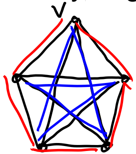
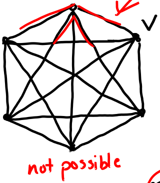
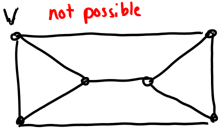
6.1: Eulerian Circuits

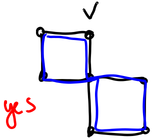
Observation: Odd degree is an obstacle!

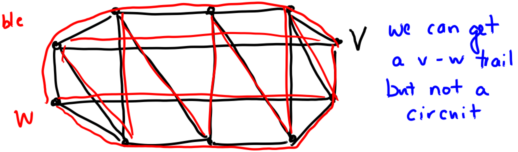
Question: Can you start and stop at vertex v in the graphs below and visit every edge exactly once?











Definition

An *Eulerian circuit* in a graph G is a circuit that contains every edge of G. If G contains an Eulerian circuit, then we say G is *Eulerian*. (We DON'T say that G IS an Euler circuit.)

A characterization

Theorem TFAE

For a connected graph G, the following are equivalent.

1. *G is Eulerian.*
2. *Every vertex of G has even degree.*
3. *The edges of G can be partitioned into cycles.*

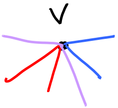
How to show a list of statements are equivalent?

- Show (1) $\Rightarrow$ (2), then
- Show (2) $\Rightarrow$ (3), then
- Show (3) $\Rightarrow$ (1).

Then we know, for example, that (3) $\Rightarrow$ (2), because (3) $\Rightarrow$ (1) and (1) $\Rightarrow$ (2).

Idea behind (1) implies (2)

You already told me:

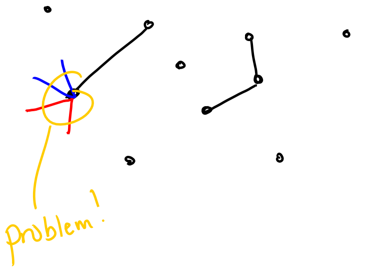


The circuit creates a natural pairing of the edges at v.
We can't ever end up with one edge left over, otherwise
we won't be able to finish a circuit.

Idea behind (2) implies (3)

Even degree implies cycle partition:

- Start with a collection of cycles in G, and expand it as much as possible, until it can't be made any bigger.
- Whatever is left over must be acyclic, so it's a forest.



But can it really look like this, because an even number of edges were removed at each vertex!

Idea behind (2) implies (3)

Even degree implies cycle partition:

- Start with a collection of cycles in G, and expand it as much as possible, until it can't be made any bigger.
- Whatever is left over must be acyclic, so it's a forest.
- But all vertices have even degree, and each cycle eats up two edges per vertex, so what's left must have all vertices even degree.
- Non-empty forests have leaves (not even degree!), so what's left must be empty.
- Claim victory!

Idea behind (3) implies (1)

Make an algorithm! Start with $C_1, \ldots, C_t$, find Euler circuit.

- Start from a single cycle in the collection.
- If it doesn't cover all the edges of G, then some edge incident on a vertex of the cycle is missing (using connectivity) - must be somewhere in the cycle partition.
- Insert it! Now we have a circuit (instead of a cycle).
- Repeat. The only way it stops is by covering all of G - i.e. we have an Euler circuit.

Let G be a connected graph.
(1) $\Rightarrow$ (2): Suppose G is Eulerian. We wish to show that every vertex in G has even degree.

Let v be an arbitrary vertex in $V(G)$, and let C be an Euler circuit in G.

For every appearance v in C, two edges incident on v are contained in the circuit. Since all edges incident on v appear exactly once in C by definition of an Euler circuit, $\deg(v) = 2t$, where t is the number of times v appears in C. Therefore $\deg(v)$ is even.

Since v was arbitrary, the result follows.

(2) $\Rightarrow$ (3): Suppose every vertex in G has even degree. We wish to show that the edges of G can be partitioned into cycles.

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_t\}$ be a maximal collection of cycles in G that do not share any edges. We claim that $\mathcal{C}$ partitions the edge set.

Since $\mathcal{C}$ is maximal, when the edges of the cycles in $\mathcal{C}$ are removed, what remains is acyclic, and is therefore a forest.

Every vertex had an even number of edges removed when the edges of the cycles in $\mathcal{C}$ were removed, and since each vertex started with even degree, each vertex in the forest that remains has even degree.

Every nonempty forest has leaves, which would imply the existence of vertices of odd degree. Therefore what remains is an empty forest, and every edge appears in some cycle in $\mathcal{C}$.

(3) $\Rightarrow$ (1): Suppose the edges of G can be partitioned into cycles $C_1, C_2, \ldots, C_t$. We wish to show that G is Eulerian.

We describe an algorithm that builds an Eulerian circuit. Begin with $C = C_1$.

1. If C contains all of $E(G)$, then C is an Eulerian circuit; stop.
2. Otherwise, since G is connected, there is some vertex v in C with an incident edges uv that does not appear in C. Since the given cycles partition $E(G)$, uv is in some C_i. Insert C_i into C at vertex v, and update C.

Continue these two steps; since G is finite, the process must terminate at an Eulerian circuit. Therefore G is Eulerian.

Not much more to say...

linear in # of edges

The question of whether a graph contains an Eulerian circuit is solved.

- A computer can check *very* quickly if a graph is Eulerian by just check the degree of all the vertices.
- Property (3) above gives rise to a very fast algorithm for finding an Eulerian circuit if it exists.
- Determining whether a graph has an Eulerian circuit is a problem in P.