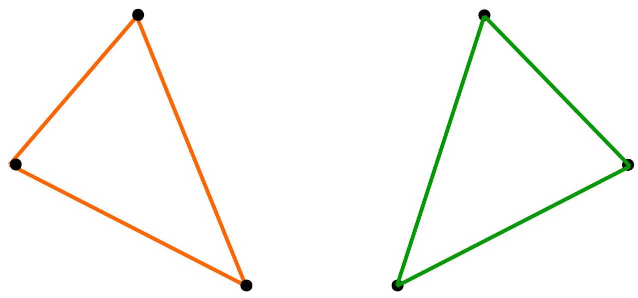


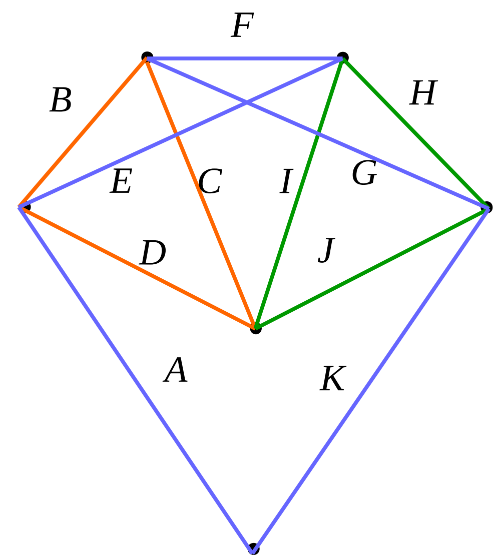
Every vertex of this graph has an even degree. Therefore, this is an Eulerian graph. Following the edges in alphabetical order gives an Eulerian circuit/cycle.

https://en.wikipedia.org/wiki/Eulerian_path

Edge Disjoint Cycle Decomposition







All even vertices ⇔ **Euerian Cycle** ⇔ **Edge Disjoint Cycles**

Eulerian Paths of Undirected Graphs

An undirected graph has an **Eulerian** **trail**
if and only if exactly **zero** or **two vertices** have **odd degree**,
and all of its vertices with **nonzero degree**
belong to a **single connected component**.

Here, the following definitions are used.

Trail : A **walk** without repeated edges. (closed or open)

This definition includes **trail** (open **walk**) and **circuit** (closed **walk**)
All of which contain no repeating edges.

https://en.wikipedia.org/wiki/Eulerian_path

Eulerian Cycles of DiGraphs

A **directed** graph has an **Eulerian** **cycle**
if and only if every vertex has
equal **in degree** and **out degree**,
and all of its vertices with nonzero degree
belong to a **single strongly connected component**.

Equivalently, a **directed** graph has an Eulerian cycle
if and only if it can be decomposed
into **edge-disjoint directed cycles**
and all of its vertices with nonzero degree
belong to a **single strongly connected component**.

https://en.wikipedia.org/wiki/Eulerian_path

Eulerian Paths of DiGraphs

A directed graph has an **Eulerian path**

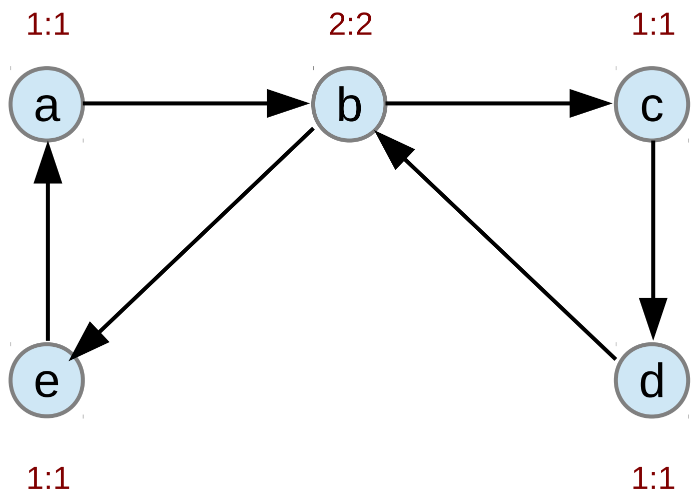
if and only if **at most one** vertex has
(out-degree) − (in-degree) = 1,
at most one vertex has
(in-degree) − (out-degree) = 1,

every other vertex has
equal in-degree and out-degree,

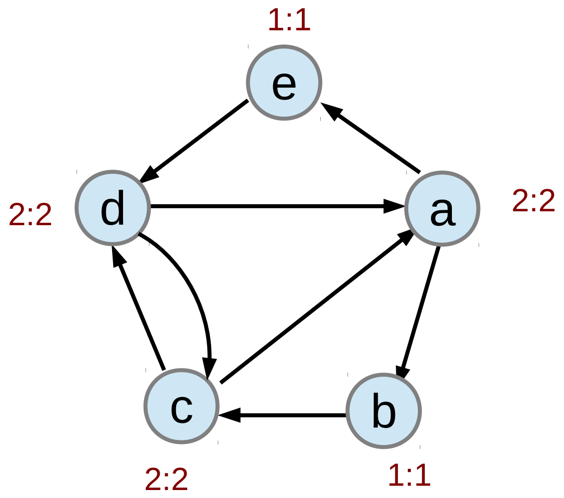
and all of its vertices with nonzero degree
belong to a single connected component
of the underlying undirected graph.

https://en.wikipedia.org/wiki/Eulerian_path

DiGraph Eulerian Cycle Examples



abcdbea

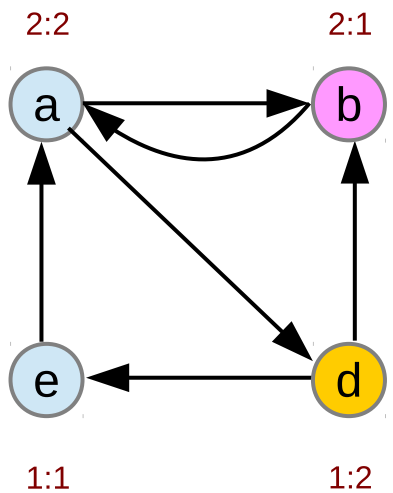


edabcdcae

https://www.geeksforgeeks.org/euler-circuit-directed-graph/

https://math.stackexchange.com/questions/1871065/euler-path-for-directed-graph

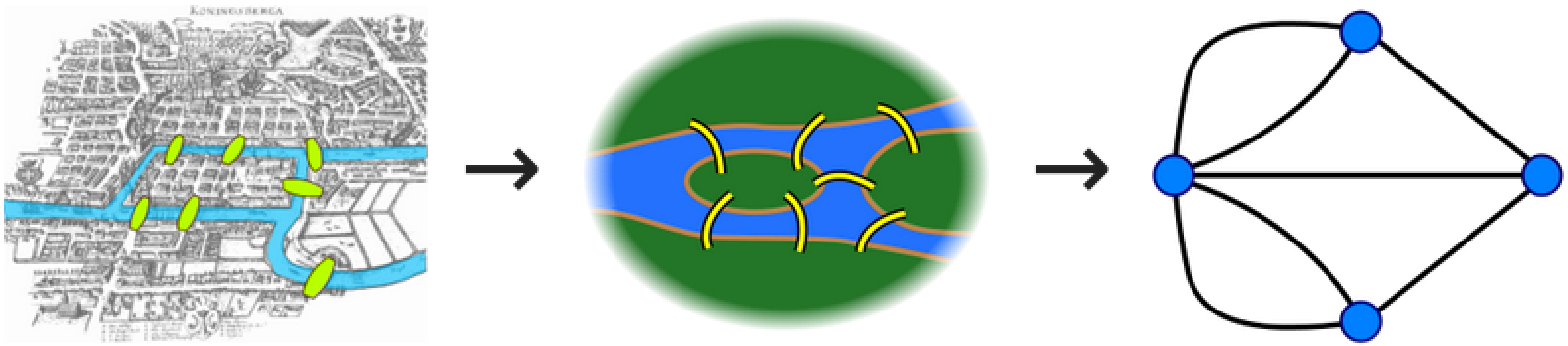
DiGraph Eulerian Path Examples



dbadeab

https://www.boost.org/doc/libs/1_58_0/libs/graph/doc/graph_theory_review.html

Seven Bridges of Königsberg

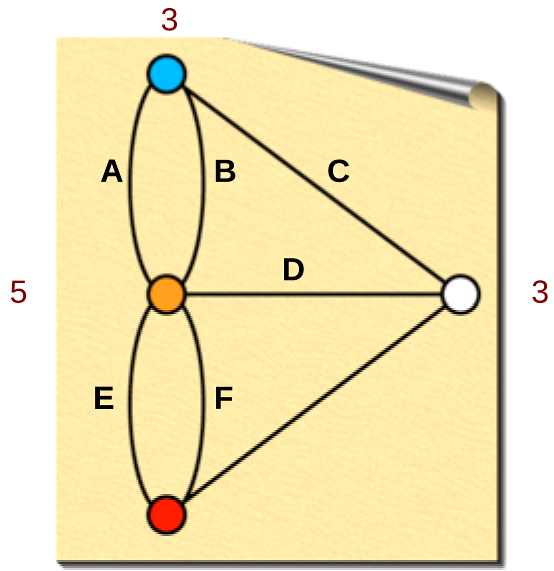


The problem was to devise a walk through the city that
would cross each of those bridges once and only once.

