Euler Paths and Circuits

Introduction

Can we travel along the edges of a graph starting at a vertex and returning to it by traversing each edge of the graph exactly once?

Similarly, can we travel along the edges of a graph starting at a vertex and returning to it while visiting each vertex of the graph exactly once?

Although these questions seem to be similar,

the first question, which asks whether a graph has an Euler circuit, can be easily answered simply by examining the degrees of the vertices of the graph,

while the second question, which asks whether a graph has a Hamilton circuit, is quite difficult to solve for most graphs.

Although both questions have many practical applications in many different areas, both arose in old puzzles.

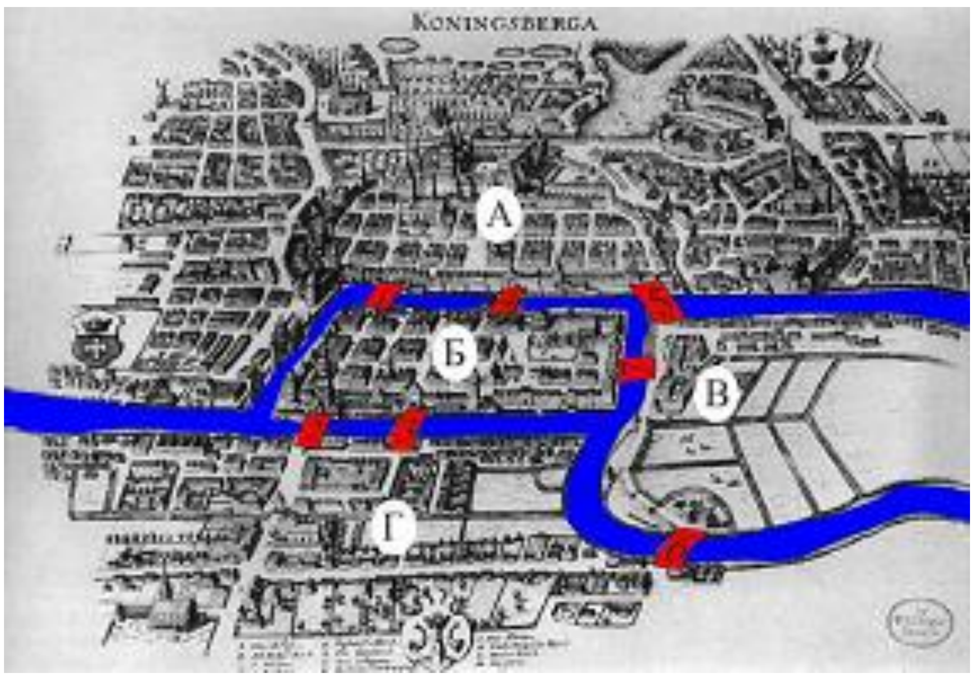
We will learn about these old puzzles as well as modern practical applications.

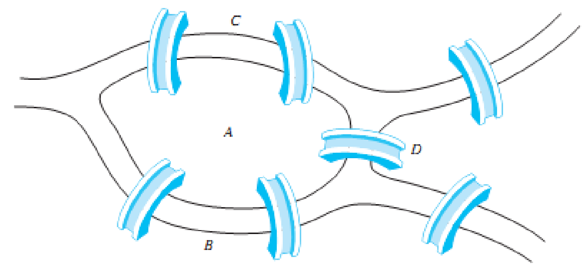
The town of Königsberg, Prussia (now called Kaliningrad and part of the Russian federation),

was divided into 4 sections by the branches of the Pregel River.

These 4 sections included the 2 regions on the banks of the Pregel, Kneiphof Island, and the region between the 2 branches of the Pregel.

In the 18th century 7 bridges connected these regions.





The Seven Bridges of Königsberg.

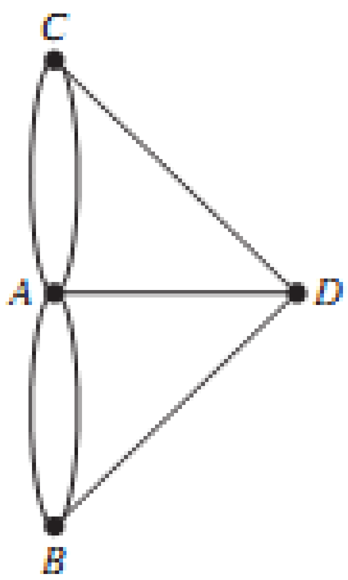
Question: Is it possible to start in some location in the town, travel across all the bridges once without crossing any bridge twice and return to the starting point.

The Swiss mathematician Leonhard Euler solved this problem.

His solution, published in 1736, may be the first use of graph theory.

Euler studied this problem using the multigraph obtained when the 4 regions are represented by vertices and the bridges by edges.

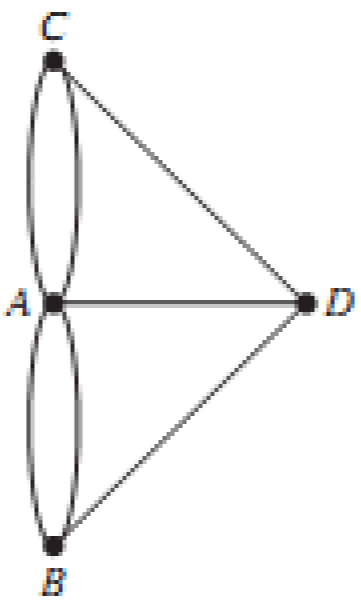
This multigraph is shown bellow.



