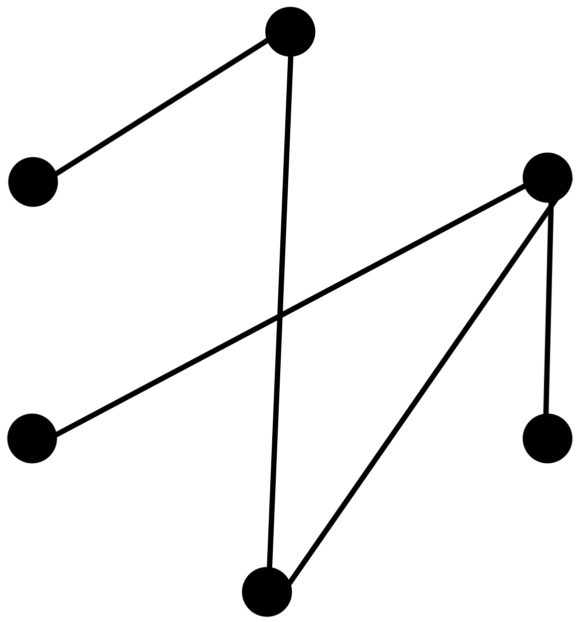
Lecture 10. Graphs

DR. YARASLAU ZADVORNY

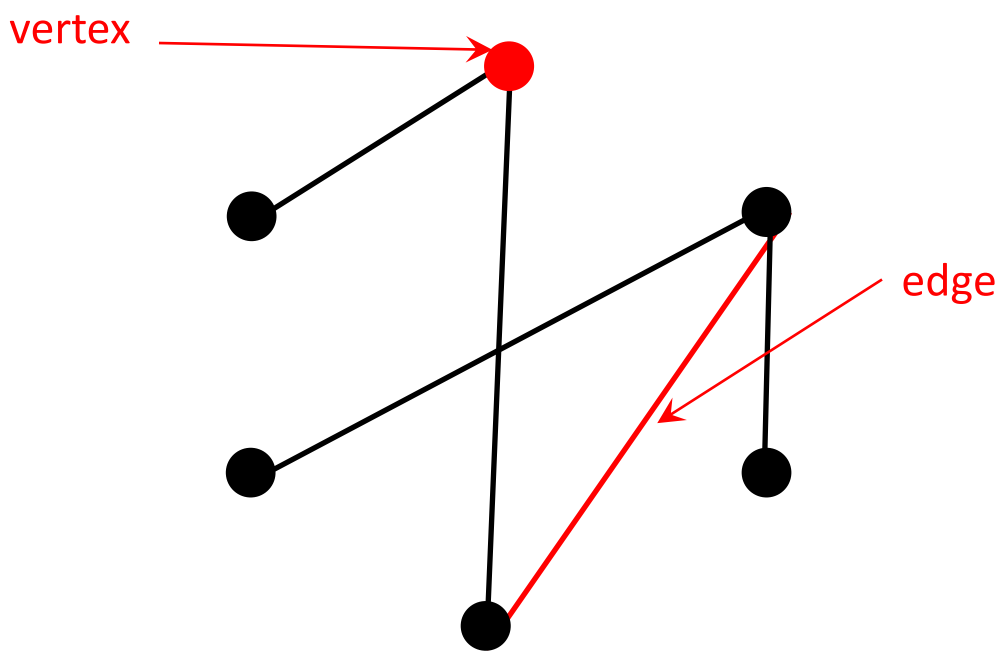
Graphs

Conceptually, a graph is formed by vertices and edges connecting the vertices.

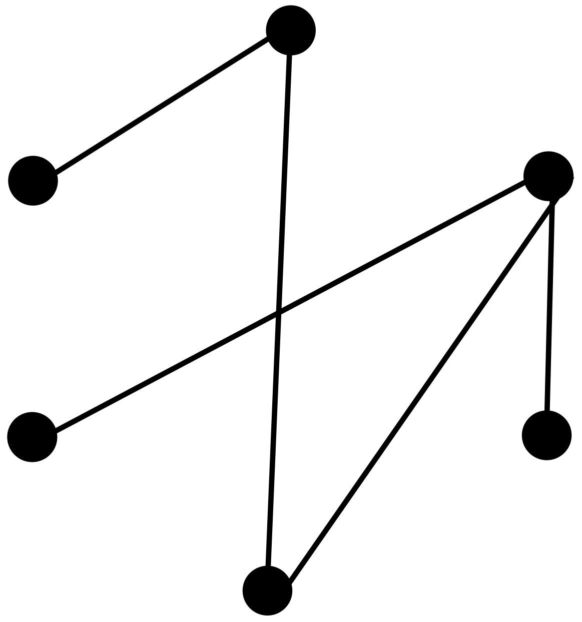


Graphs

Conceptually, a graph is formed by vertices and edges connecting the vertices.



Graphs: what are they in real live?

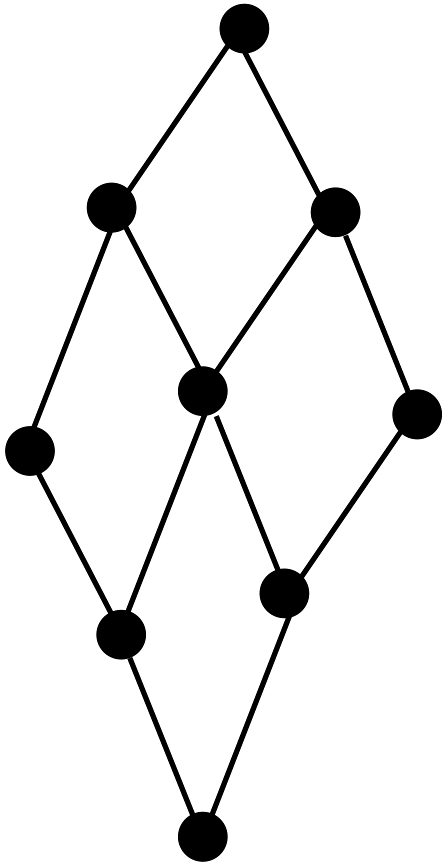


1) Suppose there is a company of friends. Then the vertices are the people in the company, and we join two vertices with an edge if the corresponding persons are friends.

2) Suppose there is a chess tournament. Each player should play a game with each of the other players. Then the vertices are the players, and we join two vertices with an edge if the corresponding players have already played a game against each other.

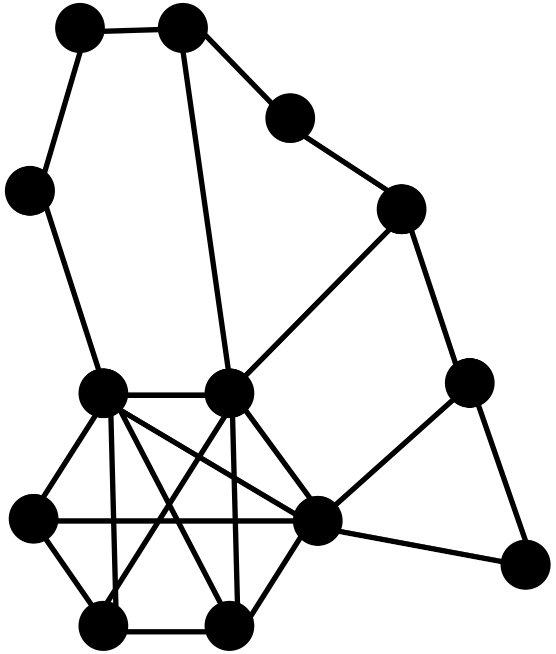
3) Suppose there is a country, and there are several cities in it. Then the vertices are the cities, and we join two vertices with an edge if the corresponding cities are joined with a road.

Graphs: what are they in practice?



4) Suppose you want to design a GPS-navigator. Then you need to study the graph of the city: the squares and the crossroads are the vertices, and the streets are the edges.

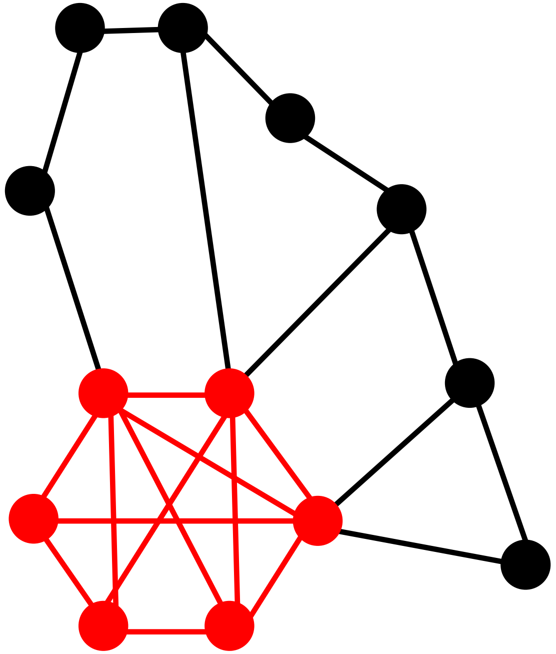
Graphs: what are they in practice?



4) Suppose you want to design a GPS-navigator. Then you need to study the graph of the city: the squares and the crossroads are the vertices, and the streets are the edges.

5) Suppose have made your social network. Then the users are vertices, and the edges are “friendships” among them.

Graphs: what are they in practice?

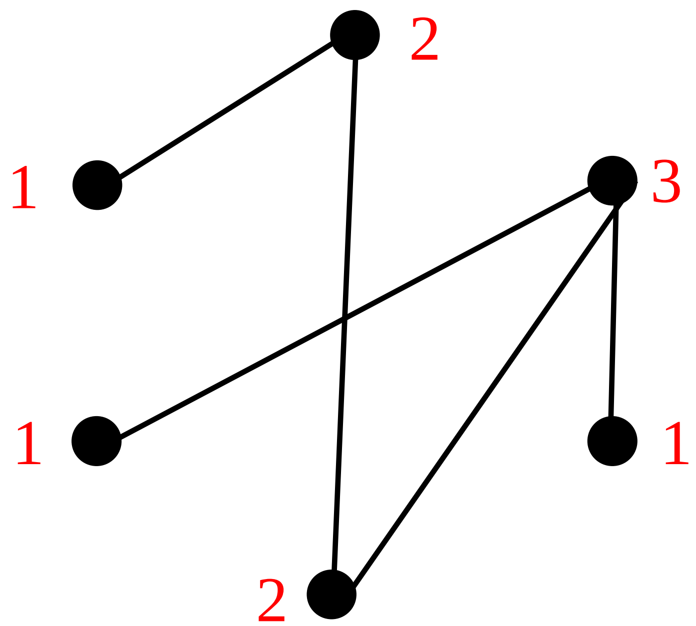


4) Suppose you want to design a GPS-navigator. Then you need to study the graph of the city: the squares and the crossroads are the vertices, and the streets are the edges.

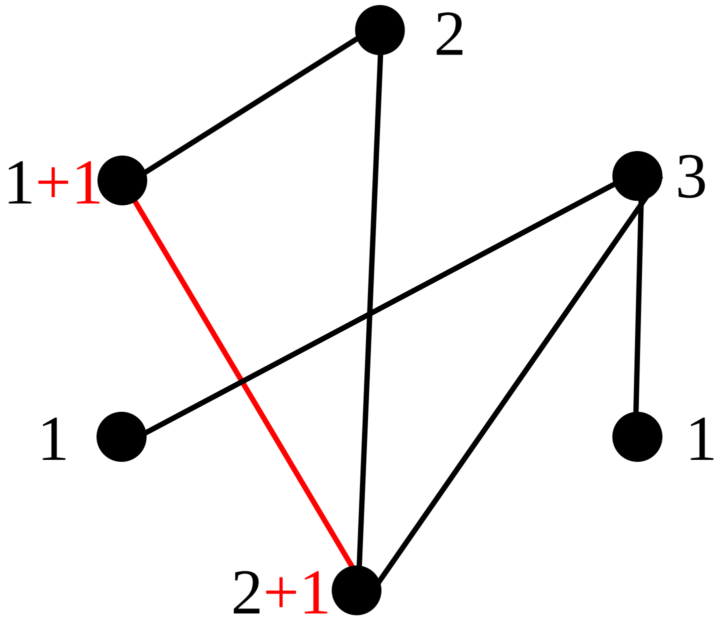
5) Suppose have made your social network. Then the users are vertices, and the edges are "friendships" among them.

Vertices and edges

The number of edges which are adjacent to a vertex is a **degree** of the vertex.



Handshacking Lemma

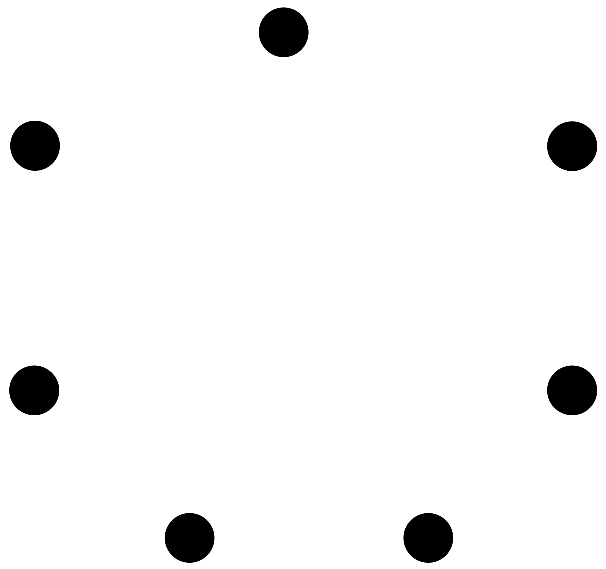


The sum of degrees of the vertices in any graph is an even number.

In other words, the number of vertices of odd degree in any graph is even.