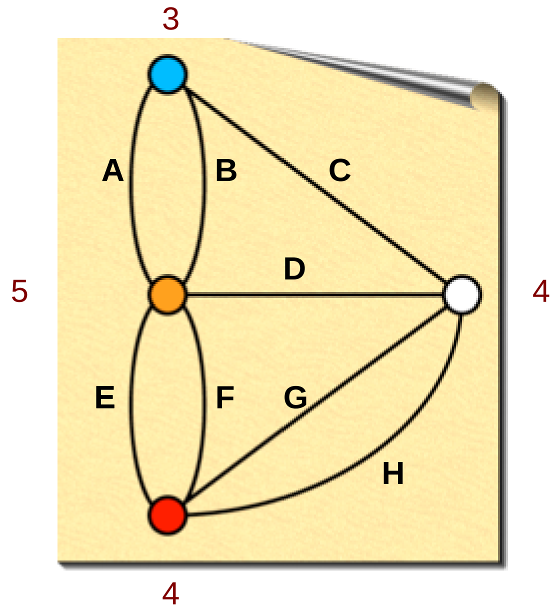
https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Seven and Eight Bridges Problems

7 bridges problem



8 bridges problem



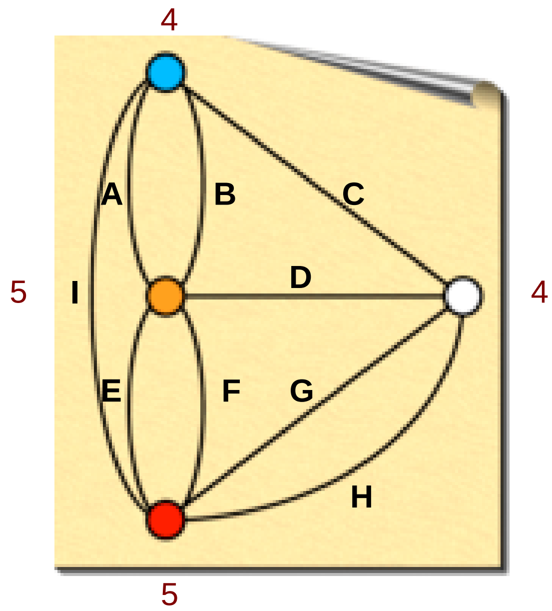
Eulerian Path

● AEHGFDCB ●

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Nine and Ten Bridges Problems

