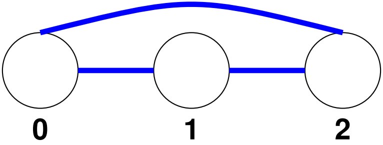
CS250: Discrete Math for Computer Science

L19: Eulerian Graphs

Lecture 19: Eulerian Graphs

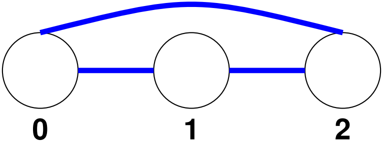
Last time we introduced Mathematical Induction. Today, we will use induction to prove things about graphs.



Lecture 19: Eulerian Graphs

Last time we introduced Mathematical Induction. Today, we will use induction to prove things about graphs.

Today, all graphs are undirected



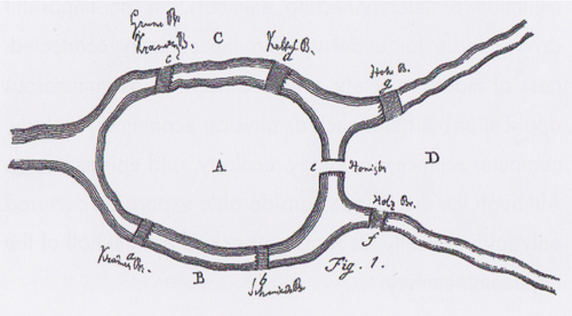
Leonhard **Euler** (1707–1783) made contributions to number theory, geometry, numerical analysis, combinatorics, calculus and complex analysis. His solution to the **Königsberg Bridge Problem** is considered the earliest result in graph theory.

Leonhard **Euler** (1707–1783) made contributions to number theory, geometry, numerical analysis, combinatorics, calculus and complex analysis. His solution to the **Königsberg Bridge Problem** is considered the earliest result in graph theory.

Is it possible to **traverse each** of the **seven bridges** of the city of Königsberg returning to your starting point, **without crossing** a single **bridge twice**?

Leonhard **Euler** (1707–1783) made contributions to number theory, geometry, numerical analysis, combinatorics, calculus and complex analysis. His solution to the **Königsberg Bridge Problem** is considered the earliest result in graph theory.

Is it possible to **traverse each** of the **seven bridges** of the city of Königsberg returning to your starting point, **without crossing** a single **bridge twice**?



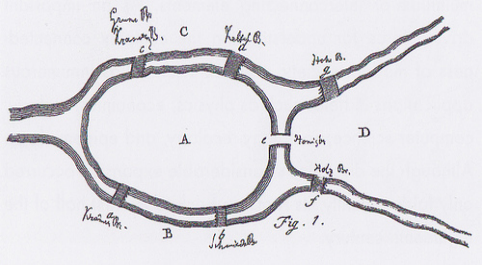
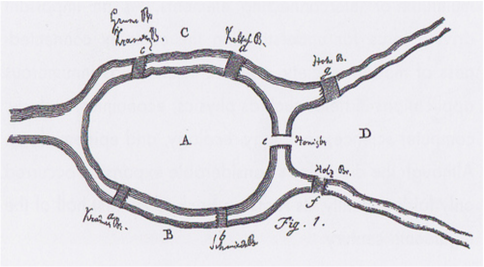
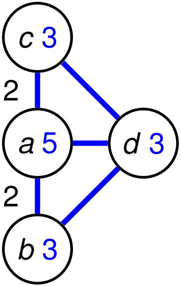
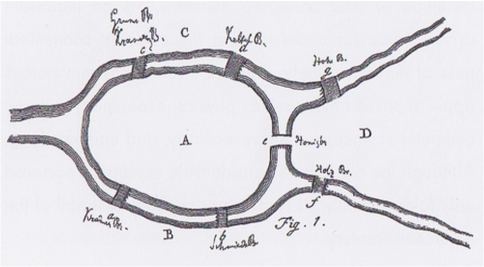
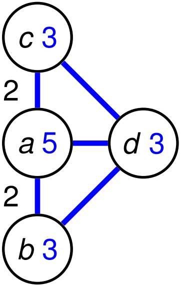


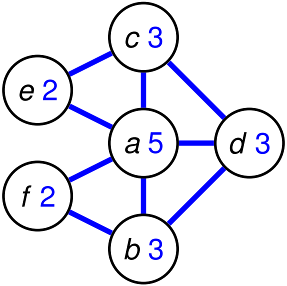
Fig. 1.


