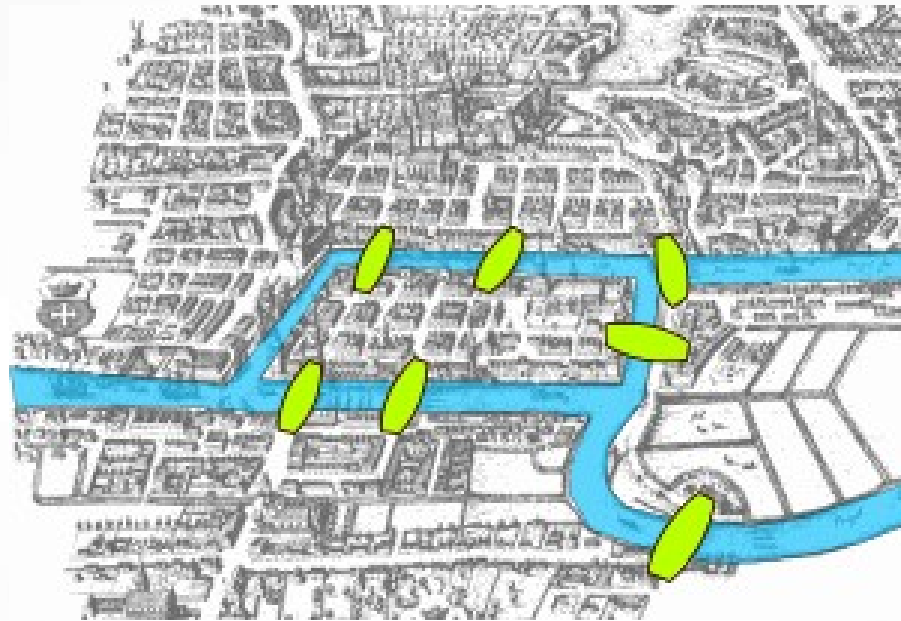


Eulerian tours

	Miles Jones	MTThF 8:30-9:50am	CSE 4140

August 15, 2016

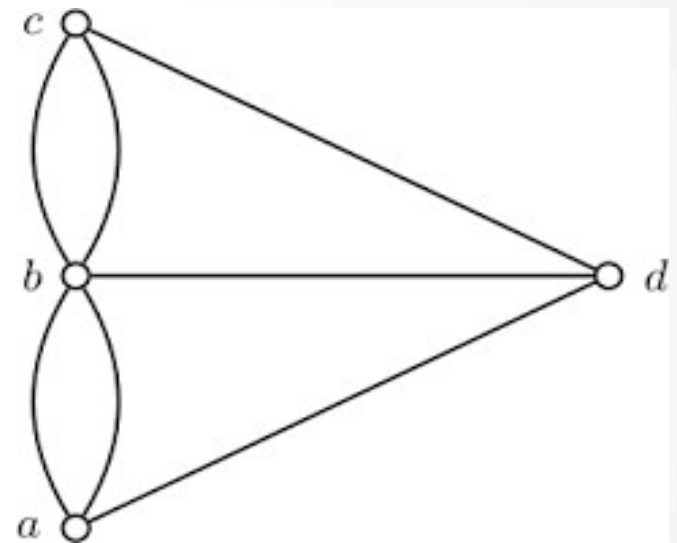
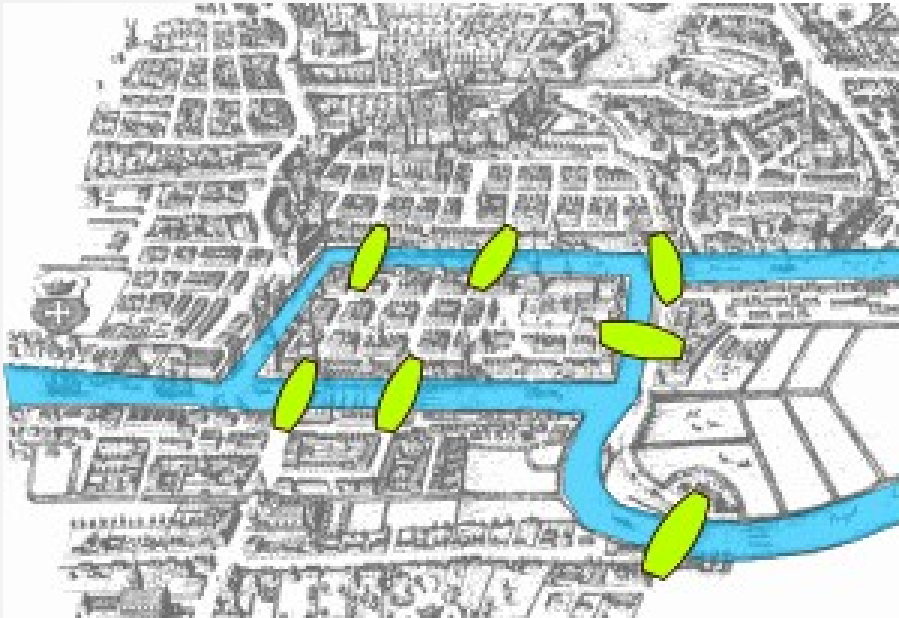
Seven Bridges of Königsberg



Is there a path that crosses each bridge (exactly) once?

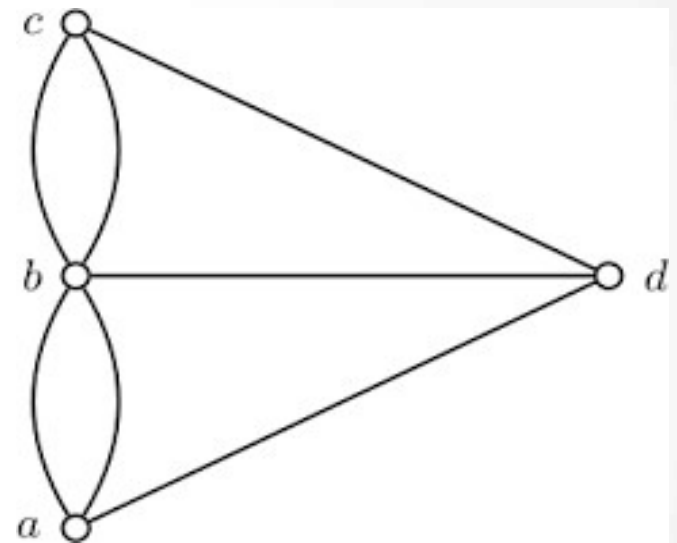
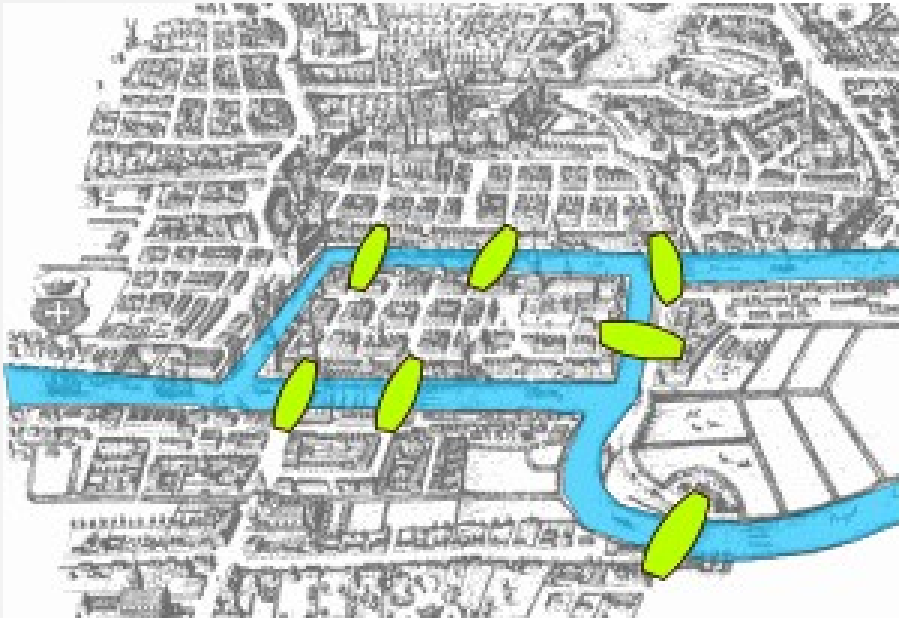
Rosen p. 693

Seven Bridges of Königsberg



Observe: exact location on the north side doesn't matter because must come & go via a bridge. Can represent each bridge as an edge.