Multigraph Model of the Town of Königsberg (Ball, 1892).

The problem of traveling across every bridge without crossing any bridge more than once can be rephrased in terms of this model.

The question becomes: Is there a simple circuit in this multigraph that contains every edge?



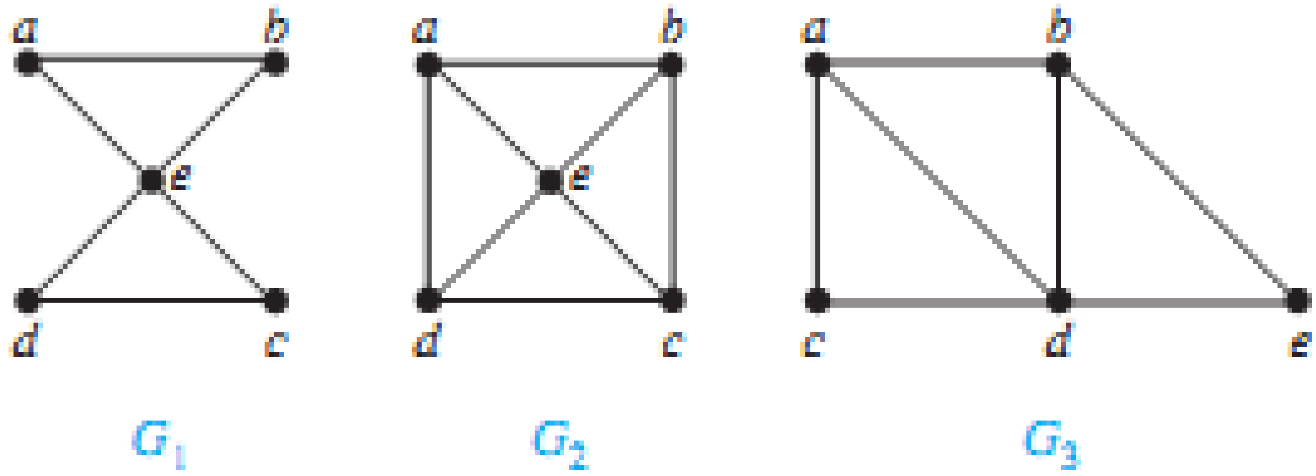
DEFINITION 1

An Euler circuit (эйлеров цикл) in a graph *G* is a simple circuit containing every edge of G.

An Euler path (эйлеров путь) in *G* is a simple path containing every edge of G.

EXAMPLE 1 Which of the undirected graphs bellow have an Euler circuit? Of those that do not, which have an Euler path?

Solution: The graph G_1 has an Euler circuit, for example, *a, e, c, d, e, b, a.*
Neither of the graphs G_2 or G_3 has an Euler circuit.
However, G_3 has an Euler path, namely, *a, c, d, e, b, d, a, b*.
G_2 does not have an Euler path.



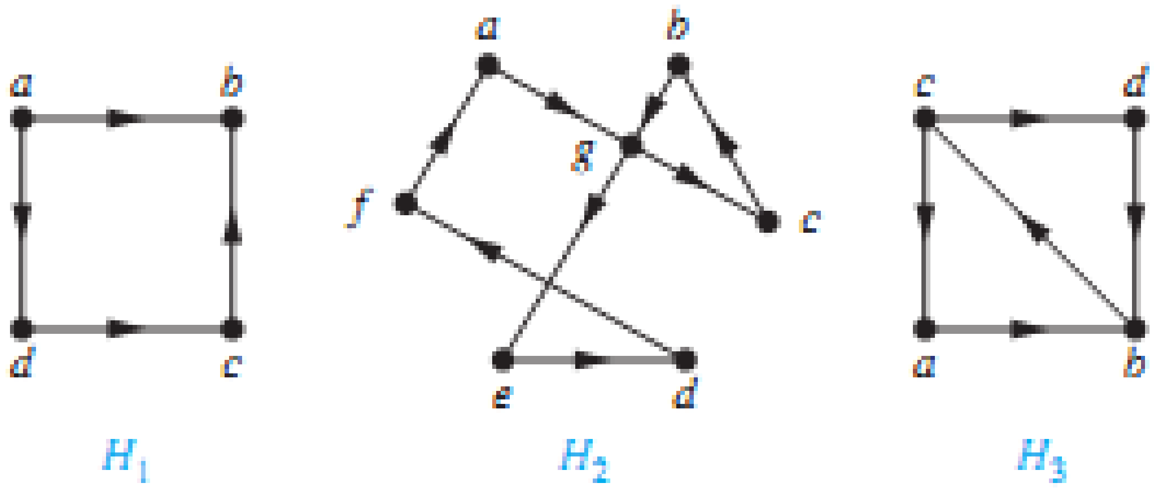
EXAMPLE 2 Which of the directed graphs bellow have an Euler circuit?

Of those that do not, which have an Euler path?

Solution: The graph H_2 has an Euler circuit, for example, *a, g, c, b, g, e, d, f, a.*

Neither H_1 nor H_3 has an Euler circuit.

H_3 has an Euler path, namely, *c, a, b, c, d, b*, but H_1 does not.



NECESSARY AND SUFFICIENT CONDITIONS FOR EULER CIRCUITS AND PATHS

There are simple criteria for determining whether a multigraph has an Euler circuit or an Euler path.

We will assume that all graphs discussed in this section have a finite number of vertices and edges.