C
A
D
B
Fig. 1.


c 3
2
a 5
d 3
2
b 3


C
A
D
B
Fig. 1.


c 3
2
a 5
d 3
2
b 3

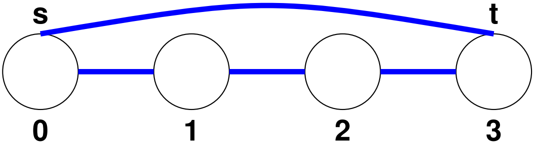

c 3
e 2
a 5
d 3
f 2
b 3

Walks on Graphs

Def. A **walk** of **length** r on graph G from s to t is a sequence of $r+1$ vertices starting at s and ending at t so that each step is along an edge.

$$w \ = \ (s = v_0, v_1, v_2, \ldots, v_{r-1}, v_r = t) \quad (v_i, v_{i+1}) \in E^G, 0 \le i < r$$

G_1

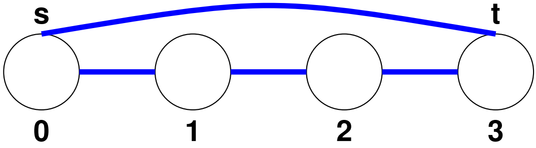


Walks on Graphs

Def. A **walk** of **length** r on graph G from s to t is a sequence of $r+1$ vertices starting at s and ending at t so that each step is along an edge.

$$w \;=\; (s = v_0, v_1, v_2, \ldots, v_{r-1}, v_r = t) \qquad (v_i, v_{i+1}) \in E^G, 0 \leq i < r$$

G_1



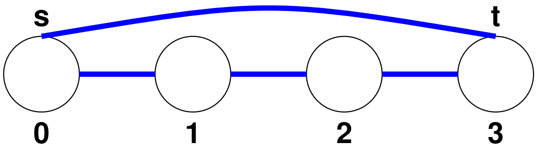
$(0,\ 1,\ 2,\ 3)$ is a **walk** of length 3 from s to t on G_1

Walks on Graphs

Def. A **walk** of **length** r on graph G from s to t is a sequence of $r+1$ vertices starting at s and ending at t so that each step is along an edge.

$$w \;=\; (s = v_0, v_1, v_2, \ldots, v_{r-1}, v_r = t) \qquad (v_i, v_{i+1}) \in E^G, 0 \leq i < r$$

G_1

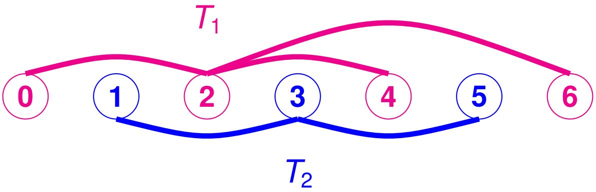


$(0,\ 1,\ 2,\ 3)$ is a **walk** of length 3 from s to t on G_1

$(0,\ 3)$ is a **walk** of length 1 from s to t

Def. A **path**, $p = (v_0, v_1, \ldots, v_r)$, is a walk with no repeated vertices or edges (except the start and end points may be equal).

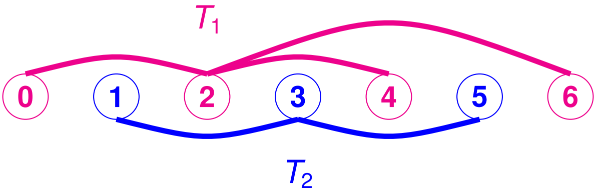
$F_1 = T_1 \cup T_2$



Def. A **path**, $p = (v_0, v_1, \ldots, v_r)$, is a walk with no repeated vertices or edges (except the start and end points may be equal).