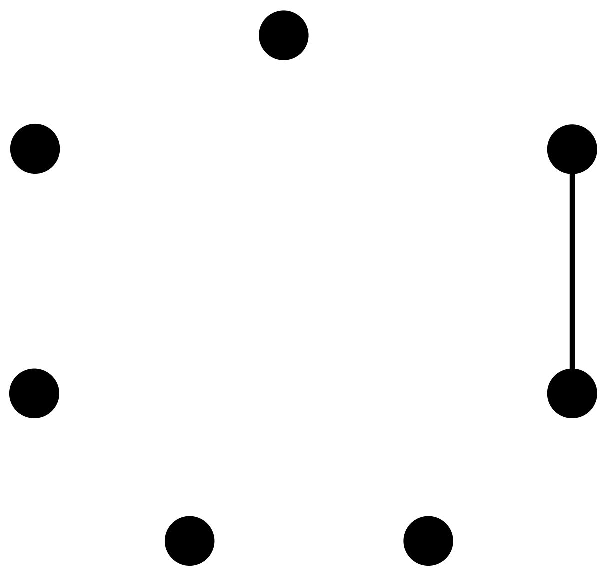
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



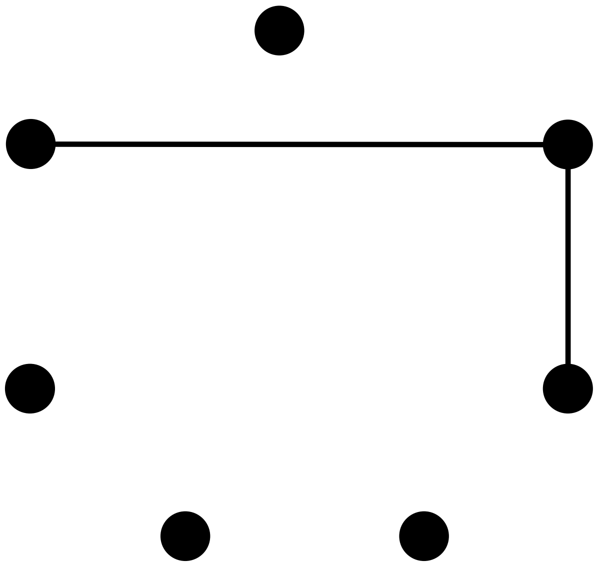
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



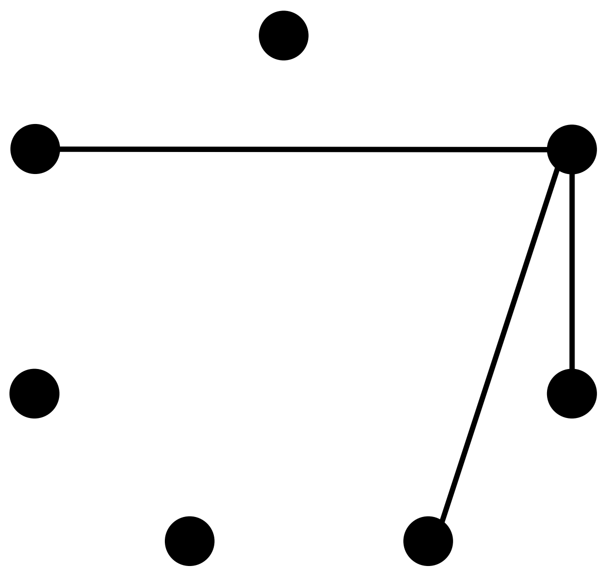
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



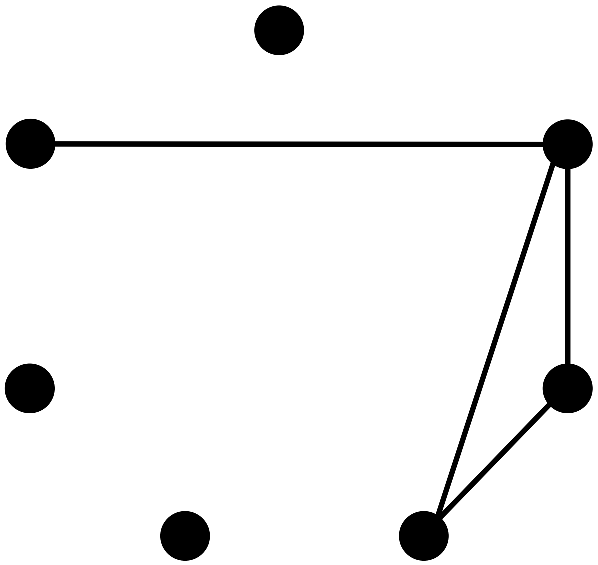
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



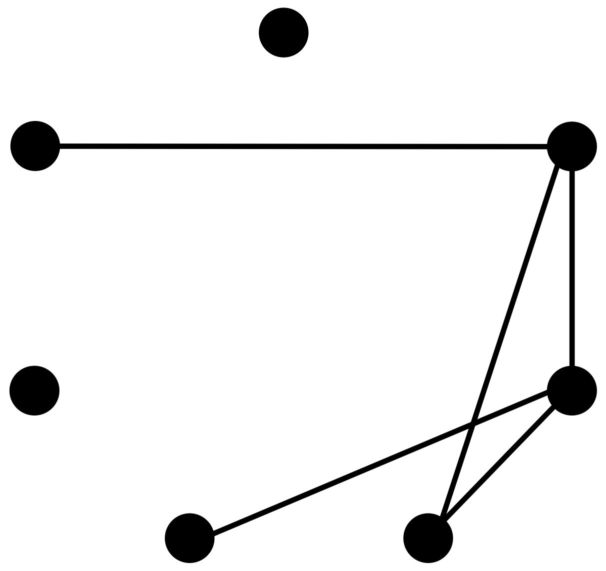
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



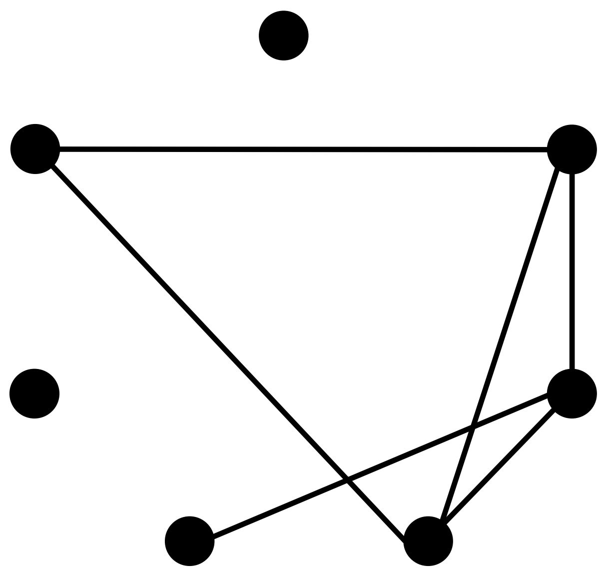
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



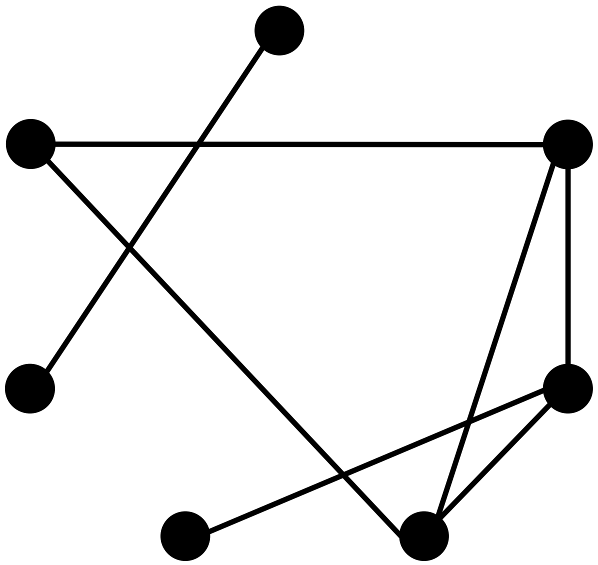
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



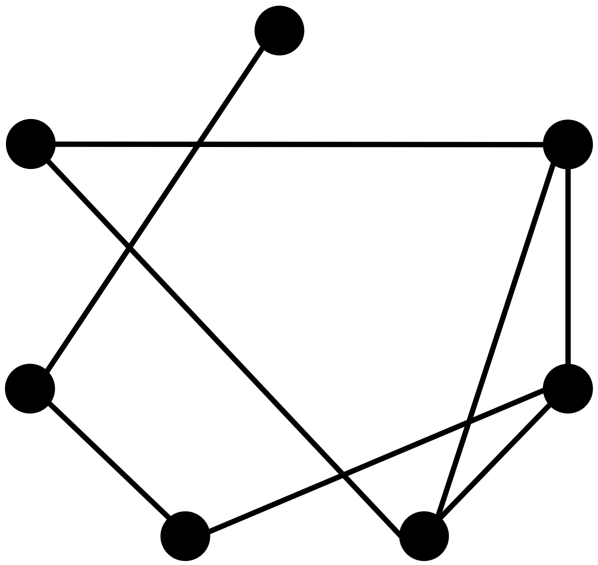
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



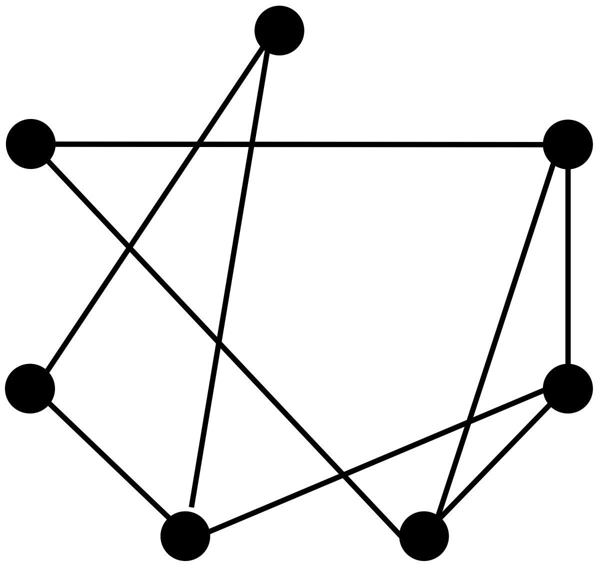
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



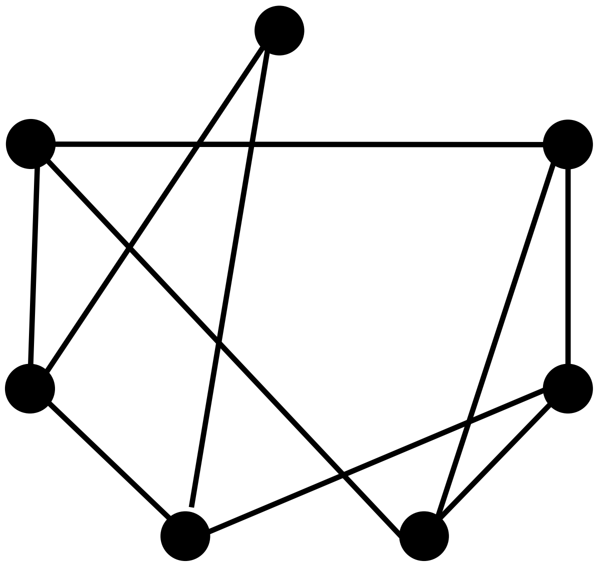
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



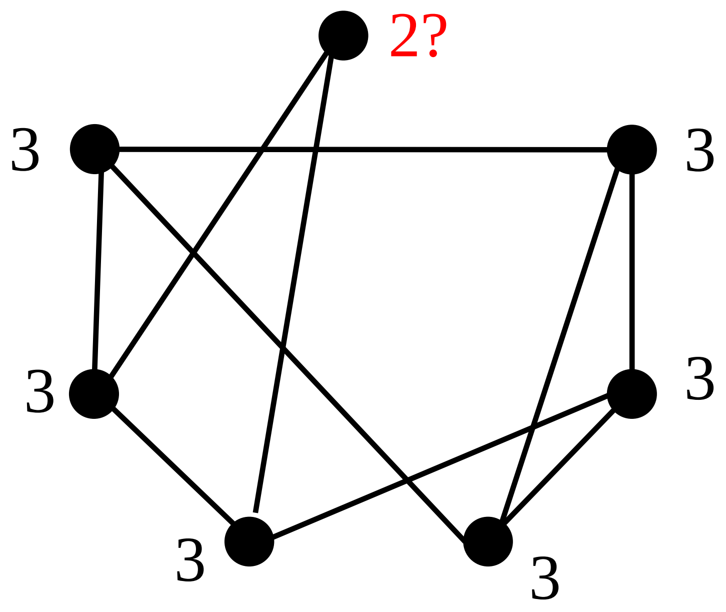
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



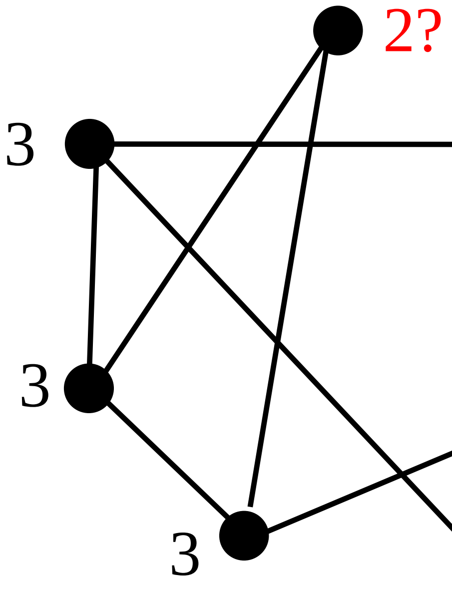
A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



A question

Suppose there are 7 students in a group. Each of them has 3 friends in this group. Is it possible?



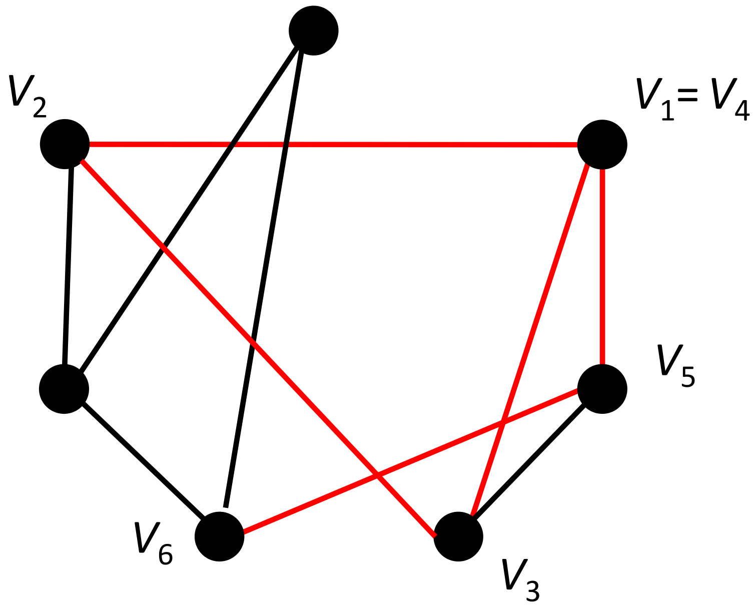
If it was possible, then the sum of the degrees of the vertices would be an even number, but it is required the sum of the degrees to be equal to

$$3 \times 7 = 21,$$

which is impossible.

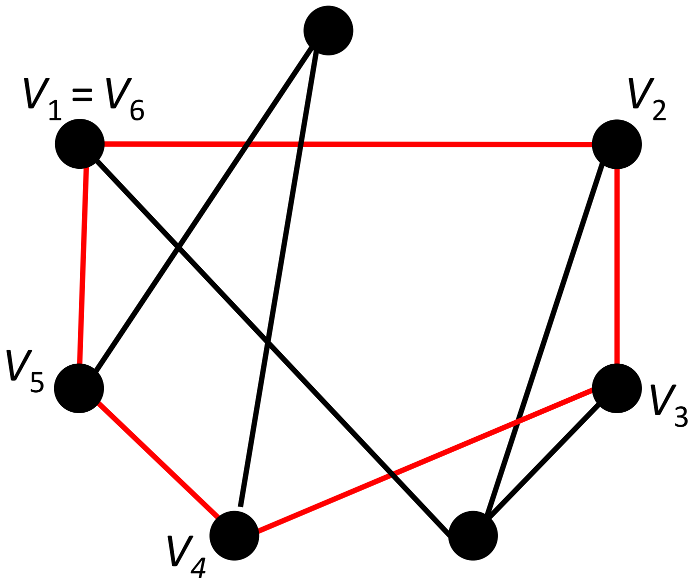
Several definitions

A **path** is some sequence of vertices V_1, V_2, …, V_k, such that V_i is connected to V_{i+1}.



Several definitions

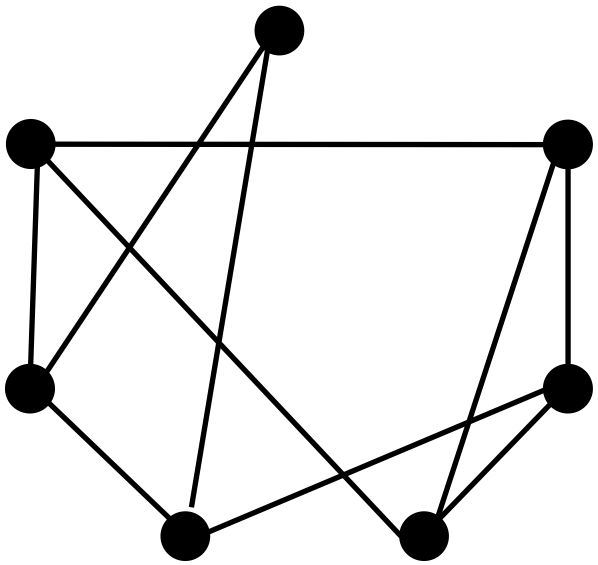
A **path** is some sequence of vertices V_1, V_2, ..., V_k, such that V_i is connected to V_{i+1}.



A **cycle** is a path $\{V_1, V_2, ..., V_k\}$ such that $V_1 = V_k$.

