**4.4Euler Paths and Circuits**[**¶ permalink**](http://discretetext.oscarlevin.com/dmoi/sec_paths.html)

**Investigate!**

An *Euler path*, in a graph or multigraph, is a walk through the graph which uses every edge exactly once. An *Euler circuit* is an Euler path which starts and stops at the same vertex. Our goal is to find a quick way to check whether a graph (or multigraph) has an Euler path or circuit.

1. Which of the graphs below have Euler paths? Which have Euler circuits?
2. List the degrees of each vertex of the graphs above. Is there a connection between degrees and the existence of Euler paths and circuits?
3. Is it possible for a graph with a degree 1 vertex to have an Euler circuits? If so, draw one. If not, explain why not. What about an Euler path?
4. What if every vertex of the graph has degree 2. Is there an Euler path? An Euler circuit? Draw some graphs.
5. Below is *part* of a graph. Even though you can only see some of the vertices, can you deduce whether the graph will have an Euler path or circuit?

If we start at a vertex and trace along edges to get to other vertices, we create a *walk* through the graph. More precisely, a *walk* in a graph is a sequence of vertices such that every vertex in the sequence is adjacent to the vertices before and after it in the sequence. If the walk travels along every edge exactly once, then the walk is called an *Euler path* (or *Euler walk*). If, in addition, the starting and ending vertices are the same (so you trace along every edge exactly once and end up where you started), then the walk is called an *Euler circuit* (or *Euler tour*). Of course if a graph is not connected, there is no hope of finding such a path or circuit. For the rest of this section, assume all the graphs discussed are connected.

The bridges of Königsberg problem is really a question about the existence of Euler paths. There will be a route that crosses every bridge exactly once if and only if the graph below has an Euler path:

This graph is small enough that we could actually check every possible walk that does not reuse edges, and in doing so convince ourselves that there is no Euler path (let alone an Euler circuit). On small graphs which do have an Euler path, it is usually not difficult to find one. Our goal is to find a quick way to check whether a graph has an Euler path or circuit, even if the graph is quite large.

One way to guarantee that a graph does *not* have an Euler circuit is to include a “spike,” a vertex of degree 1.

The vertex *a*

has degree 1, and if you try to make an Euler circuit, you see that you will get stuck at the vertex. It is a dead end. That is, unless you start there. But then there is no way to return, so there is no hope of finding an Euler circuit. There is however an Euler path. It starts at the vertex *a*

, then loops around the triangle. You will end at the vertex of degree 3.

You run into a similar problem whenever you have a vertex of any odd degree. If you start at such a vertex, you will not be able to end there (after traversing every edge exactly once). After using one edge to leave the starting vertex, you will be left with an even number of edges emanating from the vertex. Half of these could be used for returning to the vertex, the other half for leaving. So you return, then leave. Return, then leave. The only way to use up all the edges is to use the last one by leaving the vertex. On the other hand, if you have a vertex with odd degree that you do not start a path at, then you will eventually get stuck at that vertex. The path will use pairs of edges incident to the vertex to arrive and leave again. Eventually all but one of these edges will be used up, leaving only an edge to arrive by, and none to leave again.

What all this says is that if a graph has an Euler path and two vertices with odd degree, then the Euler path must start at one of the odd degree vertices and end at the other. In such a situation, every other vertex *must* have an even degree since we need an equal number of edges to get to those vertices as to leave them. How could we have an Euler circuit? The graph could not have any odd degree vertex as an Euler path would have to start there or end there, but not both. Thus for a graph to have an Euler circuit, all vertices must have even degree.

The converse is also true: if all the vertices of a graph have even degree, then the graph has an Euler circuit, and if there are exactly two vertices with odd degree, the graph has an Euler path. To prove this is a little tricky, but the basic idea is that you will never get stuck because there is an “outbound” edge for every “inbound” edge at every vertex. If you try to make an Euler path and miss some edges, you will always be able to “splice in” a circuit using the edges you previously missed.

**Euler Paths and Circuits**

* A graph has an Euler circuit if and only if the degree of every vertex is even.
* A graph has an Euler path if and only if there are at most two vertices with odd degree.

Since the bridges of Königsberg graph has all four vertices with odd degree, there is no Euler path through the graph. Thus there is no way for the townspeople to cross every bridge exactly once.

**Hamilton Paths**[**¶ permalink**](http://discretetext.oscarlevin.com/dmoi/sec_paths.html#subsection-40)

Suppose you wanted to tour Königsberg in such a way where you visit each land mass (the two islands and both banks) exactly once. This can be done. In graph theory terms, we are asking whether there is a path which visits every vertex exactly once. Such a path is called a *Hamilton path* (or *Hamiltonian path*). We could also consider *Hamilton cycles*, which are Hamliton paths which start and stop at the same vertex.

**Example4.4.1**

Determine whether the graphs below have a Hamilton path.