A path that starts and ends at the same vertex is called a **cycle**. A graph that has no cycles is **acyclic**.

$F_1 = T_1 \cup T_2$

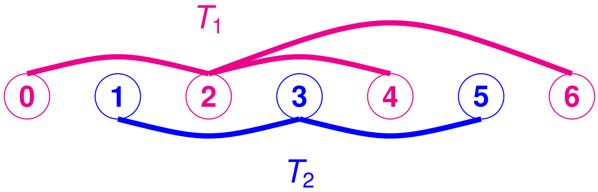


Def. A **path**, $p = (v_0, v_1, \ldots, v_r)$, is a walk with no repeated vertices or edges (except the start and end points may be equal).

A path that starts and ends at the same vertex is called a **cycle**. A graph that has no cycles is **acyclic**.

A graph is **connected** iff it has a path between every pair of vertices. A **connected component**, C, of an undirected graph, G, is a maximal subgraph of G that is connected.

$F_1 = T_1 \cup T_2$



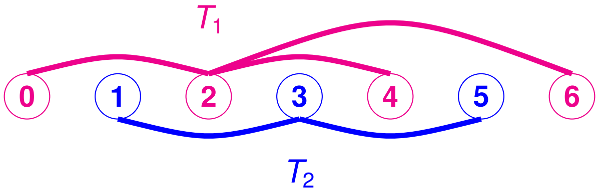
Def. A **path**, $p = (v_0, v_1, \ldots, v_r)$, is a walk with no repeated vertices or edges (except the start and end points may be equal).

A path that starts and ends at the same vertex is called a **cycle**. A graph that has no cycles is **acyclic**.

A graph is **connected** iff it has a path between every pair of vertices. A **connected component**, C, of an undirected graph, G, is a maximal subgraph of G that is connected.

An undirected **forest** is an acyclic undirected graph. A **tree** is a connected forest.

$F_1 = T_1 \cup T_2$



Walks on Graphs

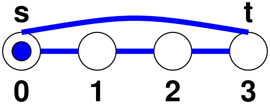
Today we want to walk from s to t, but **never cross the same edge twice**.

Walks on Graphs

Today we want to walk from s to t, but **never cross the same edge twice**. We call such a walk a **bb** walk. ("bb" stands for "**bridge burning**".) $(0, 1, 2, 3)$ is a **bb** walk from s to t.

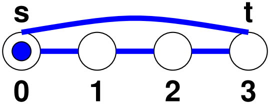
Walks on Graphs

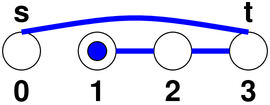
Today we want to walk from s to t, but **never cross the same edge twice**. We call such a walk a **bb** walk. ("bb" stands for "**bridge burning**".) $(0, 1, 2, 3)$ is a **bb** walk from s to t.



Walks on Graphs

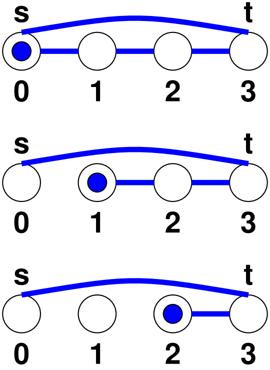
Today we want to walk from s to t, but **never cross the same edge twice**. We call such a walk a **bb** walk. ("bb" stands for "**bridge burning**".) $(0, 1, 2, 3)$ is a **bb** walk from s to t.





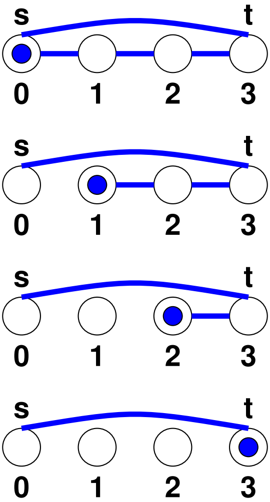
Walks on Graphs

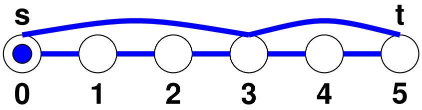
Today we want to walk from s to t, but **never cross the same edge twice**. We call such a walk a **bb** walk. ("bb" stands for "**bridge burning**".) $(0, 1, 2, 3)$ is a **bb** walk from s to t.