Several definitions

A **path** is some sequence of vertices V_1, V_2, …, V_k, such that V_i is connected to V_{i+1}.

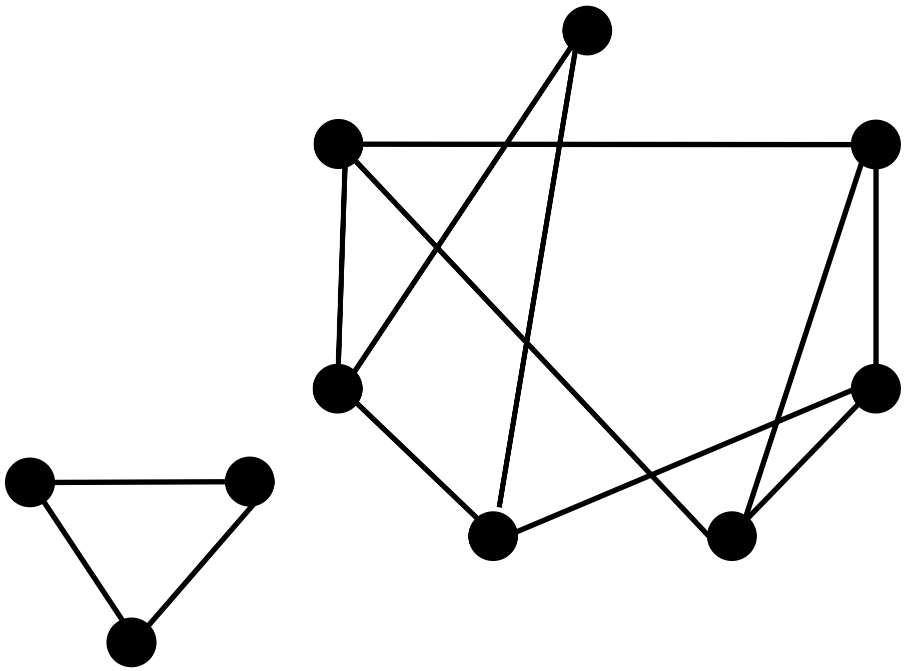


A **cycle** is a path $\{V_1, V_2, \ldots, V_k\}$ such that $V_1 = V_k$.

A graph is **connected**, if for any two vertices there is a path which joins these vertices.

Several definitions

A **path** is some sequence of vertices V_1, V_2, …, V_k, such that V_i is connected to V_{i+1}.

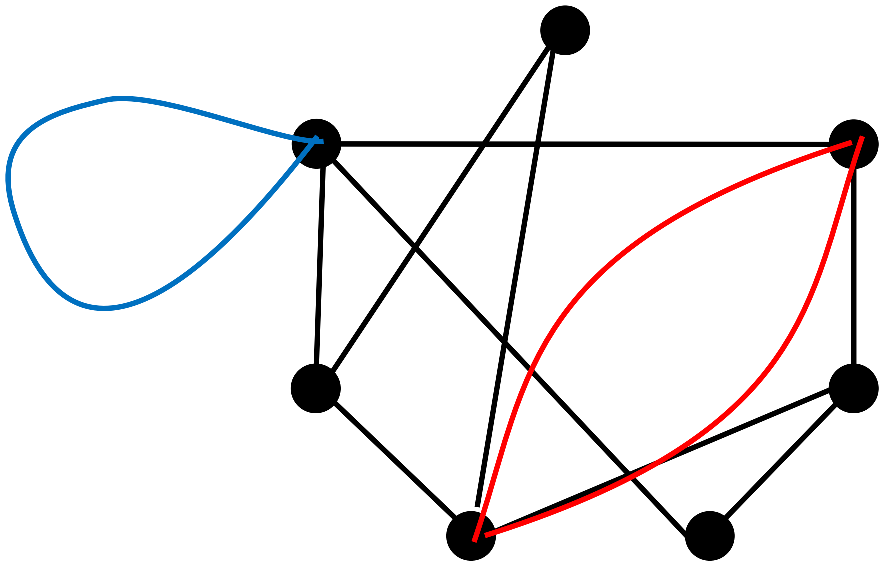


A **cycle** is a path $\{V_1, V_2, \ldots, V_k\}$ such that $V_1 = V_k$.

A graph is **connected**, if for any two vertices there is a path which joins these vertices.

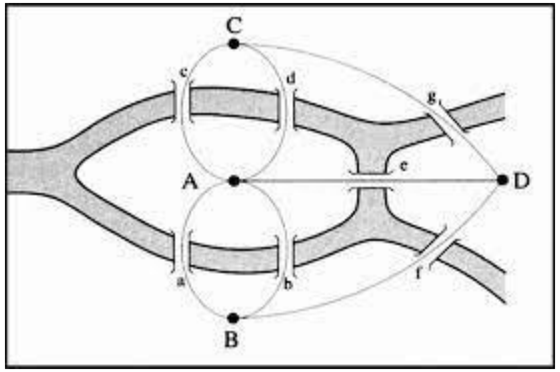
This graph is not connected and consists of two **components**.

Several definitions



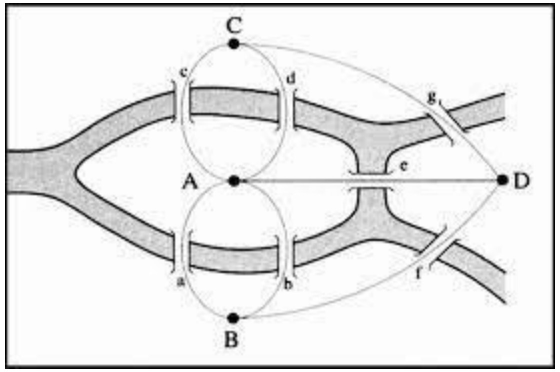
A **multigraph** is a graph in which **multiple edges** (that is, two or more edges which join the same pair of vertices) and **loops** (the edges which join the vertex with itself) are permitted.

Eulerian Graphs: Seven Bridges of Königsberg

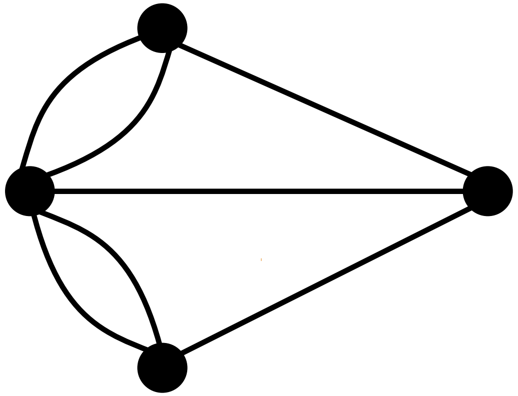


Is it possible to walk by the bridges in such a way that any bridge is been crossed exactly once?

Eulerian Graphs: Seven Bridges of Königsberg

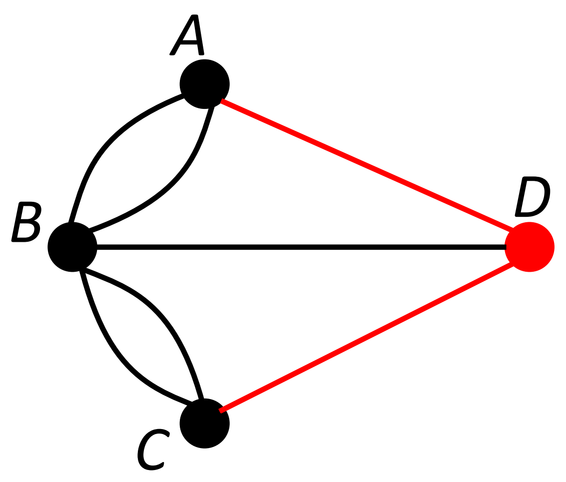


Is it possible to walk by the bridges in such a way that any bridge is been crossed exactly once?



That is, does there exist a path in the graph which contains each of its edges?

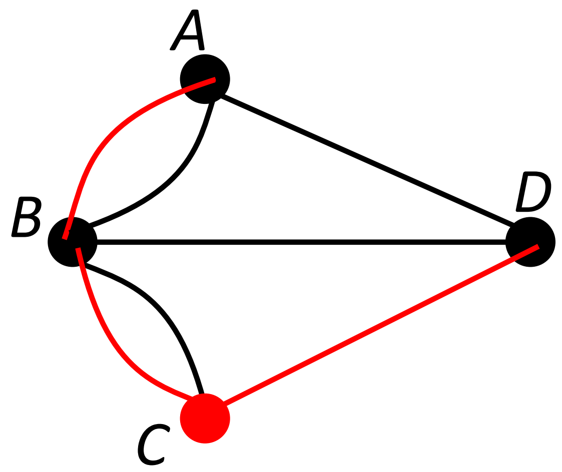
Eulerian Graphs: Seven Bridges of Königsberg



That is, does there exist a cycle in the graph which contains each of its edges?

Suppose we begin in the vertex *A* and have once gone throw the vertex *D*. Then we have to finish our walk in the vertex *D*.

Eulerian Graphs: Seven Bridges of Königsberg



That is, does there exist a cycle in the graph which contains each of its edges?

Suppose we begin in the vertex *A* and have once gone throw the vertex *D*. Then we have to finish our walk in the vertex *D*.

But the same happens to the vertex *C*. Thus, the root we are looking for doesn’t exist.

Eulerian graphs

A path in a graph is said to be the **Eulerian path** if it contains each edge exactly once.

A cycle in a graph is said to be the **Eulerian cycle** if it contains each edge exactly once.

A graph is said to be **Eulerian** if it has an Eulerian cycle.

Eulerian graphs

A path in a graph is said to be the **Eulerian path** if it contains each edge exactly once.

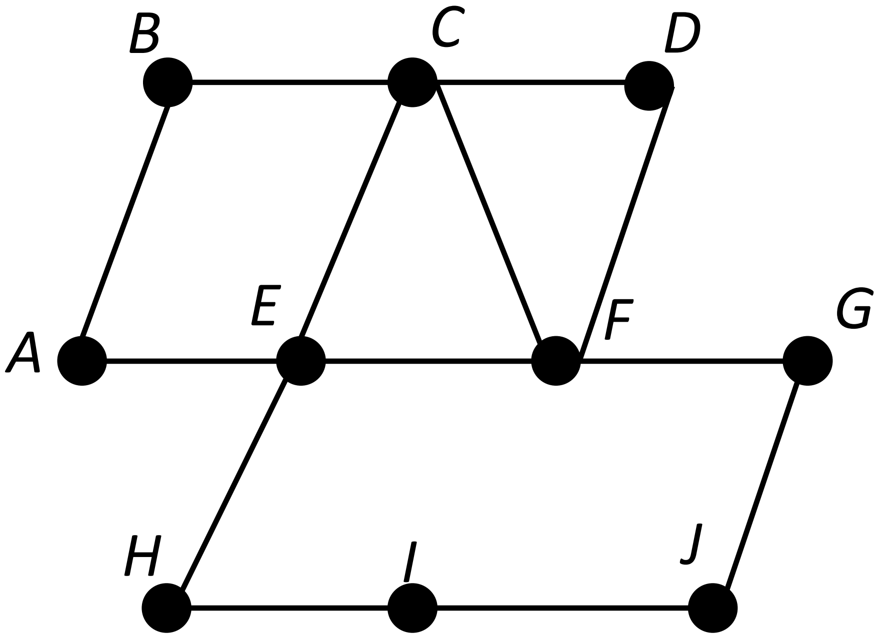
A cycle in a graph is said to be the **Eulerian cycle** if it contains each edge exactly once.

A graph is said to be **Eulerian** if it has an Eulerian cycle.

If there is at least one (and, thus, at least two) vertices with odd degrees in the graph then there is no Eulerian cycle.

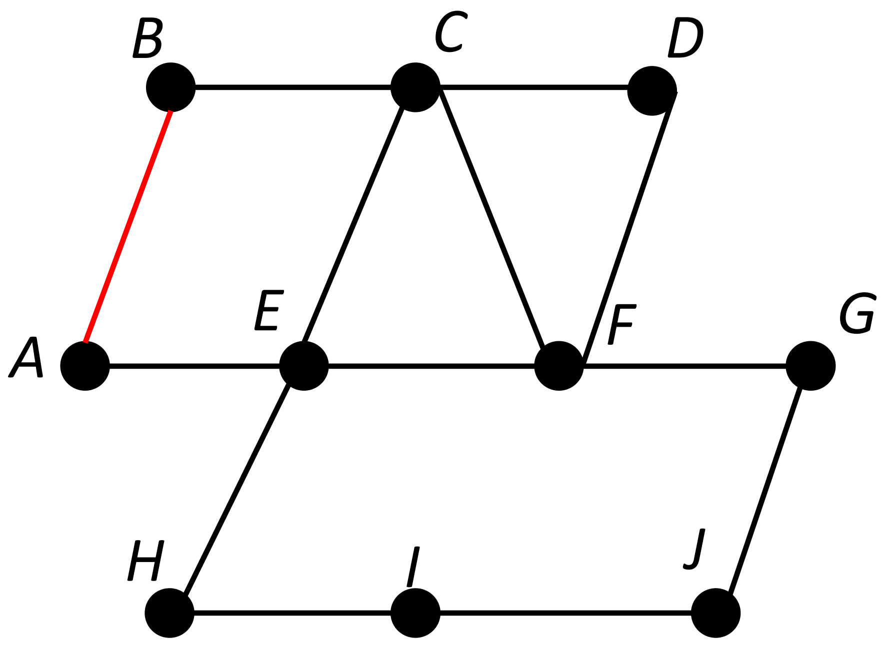
If there is at least three (and, thus, at least four) vertices with odd degrees in the graph then there is no Eulerian path.