Seven Bridges of Königsberg



Is there a path where each **edge** occurs exactly once? **Eulerian tour**

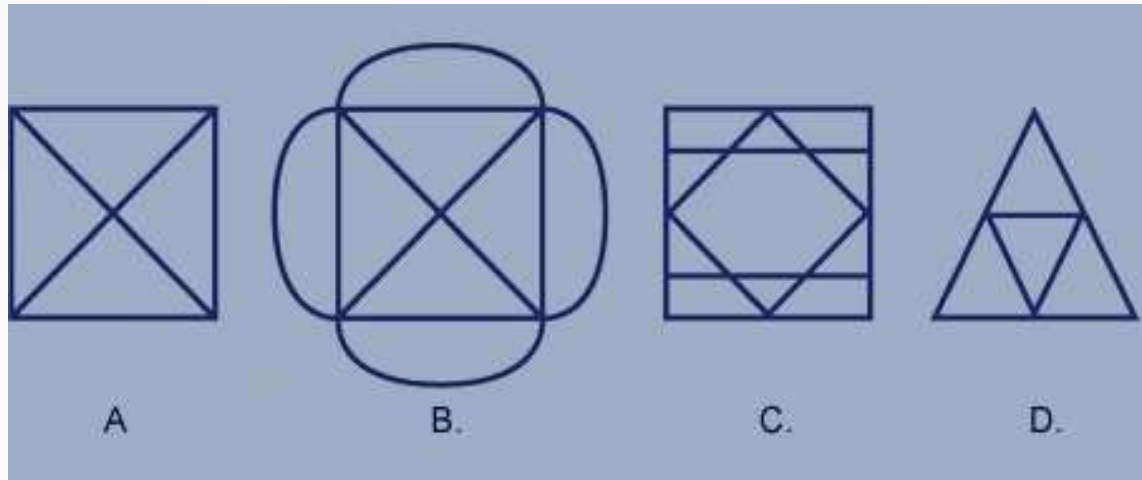
Rosen p. 693

Eulerian tour and Eulerian cycle (or circuit)

Eulerian tour (or path): a path in a graph that passes through every edge exactly once.

Eulerian cycle (or circuit): a path in a graph that pass through every edge exactly once and starts and ends on the same vertex.

Seven Bridges of Königsberg redux



Which of these puzzles can you draw without lifting your pencil off the paper?

Algorithmic questions related to Euler tours

Existence:

Does the given graph G contain an Euler tour?

Path:

Find an Euler tour for the given graph G , if possible.

Turns out there are great algorithms for each of these ... next!

Algorithmic questions related to ~~Euler tours~~

Hamiltonian tour

A **Hamiltonian tour** is a path where each **vertex** occurs exactly once.

Existence:

Does the given graph G contain a Hamiltonian tour?

Path:

Find a Hamiltonian tour for the given graph G , if possible.

These questions turn out to be intractable!!!

Algorithmic questions related to ~~Euler~~ tours

Hamiltonian tour

Actually, it is not known how to determine in any reasonable amount of time whether a graph G has a Hamilton Tour, or how to find one.

An Opportunity:

You can earn \$1,000,000 if you can give an algorithm that finds a Hamilton Tour (if one exists) in an arbitrary graph on n vertices that takes time $O(n^k)$ for constant k .

Light Switches

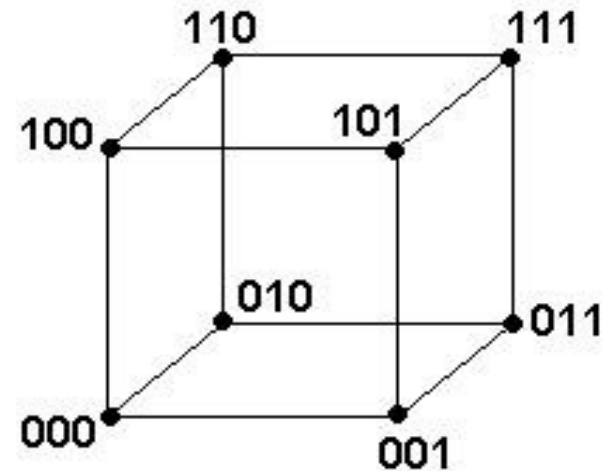
A light bulb is connected to 3 switches in such a way that it lights up only when all the switches are in the proper position. But you don't know what the proper position of each switch is!

What's the minimum number of single flips of a switch to guarantee that the light bulb turns on?

- A. 4
- B. 8
- C. 16
- D. 64
- E. None of the above



Light Switches



Configuration 1 if switch is UP, 0 if DOWN

Connect configuration if off by one switch.

Rephrasing the problem:

Looking for **Hamiltonian tour** through graph.

Our Strategy

Puzzle / Problem

Model as a graph

Choose vertex set & edge set ... sometimes many possible options

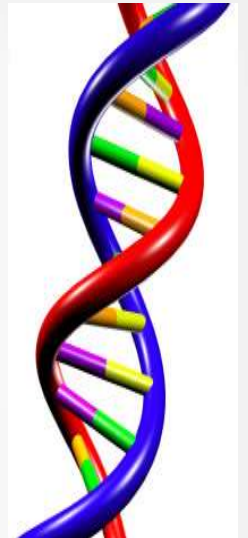
Use graph algorithms to solve puzzle / problem

DNA Reconstruction

Problem

Given collection of short DNA strings.

Find longer DNA string that includes them all.



Many possible formulations as a graph problem.

Successful solution was a big step in Human Genome Project.

DNA Reconstruction: as a Hamiltonian tour

Given collection of short DNA strings.

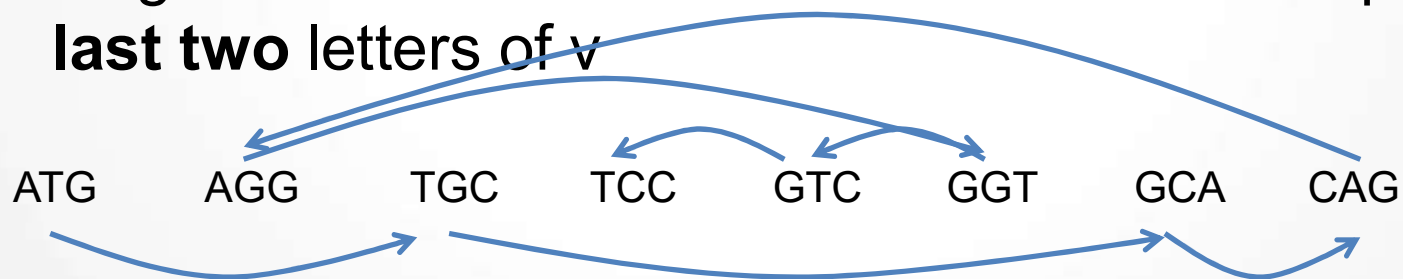
$S = \{ \text{ATG}, \text{AGG}, \text{TGC}, \text{TCC}, \text{GTC}, \text{GGT}, \text{GCA}, \text{CAG} \}$

Find longer DNA string that includes them all.



Vertex set: S , i.e. a vertex for each short DNA string

Edge set: edge from v to w if the **first two** letters of w equal the **last two** letters of v



DNA Reconstruction: as a Hamiltonian tour

Given collection of short DNA strings.

