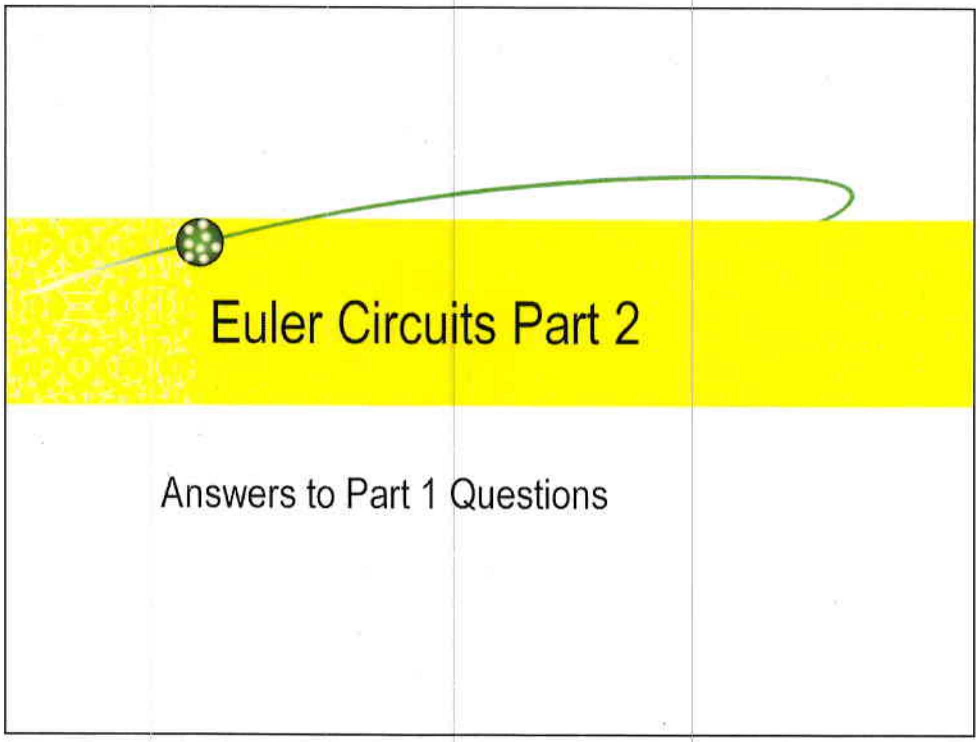


Chapter 5b.ppt

5.5 – 5.7

- Now we'll move forward, beginning to answer the questions we left off with.
-

Euler's Circuit Theorem

- If a graph is *connected* and every vertex is *even*, then it has an Euler Circuit
- If a graph has *any* odd vertices, then it does not have an Euler Circuit.

Remind me – what is an Euler circuit?

Remember what connected means – no pieces dangling.

If *every vertex* is even, there is an Euler Circuit. If *any vertex* is odd, there is no Euler circuit.

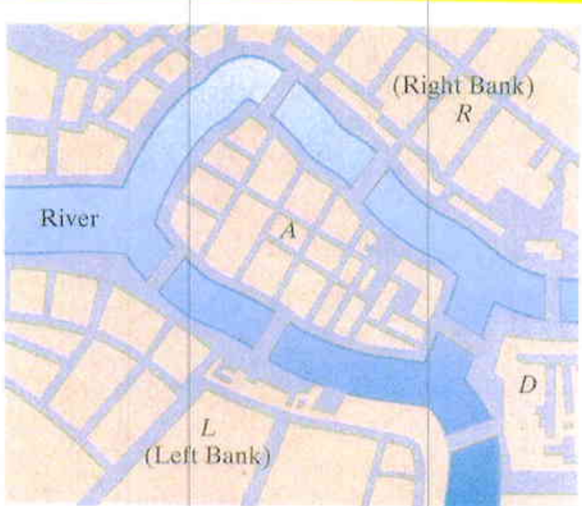
Euler's Path Theorem

- If a graph is *connected* and has exactly *two* odd vertices, then it has an Euler Path. Any such path must begin at one of the odd vertices and end at the other
- If a graph has *more than two* odd vertices, then it does not have an Euler Path.

Once again, what is an Euler Path?

This theorem tells us that if *exactly two vertices* are odd, then the graph has an Euler path. If the graph has *more than two odd vertices*, then there is no Euler path. Why two odd vertices? One to start and one to end the path.

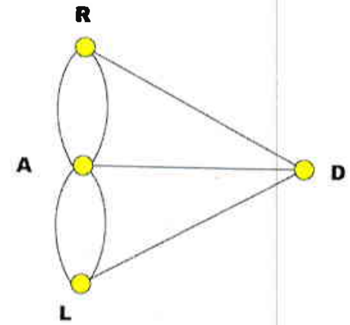
Königsberg Again



Let's go back to Königsberg again. We saw it could be represented by this graph:

Königsberg Again

- The bridges in the old city can be modeled by this graph. What is the degree of each vertex?



