“Notes” slides from before lecture

CSE 21, Winter 2017, Section A00

Lecture 8 Notes

Class URL: http://vlsicad.ucsd.edu/courses/cse21-w17/
Notes February 6 (1)

- HW4 is now due on FRIDAY (extended by a day)
- Graded HW3 appears delayed beyond tonight’s 5-day return target
- MT1 should be back to you this week (target = Wednesday night)
- Slides for Days 12, 13 (this week) have been posted
- Today: Day 11 + 12 of posted slides
- Any logistic, other issues?
Seven Bridges of Konigsberg

Is there a path where each edge occurs exactly once? **Euler (or, Eulerian) tour**

Rosen p. 693
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!!!
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!

Notation: 0 = down; 1 = up

All the switches are down. How many times must you press a switch if you want to guarantee that the light bulb will turn on?

A. 4  B. 7  C. 8  D. 16

"Gray Code"

Only one bit flip (≈ pressing a switch) between adjacent configurations
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. (starting at the “000” vertex)
Our Strategy

Puzzle / Problem

Model as a graph
Choose vertex set & edge set ... sometimes many possible options

Use graph algorithms to solve puzzle / problem

A general approach to solving problems modeled/abstracted using graphs

Representation is key... (will see this in next example)
DNA Reconstruction

Problem

Given collection of short DNA strings.

(Shortest possible)

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.
Given collection of short DNA strings. S = { ATG, AGG, TGC, TCC, GTC, GGT, GCA, CAG } Find longer DNA string that includes them all.

A possible “representation”??????

**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

* Intuitively: short strings can be overlapped in the longer (containing) string...
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?**

Is a Hamiltonian tour meaningful?

What if we had more edges, with weights?
Given collection of short DNA strings.
S = { 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?

For a given set S of DNA strings, will such a graph always have an Eulerian tour? How about S = {AAA, GGG}?
Algorithmic questions related to tours

An **Eulerian tour** is a path where each **edge** occurs exactly once.
- **Eulerian cycle**: Eulerian tour that starts and ends at same vertex.

A **Hamiltonian tour** is a path where each **vertex** occurs exactly once.
- **Hamiltonian cycle**: Ham. tour that starts and ends at same vertex.

Which is true of the graph to the right?
- A. There's only one Eulerian cycle in this graph.
- B. There's no Hamiltonian tour in this graph.
- C. Each Hamiltonian tour is an Eulerian cycle.
- D. None of the above.
- E. More than one of the above.

Note: Other classes may use terminology whereby “tour” = “cycle” = “circuit”
Number of Paths

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.

A - B
A - C - B
A - C - D - F - E - B
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.
Number of Paths

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.

A. None.

no possible connection...
("unreachable")
An undirected graph $G$ is **connected** if for any ordered pair of vertices $(v, w)$ there is a path from $v$ to $w$. (i.e., "every")

**Connected**

**Not connected**
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 (maximal) connected piece of the graph is a connected component.

“This graph has 3 connected components.”

“maximal” means that if you make it any larger, it will lose the given (in this case, “connected”) property.
Eulerian Tours and Fleury’s Algorithm

CSE21 Winter 2017, Day 8 (A00)
February 8, 2017

http://visicad.ucsd.edu/courses/cse21-w17

On Wednesday 2/8, we will go through Fleury’s Algorithm “quickly” – please read the Day 12 and Day 13 slides in advance. Thanks!!
Vocabulary

Path (or walk): describes a route from one vertex to another
\((v_1, e_1, v_2, e_2, \ldots, v_k)\)

Length of path: number of edges

Simple path: path that doesn’t repeat vertices

Circuit (or cycle or closed walk): path that starts and ends at the same vertex, with length greater than zero

Loop (or self-loop): an edge from a vertex to itself
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.
Disconnected graphs can be broken up into pieces where each is connected.

Each connected piece of the graph is a connected component.

Does this graph have an Eulerian tour?
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.
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** \( v \).

An Eulerian tour contains all edges incident on \( v \).
So, half of the edges are used to enter \( v \), half to leave.

* 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** \( v \).

An Eulerian tour contains all edges incident on \( v \).
So, half of the edges are used to enter \( v \), half to leave.

* **Degree of** \( v \) **must be even!**

* not the start vertex, not the end vertex.
Finding Eulerian tours

(Summary of) Observation:

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

In a circuit, all vertices are intermediate so all have even degree.

In a tour starting and ending at different vertices (not a circuit), the starting and ending vertices will have odd degree; all others will have even degree.
[Side note: remember “contrapositive”?

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p → q</th>
<th>¬q → ¬p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
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.
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?

*We’ll see!!! (Fleury’s Algorithm)*