It appears that finding Hamilton paths would be easier because graphs often have more edges than vertices, so there are fewer requirements to be met. However, nobody knows whether this is true. There is no known simple test for whether a graph has a Hamilton path. For small graphs this is not a problem, but as the size of the graph grows, it gets harder and harder to check wither there is a Hamilton path. In fact, this is an example of a question which as far as we know is too difficult for computers to solve; it is an example of a problem which is NP-complete.

**4.4Exercises**[**¶ permalink**](http://discretetext.oscarlevin.com/dmoi/sec_paths.html#exercises_gt-paths)

**1**

You and your friends want to tour the southwest by car. You will visit the nine states below, with the following rather odd rule: you must cross each border between neighboring states exactly once (so, for example, you must cross the Colorado-Utah border exactly once). Can you do it? If so, does it matter where you start your road trip? What fact about graph theory solves this problem?

**2**

Which of the following graphs contain an Euler path? Which contain an Euler circuit?

1. *K*4

  *K*5

 .

 *K*5,7

  *K*2,7

  *C*7

  *P*7

**3**

Edward A. Mouse has just finished his brand new house. The floor plan is shown below:

1. Edward wants to give a tour of his new pad to a lady-mouse-friend. Is it possible for them to walk through every doorway exactly once? If so, in which rooms must they begin and end the tour? Explain.
2. Is it possible to tour the house visiting each room exactly once (not necessarily using every doorway)? Explain.
3. After a few mouse-years, Edward decides to remodel. He would like to add some new doors between the rooms he has. Of course, he cannot add any doors to the exterior of the house. Is it possible for each room to have an odd number of doors? Explain.

**4**

For which *n*

does the graph *Kn*

contain an Euler circuit? Explain.

**5**

For which *m*

and *n* does the graph *Km*,*n*

contain an Euler path? An Euler circuit? Explain.

**6**

For which *n*

does *Kn*

contain a Hamilton path? A Hamilton cycle? Explain.

**7**

For which *m*

and *n* does the graph *Km*,*n*

contain a Hamilton path? A Hamilton cycle? Explain.

**8**

A bridge builder has come to Königsberg and would like to add bridges so that it *is* possible to travel over every bridge exactly once. How many bridges must be built?

**9**

Below is a graph representing friendships between a group of students (each vertex is a student and each edge is a friendship). Is it possible for the students to sit around a round table in such a way that every student sits between two friends? What does this question have to do with paths?

**10**

1. Suppose a graph has a Hamilton path. What is the maximum number of vertices of degree one the graph can have? Explain why your answer is correct.
2. Find a graph which does not have a Hamilton path even though no vertex has degree one. Explain why your example works.

**11**

Consider the following graph:

1. Find a Hamilton path. Can your path be extended to a Hamilton cycle?
2. Is the graph bipartite? If so, how many vertices are in each “part”?
3. Use your answer to part (b) to prove that the graph has no Hamilton cycle.
4. Suppose you have a bipartite graph *G*

in which one part has at least two more vertices than the other. Prove that *G* does not have a Hamilton path.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Euler and Hamiltonian Paths**

* [Euler Paths and Circuits](https://www.cs.sfu.ca/~ggbaker/zju/math/euler-ham.html#euler)
* [Hamilton Paths and Circuits](https://www.cs.sfu.ca/~ggbaker/zju/math/euler-ham.html#ham)
* [Travelling Salesman](https://www.cs.sfu.ca/~ggbaker/zju/math/euler-ham.html#tsp)

**Euler Paths and Circuits**

* An *Euler circuit* (or *Eulerian circuit*) in a graph *G* is a simple circuit that contains every edge of *G*.
	+ Reminder: a simple circuit doesn't use the same edge more than once.
	+ So, a circuit around the graph passing by every edge exactly once.
	+ We will allow simple or multigraphs for any of the Euler stuff.
* Euler circuits are one of the oldest problems in graph theory.
	+ People wanted to walk around the city of Königsberg and cross every bridge as they went.
	+ … and end up back at home.
	+ The problem turns into one about graphs if you let each bridge be an edge and each island/land mass be a vertex.
	+ And Euler solved it, so he gets his name on it.
* In the modern world: you want to walk around the mall without missing any stores, or wasting time walking the same hall again.
* For example, the first graph has an Euler circuit, but the second doesn't.
	+ Note: you're allowed to use the same **vertex** multiple times, just not the same edge.
* An *Euler path* (or *Eulerian path*) in a graph *G* is a simple path that contains every edge of *G*.
	+ The same as an Euler circuit, but we don't have to end up back at the beginning.
	+ The other graph above does have an Euler path.
* *Theorem:* A graph with an Eulerian circuit must be connected, and each vertex has even degree.

*Proof:* If it's not connected, there's no way to create a circuit.

When the Eulerian circuit arrives at an edge, it must also leave. This visits two edges on the vertex. When it returns to its starting point, it has visited an even number of edges at each vertex.∎

* *Theorem:* A connected graph with even degree at each vertex has an Eulerian circuit.

