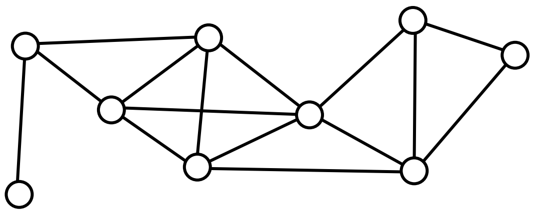
Topics in Discrete Mathematics
Part 2: Introduction to Graph Theory

Lecture 2
Paths, Circuits, and Cycles

- Definitions: walk, trail, path, closed walk, circuit, cycle
- Connectedness
- Eulerian circuits, and their existence in a connected graph iff all degrees even
- Hamiltonian cycles and Dirac's theorem
- Trees

2.1 Defintions

Think of this graph as denoting some towns linked together by roads.



Another example might be sending a packet of information around a computer network.

A natural thing to do might be to walk from town to town along the edges of the graph.

Definition 2.1. A *walk* of length k from $v_0 \in V$ to $v_k \in V$ is a sequence of vertices $v_0v_1v_2 \cdots v_{k-1}v_k$ such that the adjacent pairs $v_0v_1, v_1v_2, \dots, v_{k-1}v_k$ are all edges.

A *trail* is a walk with all edges distinct.

A *path* is a walk with all vertices (and hence all edges) distinct.

In the example of the walk around towns, it seems natural for the walker to want to end up back where she started.

Definition 2.2. A *closed walk* is a walk $v_0v_1v_2 \cdots v_{k-1}v_0$ from a vertex v_0 back to itself.

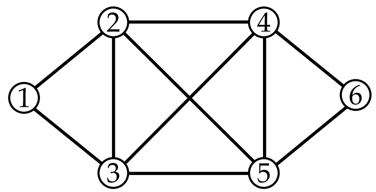
A *circuit* is a trail from a vertex back to itself. Equivalently, a circuit is a closed walk with all edges distinct.

A *cycle* is a path from a vertex back to itself (so the first and last vertices are not distinct). Equivalently, a cycle is a closed walk with all vertices (and hence all edges) distinct (except the first and last vertices).

To summarize these definitions:

	Any reuse	Don't reuse edges	Don't reuse vertices (or edges)
Start and end anywhere	walk	trail	path
Start and end at the same place	closed walk	circuit	cycle

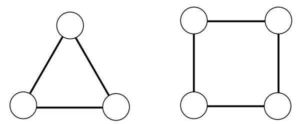
So in this picture



- 124523 is walk and a trail, but not a path;
- 124231 is a walk and a closed walk;
- 1231 is a walk, trail, closed walk, circuit and cycle.

2.2 Connectivity

Last time, we saw this graph



and noted that this seemed to be more like two graphs than one.

With our new definitions, we can formalise way they seem separate: it's that there isn't a walk from a vertex on the left K_3 to the right C_4. When this happens, we say that the graph is not *connected*.

Definition 2.3. Consider a graph $G = (V, E)$.

For two vertices $u, v \in V$, we write $u \to v$ if there exists a walk from u to v.

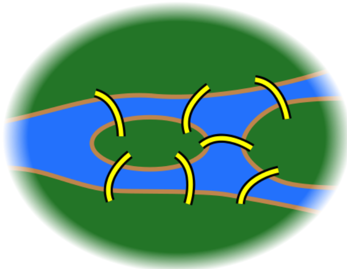
If $u \to v$ for every pair of vertices $u, v \in V$, we say that G is *connected*

It's easy to check that $\to$ is an equivalence relation on V. This means that $\to$ partitions V into what we call the *connected components*.

So the graph above has two connected components – the K_3 and the C_4.

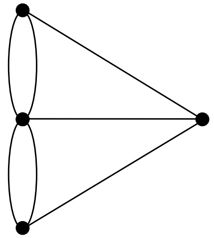
Often in this course, we shall restrict ourselves to connected graphs, knowing that for a disconnected graph we can direct our attention to each of the connected components separately.

2.3 Eulerian circuits

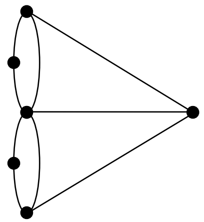


Above is a picture (and a cartoon) of the city of Königsberg (now called Kalin-ingrad) in Russia, as it stood in the 18th century, with its famous seven bridges over the river Pregel. The story has it that the Königsbergers wanted to be able to go for an afternoon walk that would take them over each bridge exactly once and end up back at the house they started from. However as they discovered – and as you might notice from the picture above – it seemed difficult to manage this. So they called in Leonhard Euler, the greatest mathematician of the time (or perhaps any time) to help solve the problem.