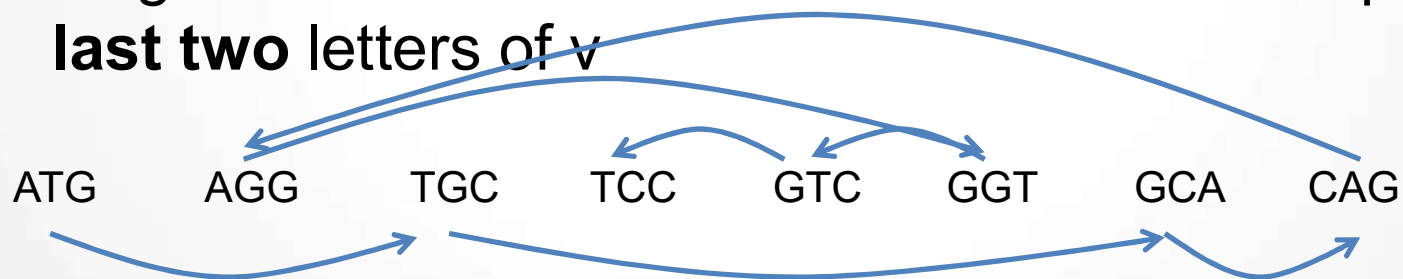
$S = \{ \text{ATG, AGG, TGC, TCC, GTC, GGT, GCA, CAG} \}$

Find longer DNA string that includes them all.



Vertex set: S , i.e. a vertex for each short DNA string

Edge set: edge from v to w if the **first two** letters of w equal the **last two** letters of v



What's a Hamiltonian tour?

DNA Reconstruction: as an Eulerian tour

Given collection of short DNA strings.

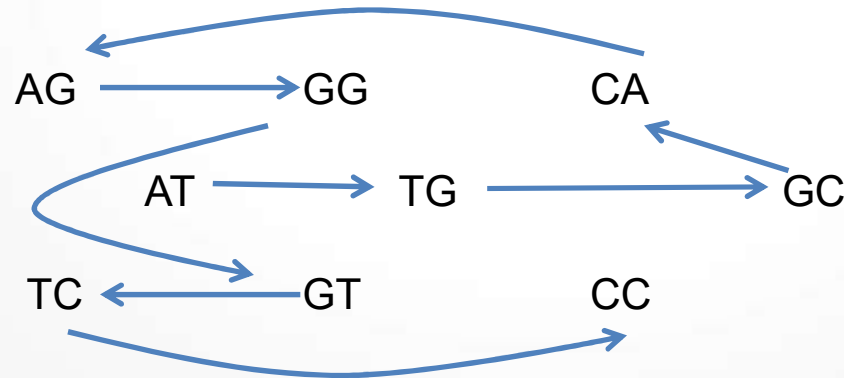
$S = \{ \text{ATG, AGG, TGC, TCC, GTC, GGT, GCA, CAG} \}$

Find longer DNA string that includes them all.



Vertex set: All length-two strings that appear in a word in S

Edge set: edge from ab to bc if abc is in S .



What's an Eulerian tour?

Finding Eulerian tours

Consider only **undirected** graphs.

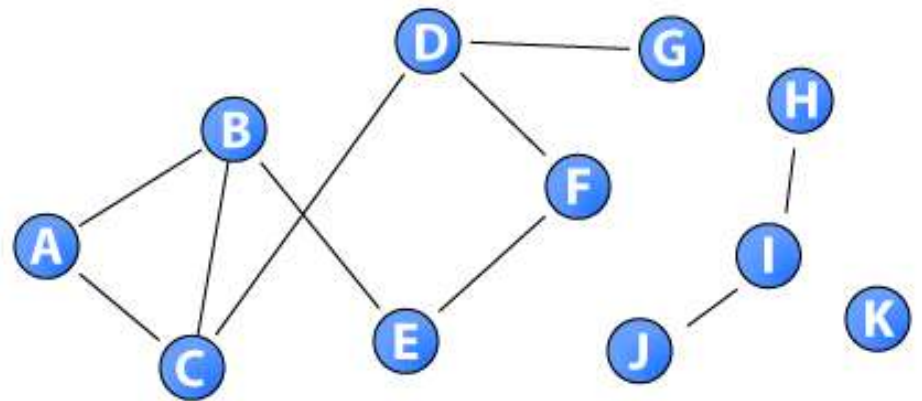
1st goal: Determine whether a given undirected graph G has an Eulerian tour.

2nd goal: Actually find an Eulerian tour in an undirected graph G , when possible.

Finding Eulerian tours

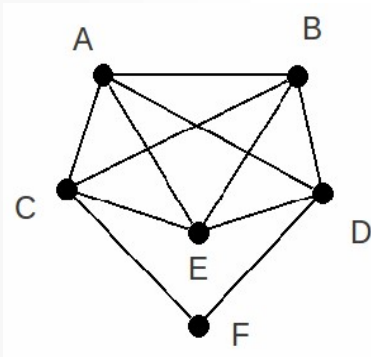
How many paths are there between vertex A and vertex I?

- A. None.
- B. Exactly one.
- C. Exactly two.
- D. More than two.
- E. None of the above.

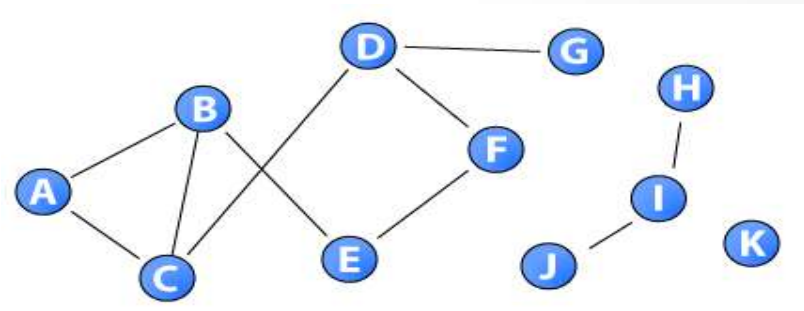


Connectedness

An undirected graph G is **connected** if *for any* ordered pair of vertices (v,w) there is a path from v to w .



Connected



Not connected

Connectedness

An undirected graph G is **connected** if *for any* ordered pair of vertices (v,w) there is a path from v to w .

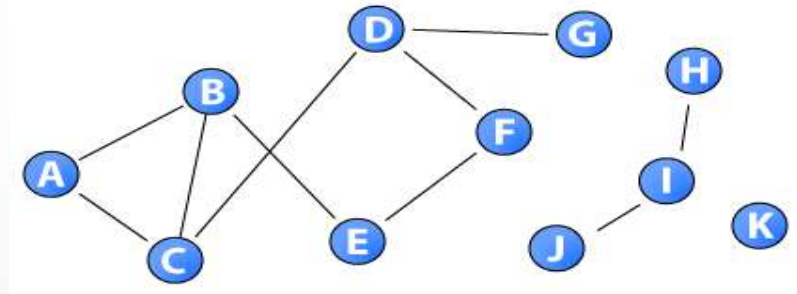
An undirected graph G is **disconnected** if

- A. for any ordered pair of vertices (v,w) there is no path from v to w .
- B. there is an ordered pair of vertices (v,w) with a path from v to w .
- C. there is an ordered pair of vertices (v,w) with no path from v to w .
- D. for every ordered pair of vertices (v,w) there is a path from v to w .
- E. None of the above.

Connected Components

Disconnected graphs can be broken up into pieces where each is connected.

Each connected piece of the graph is a **connected component**.



Finding Eulerian tours

Let $G = (V, E)$ be an

- **undirected**
- connected

graph with n vertices.

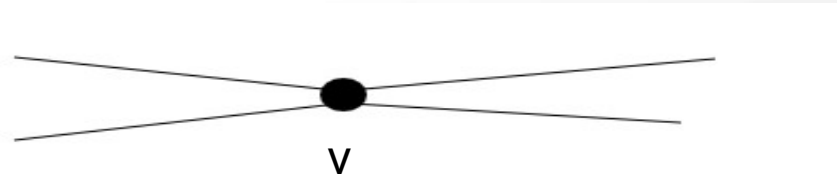
1st goal: Determine whether G has an Eulerian tour.