Walks on Graphs

Today we want to walk from s to t, but **never cross the same edge twice**. We call such a walk a **bb** walk. ("bb" stands for "**bridge burning**".) $(0, 1, 2, 3)$ is a **bb** walk from s to t.





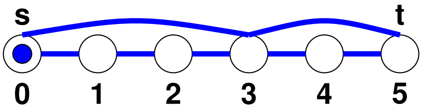
From *s*, take an

exhaustive bb walk

on *G*, i.e., **until**

you can go

no further.



From s, take an

exhaustive bb walk

on G, i.e., **until**

you can go

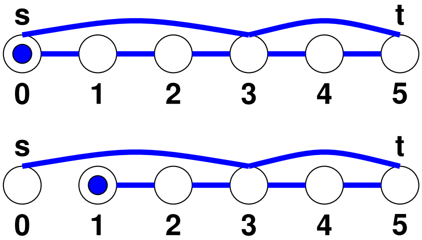
no further.

Must you end up

back at s?

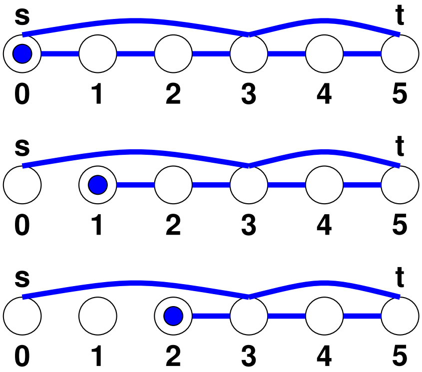
From *s*, take an **exhaustive** bb walk on *G*, i.e., **until you can go no further**.

Must you end up back at *s*?



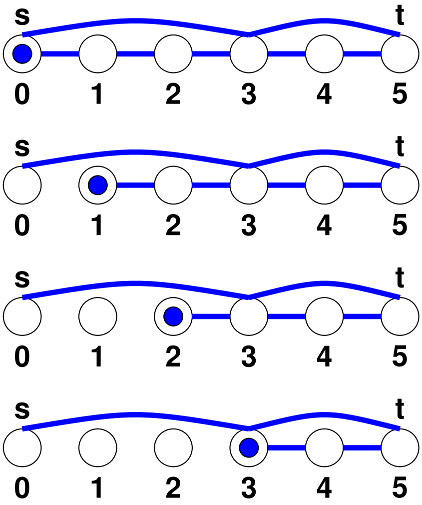
From s, take an **exhaustive** bb walk on G, i.e., **until you can go no further**.

Must you end up back at s?



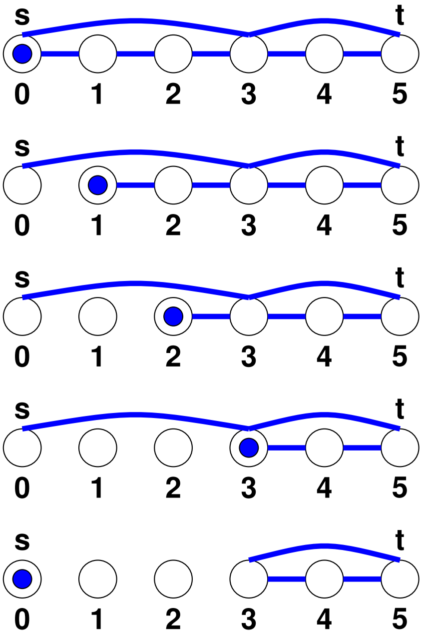
From *s*, take an **exhaustive** bb walk on *G*, i.e., **until you can go no further**.

Must you end up back at *s*?

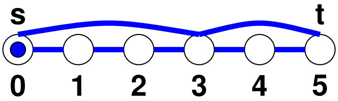


From s, take an **exhaustive** bb walk on G, i.e., **until you can go no further**.