Euler then invented graph theory, by noting that Königsberg could be represented by a graph where the landmasses are vertices and the bridges edges.



(In fact this graph has multiple edges, which we said weren't allowed in this course. You can either check that the results in this section also apply to multigraphs, or cunningly subdivide two of the edges by adding extra vertices, which clearly doesn't change the problem.)



So a solution to the Königsberg problem would be a circuit (which can use each edge at most once and must end where it start) that uses every edge (necessarily exactly once).

Definition 2.4. An *Eulerian circuit* on a graph is a circuit that uses every edge.

What Euler worked out is that there is a very simple necessary and sufficient condition for an Eulerian circuit to exist.

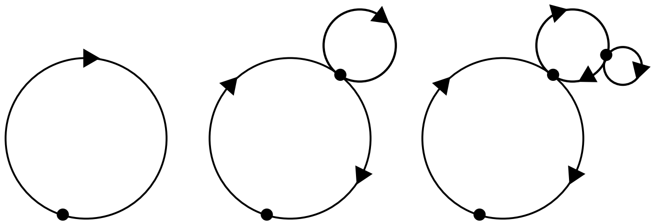
Theorem 2.5. *A graph $G = (V, E)$ has an Eulerian circuit if and only if G is connected and every vertex $v \in V$ has even degree $d(v)$.*

Note that the Königsberg graph has four vertices of odd degree, so no Eulerian circuit exists.

Proof. (*Only if*) Clearly G must be connected. Note that everytime the Eulerian circuit visits a vertex v, it uses one edge to enter and other to exit. Hence, the degree $d(v)$ must be twice the number of visits to v, an even number.

Before beginning the 'if' part of the proof, we outline the idea of the proof. Given a connected graph with all even degrees, we want to construct an Eulerian circuit. We will do this as follows:

Pick a starting vertex, and pick any circuit back to that vertex. If this is Eulerian, we are done. Otherwise, we can find another circuit that intersects a first one, and splice the second circuit into the first to make a new, larger circuit. Again, if this is Eulerian, we're done; else we splice in another circuit. Continuing like this, we eventually get an Eulerian circuit.



We just need to check that this process will always work.

Proof continued. (*If*) We will construct an Eulerian circuit.

Pick an arbitrary starting vertex v_1, and choose a circuit $C_1 = D_1$ starting from and returning to v_1. Note that we can indeed do this: since all vertices are even, we can exit any vertex we enter, and since the graph is finite, we must eventually make our way back to v_1.

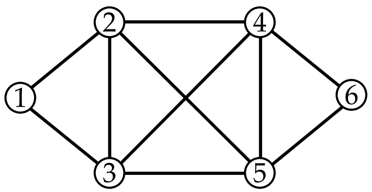
If C_1 is Eulerian, we are done. Else, choose a vertex v_2 on C_1 that has spare edges left – if we were unable to do this, G would be disconnected. By the same logic, we can pick a circuit D_2 from the remaining edges starting from and returning to v_2. We then create a new circuit C_2 as follows: we begin at v_1, walk to v_2 on the first part of C_1, walk around D_2 back to v_2, then walk back from v_2 to v_1 on the second part of C_1.

Again, if C_2 Eulerian, we are done. Else we pick a vertex v_3 with edges remaining, choose another circuit D_3 containing v_3, and splice D_3 into C_2 to make a new circuit C_3.

Since our graph is finite, this process must eventually use up all the edges, giving us an Eulerian circuit. ☐

Given a connected graph with all even degrees, we can use the algorithm in the proof above to construct an Eulerian circuit.

Example 2.6. Consider the following graph:

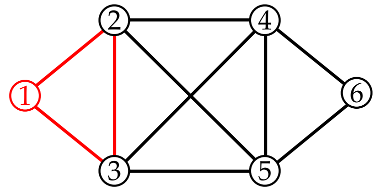


Note that it is connected and has all even degrees, so we know an Eulerian circuit exists. Now we want to build one.

We pick **1** as our starting vertex and

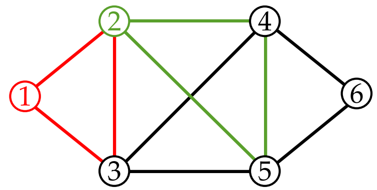
1231

as the first circuit.



