When are two graphs really the same?

Ted Dobson

University of Primorska

January 25, 2023

What is a graph?

A **graph** consists of dots that we will call vertices and lines between them that we will call edges.

What is a graph?

A **graph** consists of dots that we will call vertices and lines between them that we will call edges. Let's look at some examples.

What is a graph?

A **graph** consists of dots that we will call vertices and lines between them that we will call edges. Let's look at some examples.

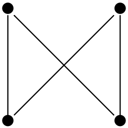


Figure: A graph

What is a graph?

A **graph** consists of dots that we will call vertices and lines between them that we will call edges. Let's look at some examples.

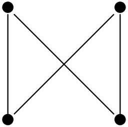


Figure: A graph

There is a more formal definition which gives the vertices of a graph, and then specifies which vertices have a line between them

What is a graph?

A **graph** consists of dots that we will call vertices and lines between them that we will call edges. Let's look at some examples.

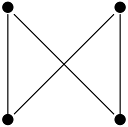


Figure: A graph

There is a more formal definition which gives the vertices of a graph, and then specifies which vertices have a line between them (in a mathy way).

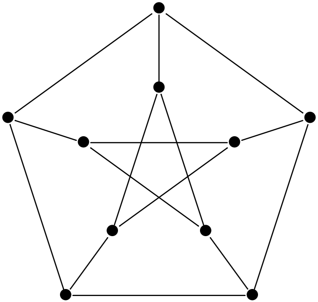


Figure: The Petersen graph

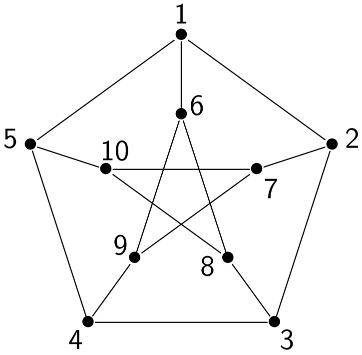


Figure: The Petersen graph with vertices labeled

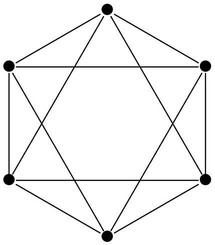


Figure: Octahedron

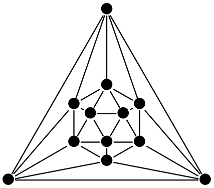


Figure: Icosahedron

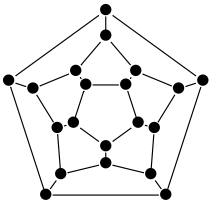


Figure: Dodecahedron

The definition of a graph does not specify how a graph should be drawn, or give any information about what a line looks like.

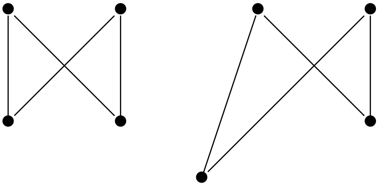


Figure: Two drawings of the same graph

The definition of a graph does not specify how a graph should be drawn, or give any information about what a line looks like.

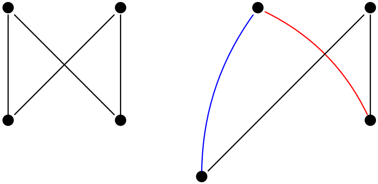


Figure: Two drawings of the same graph

The definition of a graph does not specify how a graph should be drawn, or give any information about what a line looks like.

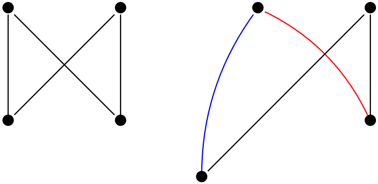


Figure: Two drawings of the same graph

So we can draw an edge in any shape, length or color, and that does not change the graph.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way,

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem. The first way, which we will spend our time on, is how a human would decide if two graphs are really the same.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem. The first way, which we will spend our time on, is how a human would decide if two graphs are really the same. But humans can't really decide with very large graphs, so we need algorithms (computer programs) to have a computer decide.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem. The first way, which we will spend our time on, is how a human would decide if two graphs are really the same. But humans can't really decide with very large graphs, so we need algorithms (computer programs) to have a computer decide. We will very briefly at the end touch on this.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem. The first way, which we will spend our time on, is how a human would decide if two graphs are really the same. But humans can't really decide with very large graphs, so we need algorithms (computer programs) to have a computer decide. We will very briefly at the end touch on this.

Formally, we say that two graphs that are really the same are **isomorphic**.