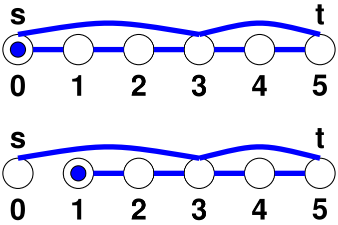
Must you end up back at s?



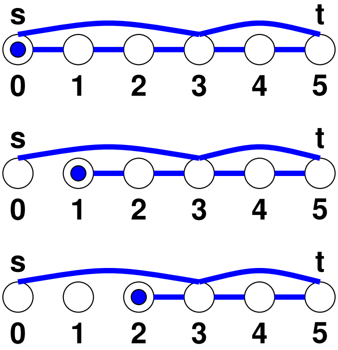
Must you end up back at *s*?



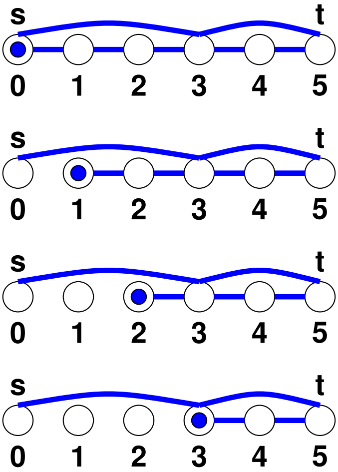
Must you end up back at *s*?



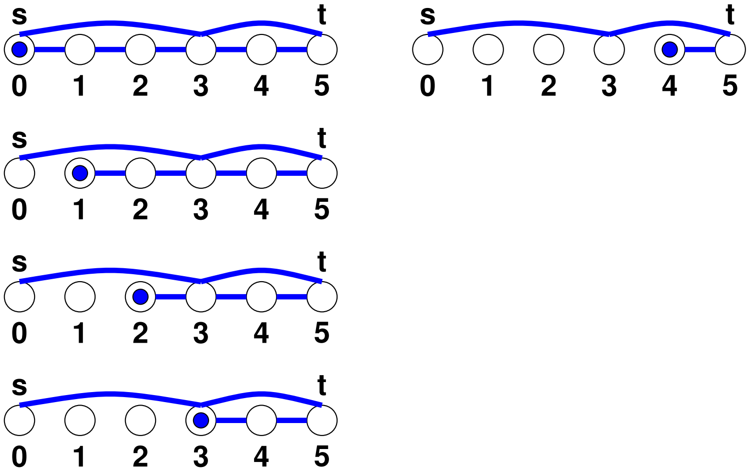
Must you end up back at *s*?



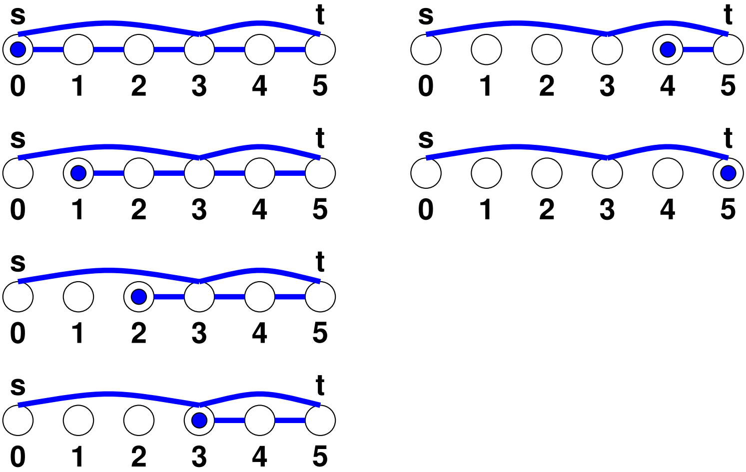
Must you end up back at *s*?