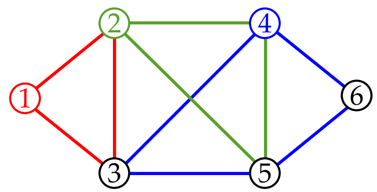
This is not Eulerian. Vertex **2** still has edges left over, so we pick **2452** as our second circuit. Splicing this into the first, we get

12 **452** 31.



This is still not Eulerian, but **4** has spare edges, and is in the circuit **46534**. Splicing this in, we get

12**4** **6534** 5231.



This is Eulerian, so we have an Eulerian circuit 12465345231.

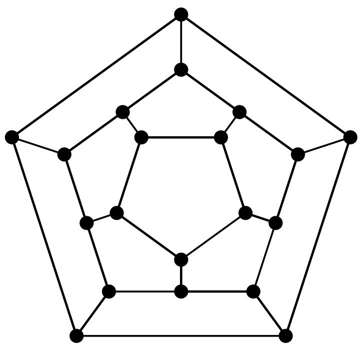
2.4 Hamiltonian cycles

The Bridges of Königsberg problem is a bit artificial – more often, it seems natural to want a walk to visit every vertex. For example, if the graph is map of towns, we may want to visit each of the towns (not each of the roads); if we think of information going around a network, we want it to get to every computer (not be transmitted down every wire). In the example of towns, we'll also probably want a cycle, so we end up where we started.

Definition 2.7. A *Hamiltonian cycle* on a graph is a cycle that visits every vertex.

Note that, by definition, a cycle can visit each vertex at most once, so a Hamiltonian cycle will visit every vertex exactly once (except that the first and last vertex are the same).

Hamiltonian cycles are named after William Rowan Haimlton, who invented the 'icosian game', which asked if there is a Hamiltonian cycle on the graph of the dodecahedron.



(He got the game made out of wood and sold it – not very successfully.)

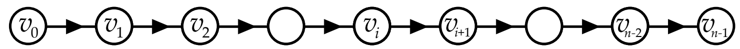
Unfortunately, unlike for Eulerian circuits, there is no easy way in general to tell whether or not a graph has a Hamiltonian cycle. (It's an NP-complete problem.) However, we can say that if there are lots of edges incident at every vertex, then a Hamiltonian cycle must exist. This result is called Dirac's theorem, after GA Dirac (stepson of the famous quantum theorist and Bristolian Paul Dirac).

Theorem 2.8 (Dirac's theorem)**.** *Consider a graph $G = (V, E)$ with $n = |V| \geq 3$ vertices. If $d(v) \geq n/2$ for all $v \in V$, then G has a Hamiltonian cycle.*

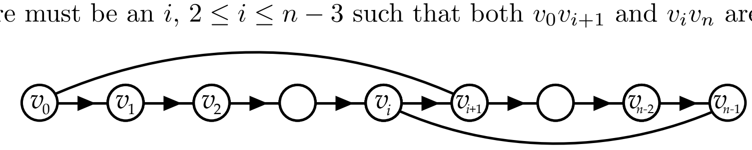
Proof. Suppose G has no Hamiltonian cycle. We will show that we cannot have all degrees $n/2$ or bigger.

Without loss of generality, we can add edges to G until it is maximal non-Hamiltonian; that is, until adding any extra edge would form a Hamiltonian cycle.

Therefore G must have a path $v_0 v_1 \ldots v_{n-1}$ that goes through every vertex exactly once (a Hamiltonian cycle minus an edge).



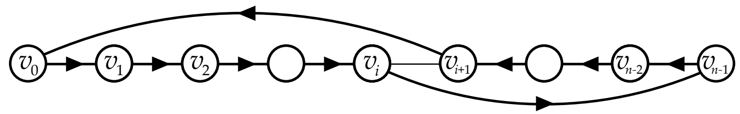
Suppose, seeking a contradiction, that $d(v_0) \geq n/2$ and $d(v_{n-1}) \geq n/2$. We claim that there must be an i, $2 \leq i \leq n-3$ such that both $v_0 v_{i+1}$ and $v_i v_n$ are edges.



Let's prove this claim. Since v_0 is already adjacent to v_1 and not adjacent to v_{n-1}, then it is joined to at least $n/2 - 1$ of the vertices $v_2, v_3, \ldots, v_{n-2}$. If we wanted the claim to be false, then v_{n-1} would have to avoid the $n/2 - 1$ or more vertices following these (including v_2, but perhaps losing one by not counting $i = n - 2$). But if $d(v_{n-1}) \geq n/2$, it has to take $n/2 - 1$ of these $n - 3$ vertices too. But $2(n/2 - 1) > n - 3$, so the claim is unavoidable.