When are two graphs the same?

We have seen that there are ways to make graphs look different, but that do not really change the graph. We could draw the graph in a different way, or label the vertices differently. There are two ways one can think about this problem. The first way, which we will spend our time on, is how a human would decide if two graphs are really the same. But humans can't really decide with very large graphs, so we need algorithms (computer programs) to have a computer decide. We will very briefly at the end touch on this.

Formally, we say that two graphs that are really the same are **isomorphic**. The roots of this word are Greek, where iso means "same", and morph means "form or shape".

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in "structural properties".

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in "structural properties". What is a structural property?

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in “structural properties”. What is a structural property? Well, it is something that doesn’t change no matter how the graph is drawn or labeled.

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in "structural properties". What is a structural property? Well, it is something that doesn't change no matter how the graph is drawn or labeled. That sounds a lot like asking if the two graphs are the same.

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in “structural properties”. What is a structural property? Well, it is something that doesn’t change no matter how the graph is drawn or labeled. That sounds a lot like asking if the two graphs are the same. But a structural property usually refers not to the whole graph, but some piece of information about the graph or even small parts of the graph itself.

When trying to see if two graphs are really the same, usually the first thing one looks for are differences (if the graphs are really different, or not isomorphic) or similarities (if the graphs are really the same, or isomorphic) in "structural properties". What is a structural property? Well, it is something that doesn't change no matter how the graph is drawn or labeled. That sounds a lot like asking if the two graphs are the same. But a structural property usually refers not to the whole graph, but some piece of information about the graph or even small parts of the graph itself. We will now give some examples. We start with the most basic, which hopefully will be obvious to you, and then look at more complicated things.

Number of edges and vertices

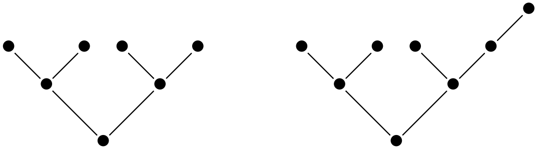
The easiest way to see that two graphs are NOT isomorphic, is if they have different numbers of vertices or edges.

Number of edges and vertices

The easiest way to see that two graphs are NOT isomorphic, is if they have different numbers of vertices or edges. So the graphs below are not isomorphic.

Number of edges and vertices

The easiest way to see that two graphs are NOT isomorphic, is if they have different numbers of vertices or edges. So the graphs below are not isomorphic.



How many pieces is the graph made out of?

I think it is a bit obvious that this is a structural property.

How many pieces is the graph made out of?

I think it is a bit obvious that this is a structural property. So the graphs below are not isomorphic as they have a different number of pieces.

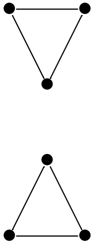


Figure: A graph that is two pieces

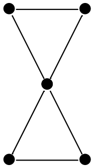


Figure: A graph that is one piece

How many pieces is the graph made out of?

I think it is a bit obvious that this is a structural property. So the graphs below are not isomorphic as they have a different number of pieces.

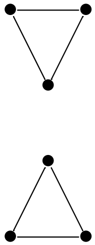


Figure: A graph that is two pieces

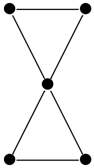


Figure: A graph that is one piece

Notice that the number of vertices of these two graphs are different. These graphs differ in other structural properties, and so there are other reasons why these graphs are not isomorphic.

A definition

A **cycle** in a graph is a sequence of vertices, say, $v_1, v_2, \ldots, v_r, v_1$ which, except for the first and last vertices, are all different, and there is an edge in the graph between successive pairs (i.e. there is an edge between v_1 and v_2, v_2 and v_3, etc., and v_r and v_1). Such a cycle is usually called an r-**cycle**. Here are some examples:

A definition

A **cycle** in a graph is a sequence of vertices, say, $v_1, v_2, \ldots, v_r, v_1$ which, except for the first and last vertices, are all different, and there is an edge in the graph between successive pairs (i.e. there is an edge between v_1 and v_2, v_2 and v_3, etc., and v_r and v_1). Such a cycle is usually called an r-**cycle**. Here are some examples:

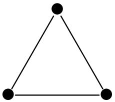


Figure: A 3-cycle or triangle

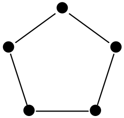


Figure: A 5-cycle or pentagon

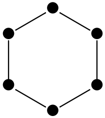


Figure: A 6-cycle or a hexagon

A definition

A **cycle** in a graph is a sequence of vertices, say, $v_1, v_2, \ldots, v_r, v_1$ which, except for the first and last vertices, are all different, and there is an edge in the graph between successive pairs (i.e. there is an edge between v_1 and v_2, v_2 and v_3, etc., and v_r and v_1). Such a cycle is usually called an r-**cycle**. Here are some examples:

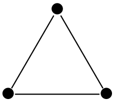


Figure: A 3-cycle or triangle

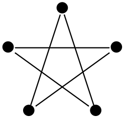


Figure: A 5-cycle or pentagon

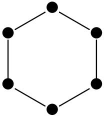


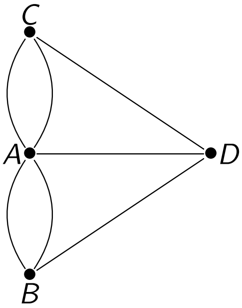
Figure: A 6-cycle or a hexagon

Another definition

The valency is the number of edges at a vertex.

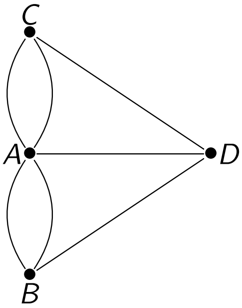
Another definition

The valency is the number of edges at a vertex.



Another definition

The valency is the number of edges at a vertex.



The valency of A is 5, and the valencies of B, C, and D are 3.

Are the valencies the same?

Moving a vertex to a different position in a graph or relabeling it doesn't change its valency.

Are the valencies the same?

Moving a vertex to a different position in a graph or relabeling it doesn't change its valency. We usually describe the valencies of all of the vertices in a graph with a list of all valencies, typically from smallest valency to largest valency.

Are the valencies the same?

Moving a vertex to a different position in a graph or relabeling it doesn't change its valency. We usually describe the valencies of all of the vertices in a graph with a list of all valencies, typically from smallest valency to largest valency. If the lists are the same, the graphs MIGHT be isomorphic. If the lists are not the same, they are not isomorphic.

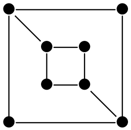


Figure: A graph with degree sequence $(2, 2, 2, 2, 3, 3, 3, 3)$

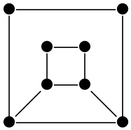


Figure: Another graph with degree sequence $(2, 2, 2, 2, 3, 3, 3, 3)$

Are the valencies the same?

Moving a vertex to a different position in a graph or relabeling it doesn't change its valency. We usually describe the valencies of all of the vertices in a graph with a list of all valencies, typically from smallest valency to largest valency. If the lists are the same, the graphs MIGHT be isomorphic. If the lists are not the same, they are not isomorphic.

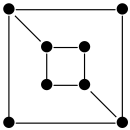


Figure: A graph with degree sequence $(2, 2, 2, 2, 3, 3, 3, 3)$

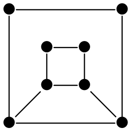


Figure: Another graph with degree sequence $(2, 2, 2, 2, 3, 3, 3, 3)$

These graphs are not isomorphic as on the right hand graph every vertex of degree 3 forms a cycle while on the left hand side they do not.

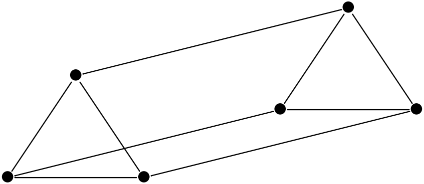
Focus on cycles

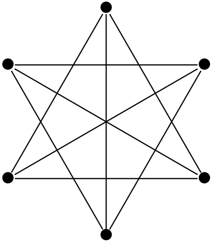
If none of the previous techniques we have discussed give that a pair of graphs are not isomorphic, it is probably time to focus on some small graph that is part of both the graphs and see if this can help. It is usually a good idea to choose something that is not too small (there could be too many of them to do much good) and not too large (they can be hard to find).

Focus on cycles

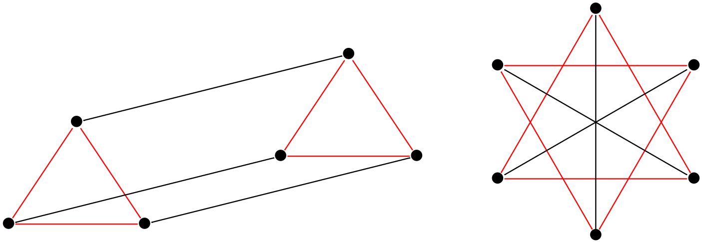
If none of the previous techniques we have discussed give that a pair of graphs are not isomorphic, it is probably time to focus on some small graph that is part of both the graphs and see if this can help. It is usually a good idea to choose something that is not too small (there could be too many of them to do much good) and not too large (they can be hard to find). Cycles are a good feature to focus on.

