Eulerian Tours and Fleury’s Algorithm

<table>
<thead>
<tr>
<th>Lecture A</th>
<th>Tiefenbruck</th>
<th>MWF 9-9:50am</th>
<th>Center 212</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture B</td>
<td>Jones</td>
<td>MWF 2-2:50pm</td>
<td>Center 214</td>
</tr>
<tr>
<td>Lecture C</td>
<td>Tiefenbruck</td>
<td>MWF 11-11:50am</td>
<td>Center 212</td>
</tr>
</tbody>
</table>

http://cseweb.ucsd.edu/classes/wi16/cse21-abc/

February 3, 2016
Finding Eulerian tours

Consider only **undirected** graphs.

1\textsuperscript{st} goal: Determine whether a given undirected graph $G$ has an Eulerian tour.

2\textsuperscript{nd} 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 B?

A. None.
B. Exactly one.
C. Exactly two.
D. More than two.
E. None of the above.
Finding Eulerian tours

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

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

inf many
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.
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 **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.
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.

1$^{st}$ goal: Determine whether $G$ has an Eulerian tour.

2$^{nd}$ goal: If yes, find the tour itself.
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.

Rosen p. 652
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 $p$.

* not the start vertex, not the end vertex.
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 $p$.

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 **equivalent** to the theorem (using the facts we know so far about graphs)?

A. If the number of odd degree vertices in \( G \) is anything other than 0 or 2, then \( G \) has no Eulerian tour.
B. If \( G \) has three or more odd degree vertices, then \( G \) does not have an Eulerian tour.
C. If \( G \) has an Eulerian tour, then \( G \) has either all vertices with even degree or \( G \) has exactly two vertices with odd degree.
D. All of the above.
E. None of the above.
**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.
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?

add most 2 odds
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.
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.
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?**
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
6. Else, cross the only edge available from v.
7. Delete the edge just crossed from G, update v.
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
6. Else, cross the only edge available from v.
7. Delete the edge just crossed from G, update v.

What Eulerian tour do you get when following Fleury's algorithm, starting at vertex 4?
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?

Will go through each edge at most once, so if while loop iterates |E| times, get an Eulerian tour.
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,
   Cross any edge incident with v that is not a bridge
5. Else, cross the only edge available from v.
6. 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.
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?

Need to show (loop invariant): Connected graph with no more than one bridge from v.

If we enter the while loop, then after t^{th} iteration of while loop,
• the (remaining) graph is still connected, AND
• there is at most one other odd degree vertex in the (remaining) graph other than v, AND
• across every bridge from v there is an odd degree vertex (and hence there is at most one bridge incident with 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 with v, Cross any edge incident with v that is not a bridge
5. Else, cross the only edge available from v.
6. Delete the edge just crossed from G, update v.

Why is more than one bridge from v bad?
Is there a path where each edge occurs exactly once? **Eulerian tour**

No!
Which of these puzzles can you draw without lifting your pencil off the paper?

A. No
B. No
C. No
D. Yes
Consider only **undirected** graphs.

1\textsuperscript{st} goal: Determine whether a given undirected graph \( G \) has an Eulerian tour.

\textbf{G has an Eulerian tour if and only if \( G \) has at most 2 odd-degree vertices.}

2\textsuperscript{nd} goal: Actually find an Eulerian tour in an undirected graph \( G \), when possible.

\textbf{Fleury's Algorithm: don't burn your bridges.}
## Eulerian Tours: recap

<table>
<thead>
<tr>
<th>Number of odd degree vertices</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>&gt;2, odd</th>
<th>&gt;2, even</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is such a graph possible?</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Is an Eulerian tour possible?</td>
<td>yes, an Eulerian circuit</td>
<td>yes</td>
<td></td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>