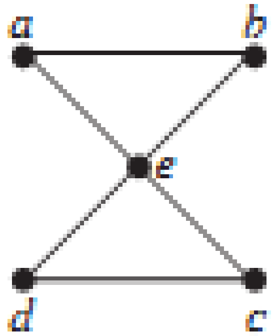
What can we say if a connected multigraph has an Euler circuit?

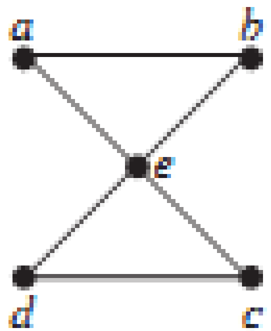
We can show that every vertex must have EVEN degree.

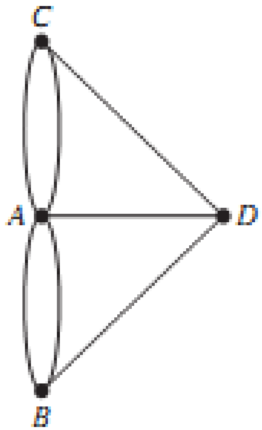
To do this, first note that an Euler circuit begins with a vertex a and continues with an edge incident with a, say $\{a, b\}$.

The edge $\{a, b\}$ contributes 1 to $\deg(a)$.

Each time the circuit passes through a vertex it contributes 2 to the vertex's degree, because the circuit enters via an edge incident with this vertex and leaves via another such edge.







Finally, the circuit terminates where it started, contributing 1 to deg(a).

Therefore, deg(a) must be even, because the circuit contributes 1 when it begins, 1 when it ends, and 2 every time it passes through a (if it ever does).

A vertex other than a has even degree because the circuit contributes 2 to its degree each time it passes through the vertex.

We conclude that if a connected graph has an Euler circuit, then every vertex must have even degree.

Is this necessary condition for the existence of an Euler circuit also sufficient?

That is, must an Euler circuit exist in a connected multigraph if all vertices have even degree?

This question can be settled affirmatively with a construction.

Suppose that G is a connected multigraph with at least 2 vertices and the degree of every vertex of G is even.

We will form a simple circuit that begins at an arbitrary vertex a of G, building it edge by edge.

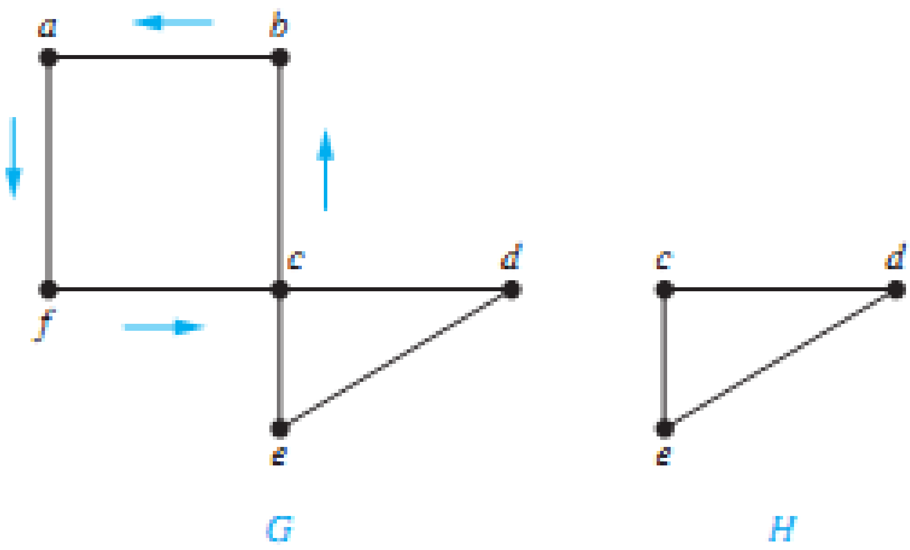
Let $x_0 = a$.

First, we arbitrarily choose an edge $\{x_0, x_1\}$ incident with a which is possible because G is connected.

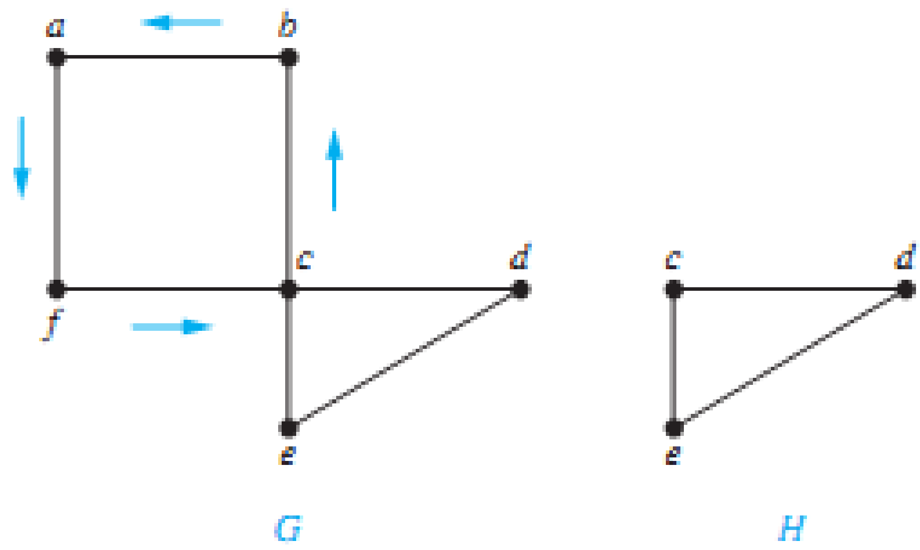
We continue by building a simple path $\{x_0, x_1\}$, $\{x_1, x_2\}$, ..., $\{x_{n-1}, x_n\}$, successively adding edges one by one to the path until we cannot add another edge to the path.