Are these graphs isomorphic?

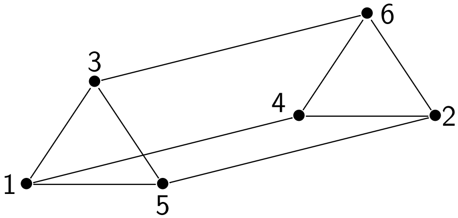


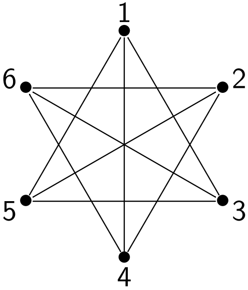


Are these graphs isomorphic?

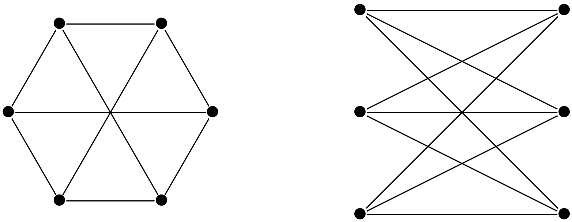


Yes, they are!

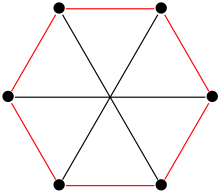


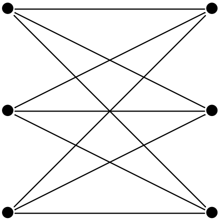


Are these two graphs isomorphic?

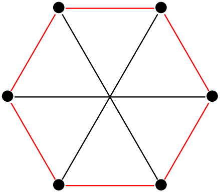


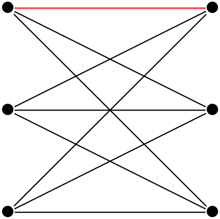
Are these two graphs isomorphic?



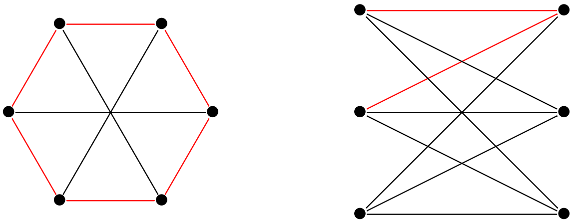


Are these two graphs isomorphic?

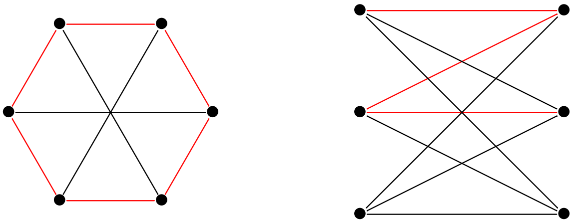




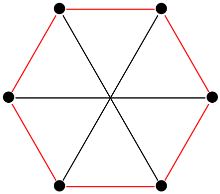
Are these two graphs isomorphic?

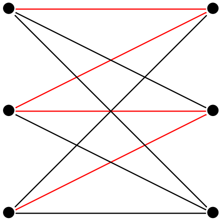


Are these two graphs isomorphic?

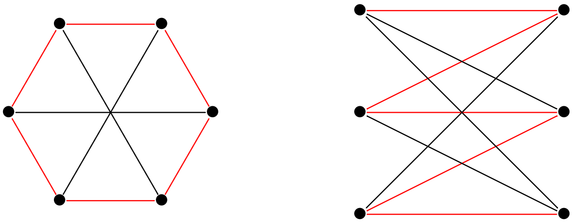


Are these two graphs isomorphic?

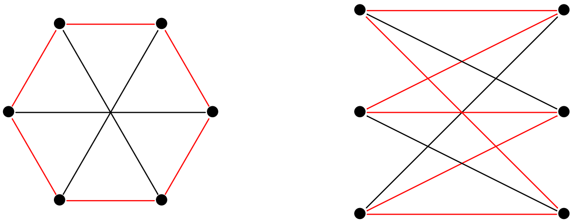




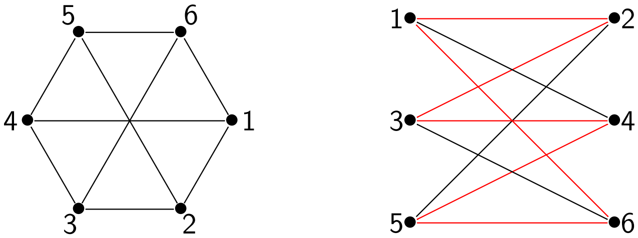
Are these two graphs isomorphic?