2nd goal: If yes, find the tour itself.

Finding Eulerian tours

Observation:

If v is an **intermediate*** vertex on a path p , then p must **enter** v the same number of times it **leaves** v .



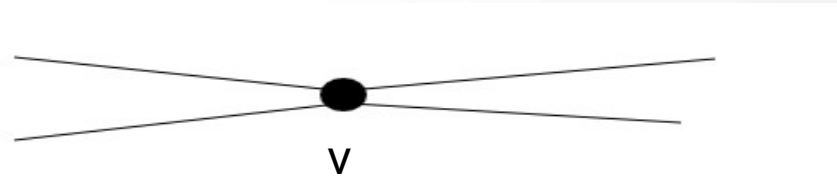
* not the start vertex, not the end vertex.

Finding Eulerian tours

Observation:

If v is an **intermediate*** vertex on a path p , then p must **enter** v the same number of times it **leaves** v .

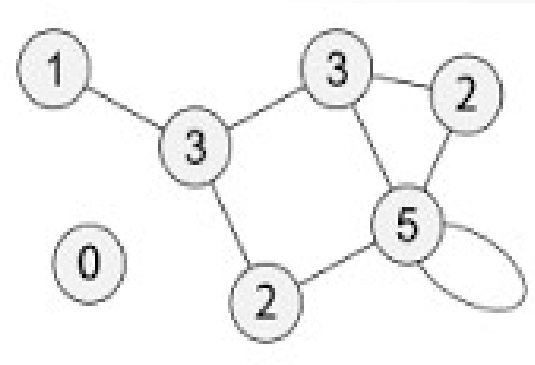
If p is an Eulerian tour, it contains all edges.
So, each edge incident with v is in p .



* not the start vertex, not the end vertex.

Recall: Degree

The **degree** of a vertex in an undirected graph is the total number of edges **incident** with it, except that a loop contributes twice.



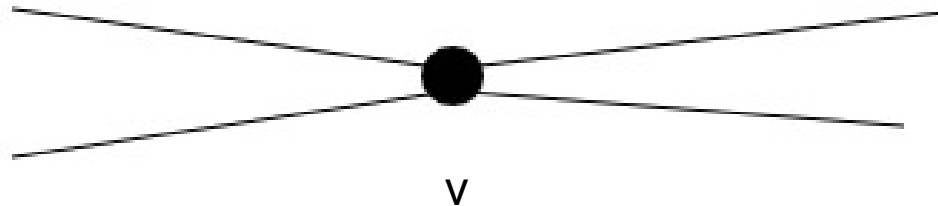
Finding Eulerian tours

Observation:

If v is an **intermediate*** vertex on a path p , then p must **enter** v the same number of times it **leaves** v .

If p is Eulerian tour, it has all edges: **each edge incident with v is in p .**

There are $\text{degree}(v)$ many of these



* not the start vertex, not the end vertex.

Finding Eulerian tours

Observation:

If v is an **intermediate*** vertex on a path p , then p must **enter** v the same number of times it **leaves** v .

If p is Eulerian tour, it has all edges: **each edge incident with v is in p .**

There are $\text{degree}(v)$ many of these

Half these edges are entering v , half are leaving v ...

$\text{degree}(v)$ is even!

Finding Eulerian tours

(Summary of) Observation:

In an Eulerian tour, any intermediate vertex has even degree.

*If tour is a circuit, all vertices are intermediate so all have even degree.
If tour is not a circuit, starting and ending vertices will have odd degree.*

Finding Eulerian tours

Theorem: If G has an Eulerian tour, G has at most two odd degree vertices.

Which of the following statements is the **converse** to the theorem?

- A. If G does not have an Eulerian tour, then G does not have at most two odd degree vertices.
- B. If G has at most two odd degree vertices, then G has an Eulerian tour.
- C. If G does not have at most two odd degree vertices, then G does not have an Eulerian tour.
- D. More than one of the above.
- E. None of the above.

Finding Eulerian tours

Theorem: If G has an Eulerian tour, G has at most two odd degree vertices.

Question: is the converse also true? i.e

If G has at most two odd degree vertices, then must G have an Eulerian tour?

Finding Eulerian tours

Theorem: If G has an Eulerian tour, G has at most two odd degree vertices.

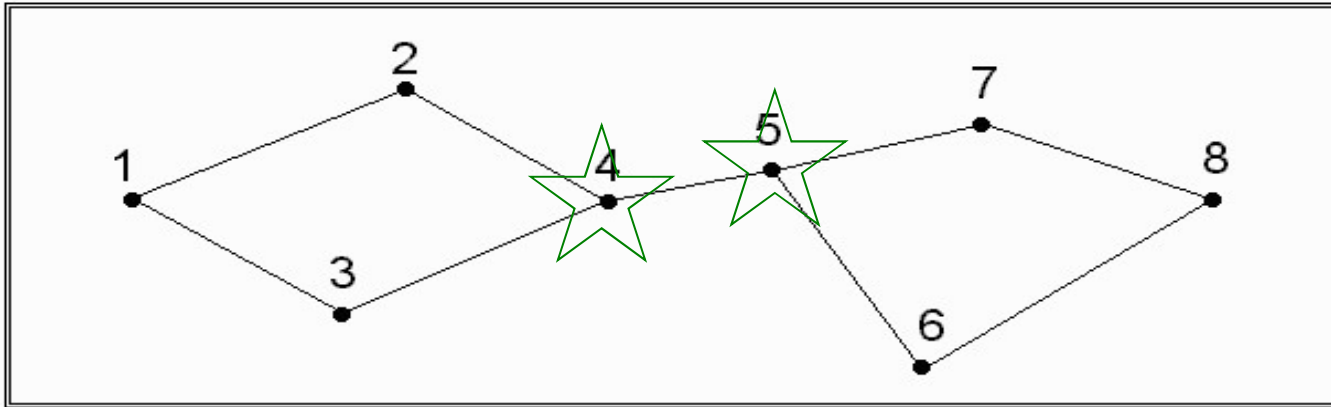
Question: is the converse also true? i.e

If G has at most two odd degree vertices, then must G have an Eulerian tour?

Answer: give algorithm to build the Eulerian tour!

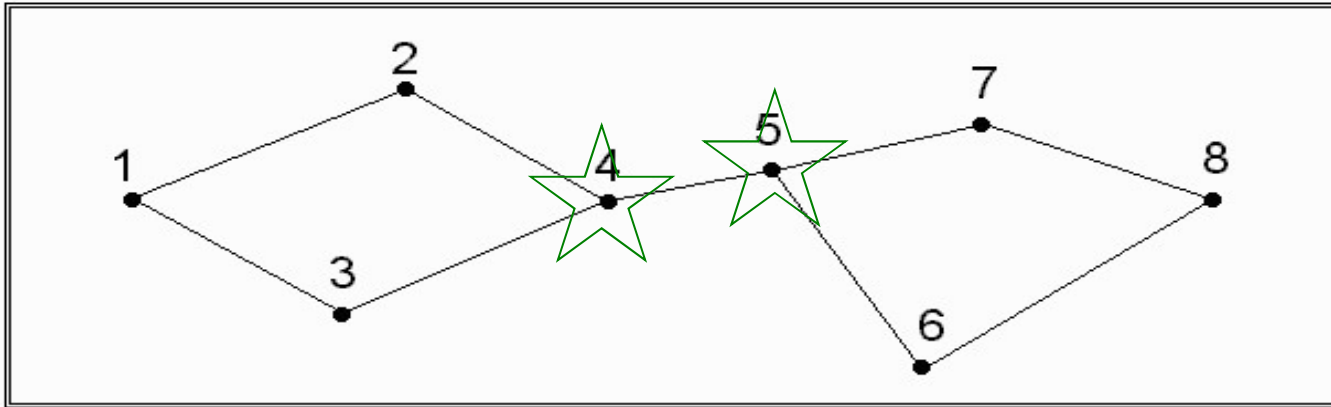
We'll develop some more graph theory notions along the way.