This happens when we reach a vertex for which we have already included all edges incident with that vertex in the path.

For instance, in the graph *G* bellow we begin at *a* and choose in succession the edges {*a, f*}, {*f, c*}, {*c, b*}, and {*b, a*}.

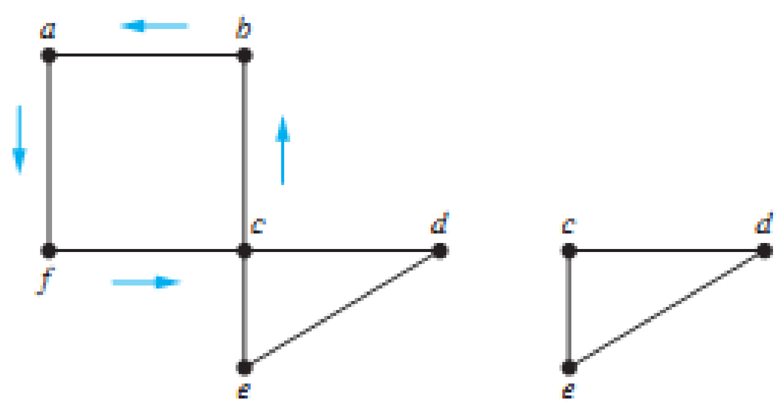


The path we have constructed must terminate because the graph has a finite number of edges, so we are guaranteed to eventually reach a vertex for which no edges are available to add to the path.



1)The path begins at *a* with an edge of the form *{a, x}*, and we now show that it must terminate at *a* with an edge of the form *{y, a}*.

To see that the path must terminate at *a*, note that each time the path goes through a vertex with even degree, it uses only 1 edge to enter this vertex, so because the degree must be at least 2, at least 1 edge remains for the path to leave the vertex.

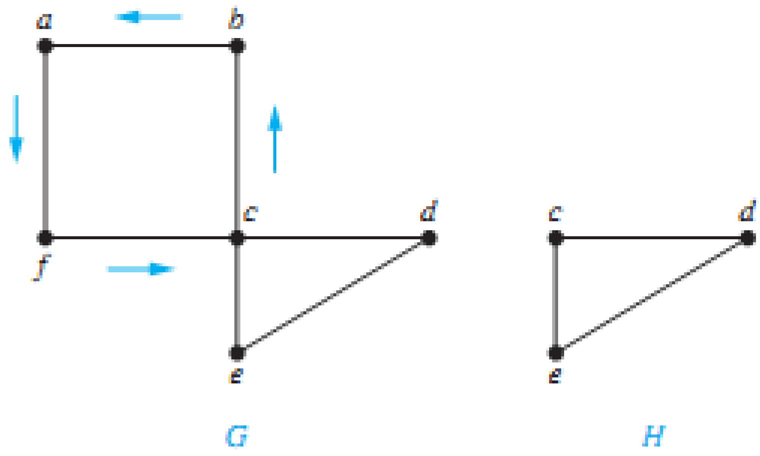


G H

Furthermore, every time we enter and leave a vertex of even degree, there are an even number of edges incident with this vertex that we have not yet used in our path.

Consequently, as we form the path, every time we enter a vertex other than *a*, we can leave it.

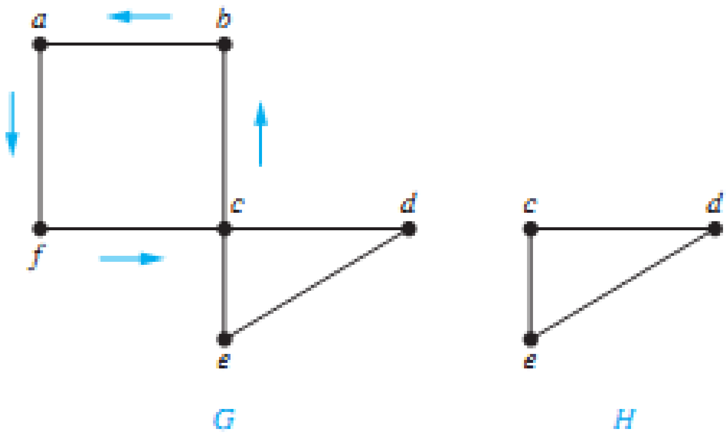
This means that the path can end only at *a*.