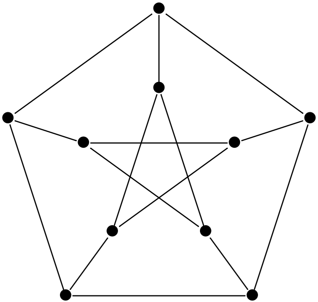
Are these two graphs isomorphic?

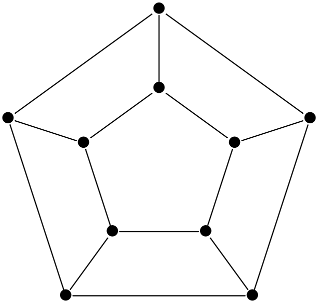


Yes they are!

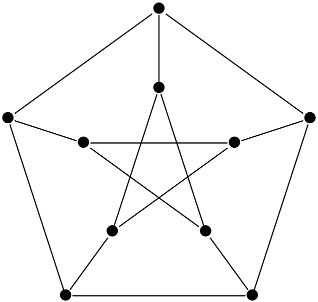


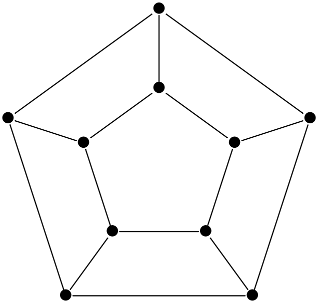
Are these two graphs isomorphic?





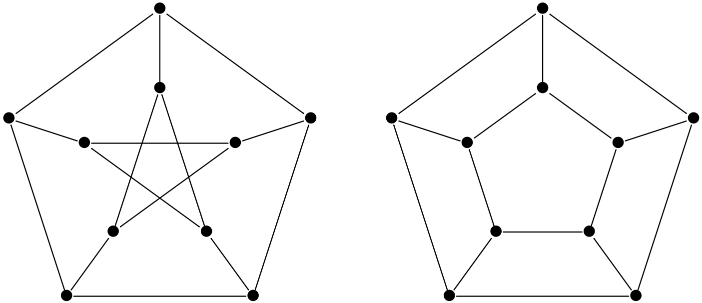
Are these two graphs isomorphic?





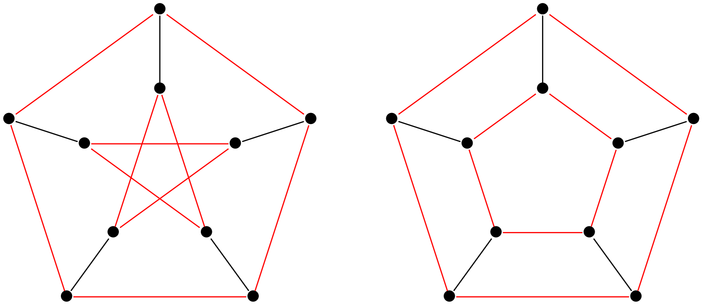
Here, we focus on 5-cycles.

Are these two graphs isomorphic?



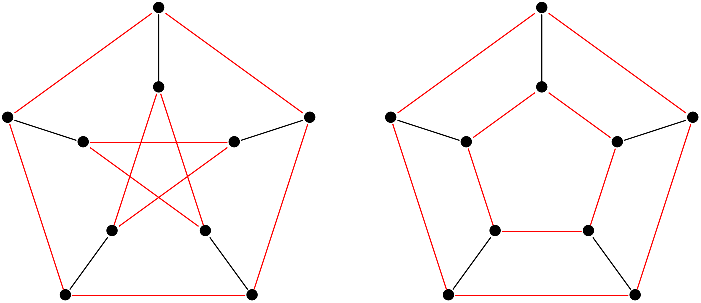
Here, we focus on 5-cycles. Both graphs have an obvious pair of 5-cycles.

Are these two graphs isomorphic?



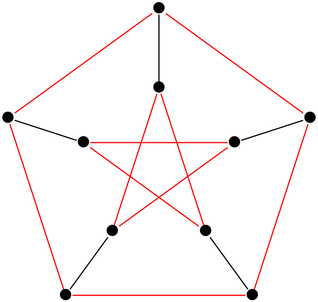
Here, we focus on 5-cycles. Both graphs have an obvious pair of 5-cycles.