How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

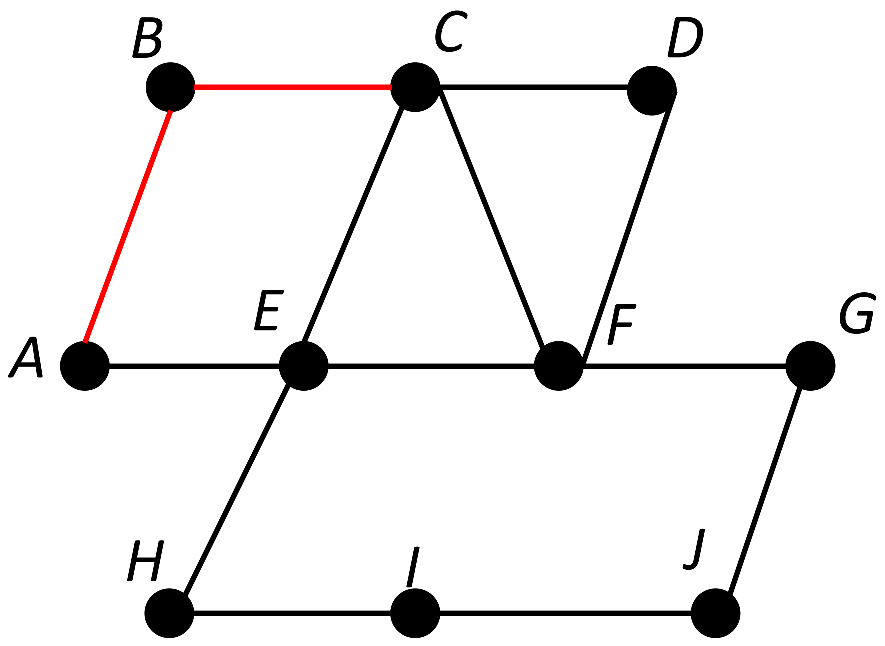
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, *B*,

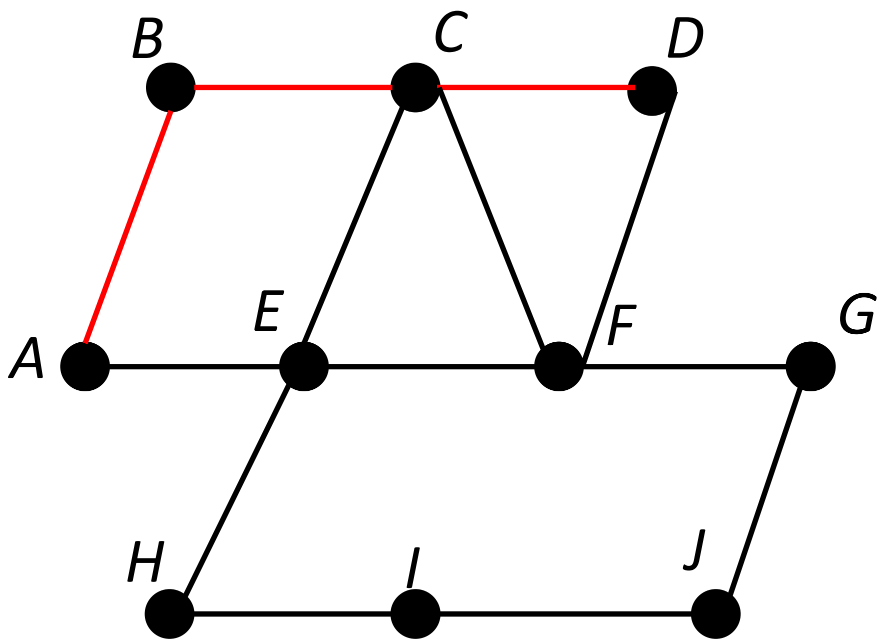
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, *B*, *C*,

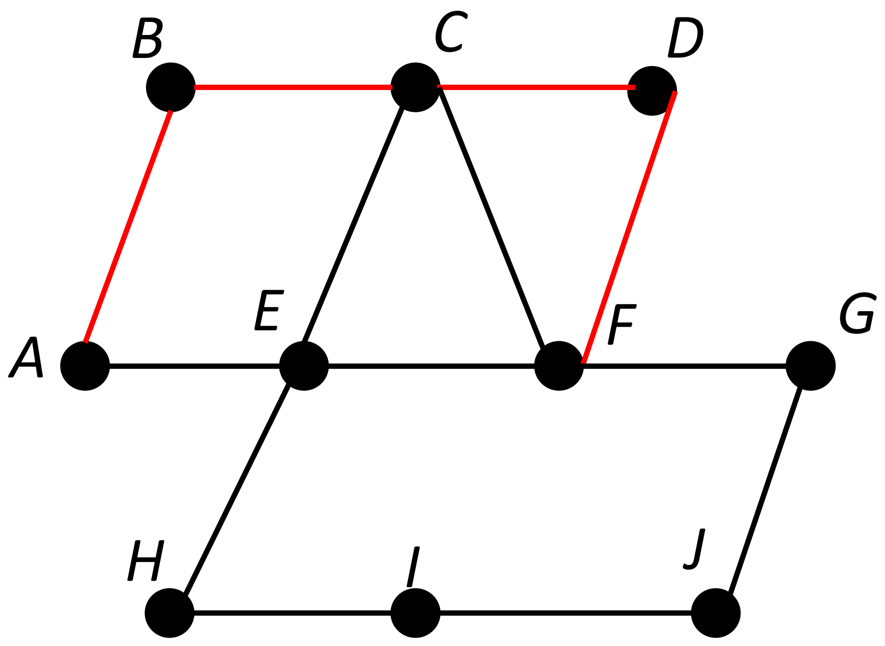
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, *B*, *C*, *D*,

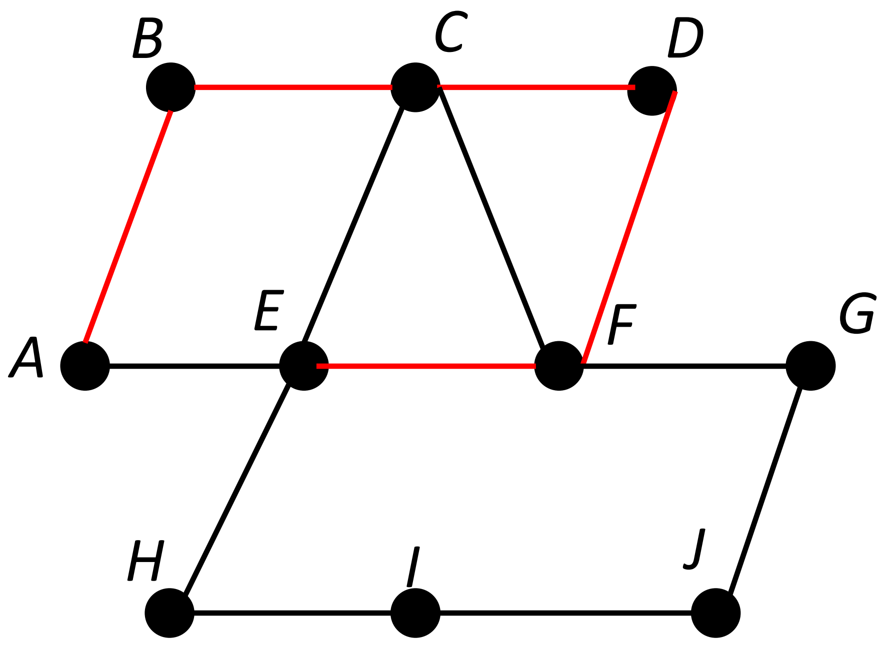
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F,

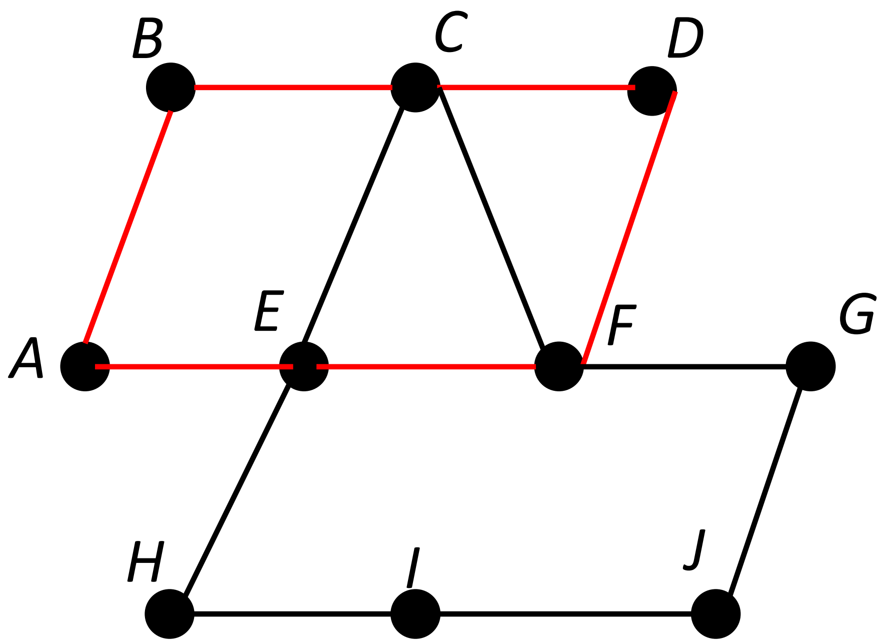
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E,

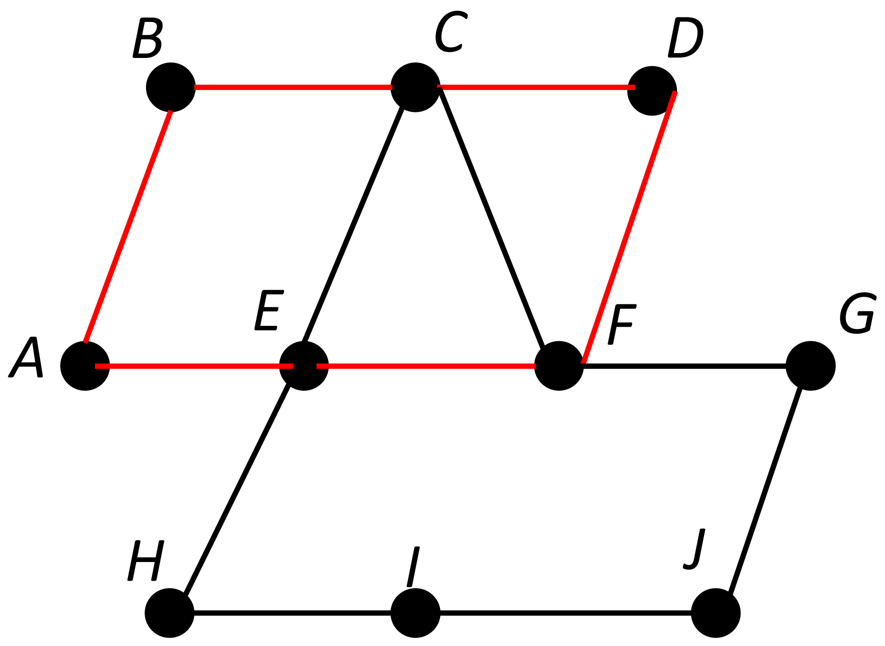
How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

How to construct an Eulerian cycle?

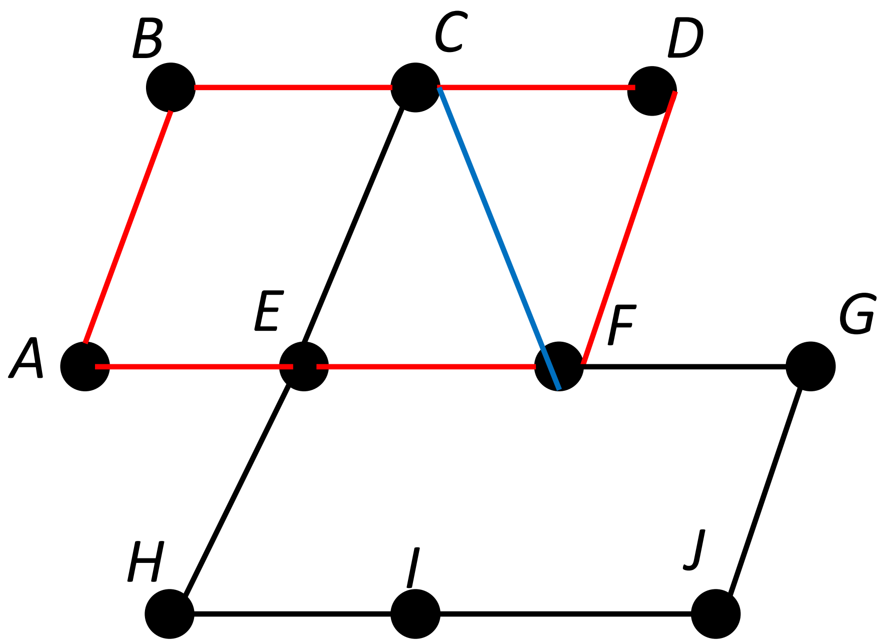


Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

How to construct an Eulerian cycle?



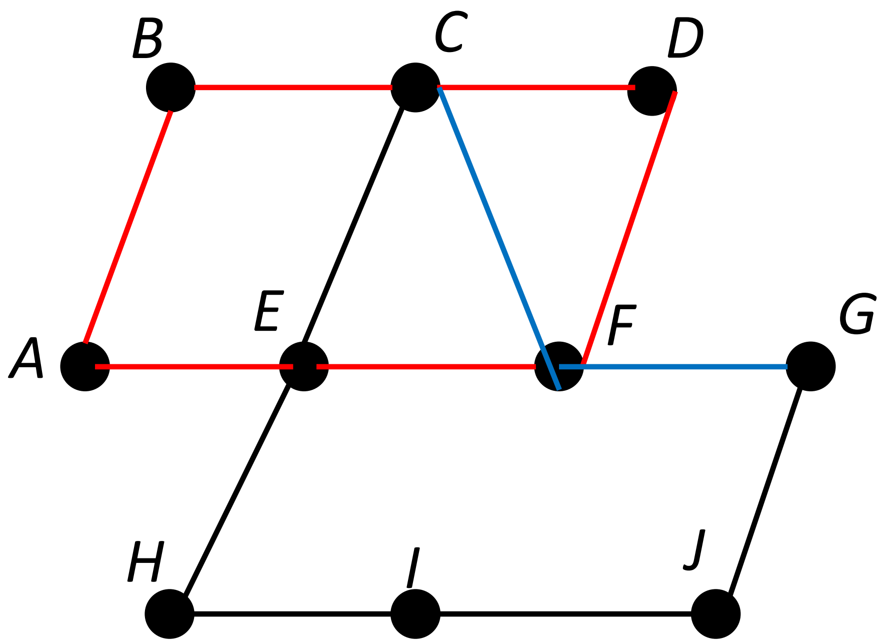
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F,

How to construct an Eulerian cycle?



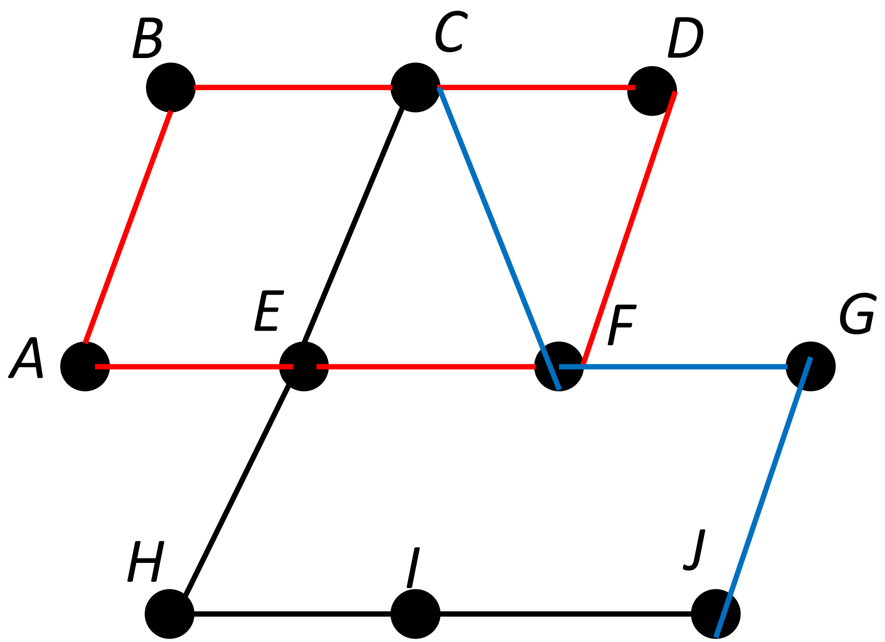
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G,

How to construct an Eulerian cycle?



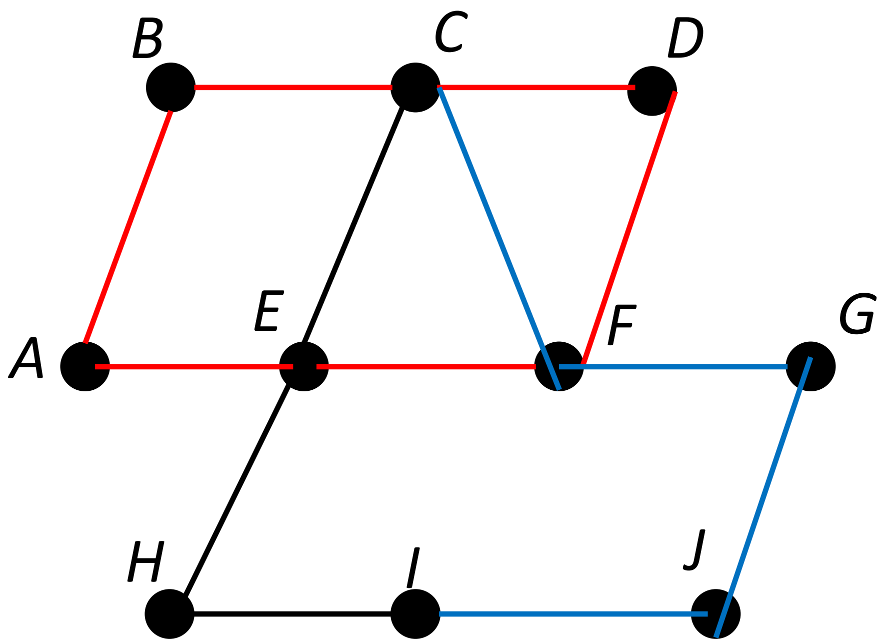
Firstly, we choose any vertex and go anywhere, while it is possible.