2) Note that the path we have constructed may use all the edges of the graph, or

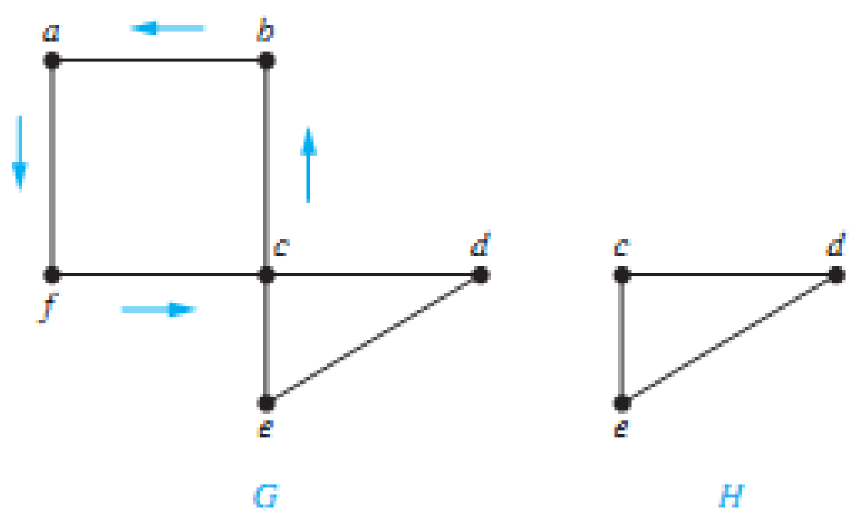
it may not if we have returned to *a* for the last time before using all the edges.

An Euler circuit has been constructed if all the edges have been used.



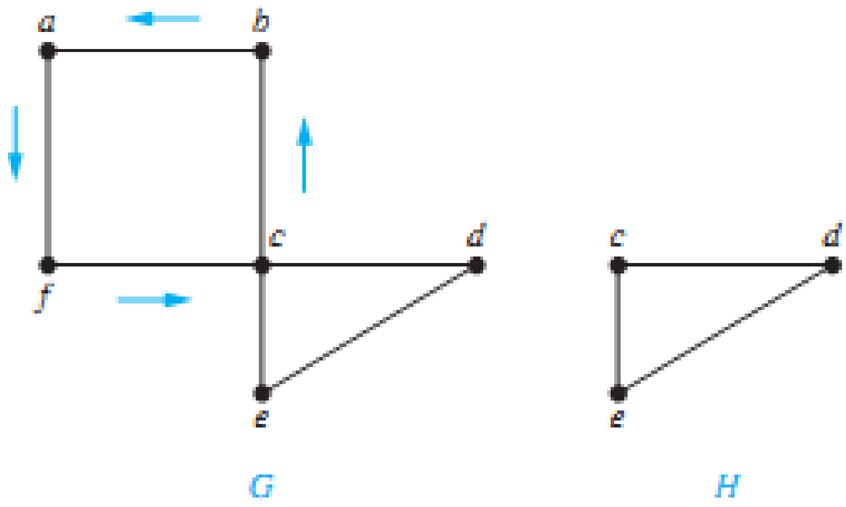
Otherwise, consider the subgraph *H* obtained from *G* by deleting the edges already used and vertices that are not incident with any remaining edges.

When we delete the *circuit a, f, c, b, a* from the graph *G*, we obtain the subgraph labeled as *H*.



Because G is connected, H has at least 1 vertex in common with the circuit that has been deleted.

Let w be such a vertex. (In our example, c is the vertex.)



Every vertex in *H* has even degree (because in *G* all vertices had even degree, and for each vertex, pairs of edges incident with this vertex have been deleted to form *H*).

Note that *H* may not be connected.

Beginning at *w,* construct a simple path in *H* by choosing edges as long as possible, as was done in *G*.

This path must terminate at *w*.

For instance, *c, d, e, c* is a path in *H* .

Next, form a circuit in *G* by splicing the circuit in *H* with the original circuit in *G.*

(this can be done because *w* is one of the vertices in this circuit).

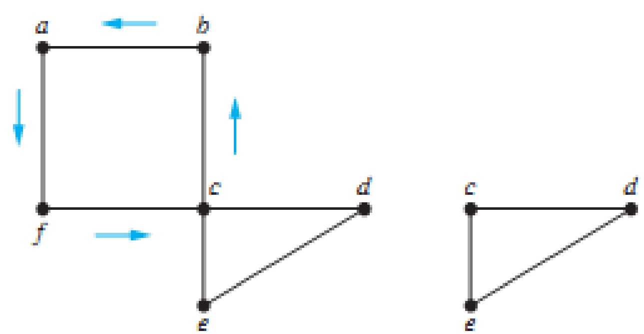
When this is done in the graph bellow, we obtain the circuit *a, f, c, d, e, c, b, a.*

Continue this process until all edges have been used.

(The process must terminate because there are only a finite number of edges in the graph.)

This produces an Euler circuit.

The construction shows that if the vertices of a connected multigraph all have even degree, then the graph has an Euler circuit.



G

H

THEOREM 1 A connected multigraph with at least 2 vertices has an Euler circuit ⇔ each of its vertices has even degree.

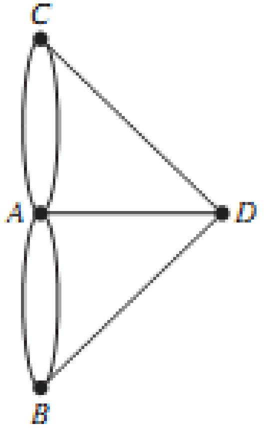
We can now solve the Königsberg bridge problem.

Because the multigraph representing these bridges, has 4 vertices of odd degree, it does not have an Euler circuit.

There is no way to start at a given point, cross each bridge exactly once, and return to the starting point.

Algorithm 1 gives the constructive procedure for finding Euler circuits given in the discussion preceding Theorem 1.

(Because the circuits in the procedure are chosen arbitrarily, there is some ambiguity. We will not bother to remove this ambiguity by specifying the steps of the procedure more precisely.)