Finding Eulerian tours



Eulerian tour?

Finding Eulerian tours



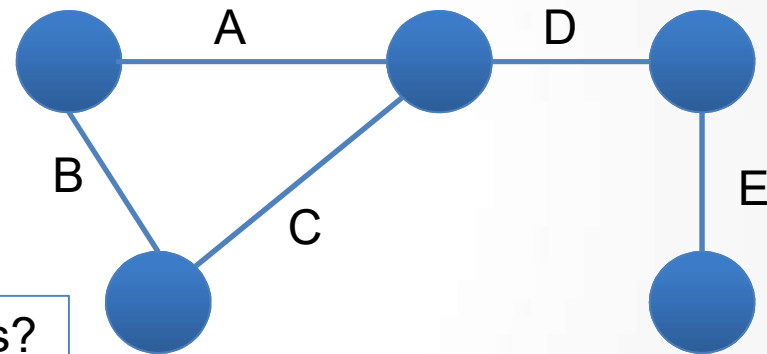
Eulerian tour?

Start at 4. Where should we go next?

- A. Along edge to 2.
- B. Along edge to 3.
- C. Along edge to 5.
- D. Any of the above.

Bridges

A **bridge** is an edge, which, if removed, would cause G to be disconnected.



Which of the edges in this graph are bridges?

- A. All of them.
- B. D, E
- C. A, B, C
- D. C, D
- E. None of the above.

Bridges

A **bridge** is an edge, which, if removed, would cause G to be disconnected.

Connection with Eulerian tours:

In an Eulerian tour, we have to visit **every edge** on one side of the bridge before we cross it (because there's no coming back).

*** Do you see divide & conquer in here?*

Eulerian Tours HOW Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident on v
5. Cross any edge incident on v that is not a bridge and delete it
6. Else,
7. Cross the only edge available from v and delete v .

Eulerian Tours WHY Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. ~~If there is more than one edge incident on v~~
5. Cross any edge incident on v that is not a bridge and delete it
6. Else,
7. Cross the only edge available from v and delete v .

Will there always be such an edge?

Will go through each edge at most once, so if while loop iterates $|E|$ times, get an Eulerian tour.

Fleury's algorithm correctness. invariants

1. If v has odd degree then there is exactly one other vertex w of odd degree. If v has even degree, all other vertices have even degree.
2. G stays connected (deleting edges as we use them.)

Fleury's algorithm correctness. invariants

1. If v has odd degree then there is exactly one other vertex w of odd degree. If v has even degree, all other vertices have even degree.

Proof:

Base Case: since G has at most 2 odd degree vertices and we pick one as v , there must only be one other.

Induction Hypothesis: Suppose after k iterations, the current vertex has odd degree and one other vertex has odd degree or all vertices have even degree.

Call the current vertex v . Then we must cross an edge to get to a neighbor of v call it y . If y is not w then y has even degree and when you delete the edge (v,y) , v now has even degree and y has odd degree and y is the new current vertex.

if y is w then when you delete the edge (v,y) v has even degree and w has even degree so all vertices have even degree.

Fleury's algorithm correctness. invariants

2. G stays connected (deleting edges as we use them.)

Proof:

Base Case:

G starts as being connected.

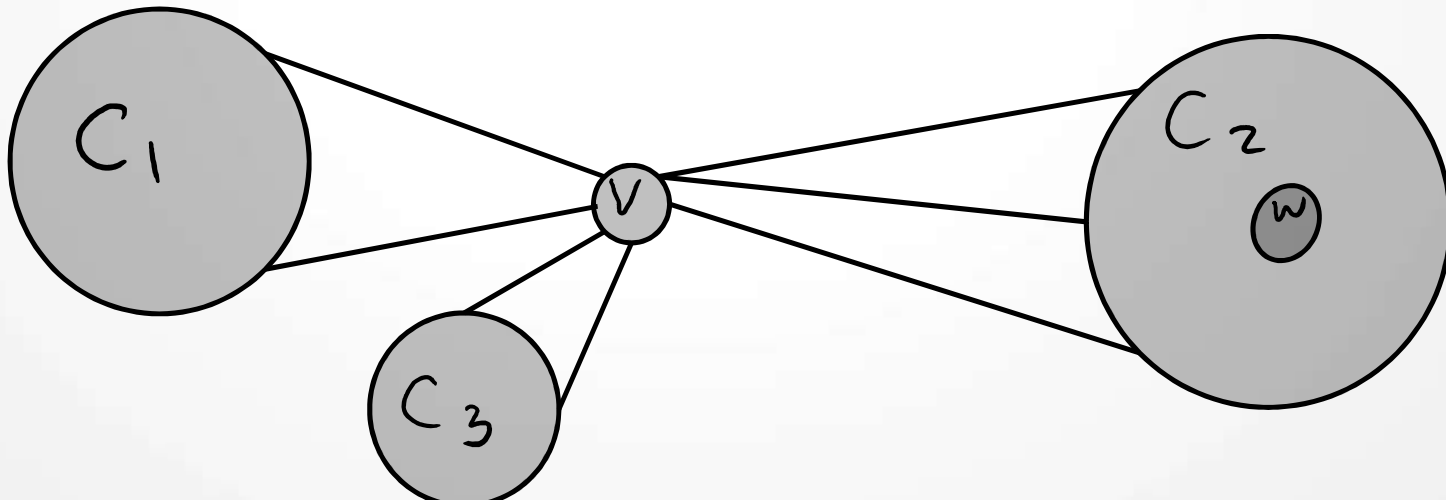
Inductive hypothesis: Assume after k iterations, G is still connected.

If we remove the current vertex v and all the edges from v , the graph could split into connected components C_1, \dots, C_m

Fleury's algorithm correctness. invariants

2. G stays connected (deleting edges as we use them.)

If we remove the current vertex v and all the edges from v , the graph could split into connected components C_1, \dots, C_m

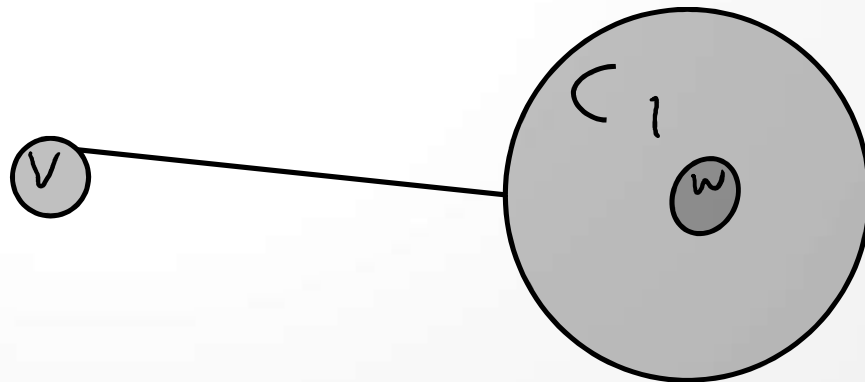


Fleury's algorithm correctness. invariants

2. G stays connected (deleting edges as we use them.)

If we remove the current vertex v and all the edges from v , the graph could split into connected components C_1, \dots, C_m