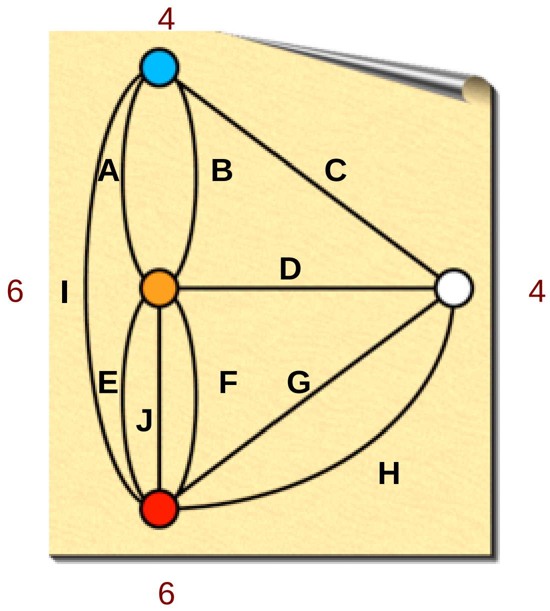
9 bridges problem



Eulerian Path

EHGFDCBAI

10 bridges problem

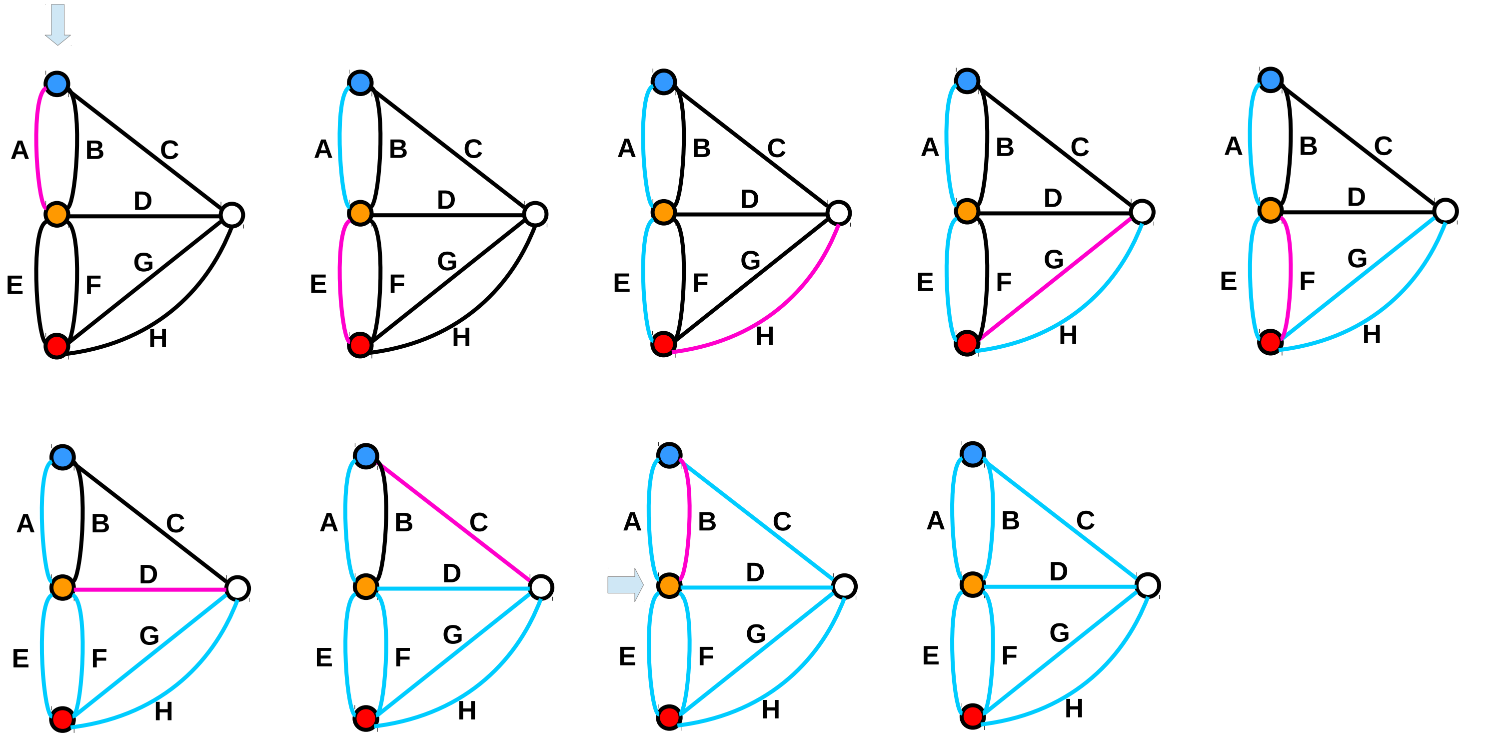


Eulerian Cycle

AEHGFDCBJI

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

8 bridges – Eulerian Path

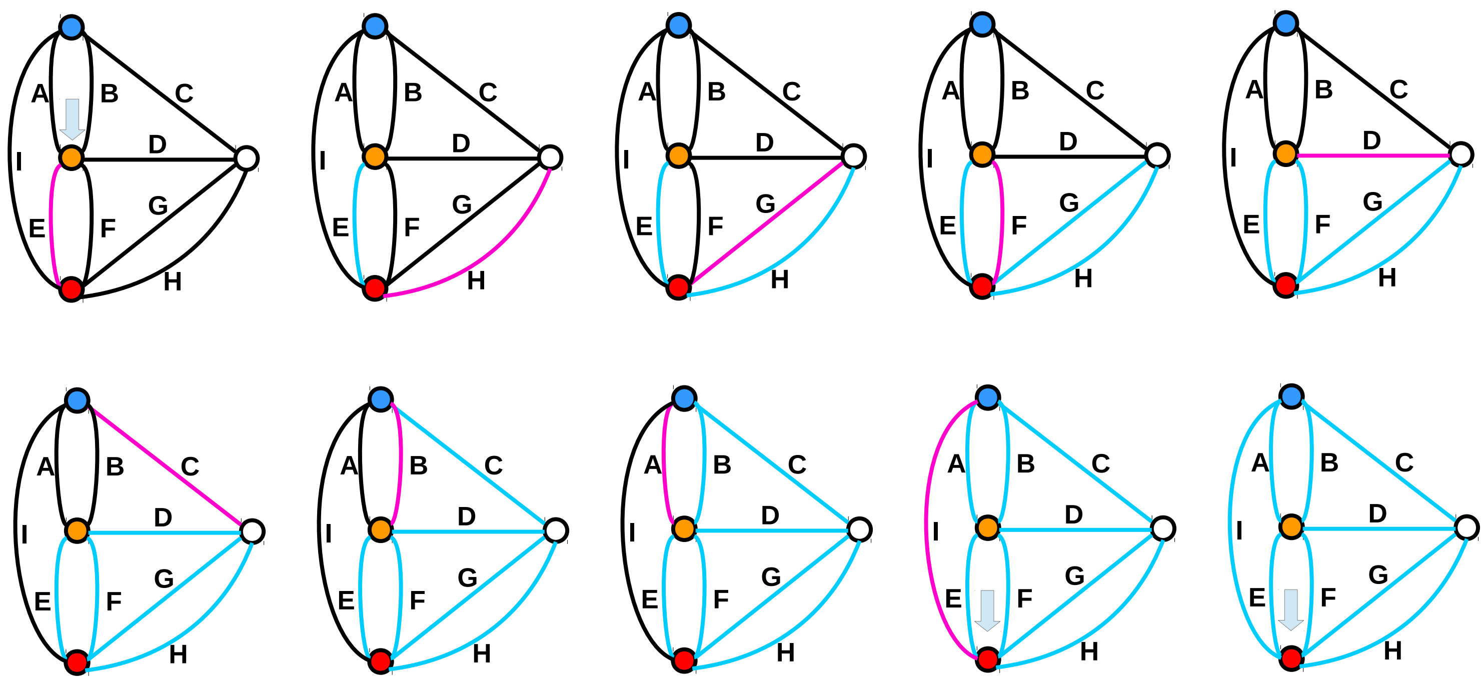


Eulerian Path

AEHGFDCB

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

9 bridges – Eulerian Path

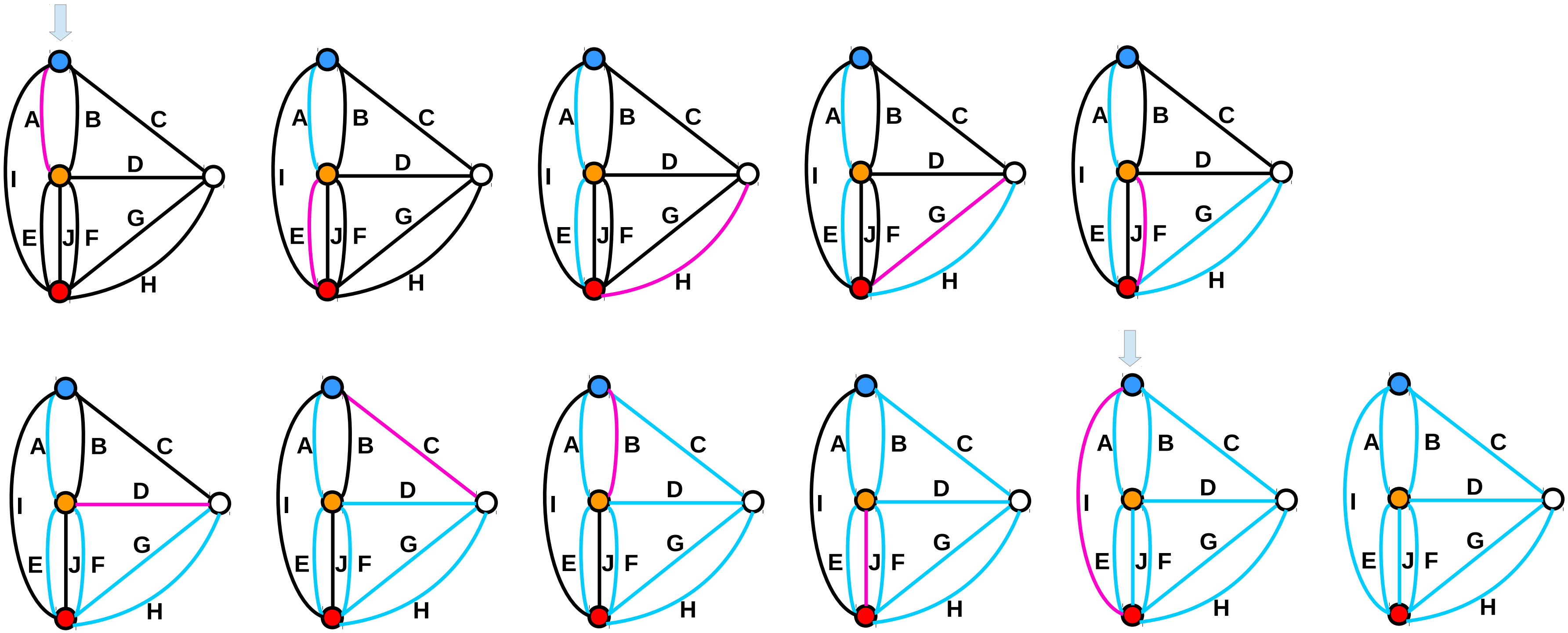


Eulerian Path

EHGFDCBAI

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

10 bridges – Eulerian Cycle



Eulerian Cycle

AEHGFDCBJI

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Fleury's Algorithm

To find an Eulerian path or an Eulerian cycle:

1. make sure the graph has either **0** or **2 odd** vertices

2. if there are **0 odd** vertex, start <u>anywhere</u>**.**
   If there are **2 odd** vertices, start at one of the <u>two</u> <u>vertices</u>

3. follow edges one at a time.
   If you have a choice between a **bridge** and a **non-bridge**,
   Always <u>choose</u> the **non-bridge**