ALGORITHM 1 Constructing Euler Circuits.

procedure *Euler*(*G:* connected multigraph with all vertices of even degree)

circuit := a circuit in *G* beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex

H := *G* with the edges of this *circuit* removed

while *H* has edges

subcircuit := a circuit in *H* beginning at a vertex in *H* that also is an endpoint of an edge of *circuit*

H := *H* with edges of *subcircuit* and all isolated vertices removed

circuit := *circuit* with *subcircuit* inserted at the appropriate vertex

return *circuit* {*circuit* is an Euler circuit}

Algorithm 1 provides an efficient algorithm for finding Euler circuits in a connected multigraph G with all vertices of even degree. The worst case complexity of this algorithm is $O(m)$, where m is the number of edges of G.

EXAMPLE 3

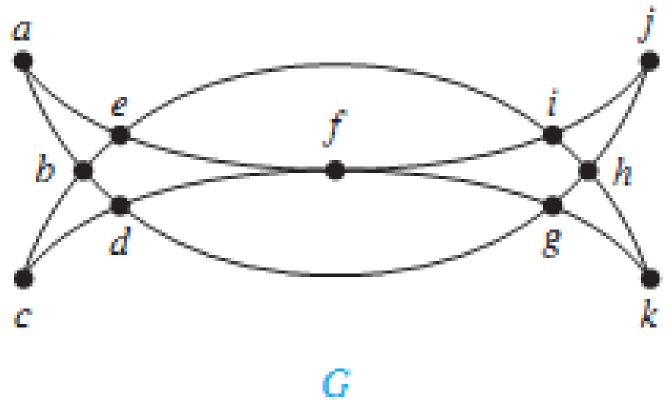
Can Mohammed’s scimitars, shown bellow, be drawn in this way, where the drawing begins and ends at the same point?

Solution: We can solve this problem because the graph *G* has an Euler circuit.

It has such a circuit because all its vertices have even degree.

We will use Algorithm1 to construct an Euler circuit.

First, we form the circuit *a, b, d, c, b, e, i, f, e, a.*



Mohammed’s Scimitars.

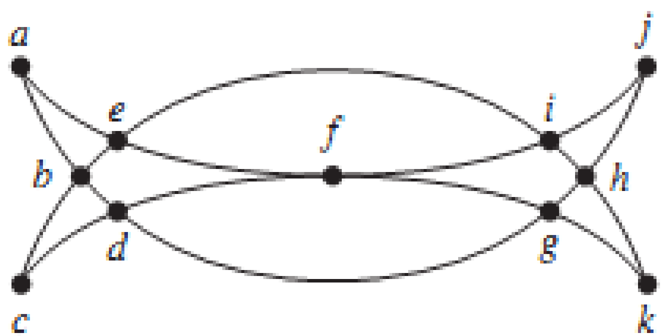
We obtain the subgraph *H* by deleting the edges in this circuit and all vertices that become isolated when these edges are removed.

Then we form the circuit *d, g, h, j, i, h, k, g, f, d* in *H*.

After forming this circuit we have used all edges in *G*.

Splicing this new circuit into the first circuit at the appropriate place produces the Euler circuit *a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a.*

This circuit gives a way to draw the scimitars without lifting the pencil or retracing part of the picture.



G

Another algorithm for constructing Euler circuits, called Fleury's algorithm, was published in 1883.

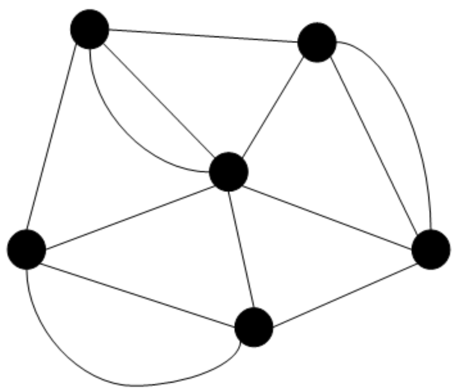
Fleury's algorithm constructs Euler circuits by first choosing an arbitrary vertex of a connected

multigraph, and then forming a circuit by choosing edges successively.

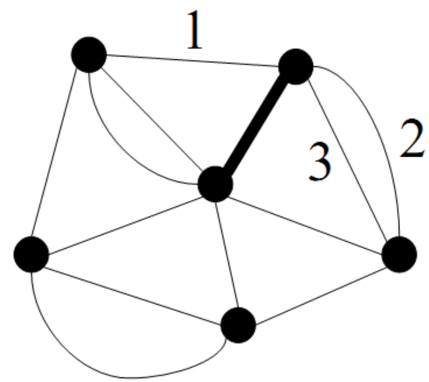
Once an edge is chosen, it is removed.

Edges are chosen successively so that each edge begins where the last edge ends, and so that this edge is not a cut edge unless there is no alternative.

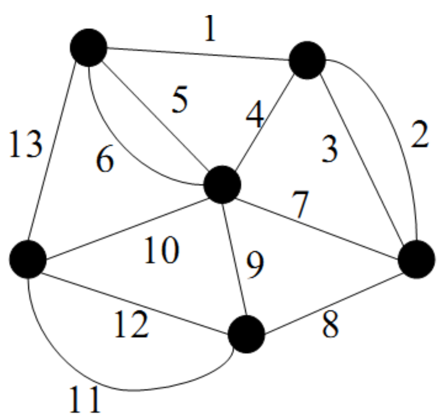
Example



(a)



(b)



(c)

We will now show that a connected multigraph has an Euler path (and not an Euler circuit) ⇔ it has exactly 2 vertices of odd degree.