A, *B*, *C*, *D*, *F*, *E*, *A*

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, *F*, *G*, *J*,

How to construct an Eulerian cycle?



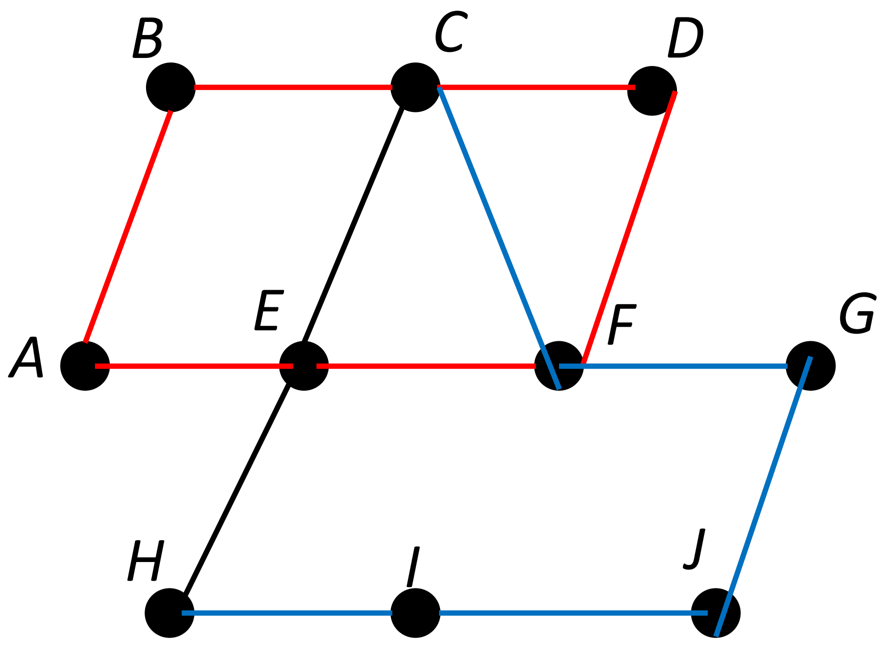
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G, J, I

How to construct an Eulerian cycle?



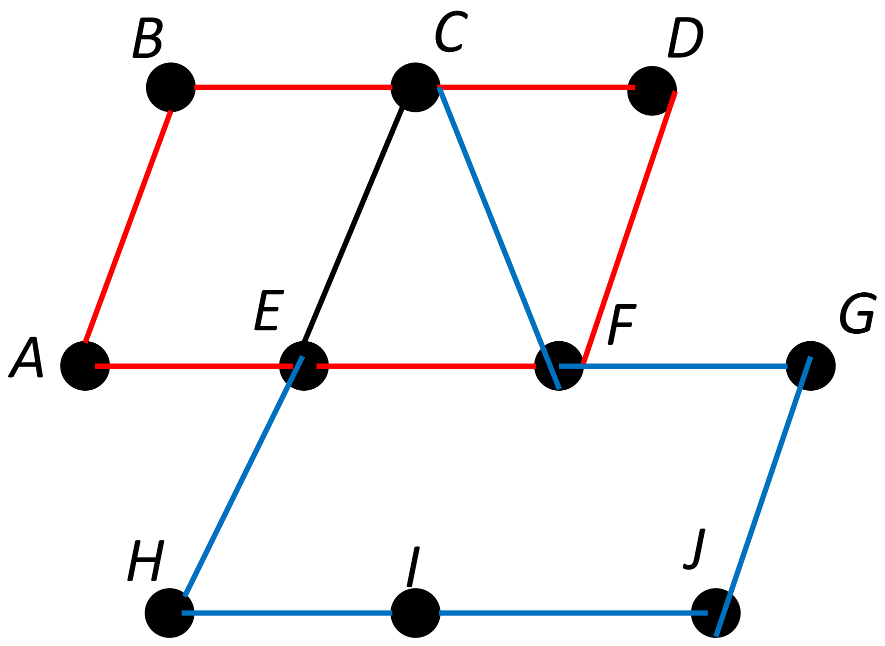
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G, J, I, H,

How to construct an Eulerian cycle?



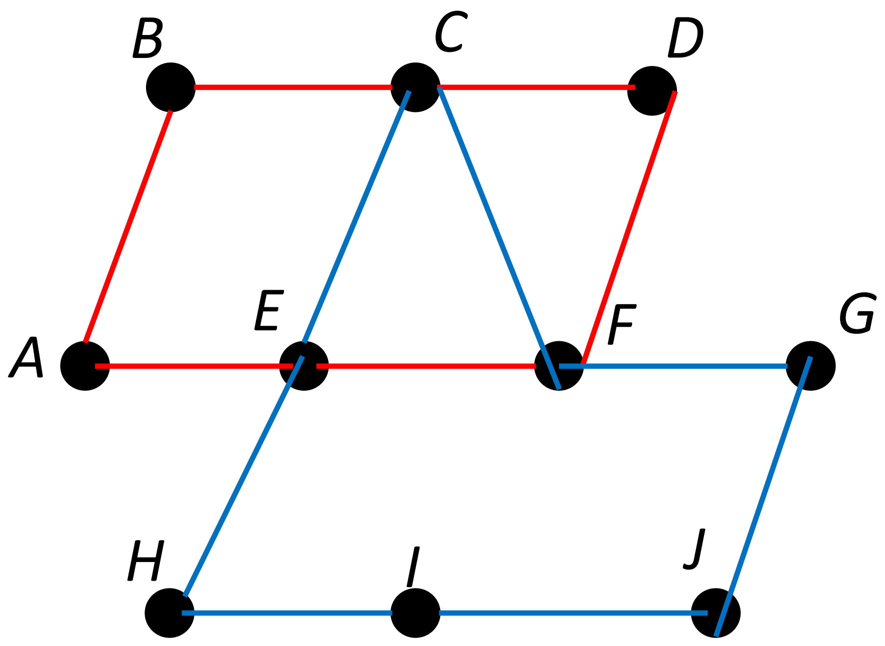
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G, J, I, H, E

How to construct an Eulerian cycle?



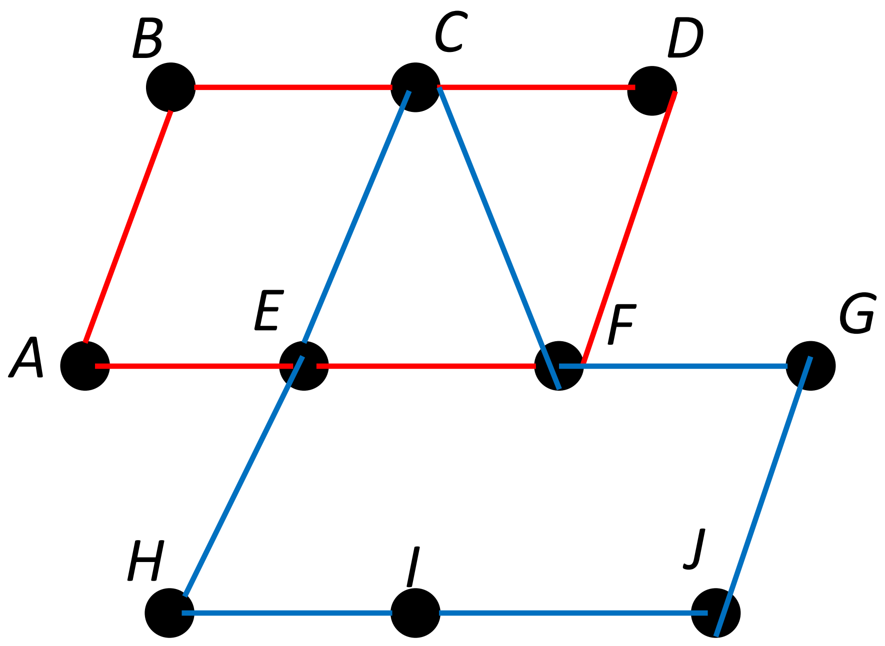
Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G, J, I, H, E, C

How to construct an Eulerian cycle?



Firstly, we choose any vertex and go anywhere, while it is possible.

A, B, C, D, F, E, A

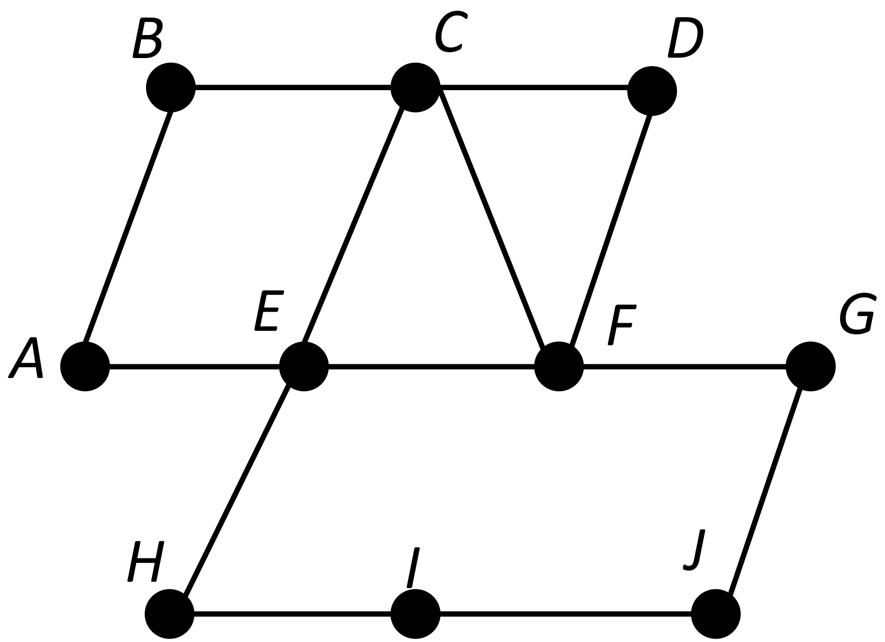
Now we choose any vertex which is adjacent to at least one black edge ant simply do the same.

C, F, G, J, I, H, E, C

The rest is to join these two cycles, say,

A, B, C, F, G, J, I, H, E, C, D, F, E, A

Hamiltonian cycle



The **Hamiltonian cycle** in a graph is a cycle which passes through any **vertex** exactly once.

Say, *A*, *B*, *C*, *D*, *F*, *G*, *J*, *I*, *H*, *E*, *A*.

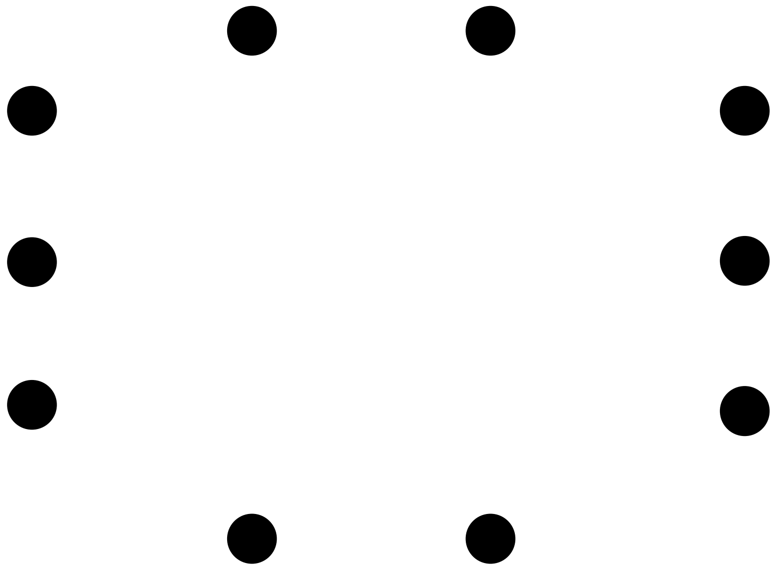
Use in cryptography

The idea is, that it is very difficult to find a Hamiltonian cycle in a big graph.

In general, almost any difficult problem can be used in such a way for saving the privacy of your information.

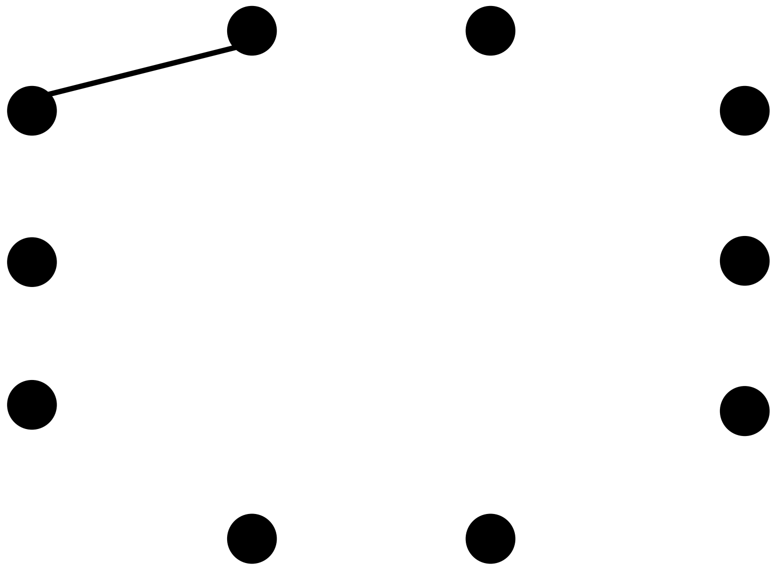
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



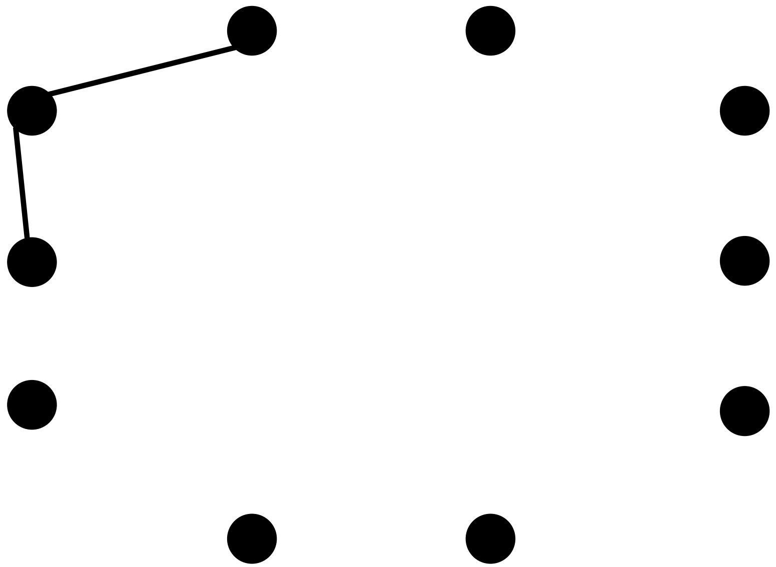
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



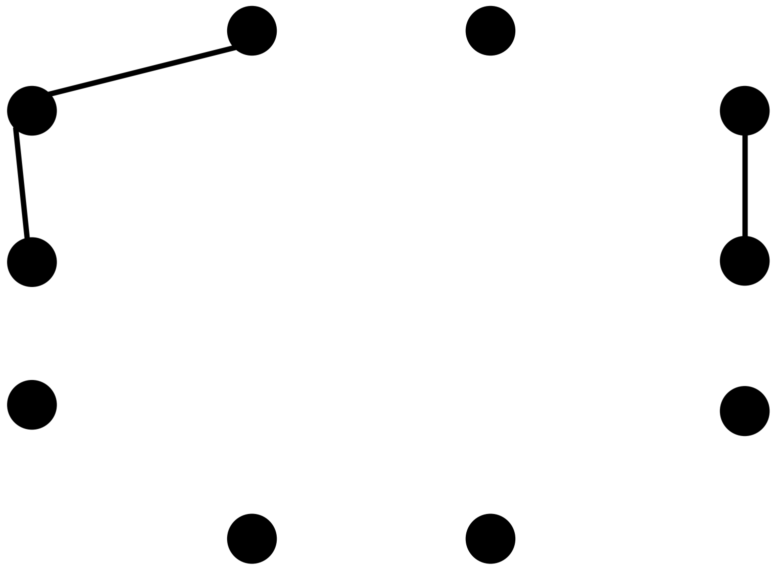
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



A Problem

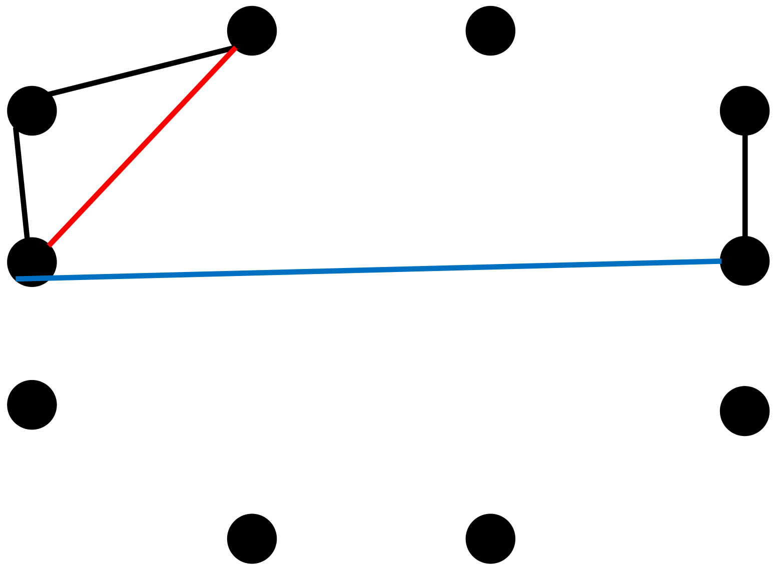
Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?