4. stop when you run out of edge

http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Bridges

A bridge edge

Removing a single edge from a connected graph
can make it disconnected

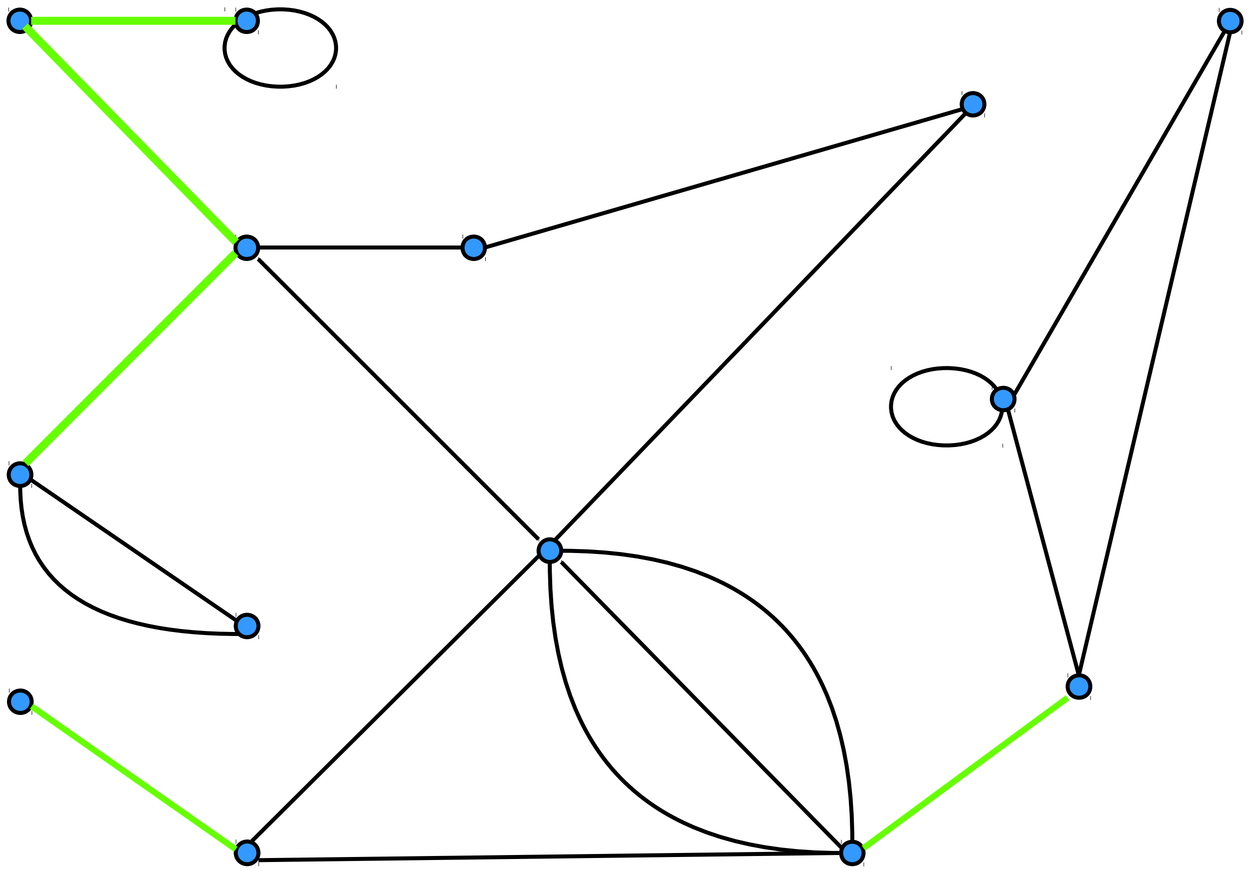
Non-bridge edges

Loops cannot be bridges

Multiple edges cannot be bridges

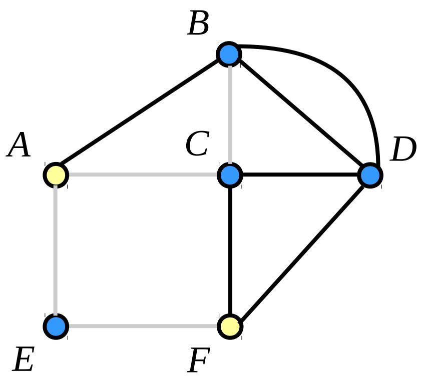
http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Bridge examples in a graph

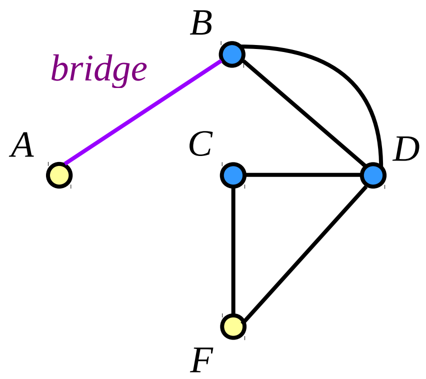


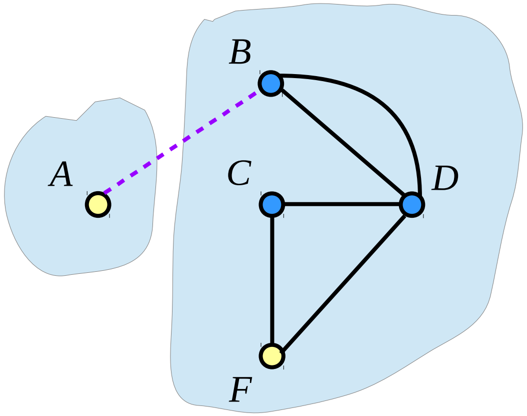
http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Bridges must be avoided, if possible

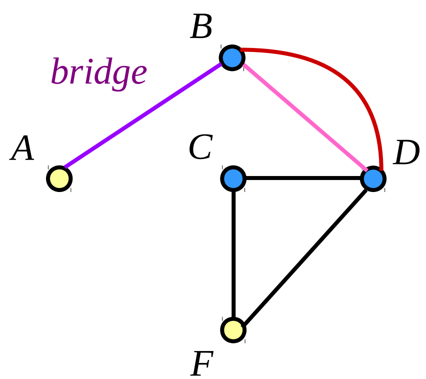


FEACB



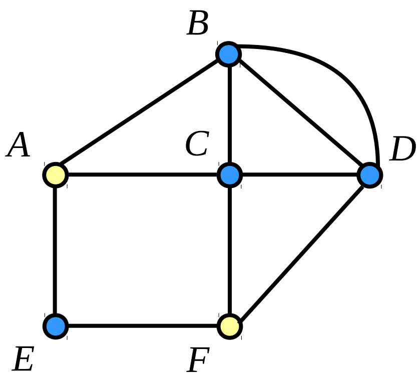


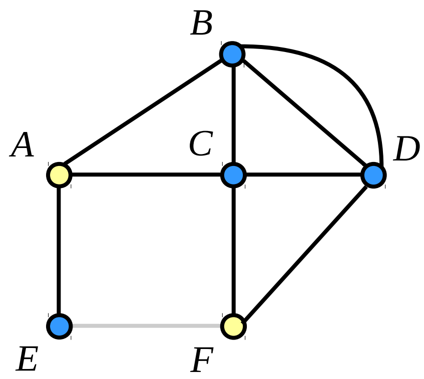
If there exists other choice other than a bridge
The bridge must <u>not</u> be chosen.



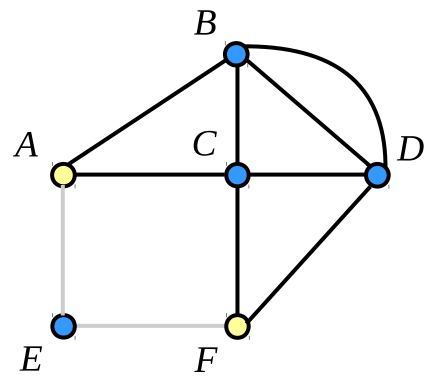
http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Fleury's Algorithm (1)