First, suppose that a connected multigraph does have an Euler path from *a* to *b*, but not an Euler circuit.

The first edge of the path contributes 1 to the degree of *a*.

A contribution of 2 to the degree of *a* is made every time the path passes through *a*.

The last edge in the path contributes 1 to the degree of *b*.

Every time the path goes through *b* there is a contribution of 2 to its degree.

Consequently, both *a* and *b* have odd degree.

Every other vertex has even degree, because the path contributes 2 to the degree of a vertex whenever it passes through it.

<= Suppose that a graph has exactly 2 vertices of odd degree, say *a* and *b*.

Consider the larger graph made up of the original graph with the addition of an edge {*a, b*}.

Every vertex of this larger graph has even degree, so there is an Euler circuit.

The removal of the new edge produces an Euler path in the original graph.

Theorem 2 summarizes these results.

THEOREM 2 A connected multigraph has an Euler path but not an Euler circuit i⇔it has exactly 2 vertices of odd degree.

EXAMPLE 4

Which graphs shown bellow have an Euler path?

Solution: G_1 contains exactly 2 vertices of *odd degree*, namely, *b* and *d*. Hence, it has an Euler path that must have *b* and *d* as its endpoints.

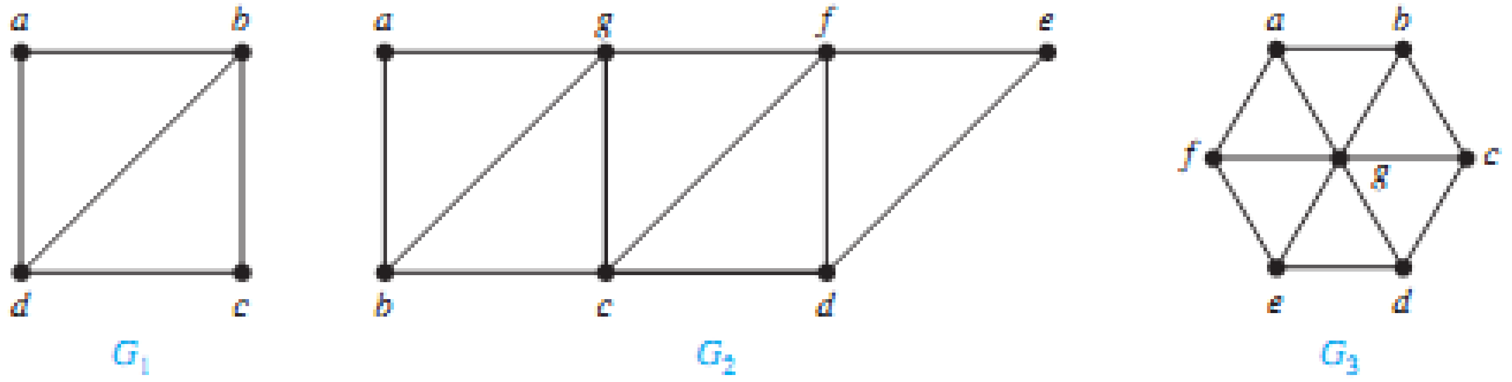
One such Euler path is *d, a, b, c, d, b.*

Similarly, G_2 has exactly 2 vertices of odd degree, namely, *b* and *d*.

So it has an Euler path that must have *b* and *d* as endpoints.

One such Euler path is *b, a, g, f, e, d, c, g, b, c, f, d*.

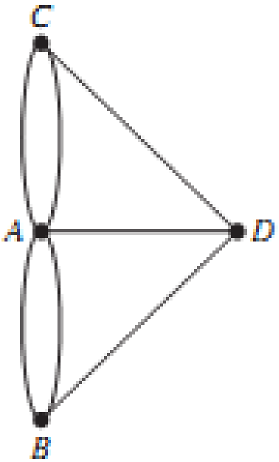
G_3 has no Euler path because it has *6* vertices of odd degree.



Returning to 18-century Königsberg, is it possible to start at some point in the town, travel across all the bridges, and end up at some other point in town?

This question can be answered by determining whether there is an Euler path in the multigraph representing the bridges in Königsberg.

Because there are 4 vertices of odd degree in this multigraph, there is no Euler path, so such a trip is impossible.



APPLICATIONS OF EULER PATHS AND CIRCUITS

Euler paths and circuits can be used to solve many practical problems.

For example, many applications ask for a path or circuit that traverses

- each street in a neighborhood,
- each road in a transportation network,
- or each link in a communications network exactly once.

Finding an Euler path or circuit in the appropriate graph model can solve such problems.

For example, if a postman can find an Euler path in the graph that represents the streets the postman needs to cover, this path produces a route that traverses each street of the route exactly once.

If no Euler path exists, some streets will have to be traversed more than once.

The problem of finding a circuit in a graph with the fewest edges that traverses every edge at least once is known as the Chinese postman problem in honor of Guan Meigu, who posed it in 1962.

Among the other areas where Euler circuits and paths are applied is in the layout of circuits, in network multicasting, and in molecular biology, where Euler paths are used in the sequencing of DNA.

Euler Traversals and RNA Chains

Proteins are encoded in living cells with ribonucleic acid (RNA, РНК) chains.

There are 4 nucleobases (азотистые основания) used in this encoding, namely, adenine (A), cytosine (C), guanine (G), and uracil (U).