*Proof:* We will show that a circuit exists by actually building it for a graph with |*V*|=*n*. For *n*=2, the graph must be two vertices connected by two edges. It has an Euler circuit.

For *n*>2, pick a vertex *v* as a starting point. Pick an arbitrary edge leaving *v*. Continue to pick edges and walk around the graph until you return to *v*. We know we'll never get stuck since every vertex has even degree: if we walk in, then there's a way to walk out.

This process forms part of our circuit. Let *Ev* be the set of edges visited in our initial loop.

Consider the graph with edges *E*−*Ev* and whatever vertices still have an edge adjacent. Each vertex in this graph has even degree (since we removed an even number from each) and it has less than *n* edges. By strong induction, we can find an Euler circuit for each connected component of this graph.

Since our graph was connected originally, each of these sub-circuits shares a vertex with out *Ev* walk. We can join these together at the shared vertex to form a circuit of all edges in *G*.∎

* An example will help. Suppose we have the graph below start at *b* and find the initial walk highlighted.
	+ That leaves us with this two-component graph to apply the inductive hypothesis to:
	+ So we find a Euler circuit in each component:
	+ We combine to form a Euler circuit of the original by following one of the component-circuits whenever we can:
* *Corollary:* A graph has an Eulerian circuit **if and only if** it is connected and all of its vertices have even degree.
* *Corollary:* A graph has an Eulerian path **but no Eulerian circuit** if and only if it has exactly two vertices with odd degree.

*Proof:* If we add an edge between the two odd-degree vertices, the graph will have an Eulerian circuit. If we remove the edge, then what remains is an Eulerian path.

Suppose a graph with a different number of odd-degree vertices has an Eulerian path. Add an edge between the two ends of the path. This is a graph with an odd-degree vertex and a Euler circuit. As the above theorem shows, this is a contradiction.∎

* The Euler circuit/path proofs imply an algorithm to find such a circuit/path.
	+ It will take Θ(|*E*|) running time: we end up traversing each edge once in the “find random cycle” phase, and again when joining cycles.
	+ I don't think we can hope for better than that.

**Hamilton Paths and Circuits**

* The Euler circuits and paths wanted to use every edge exactly once.
* It seems obvious to then ask: can we make a circuit of a graph using every **vertex** exactly once?
	+ Such a circuit is a *Hamilton circuit* or *Hamiltonian circuit*.
	+ Similarly, a path through each vertex that doesn't end where it started is a *Hamilton path*.
* It seems like finding a Hamilton circuit (or conditions for one) should be more-or-less as easy as a Euler circuit.
	+ Unfortunately, it's much harder.
* For example, the two graphs above have Hamilton paths but not circuits:
	+ … but I have no obvious proof that they don't.
* Somehow, it feels like if there “enough” edges, then we should be able to find a Hamiltonian cycle. The following two theorem give us some good-enough conditions.
* *Theorem:* (Ore's Theorem) In a graph with *n*≥3 vertices, if for each pair of vertices either deg(*u*)+deg(*v*)≥*n* or *u* and *v* are adjacent, then the graph has a Hamilton circuit.

*Proof idea:* Suppose there is any graph that had this property but no Hamilton cycle. Consider such a graph that has as the maximum number of edges without having a Hamilton cycle. Such a graph must have a Hamilton path: if not, we could add more edges without creating a cycle.

By the pigeonhole principle, there must be vertices adjacent to the ends of the path in such a way that we can construct a circuit. [Google “Ore's Theorem” for details of the proof if you're interested.]

* *Corollary:* (Dirac's Theorem) In a graph with *n*≥3 vertices, if each vertex has deg(*v*)≥*n*/2, then the graph has a Hamilton circuit.

*Proof:* If a graph has deg(*v*)≥*n*/2 for each vertex, then it meets the criteria for Ore's theorem, and thus has a Hamilton cycle.∎

* Note that these conditions are sufficient but not necessary: there are graphs that have Hamilton circuits but do not meet these conditions.
	+ *C*6 for example (cycle with 6 vertices): each vertex has degree 2 and 2<6/2, but there is a Ham cycle.
* There are no nice necessary-and-sufficient conditions known for a graph to have a Hamilton circuit.
	+ Just a few more that give imperfect conditions on one side or the other.
* There is also no good algorithm known to find a Hamilton path/cycle.
	+ The most obvious: check every one of the *n*! possible permutations of the vertices to see if things are joined up that way.
	+ There are known algorithms with running time *O*(*n*22*n*) and *O*(1.657*n*).
	+ Either way, they're exponential, so we're not going to come up with a solution for a large graph.
	+ There's no proof that no non-exponential algorithm exists, either.
* Once again: the “we just don't know” boundary has some very elementary-sounding problems in computing and discrete math.
	+ It's interesting that such similar-sounding problems have such different difficulty.