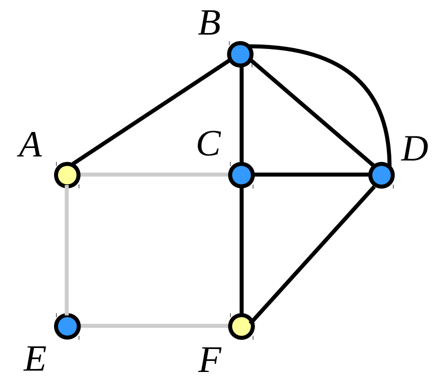




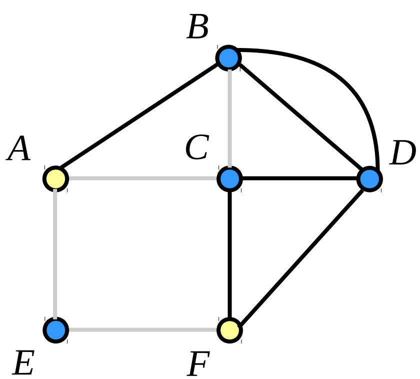
FE



FEA



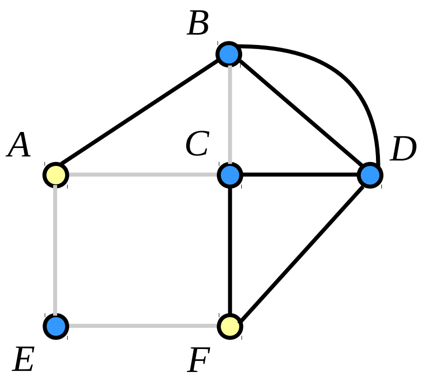
FEAC



FEACB

http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

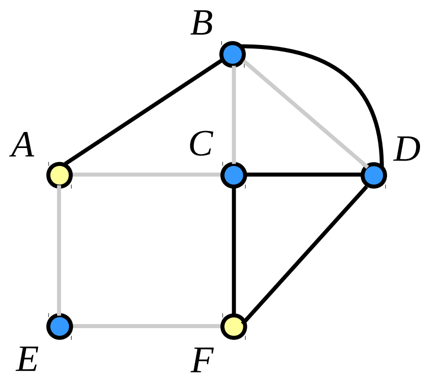
Fleury’s Algorithm (2)



FEACB

BA : *bridge*

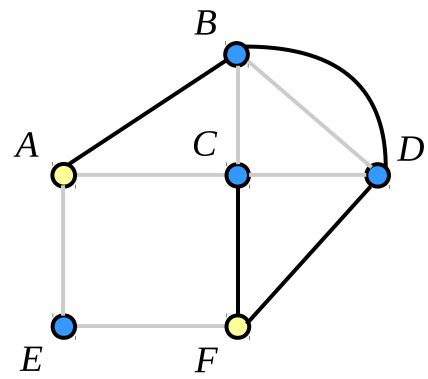
BD : *chosen*



FEACBD

DB : *bridge*

DC : *chosen*

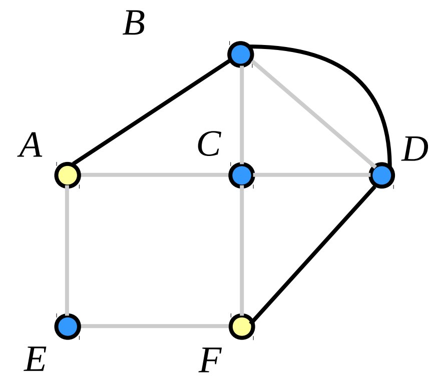


FEACBDC

CF : *bridge*

CF : *chosen*

no other choice



FEACBDCF

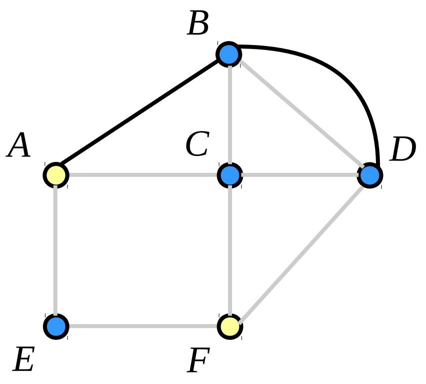
FD : *bridge*

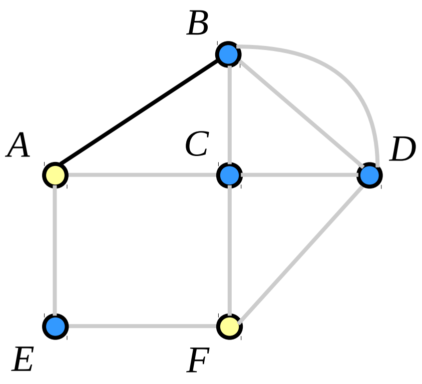
FD : *chosen*

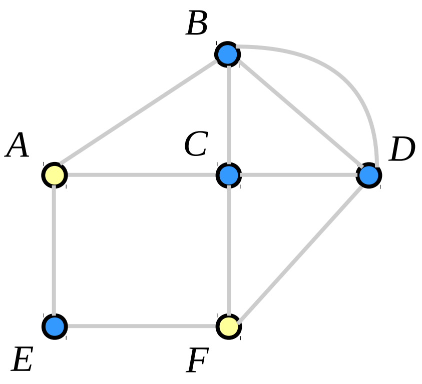
no other choice

http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

Fleury's Algorithm (3)







FEACBDCFD

DB: *bridge*

DB: *chosen*

no other choice

FEACBDCFDB

BA: *bridge*

BA: *chosen*

no other choice

http://people.ku.edu/~jlmartin/courses/math105-F11/Lectures/chapter5-part2.pdf

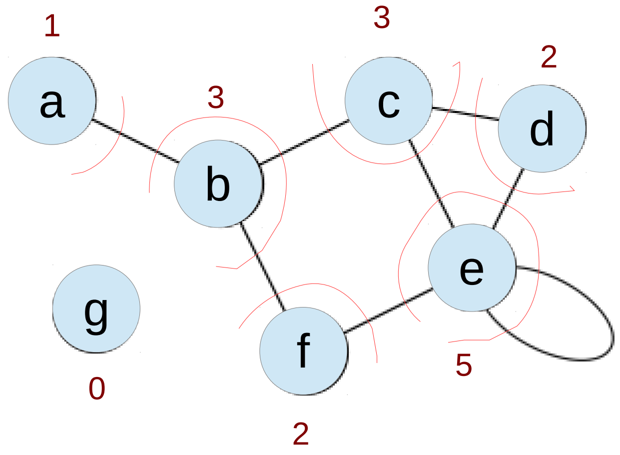
Degree of a vertex

the **degree** (or **valency**) of a vertex is
the number of edges incident to the vertex,
with loops counted twice.

The degree of a vertex v is denoted deg(v)
the maximum degree of a graph G, denoted by Δ(G)
the minimum degree of a graph, denoted by δ(G)

Δ(G) = 5
δ(G) = 0

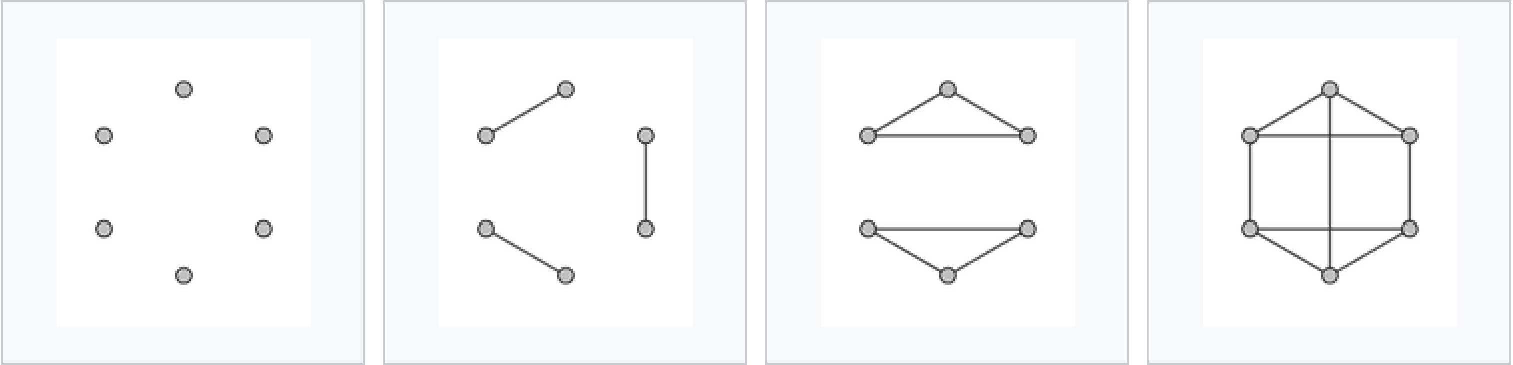
In a **regular** graph, all degrees are the same



https://en.wikipedia.org/wiki/Degree_(graph_theory)

Regular Graphs

a **regular graph** is a graph where each vertex has the same number of neighbors; i.e. every vertex has the same degree or valency.