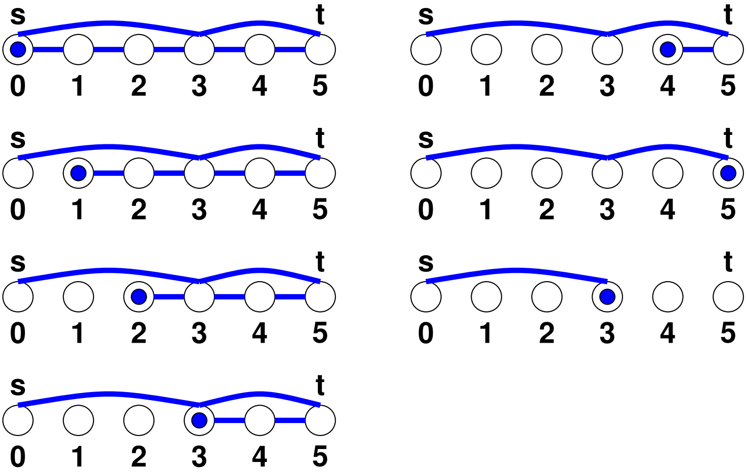
Must you end up back at *s*?



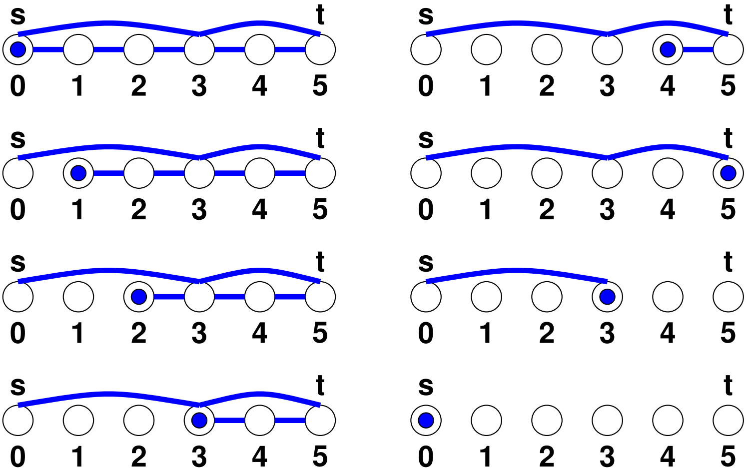
Must you end up back at *s*?



Must you end up back at *s*?



Must you end up back at *s*?



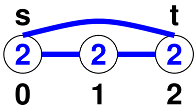
Degree of a Vertex

Def. In an undirected graph, the **degree** of a vertex v is the number of neighbors that v has, but loops count as 2:

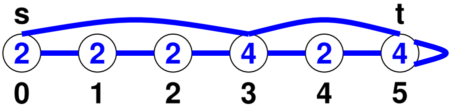
$$\mathbf{deg}(v) \quad = \quad |\{w \neq v \mid (v, w) \in E^G\}| + \begin{cases} 2 & \text{if } v \text{ has a loop} \\ 0 & \text{otherwise} \end{cases}$$

If all the vertices of G have degree d, then we say that G is **regular** of degree d.

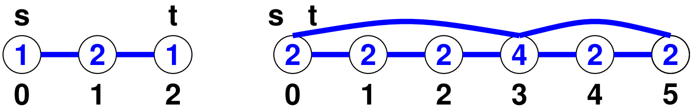
G_1



G_2



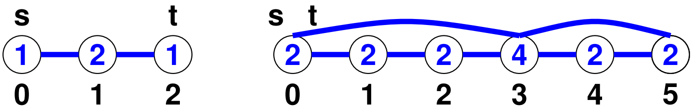
The Parity of the Degrees is Key



Def. An **Eulerian walk** in a graph *G* is a walk from *s* to *t* that traverses every edge exactly once and every vertex at least once.

$$\begin{aligned} \mathrm{EC}(G) \;\stackrel{\text{def}}{=}\; & G \text{ is connected} \;\wedge\; \forall v \notin \{s,t\}\ \deg(v) \text{ is even} \;\wedge \\ & \big(s = t \;\wedge\; \deg(s) = \deg(t) \text{ is even} \quad \vee \\ & \;s \neq t \;\wedge\; \deg(s), \deg(t) \text{ are odd} \quad \big) \end{aligned}$$

The Parity of the Degrees is Key



Def. An **Eulerian walk** in a graph G is a walk from s to t that traverses every edge exactly once and every vertex at least once.