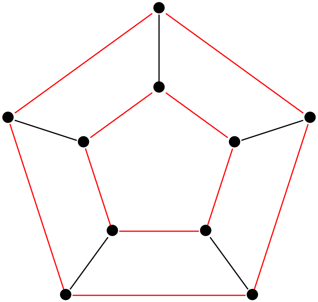
Are these two graphs isomorphic?



Here, we focus on 5-cycles. Both graphs have an obvious pair of 5-cycles. The graph on the right only has two 5-cycles.

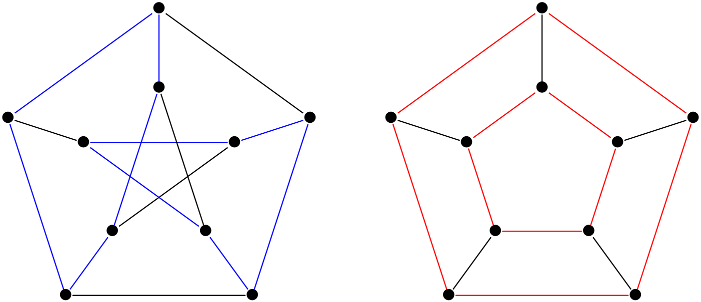
Are these two graphs isomorphic?





Here, we focus on 5-cycles. Both graphs have an obvious pair of 5-cycles. The graph on the right only has two 5-cycles. The Petersen graph on the left has twelve 5-cycles. We show two more.

Are these two graphs isomorphic?



Here, we focus on 5-cycles. Both graphs have an obvious pair of 5-cycles. The graph on the right only has two 5-cycles. The Petersen graph on the left has twelve 5-cycles. We show two more. So these graphs are not isomorphic.

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them.

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them. So the hard part would be if the graphs had NO distinguishing features.

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them. So the hard part would be if the graphs had NO distinguishing features. This would mean that we couldn't really tell the difference between two vertices other than their labels.

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them. So the hard part would be if the graphs had NO distinguishing features. This would mean that we couldn't really tell the difference between two vertices other than their labels. Such a graph is called **vertex-transitive**,

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them. So the hard part would be if the graphs had NO distinguishing features. This would mean that we couldn't really tell the difference between two vertices other than their labels. Such a graph is called **vertex-transitive**, and there is a large research group here at UP studying these graphs.

Where is the hard part?

If you think back to the examples we have seen, to show two graphs are NOT isomorphic, we look for distinguishing features of the graphs, and then compare them. So the hard part would be if the graphs had NO distinguishing features. This would mean that we couldn't really tell the difference between two vertices other than their labels. Such a graph is called **vertex-transitive**, and there is a large research group here at UP studying these graphs. Our intuition is correct in that Eugene Luks of the University of Oregon showed in 1982 that the isomorphism problem can solved quickly if it can be solved quickly for vertex-transitive graphs.

What is the state of the art?

In 2015, Lásló Babai used the framework developed by Luks to show that the graph isomorphism problem could be solved not quickly, but not really slowly (technically, he showed the problem could be solved in quasipolynomial time).

What is the state of the art?

In 2015, Lásló Babai used the framework developed by Luks to show that the graph isomorphism problem could be solved not quickly, but not really slowly (technically, he showed the problem could be solved in quasipolynomial time). It is not known if the problem can be solved quickly (in polynomial time).

What is the state of the art?

In 2015, Lásló Babai used the framework developed by Luks to show that the graph isomorphism problem could be solved not quickly, but not really slowly (technically, he showed the problem could be solved in quasipolynomial time). It is not known if the problem can be solved quickly (in polynomial time). I find it entertaining that the graph isomorphism problem is considered the most important problem in *theoretical computer science*. When I was young, Babai was a mathematician, and now he is in the theoretical computer science program at the University of Chicago.

What is the state of the art?

In 2015, Lásló Babai used the framework developed by Luks to show that the graph isomorphism problem could be solved not quickly, but not really slowly (technically, he showed the problem could be solved in quasipolynomial time). It is not known if the problem can be solved quickly (in polynomial time). I find it entertaining that the graph isomorphism problem is considered the most important problem in *theoretical computer science*. When I was young, Babai was a mathematician, and now he is in the theoretical computer science program at the University of Chicago. The best implemented program for testing graph isomorphism is called NAUTY, and was written by Brendan McKay of the Australian National University.

Thanks!