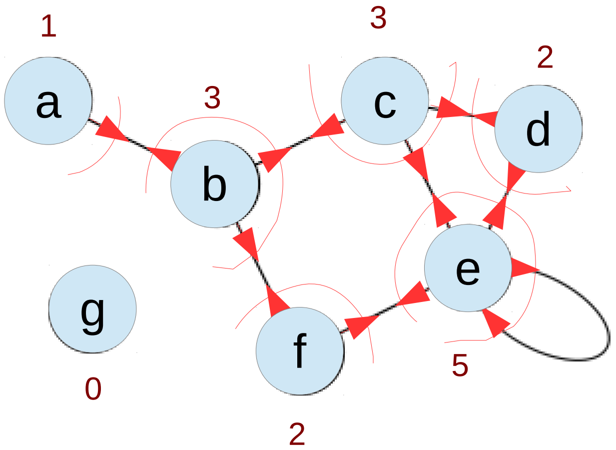
0-regular graph | 1-regular graph | 2-regular graph | 3-regular graph

https://en.wikipedia.org/wiki/Regular_graph

Handshake Lemma

The degree sum formula states that,
given a graph G = (V , E)

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



This statement (as well as the degree sum formula) is known as the **handshaking lemma**.

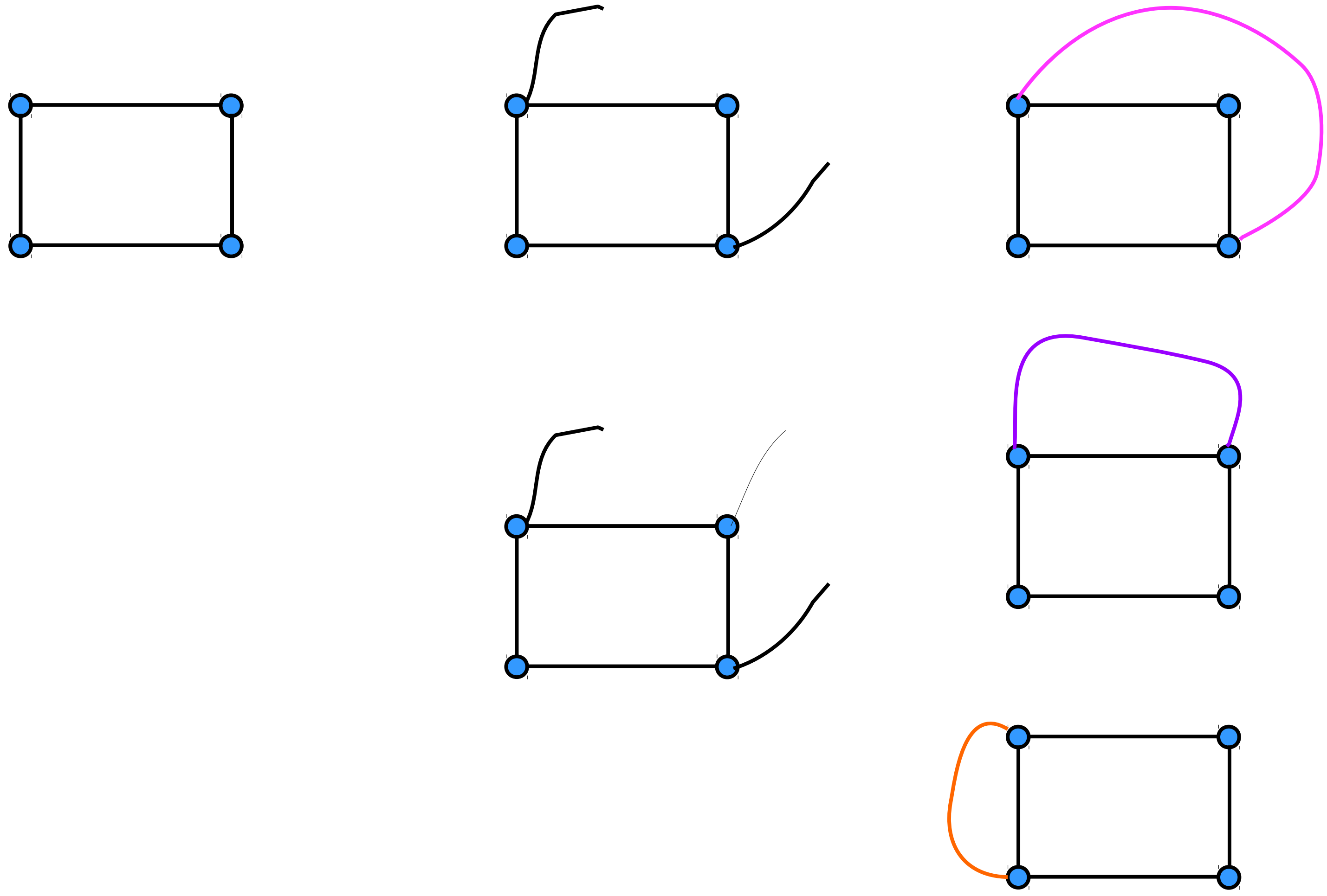
$$
\begin{aligned}
deg(a) &= 1 \\
deg(b) &= 3 \\
deg(c) &= 3 \\
deg(d) &= 2 \\
deg(e) &= 5 \\
deg(f) &= 2 \\
deg(g) &= 0
\end{aligned}
\qquad |E| = 8
$$

$$16 \qquad 2|E| = 16$$

https://en.wikipedia.org/wiki/Degree_(graph_theory)

Young Won Lim
5/25/18

Adding odd vertex



https://en.wikipedia.org/wiki/Eulerian_path

The number of odd vertices

Even vertices : $\{x_1, x_2, \cdots, x_m\}$

$$S = deg(x_1) + deg(x_2) + \cdots + deg(x_m)$$

$$deg(x_i) : even$$

$$S = even + even + \cdots + even$$

Odd vertices : $\{y_1, y_2, \cdots, y_n\}$

$$T = deg(y_1) + deg(y_2) + \cdots + deg(y_n)$$

$$deg(y_i) : odd$$

$$T = odd + odd + \cdots + odd$$

$S : even$

$S+T : even$

$$T : even = \sum n \text{ odd numbers}$$

$$n : even$$

in any graph, the number of vertices with odd degree is even.

# of **odd** vertices	Eulerian **Path**	Eulerian **Cycle**
0	No	**Yes**
2	**Yes**	No
4,6,8, …	No	No
1,3,5,7, …	No such graph	No such graph

Euler Cycle

Any connected graph
with even degree vertices → An Euler cycle

A proof by **induction** on the number of edges in G

A connected graph G with even degree vertices only and *k* edges ($k < n$)	→	An Euler cycle	Assume this is true
A connected graph G with even degree vertices only and *n* edges	→	An Euler cycle	Then this holds true

Johnsonbough, Discrete Mathematics

Euler Cycle

Any connected graph with even degree vertices which has n edge → An Euler cycle

Any connected graph with even degree vertices which has n-1 edge → An Euler cycle

Any connected graph with even degree vertices which has n-2 edge → An Euler cycle

● ● ● ● ● ●

Any connected graph with even degree vertices which has 2 edge → An Euler cycle

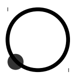
Any connected graph with even degree vertices which has 1 edge → An Euler cycle

Johnsonbough, Discrete Mathematics

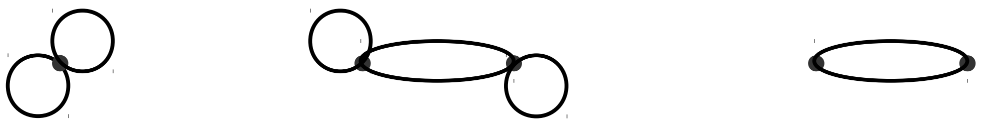
Euler Cycle – Base Cases

n = 0 edge

n = 1 edge



n = 2 edge



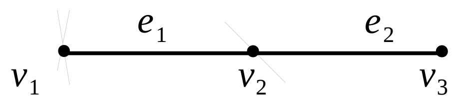
all even degree vertices

→ an Euler cycle

Johnsonbough, Discrete Mathematics

Euler Cycle – decrease the number of edges by one

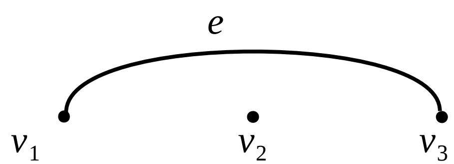
A connected graph G
with even degree vertices only
and *n* edges ($k < n$)



all even degree vertices

P: a path from v to v_1

A connected graph G'
with even degree vertices only
and *n-1* edges ($k < n$)



all even degree vertices

P': a portion of the path P
that are in G'