$$\begin{aligned} \mathrm{EC}(G) \quad &\stackrel{\text{def}}{=} \quad G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge \\ &\quad \big(s = t \ \wedge \ \deg(s) = \deg(t) \text{ is even } \quad \vee \\ &\quad\ s \neq t \ \wedge \ \deg(s), \deg(t) \text{ are odd } \quad \big) \end{aligned}$$

Thm. [Euler] G has an Eulerian walk from s to t iff $\mathrm{EC}(G)$.

$$
\begin{aligned}
\mathrm{EC}(G) \quad \stackrel{\text{def}}{=} \quad & G \text{ is connected } \wedge \forall v \notin \{s, t\} \ \deg(v) \text{ is even } \wedge \\
& \big(s = t \ \wedge \ \deg(s) = \deg(t) \text{ is even} \quad \vee \\
& \ s \neq t \ \wedge \ \deg(s), \deg(t) \text{ are odd} \quad \big)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{EC}(G) \stackrel{\text{def}}{=}\ & G \text{ is connected } \wedge \forall v \notin \{s,t\}\ \deg(v) \text{ is even } \wedge \\
& \big(s = t \ \wedge\ \deg(s) = \deg(t) \text{ is even } \quad \vee \\
& \ s \neq t \ \wedge\ \deg(s), \deg(t) \text{ are odd } \quad \big)
\end{aligned}
$$

Claim G has an Eulerian walk $\rightarrow$ $\mathrm{EC}(G)$.

$$\begin{aligned}\mathrm{EC}(G) \;\overset{\text{def}}{=}\;& G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge \\ & \big(s = t \ \wedge\ \deg(s) = \deg(t) \text{ is even } \quad \vee \\ & \ s \neq t \ \wedge\ \deg(s), \deg(t) \text{ are odd } \quad \big)\end{aligned}$$

Claim G has an Eulerian walk $\rightarrow$ EC(G).

Proof: For all vertices, v, besides s and t, the walk must leave v the same number of times that it enters v. Thus, deg(v) is even.

$$
\begin{aligned}
\mathrm{EC}(G) \overset{\text{def}}{=}\ & G \text{ is connected } \wedge \forall v \notin \{s,t\}\ \deg(v) \text{ is even } \wedge \\
& \big(s = t \wedge \deg(s) = \deg(t) \text{ is even } \quad \vee \\
& \ s \neq t \wedge \deg(s), \deg(t) \text{ are odd } \quad \big)
\end{aligned}
$$

Claim G has an Eulerian walk $\rightarrow$ $\mathrm{EC}(G)$.

Proof: For all vertices, v, besides s and t, the walk must leave v the same number of times that it enters v. Thus, $\deg(v)$ is even.

If $s = t$, then $\deg(s) = deg(t)$ is even for the same reason.

$$\mathrm{EC}(G) \overset{\text{def}}{=} G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge$$
$$\big(s = t \ \wedge \ \deg(s) = \deg(t) \text{ is even } \quad \vee$$
$$s \neq t \ \wedge \ \deg(s), \deg(t) \text{ are odd } \quad \big)$$

Claim G has an Eulerian walk $\rightarrow$ $\mathrm{EC}(G)$.

Proof: For all vertices, v, besides s and t, the walk must leave v the same number of times that it enters v. Thus, $\deg(v)$ is even.

If $s = t$, then $\deg(s) = deg(t)$ is even for the same reason.

If $s \neq t$, then s is left once more than it is reached and t is reached once more than it is left, so their degrees are both odd.

$$\mathrm{EC}(G) \overset{\text{def}}{=} G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge$$
$$\big(s = t \ \wedge \ \deg(s) = \deg(t) \text{ is even} \quad \vee$$
$$s \neq t \ \wedge \ \deg(s), \deg(t) \text{ are odd} \quad \big)$$

Claim G has an Eulerian walk $\rightarrow$ EC(G).

Proof: For all vertices, v, besides s and t, the walk must leave v the same number of times that it enters v. Thus, deg(v) is even.

If $s = t$, then deg(s) $=$ *deg*(t) is even for the same reason.