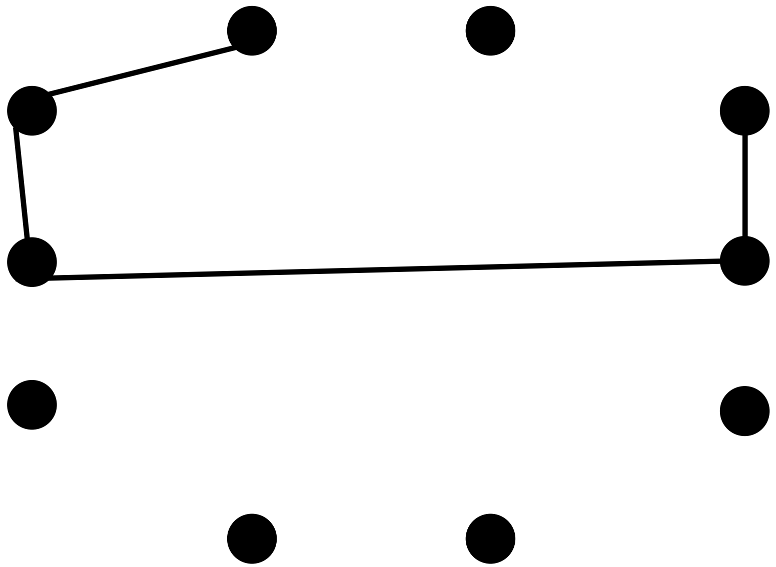
Obviously, when I draw an edge the number of components decreases by 1 or stays the same



For example, if I draw the red edge then the number of components stays the same; if I draw the blue edge then the number of components decreases by 1.

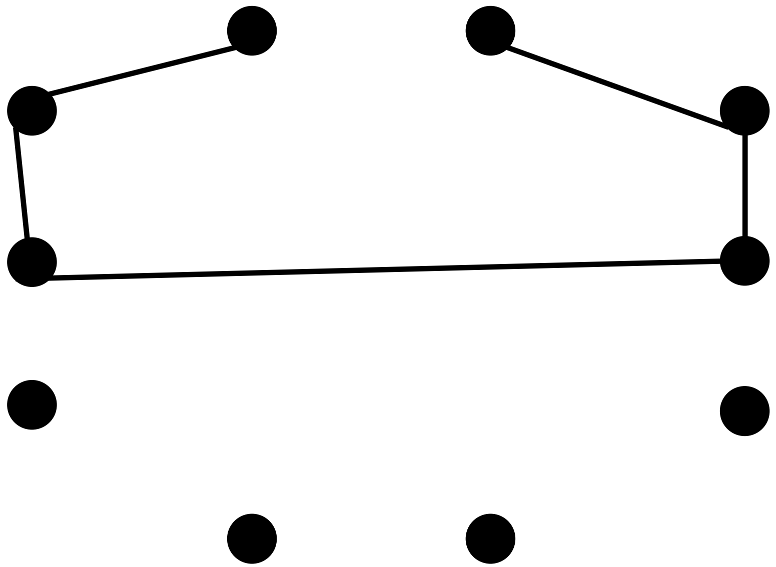
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



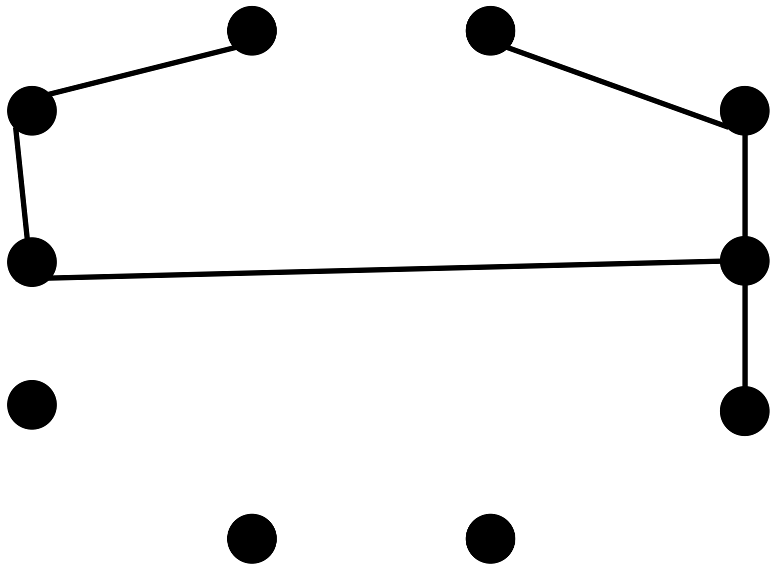
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



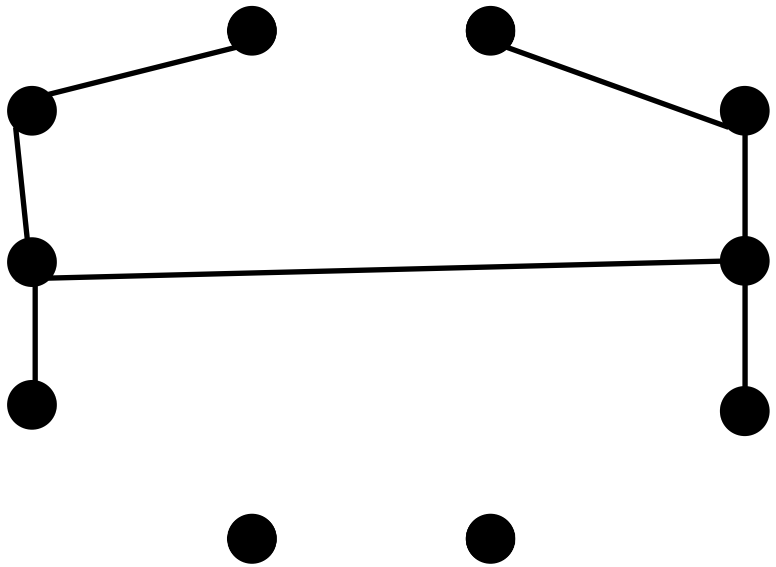
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



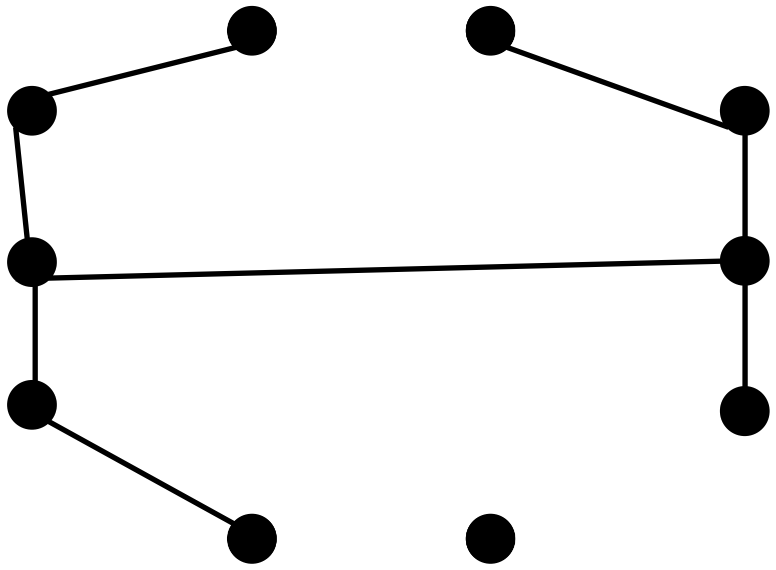
A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



A Problem

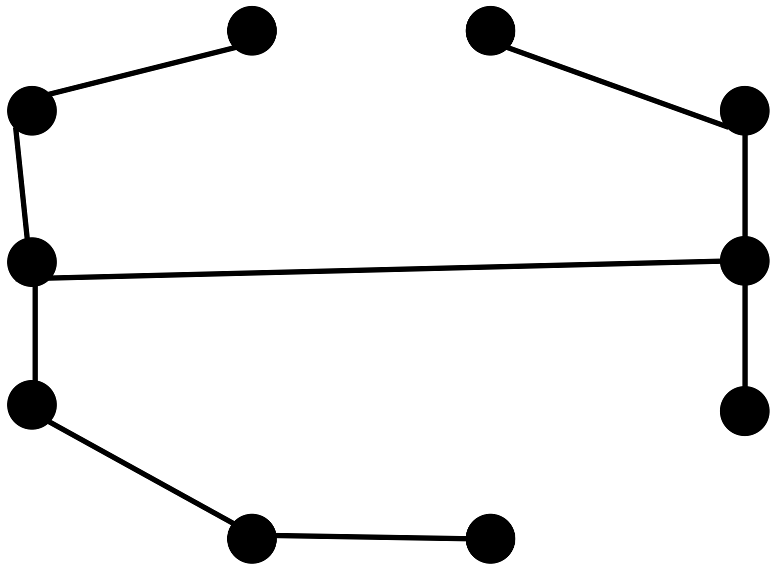
Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?



A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?

So, the smallest number of edges is 99.

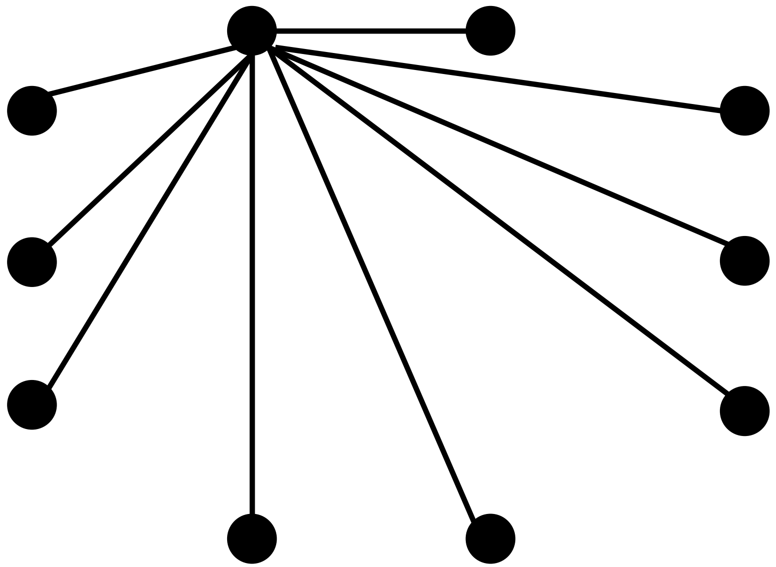


A Problem

Suppose there are 100 cities in a country. What is the smallest number of roads the king can build in such a way that for any two cities *A* and *B* there will be a way to drive from *A* to *B* (maybe throw another cities)?

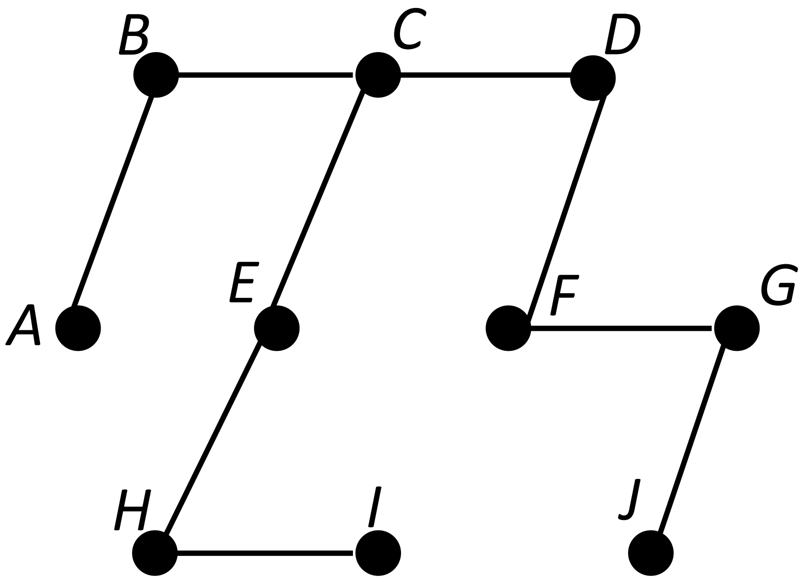
So, the smallest number of edges is 99.

Or simply:



Trees

A connected graph with no cycles is a **tree**.



Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n-1$ edges in it.

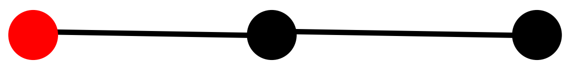
Lemma. Each tree has a vertex of degree 1.

Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

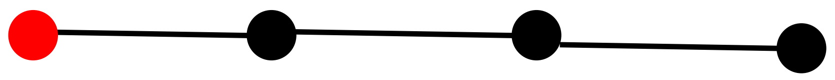


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

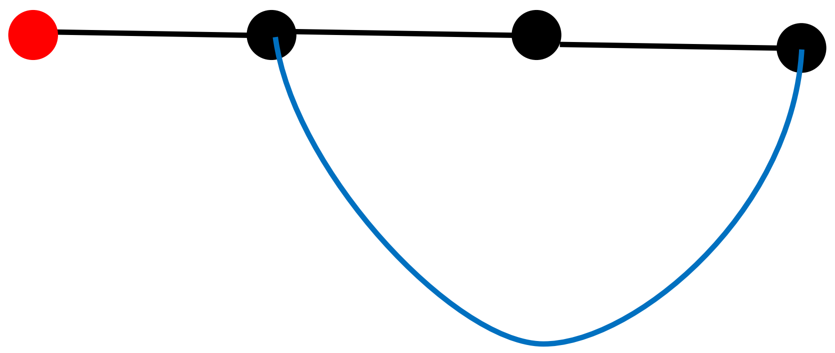


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

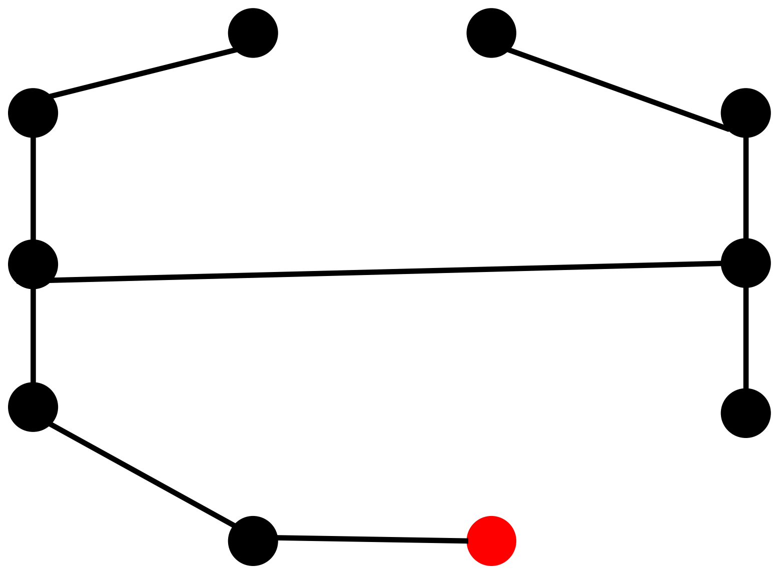


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n-1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

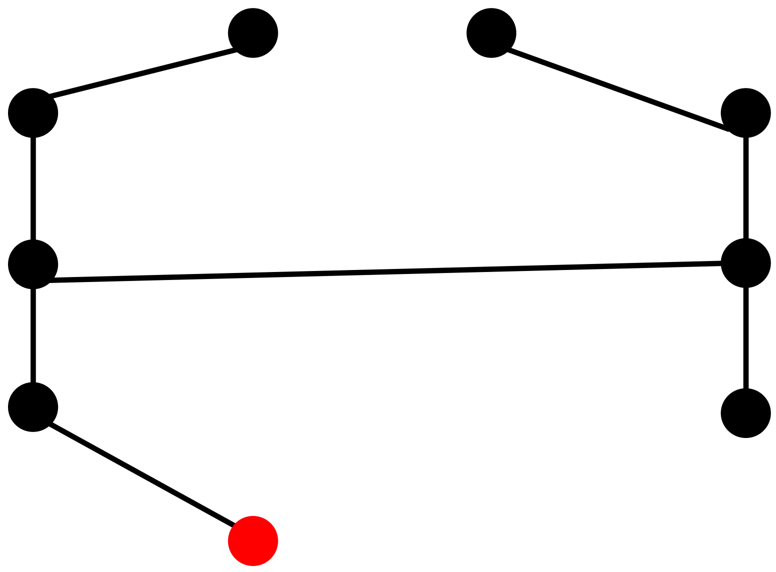


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

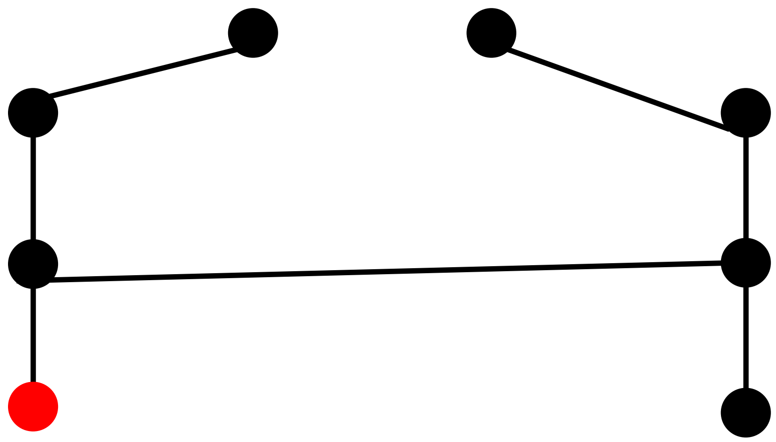


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

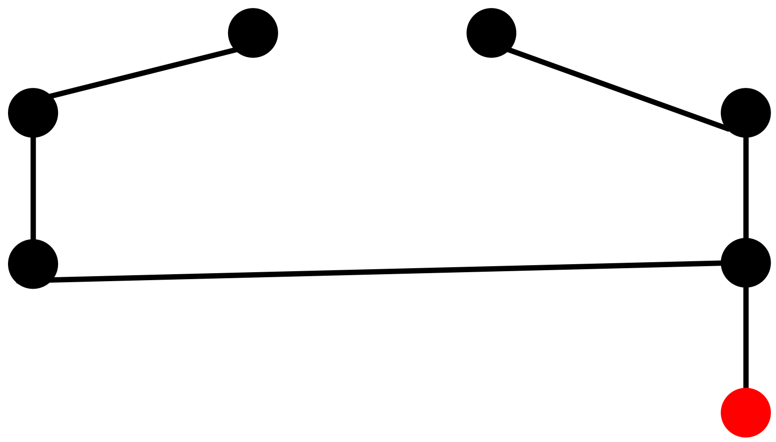


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

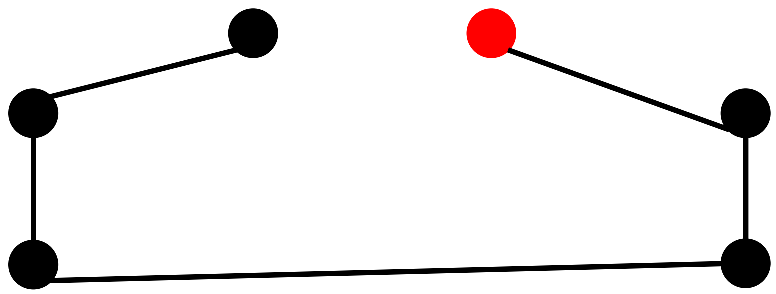


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

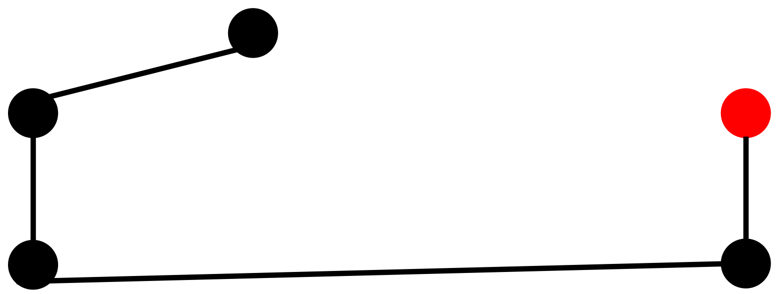


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n-1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

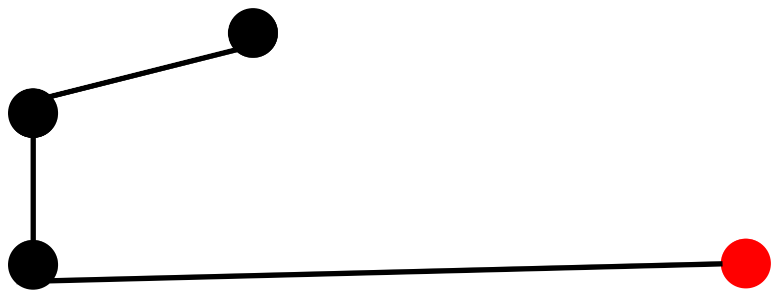


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

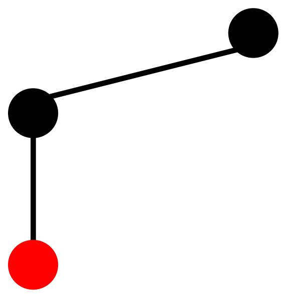


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.

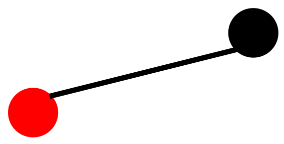


Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Lemma. Each tree has a vertex of degree 1.



Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

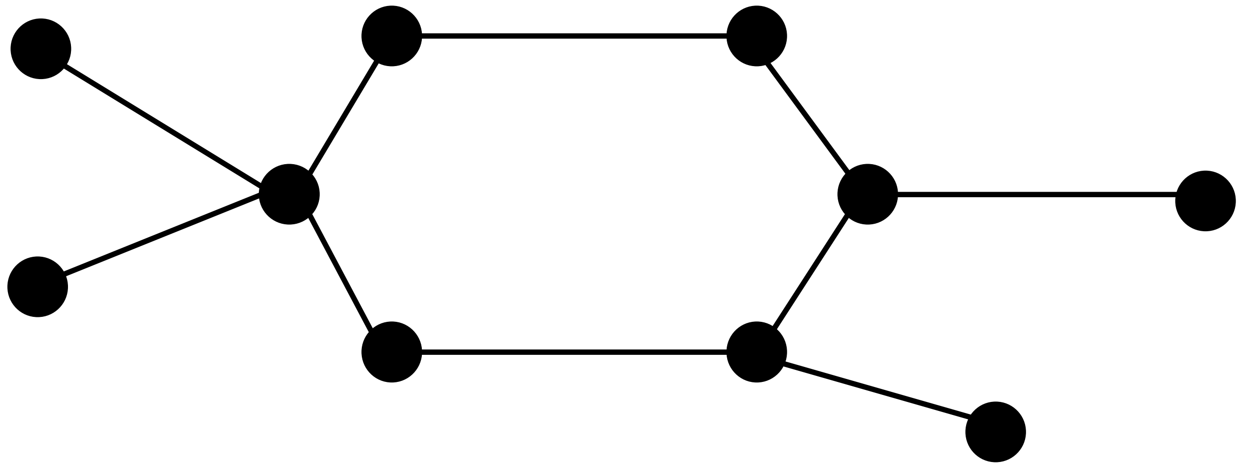
Lemma. Each tree has a vertex of degree 1.

Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Theorem. A connected graph with n vertices and $n - 1$ edges is a tree.



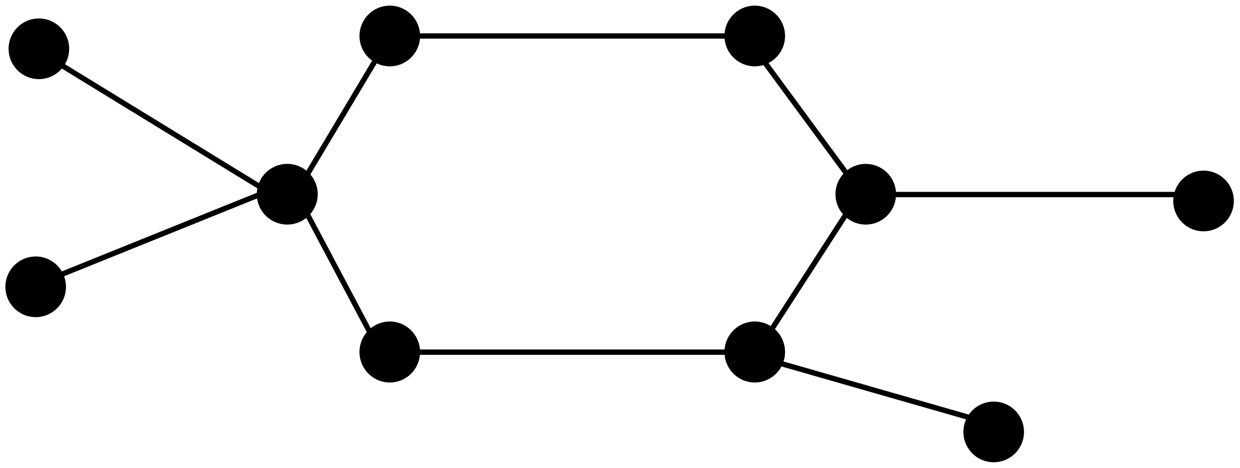
Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n – 1$ edges in it.

Theorem. A connected graph with n vertices and $n – 1$ edges is a tree.

Recall that a graph with n vertices and $n – 2$ (or less) edges cannot be connected.



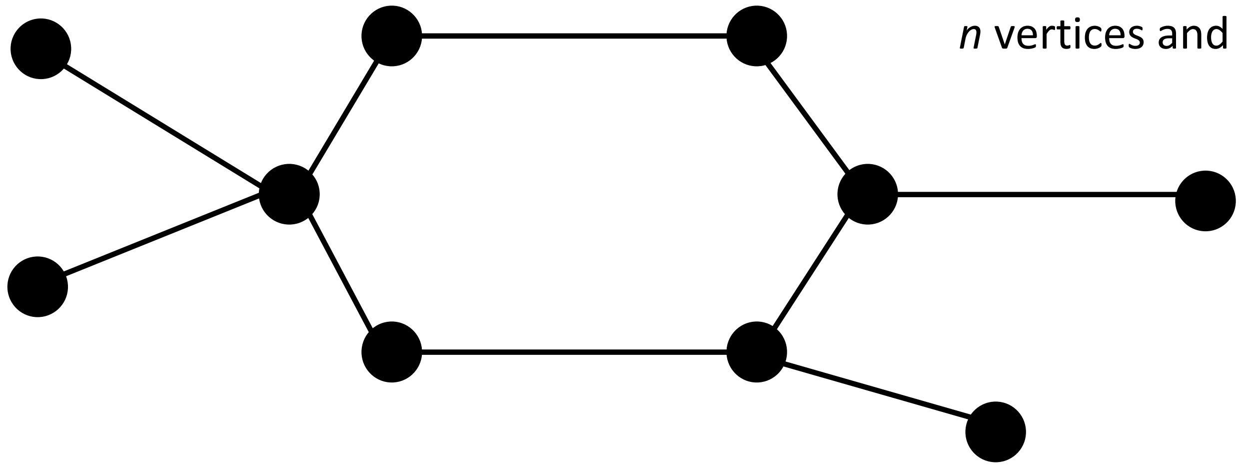
Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

Theorem. A connected graph with n vertices and $n - 1$ edges is a tree.

Recall that a graph with n vertices and $n - 2$ (or less) edges cannot be connected.



n vertices and $n - 1$ edges

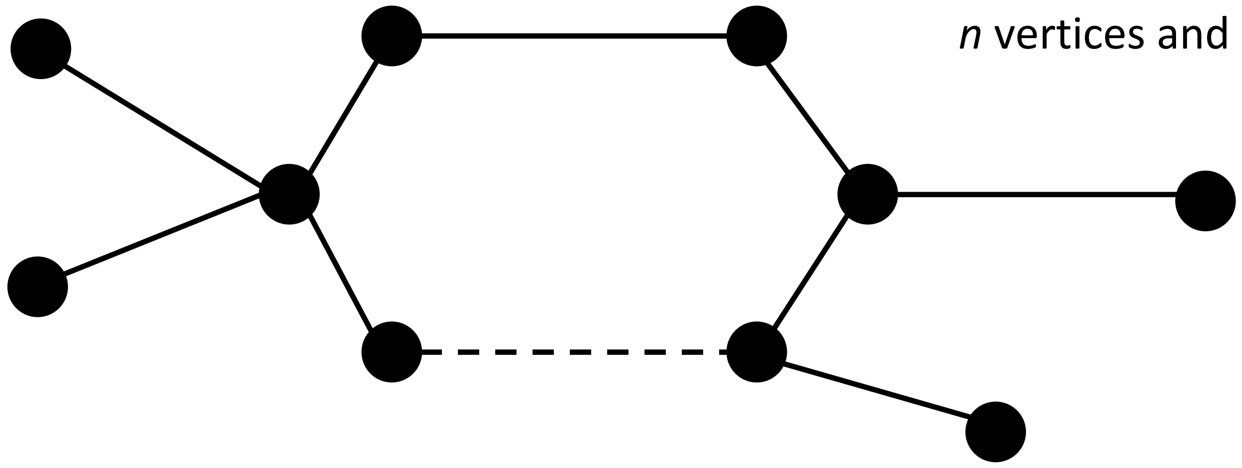
Trees

A connected graph with no cycles is a **tree**.

Theorem. If there are n vertices in a tree, then there are $n - 1$ edges in it.