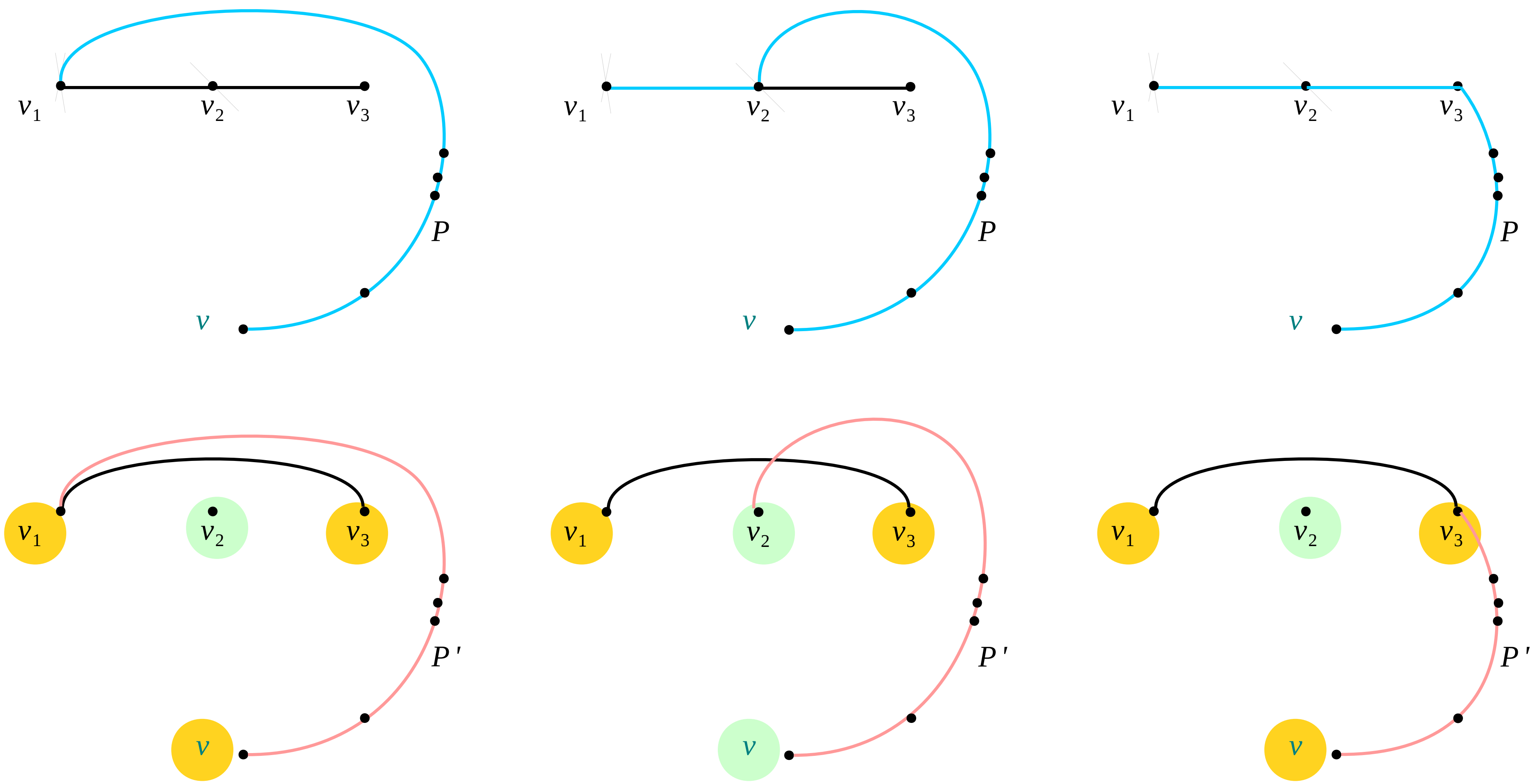
Johnsonbough, Discrete Mathematics

Euler Cycle – a path from v to v_1

Case 1: P ends at v_1

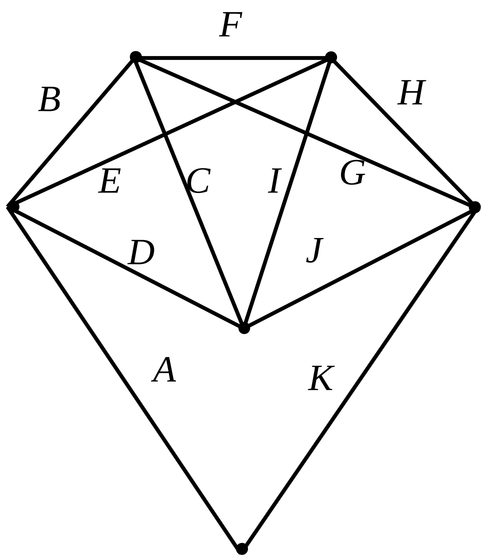
Case 2: P ends at v_2

Case 3: P ends at v_3



Johnsonbough, Discrete Mathematics

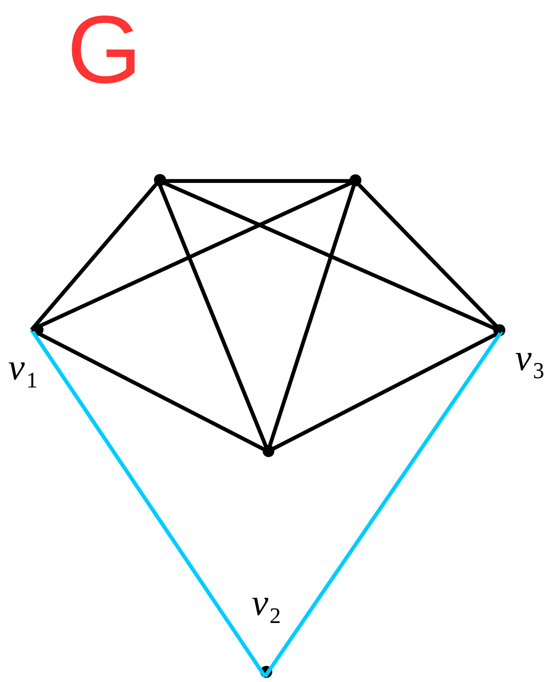
Euler Cycle

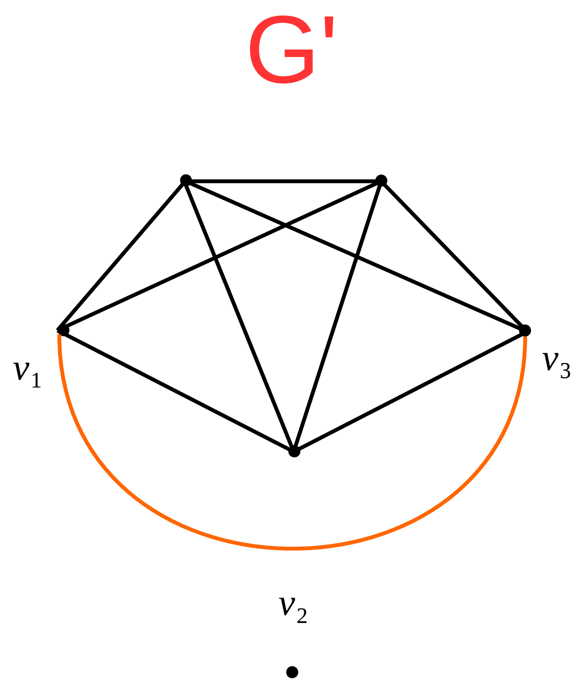


ABCDEFGHIJK

en.wikipedia.org