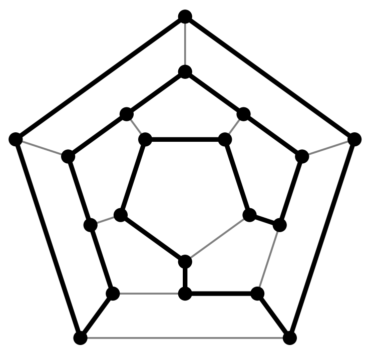
But now we have a Hamiltonian cycle

$$v_0 v_1 \cdots v_{i-1} v_i v_{n-1} v_{n-2} \cdots v_{i+2} v_{i+1} v_0.$$



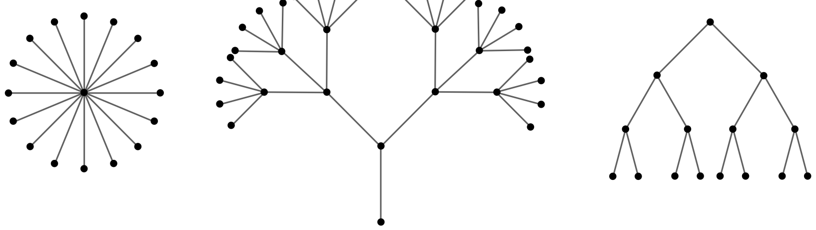
This is a contradiction, and we are done. □

Note that the condition in Dirac's theorem ia sufficient but not necessary for the existence of a Hamiltonian cycle. For example, Hamilton's icosian game has $n = 20$ vertices but all degrees $d(v) = 3 < n/2$, but it does indeed have a Hamiltonian cycle.



2.5 Trees

These graphs should be recognisable as what are called *trees*.



What makes these graphs special is that there are no cycles in them – they are *acyclic*.

Definition 2.9. An acyclic graph is called a *forest*.

A connected acyclic graph is called a *tree*.

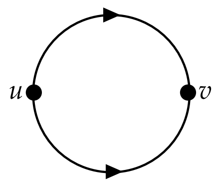
A vertex of degree 1 in a tree or forest is called a *leaf*.

(These definition are as close as we're going to get to jokes in this course.)

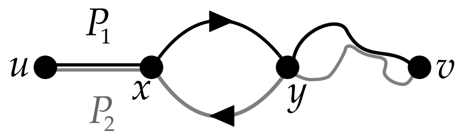
An equivalent way to define trees is by noting there is a unique path between any two vertices.

Theorem 2.10. *A graph $G = (V, E)$ is a tree if and only if for every $u, v \in V$ there is a unique path from u to v.*

Proof. (*If*) Suppose G is not a tree. We need to find u and v without a unique path. If G is not connected pick u and v in different connected components. If G has a cycle C, pick u and v two be two vertices on C. Then there are two paths from u to v: clockwise on C and anticlockwise on C.



(*Only if*) Again we prove the contrapositive. First, suppose there is no path from u to v. Then G is not connected, so not a tree. Second, suppose there are two paths P_1 and P_2 from u to v. Let x be the first place P_1 and P_2 separate and y be the next place the join back up. Then we have a cycle in G – from x to y along P_1, then back to x backwards on P_2 – so G is not a tree.



□

Looking at the trees above, you'll notice that they all have one fewer edge than they have vertices. This is always the case.

Theorem 2.11. *If $G = (V, E)$ is a tree, then $|E| = |V| - 1$.*

In fact, this can also be made an 'if and only if' too: any connected graph with $|E| = |V| - 1$ is a tree. (You might like to try and prove this.)

Proof. We proceed by induction on $|V|$. The base case $|V| = 1$ is trivial.

Consider a graph $G = (V, E)$ with $|V| = n$, and assume the theorem holds for all smaller values of $|V|$. Pick and edge e and remove it. This disconnects G into a forest of two trees, with k and $n - k$ vertices, for some $k < n$. By the inductive hypothesis, these have $k - 1$ and $n - k - 1$ edges respectively. Adding e back in again, our graph G has

$$m = (k - 1) + (n - k - 1) + 1 = n - 1$$

edges, as desired. □

Next time: *A day-before-Valentine's Day special: How to make sure everyone can get married to someone they like.*

Matthew Aldridge
m.aldridge@bristol.ac.uk