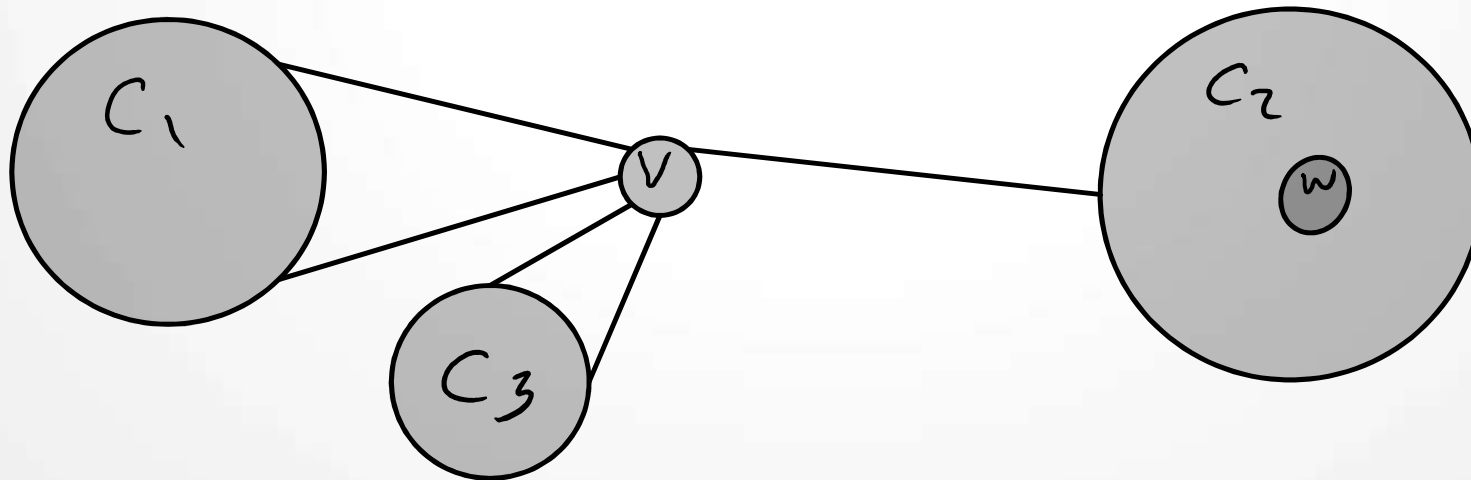
Case 1: There is a single component, $m=1$ and there is only one edge from v . Then G stays connected because it becomes C_1 and we delete v after the one edge is removed.



Fleury's algorithm correctness. invariants

2. G stays connected (deleting edges as we use them.)

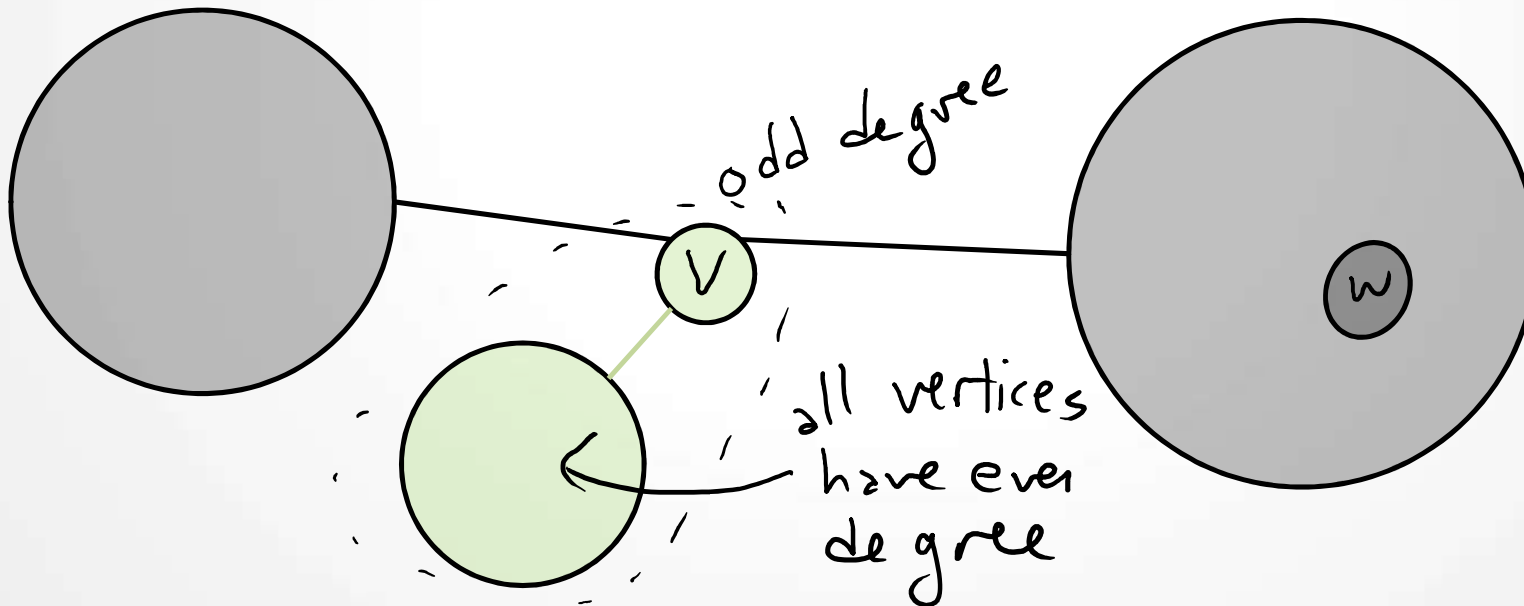
Case 2: There is a component C_i with at least 2 edges from v . Then neither is a bridge and Fleury's algorithm can take any of these edges and the graph will stay connected.



Fleury's algorithm correctness. invariants

2. G stays connected (deleting edges as we use them.)

Case 3: There are at least 2 components and all have single edges from v .
THIS CASE CANNOT HAPPEN!!!!!!!



Fleury's algorithm correctness. invariants

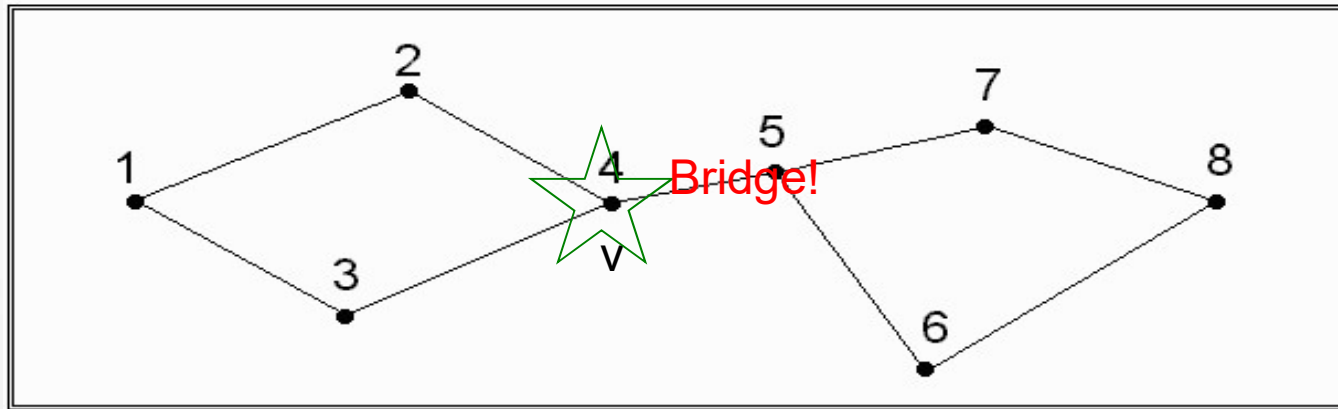
2. G stays connected (deleting edges as we use them.)

Case 3: There are at least 2 components and all have single edges from v .
THIS CASE CANNOT HAPPEN!!!!!!!

w would be in at most one of the at least two components. So pick a component C_i so that there is one edge from v to C_i which has no odd degree vertices. Consider the subgraph consisting just of v and C_i . This graph has only one odd degree vertex. But this contradicts the handshake lemma!!!!