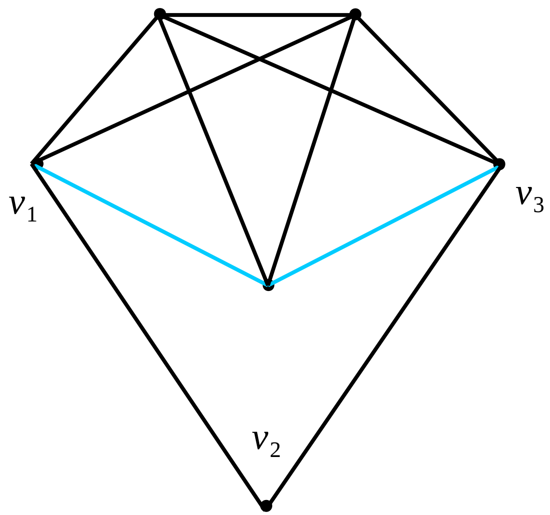
Euler Cycle



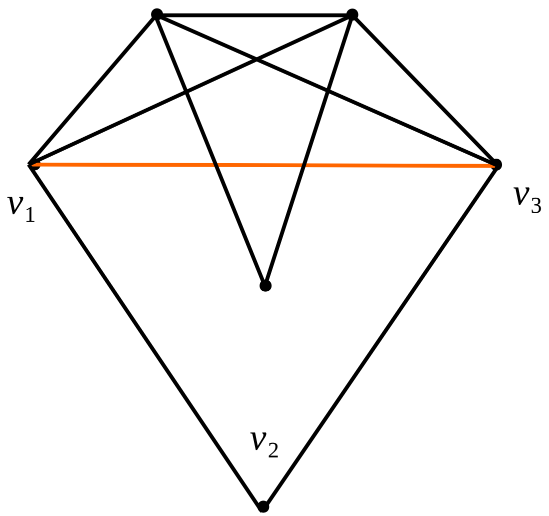


2 components

G

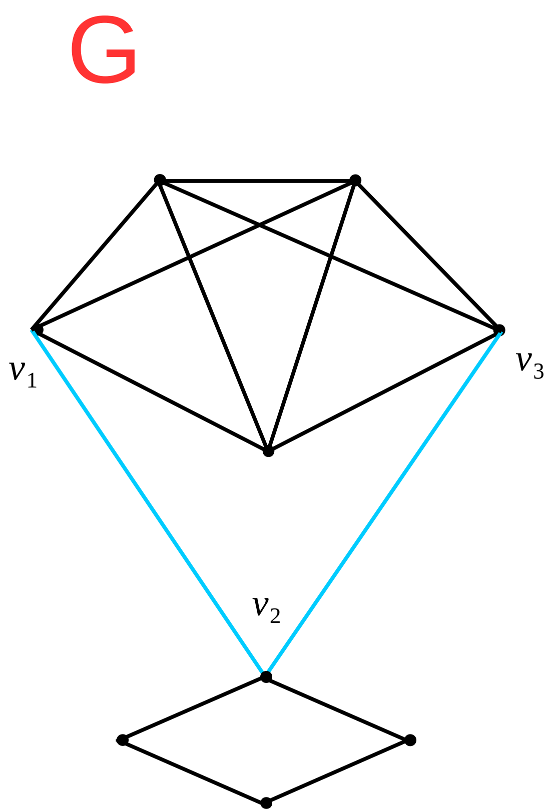


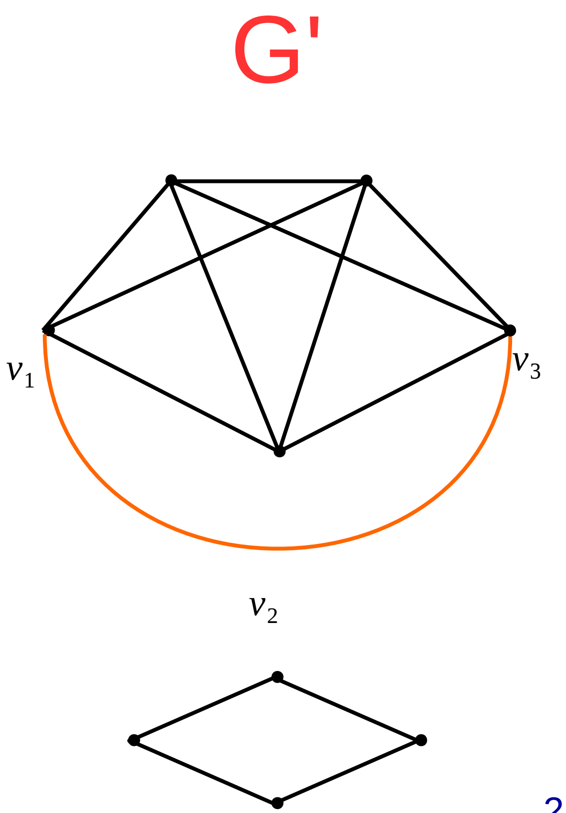
G'



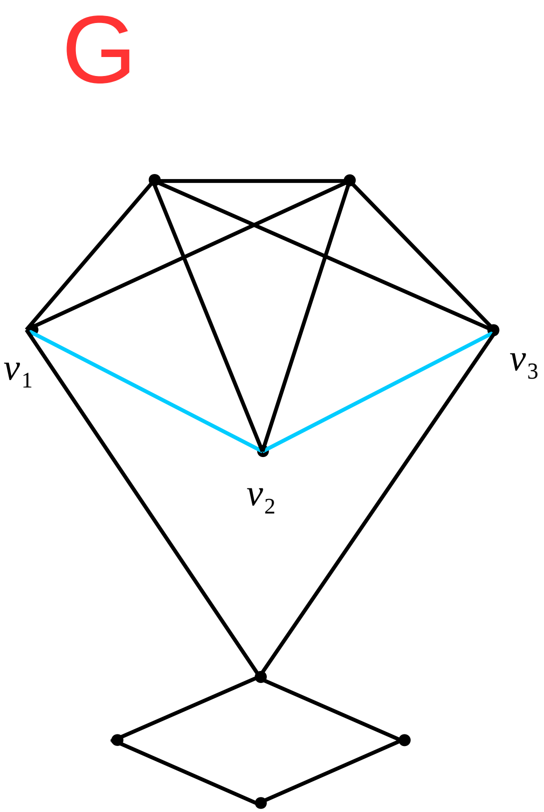
1 component

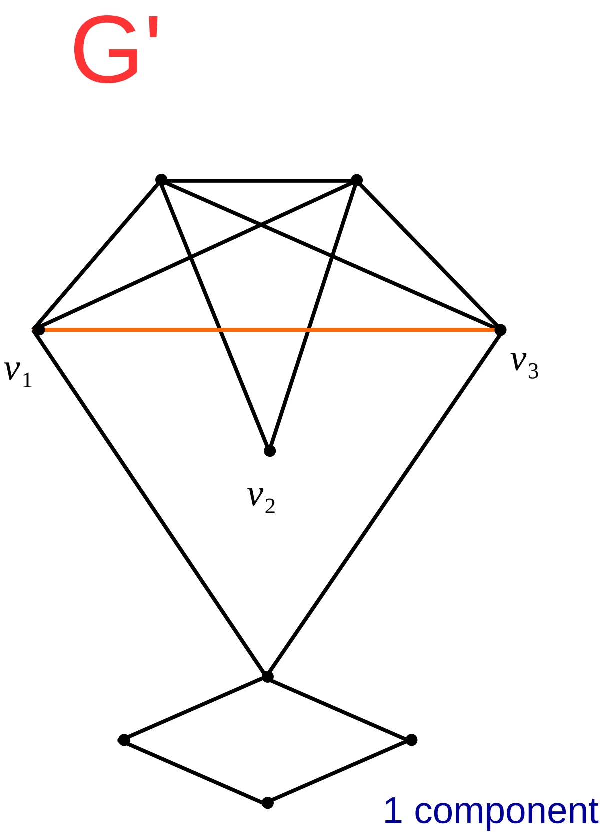
Euler Cycle





2 components


G
v_1
v_2
v_3


G'
v_1
v_2
v_3
1 component

References

[1] http://en.wikipedia.org/
[2]