Thus, an RNA chain may be denoted by a string of letters from the alphabet A, C, G, U.

There are some rules for the use of this alphabet;
the nucleobases come in triples called *codons*,

and there is one codon (AUG) that always starts a chain and

(UAA, UAG, UGA) that can end a chain.

One hypothetical RNA chain would be
AUGCAGCCUAUGGGAAAAUAG.

In order to sequence an RNA chain, biologists hit it with one or more enzymes (фермент) to break it into shorter bits.

One of these enzymes breaks a chain after every G, and another one breaks a chain after every C and after every U.

Our hypothetical RNA chain AUGCAGCCUAUGGGAAAAUAG would be broken into AUG, CAG, CCUAUG, G, G, AAAAUAG by the G-enzyme and

into AU, GC, AGC, C, U, AU, GGGAAAAU, AG by the CU-enzyme.

But of course, in reality these chain bits would not be in a nice order; they would be all jumbled together.

The mathematical task is to deduce the RNA chain sequence from the enzyme-broken bits.

Naively, there are $6! = 720$ different chains that could correspond to our G-broken example and $8! = 40{,}320$ different chains that could correspond to our CU-broken example.

First, notice that we can determine the end of the chain:

In our example, one of the CU-bits ends with G.

CU-bits: AU, GC, AGC, C, U, AU, GGGAAAAU, AG

That has to be the end bit of the chain because the CU enzyme didn’t break it.

G-bits: AUG, CAG, CCUAUG, G, G, AAAAUAG

More generally, there will be at most one CU-bit ending in A or G and at most one G-bit ending in A, C, or U.

At worst, there will be one of each type of bit ending in A, but because both bits have to end the chain, one will be a sub-bit of the other and all is well.

G-bits: AUG, CAG, CCUAUG, G, G, AAAAUAG

Now we'll take each G-bit and break it up with the CU enzyme;

for example, CCUAUG becomes C, C, U, AU, G.

AUG-> AU, G

CAG -> C, AG

CCUAUG -> C, C, U, AU, G

G

G

AAAAUAG-> AAAAU, AG-

Similarly, we'll take each CU-bit and break it up with the G-enzyme; so, for example, AGC becomes AG, C.

CU-bits: AU, GC, AGC, C, U, AU, GGGAAAAU, AG

AU,

GC ->G, C

AGC - > AG, C.

C

U

AU

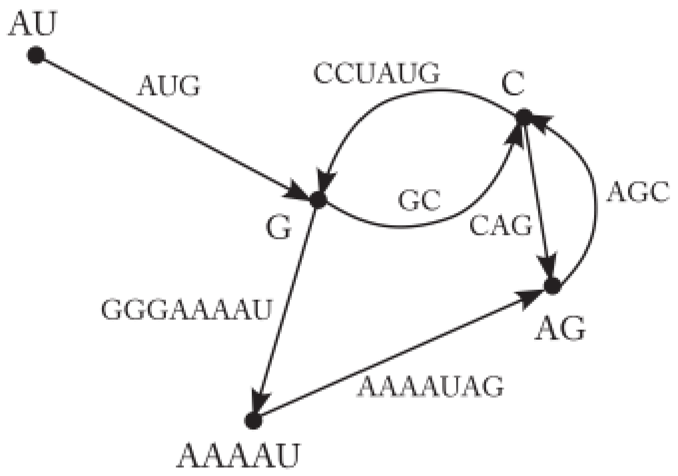
GGGAAAAU -> G, G, G, AAAAU

AG

We will make a directed edge corresponding to each G-bit and each CU bit by only paying attention to the first and last resulting bits of the double-broken sequences, so the examples from the previous two sentences become

C CCUAUG -→ G and

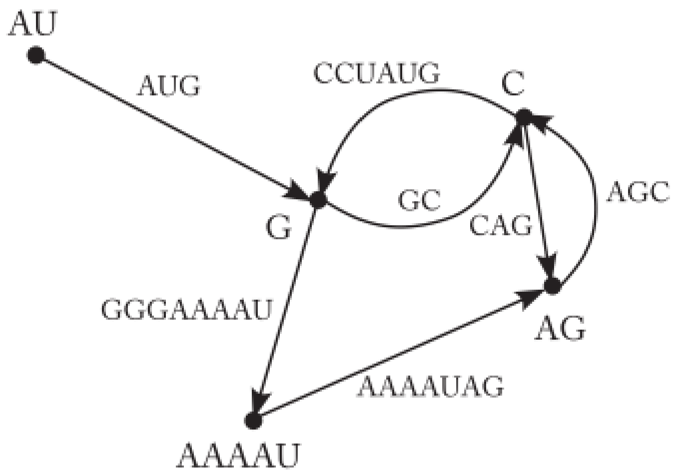
AG -→ AGC C.



Then, we glue this all together into a directed graph.

For the hypothetical RNA chain we've been using here, we obtain the digraph shown bellow.

(Notice that we do not include the fragments that are only singlebroken and not double-broken.)



Then we try to find an Euler traversal of the digraph that ends with the known end-bit.

Each Euler traversal can be "read" by listing the edge labels in the order we traverse them, eliminating the first/last letters in common.

If we are lucky, there is only one Euler traversal.

For our example digraph , there are multiple Euler traversals—we must start our reconstruction with AUG, but could then continue as AUGGGAAAAUAGCCUAUGCAG,
as AUGCAGCCUAUGGGAAAAUAG,

or as AUGCCUAUGGGAAAAUAGCAG.