Fleury's algorithm correctness. invariants

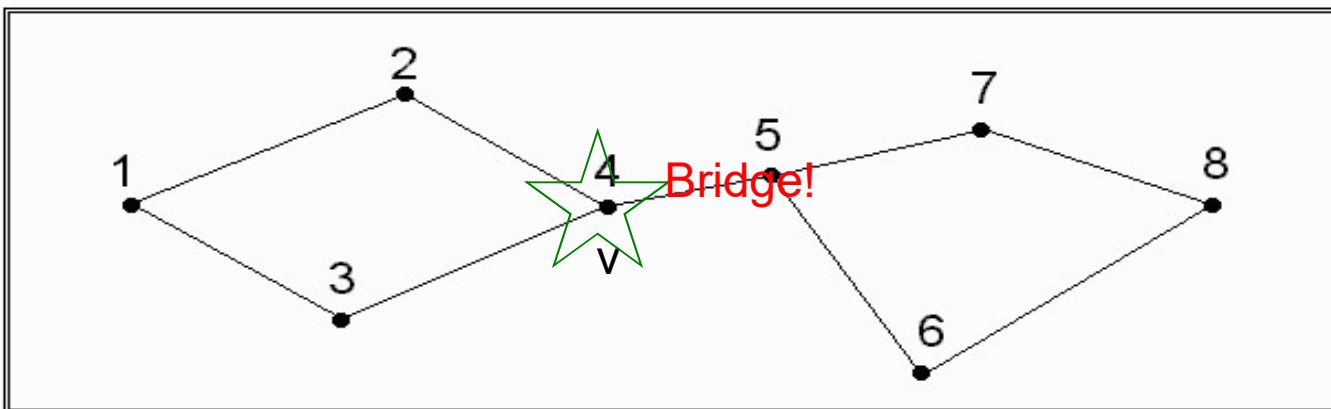
Conclusion: Throughout the algorithm, the graph stays connected. In particular, we end when there are no edges leaving our current vertex v , which means there cannot be any other vertices or the graph would be disconnected. So we have deleted all edges, and hence gone over each edge exactly once.



Eulerian Tours WHY Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident with v
5. Cross any edge incident with v that is not a bridge
6. Else, cross the only edge available from v .
7. Delete the edge just crossed from G , update v .

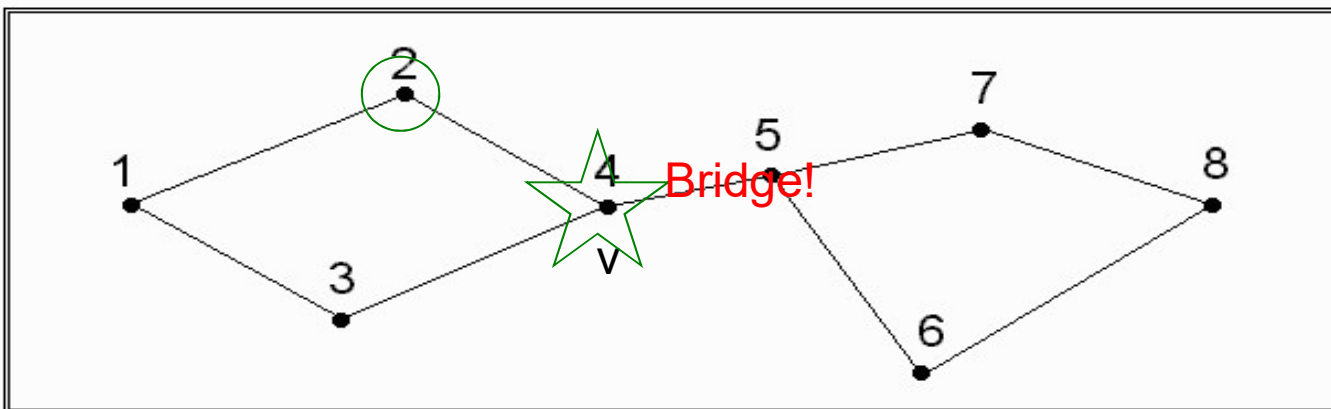
Will there always be such an edge?



Eulerian Tours WHY Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident with v
5. Cross any edge incident with v that is not a bridge
6. Else, cross the only edge available from v .
7. Delete the edge just crossed from G , update v .

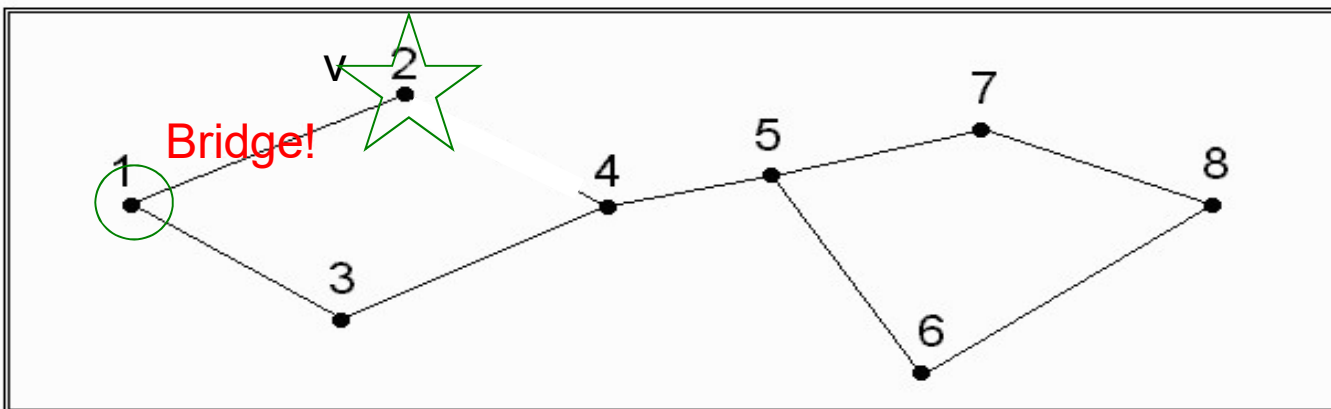
Will there always be such an edge?



Eulerian Tours WHY Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident with v
5. Cross any edge incident with v that is not a bridge
6. Else, cross the only edge available from v .
7. Delete the edge just crossed from G , update v .

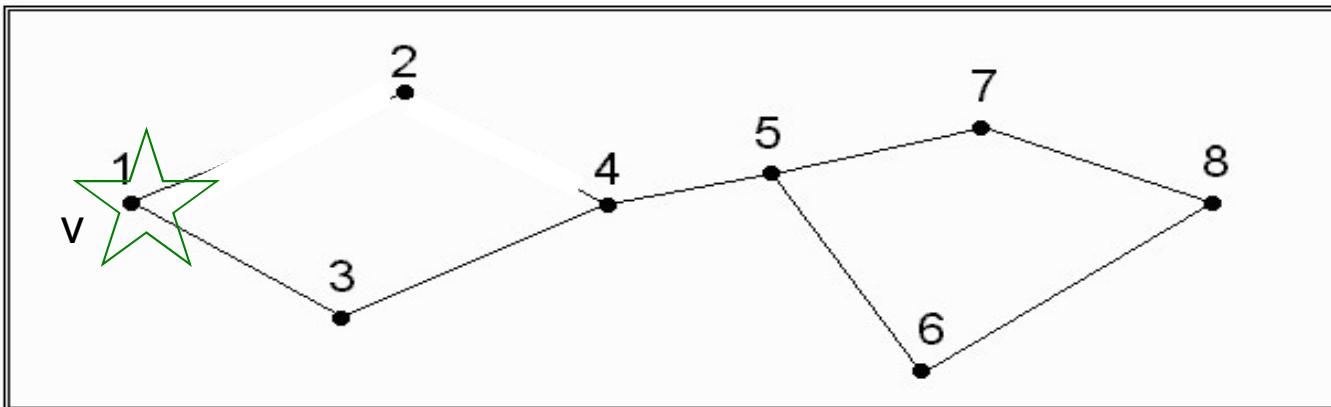
Will there always be such an edge?



Eulerian Tours WHY Fleury's Algorithm

1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident with v
5. ~~Cross any edge incident with v that is not a bridge~~
6. ~~Else, cross the only edge available from v .~~
7. Delete the edge just crossed from G , update v .