If $s \neq t$, then s is left once more than it is reached and t is reached once more than it is left, so their degrees are both odd.

Since an Eulerian walk visits every vertex, G is connected. $\square$

$$
\begin{aligned}
\mathrm{EC}(G) \stackrel{\text{def}}{=}\ & G \text{ is connected } \wedge \forall v \notin \{s, t\}\ \deg(v) \text{ is even } \wedge \\
& \big(s = t \ \wedge\ \deg(s) = \deg(t) \text{ is even } \quad \vee \\
& \ s \neq t \ \wedge\ \deg(s), \deg(t) \text{ are odd } \quad \big)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{EC}(G) \;\stackrel{\text{def}}{=}\; & G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge \\
& \big(s = t \wedge \deg(s) = \deg(t) \text{ is even } \quad \vee \\
& \ s \neq t \wedge \deg(s), \deg(t) \text{ are odd } \quad \big)
\end{aligned}
$$

Claim $\mathrm{EC}(G) \rightarrow G$ has an Eulerian walk.

$$\begin{aligned}\mathrm{EC}(G) \stackrel{\text{def}}{=}\ & G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge \\ & \big(s = t \ \wedge\ \deg(s) = \deg(t) \text{ is even } \quad \vee \\ & \ s \neq t \ \wedge\ \deg(s), \deg(t) \text{ are odd } \quad \big)\end{aligned}$$

Claim $\mathrm{EC}(G) \ \rightarrow\ G$ has an Eulerian walk.

We will prove the Claim by induction: $\mathbf{N} \models \forall x\ \alpha(x)$, where

$$\alpha(x) \stackrel{\text{def}}{=} \forall G\ (|E^G| \leq x \wedge \mathrm{EC}(G) \ \rightarrow\ G \text{ has an Eulerian walk })$$

$$\begin{aligned}\mathrm{EC}(G) \;\stackrel{\text{def}}{=}\;\; & G \text{ is connected } \;\wedge\; \forall v \notin \{s,t\}\ \deg(v) \text{ is even } \;\wedge \\ & \big(s = t \;\wedge\; \deg(s) = \deg(t) \text{ is even } \quad\vee \\ & \;s \neq t \;\wedge\; \deg(s), \deg(t) \text{ are odd } \quad\big)\end{aligned}$$

Claim $\mathrm{EC}(G) \;\rightarrow\; G$ has an Eulerian walk.

We will prove the Claim by induction: $\mathbf{N} \models \forall x\ \alpha(x)$, where

$$\alpha(x) \stackrel{\text{def}}{=} \forall G\ (|E^G| \leq x \wedge \mathrm{EC}(G) \;\rightarrow\; G \text{ has an Eulerian walk })$$

base case: $\alpha(0)$: Let G be an arbitrary graph with 0 edges and $\mathrm{EC}(G)$. Since G is connected and has no edges it must consist of a single vertex, $s = t$. Thus, the empty walk is an Eulerian walk from s to t.

$$\mathrm{EC}(G) \stackrel{\text{def}}{=} G \text{ is connected } \wedge \forall v \notin \{s,t\} \ \deg(v) \text{ is even } \wedge \\ \big(s = t \ \wedge \ \deg(s) = \deg(t) \text{ is even } \quad \vee \\ s \neq t \ \wedge \ \deg(s), \deg(t) \text{ are odd } \quad \big)$$

Claim $\mathrm{EC}(G) \ \rightarrow \ G$ has an Eulerian walk.

We will prove the Claim by induction: $\mathbf{N} \models \forall x \ \alpha(x)$, where

$$\alpha(x) \stackrel{\text{def}}{=} \forall G \ (|E^G| \leq x \wedge \mathrm{EC}(G) \ \rightarrow \ G \text{ has an Eulerian walk })$$

base case: $\alpha(0)$: Let G be an arbitrary graph with 0 edges and $\mathrm{EC}(G)$. Since G is connected and has no edges it must consist of a single vertex, $s = t$. Thus, the empty walk is an Eulerian walk from s to t.

Next time: we'll prove the **inductive case:** $\alpha(x_0) \rightarrow \alpha(x_0 + 1)$.