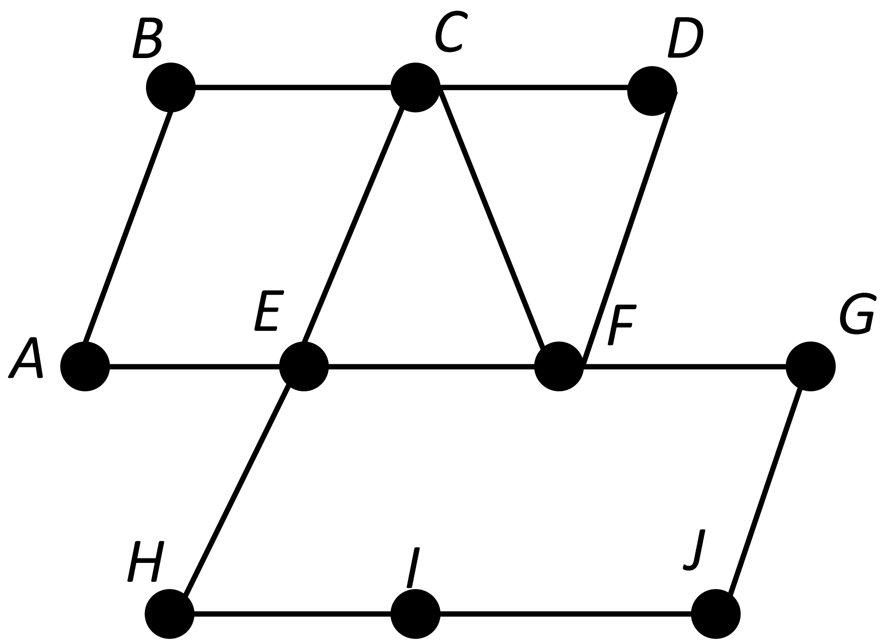
Theorem. A connected graph with n vertices and $n - 1$ edges is a tree.

Recall that a graph with n vertices and $n - 2$ (or less) edges cannot be connected.



n vertices and $n - 2$ edges

Why do we study trees?

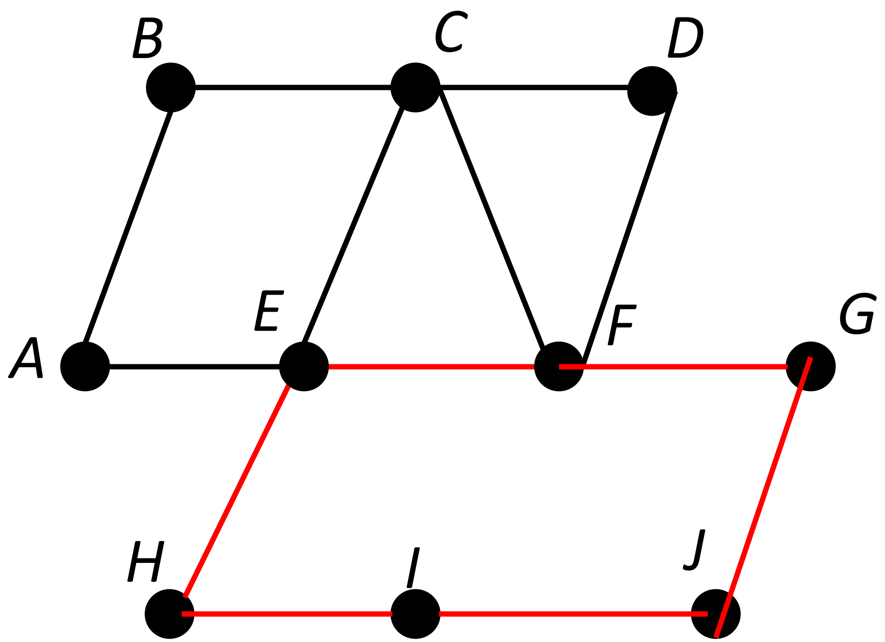


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

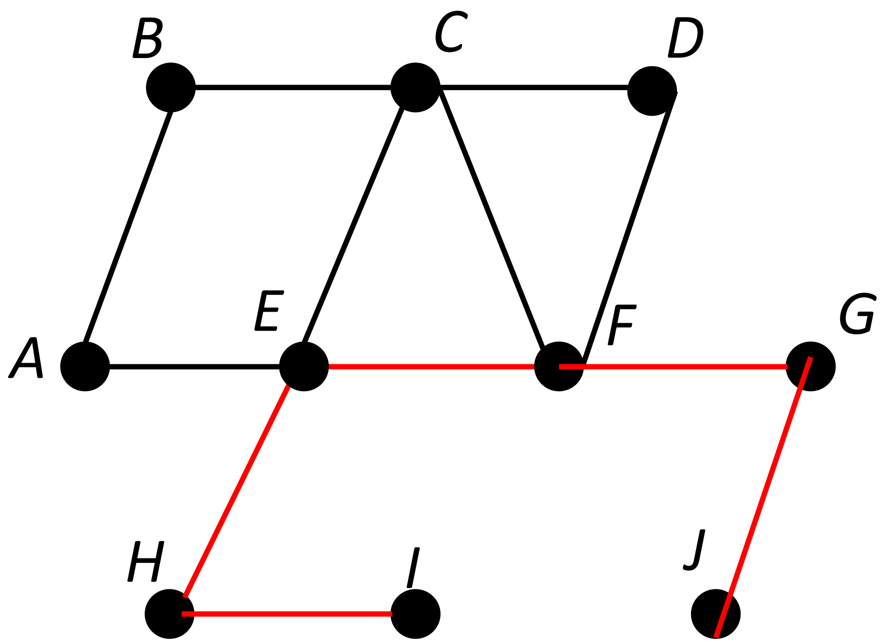


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

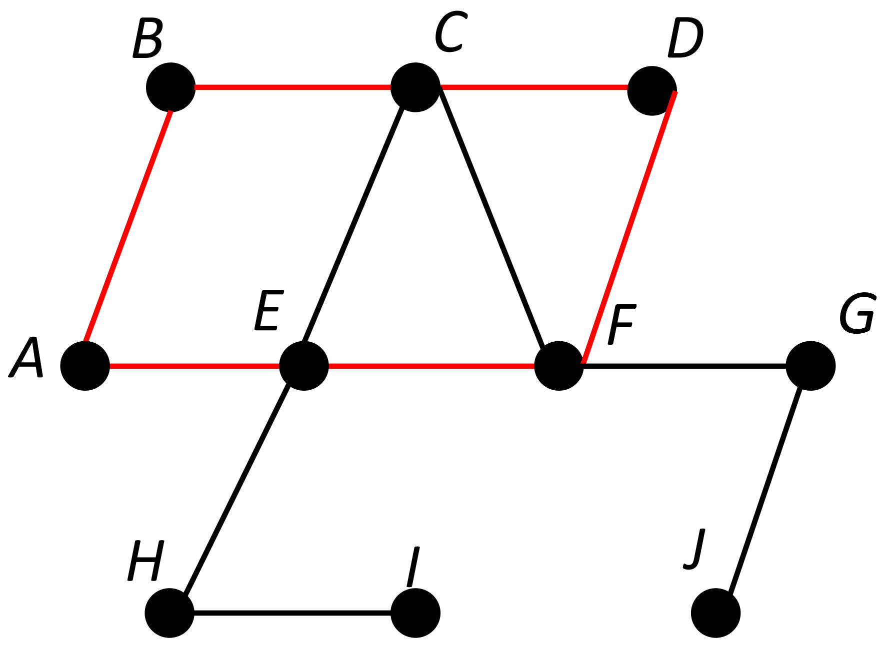


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

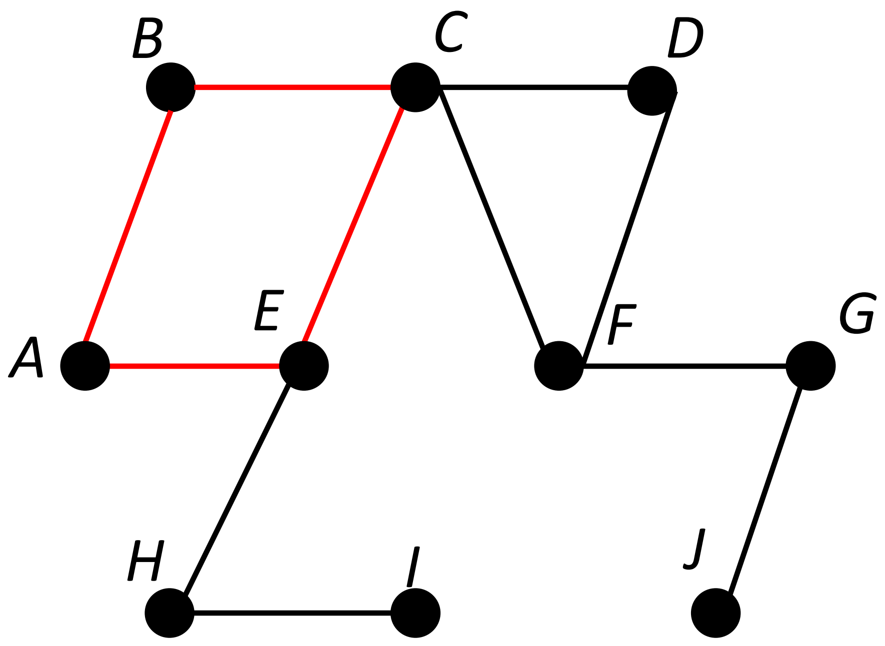


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

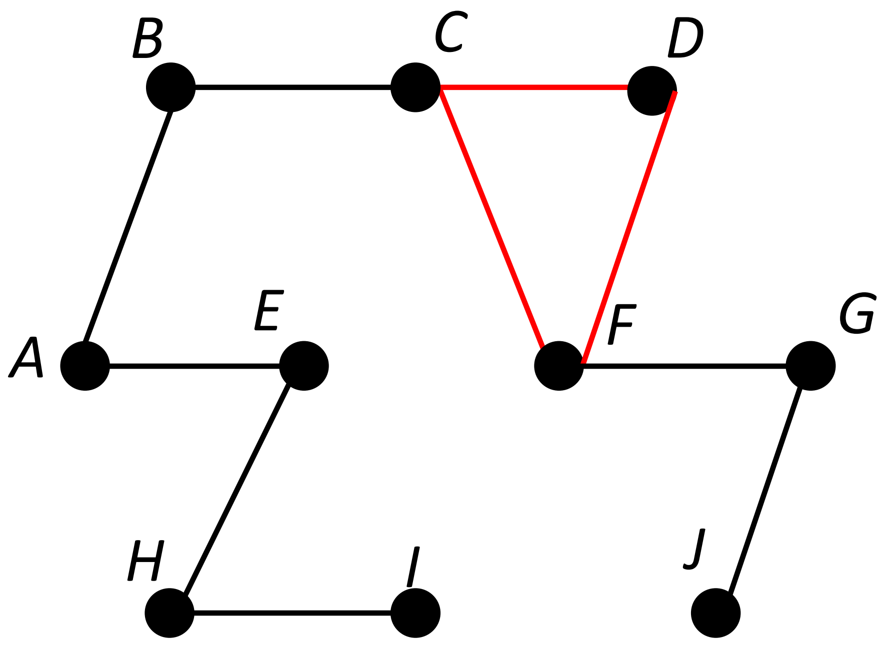


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

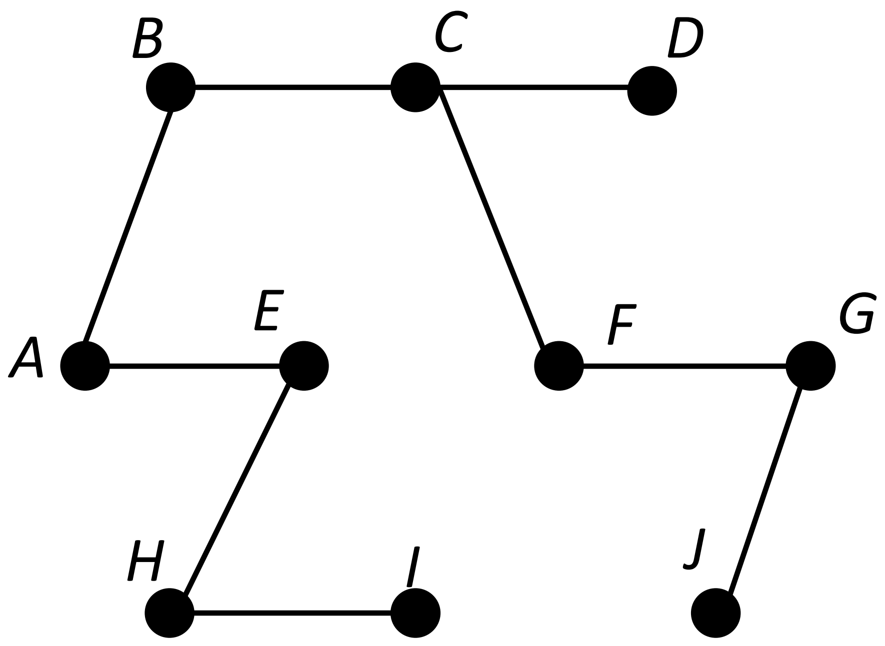


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph G is such a tree which can be obtained from the G by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?

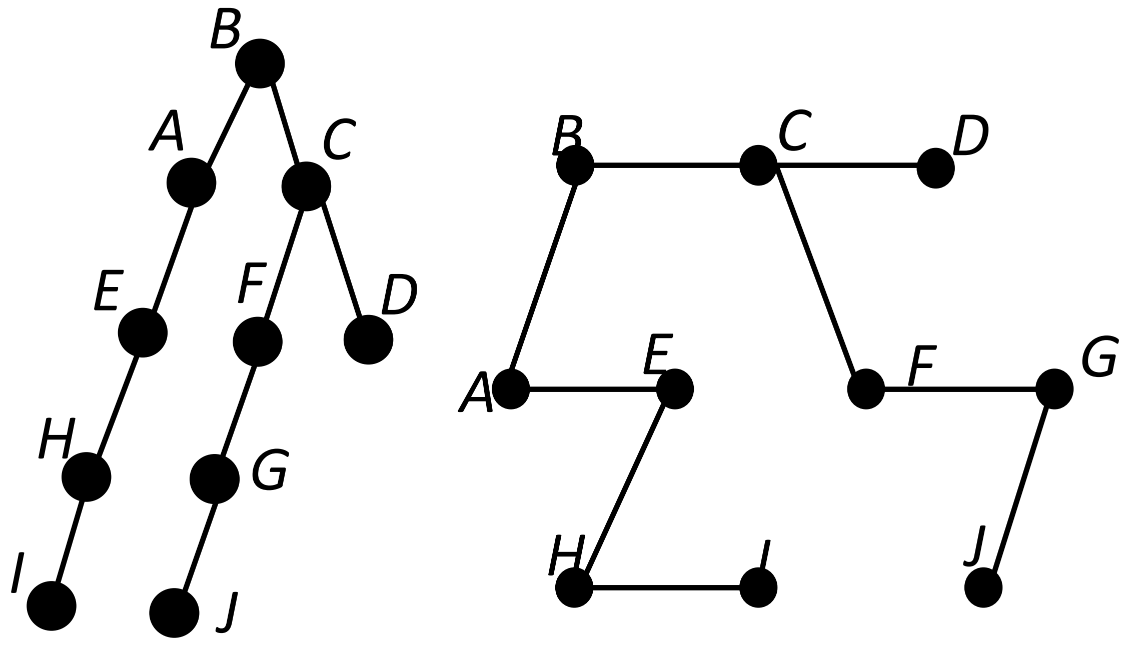


There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph G is such a tree which can be obtained from the G by excluding several edges.

Each connected graph has a spanning tree.

Why do we study trees?



There are 100 towns in a country, some of them are connected by a road. A tourist wants to visit each of the towns. In how many steps he can do it?

A **spanning tree** of the graph *G* is such a tree which can be obtained from the *G* by excluding several edges.

Each connected graph has a spanning tree.