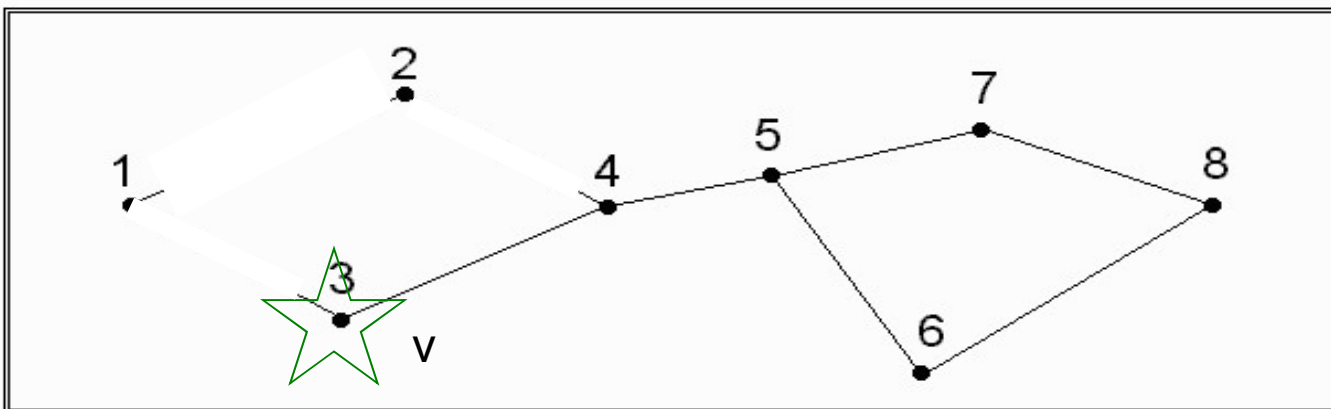
Will there always be such an edge?



Eulerian Tours WHY Fleury's Algorithm

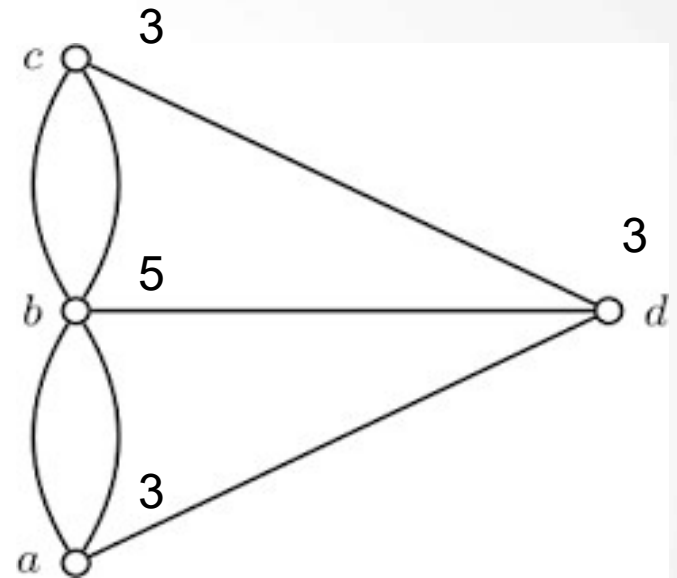
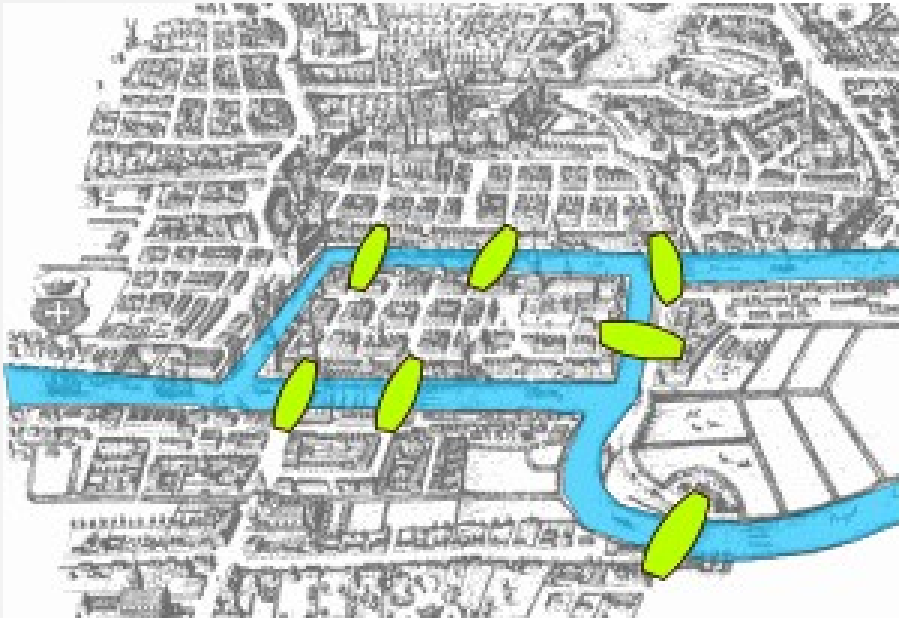
1. Check that G has at most 2 odd degree vertices.
2. Start at vertex v , an odd degree vertex if possible.
3. While there are still edges in G ,
4. If there is more than one edge incident with v
5. Cross any edge incident with v that is not a bridge
6. Else, cross the only edge available from v .
7. Delete the edge just crossed from G , update v .

Will there always be such an edge?



etc.

Seven Bridges of Königsberg

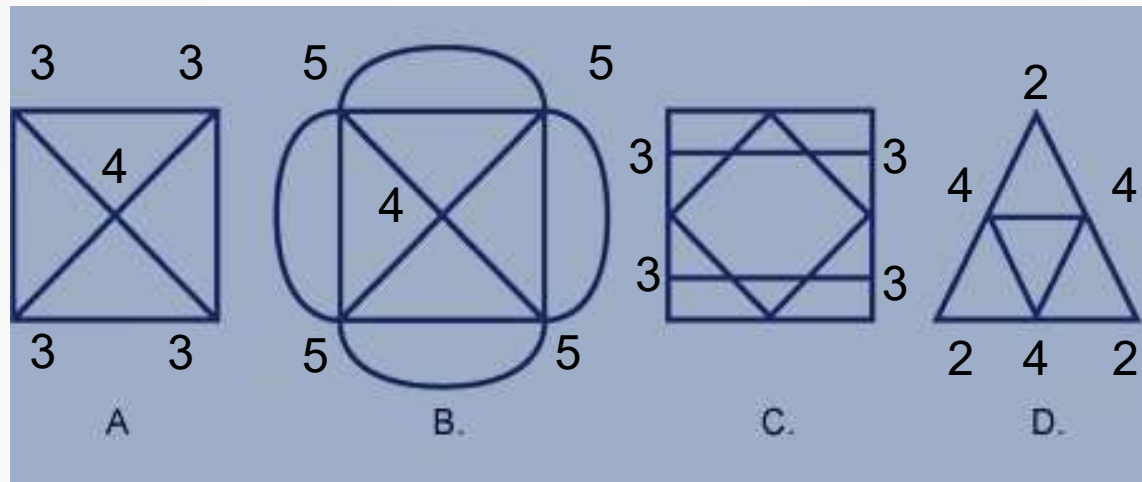


Is there a path where each **edge** occurs exactly once? **Eulerian tour**

No!

Rosen p. 693

Seven Bridges of Königsberg redux



Which of these puzzles can you draw without lifting your pencil off the paper?

- A. No
- B. No
- C. No
- D. Yes

Eulerian Tours: recap

Consider only **undirected** graphs.

1st goal: Determine whether a given undirected graph G has an Eulerian tour.

G has an Eulerian tour if and only if G has at most 2 odd-degree vertices.

2nd goal: Actually find an Eulerian tour in an undirected graph G , when possible.

Fleury's Algorithm: don't burn your bridges.

Eulerian Tours: recap

Number of odd degree vertices	0	1	2	>2, odd	>2, even
Is such a graph possible?	yes	no	yes	no	yes
Is an Eulerian tour possible?	yes, an Eulerian circuit		yes		no