R	***3***
A	***5***
L	***3***
D	***3***

What is the degree of each vertex?

R?

A?

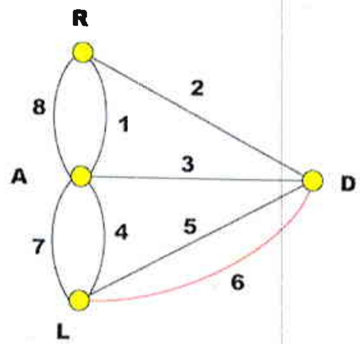
L?

D?

There are 4 odd vertices – no wonder the residents couldn't tour the city. There are neither Euler Circuits nor Euler Paths.

Königsberg Again

- To start and end in different places, we must recross one bridge.



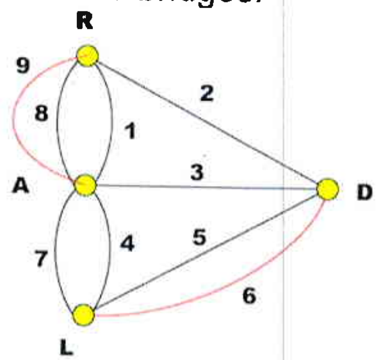
A-R-D-A-L-D—L-A-R

If it is OK to start and end in different places, it is sufficient to recross one bridge. Why?

Which bridges could we recross?

Königsberg Again

- To start and end at the same vertex, we must recross two bridges.



A-R-D-A-L-D—L-A-R—A

To solve the original problem, starting and ending in the same place, two bridges must be recrossed? Why does this work?

The process of allowing recrossings is called Eulerizing a graph. In effect it is adding multiple edges. We'll look at Eulerizing in greater detail later.

Euler's Sum of Degree Theorem

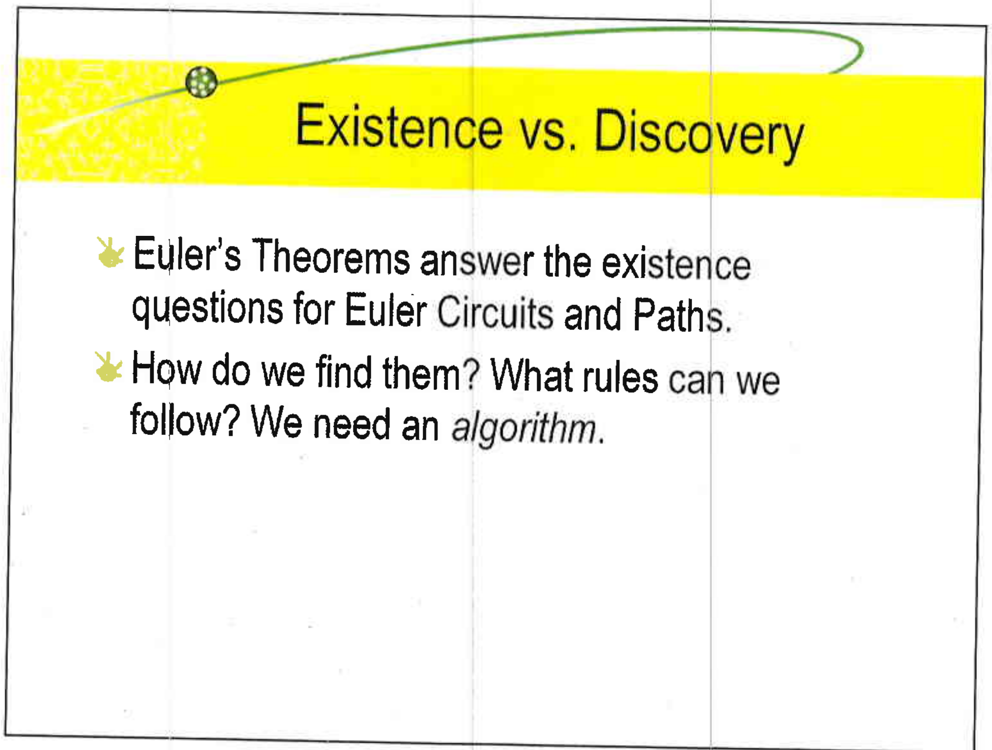
- The sum of the degrees of all the vertices of a graph equals twice the number of edges (and is therefore even)
- A graph always has an even number of *odd* vertices

Euler had some more theorems. They're very logical but were an important breakthrough in graph theory.

- The sum of the degrees of all the vertices of a graph equals twice the number of edges (and is therefore even)

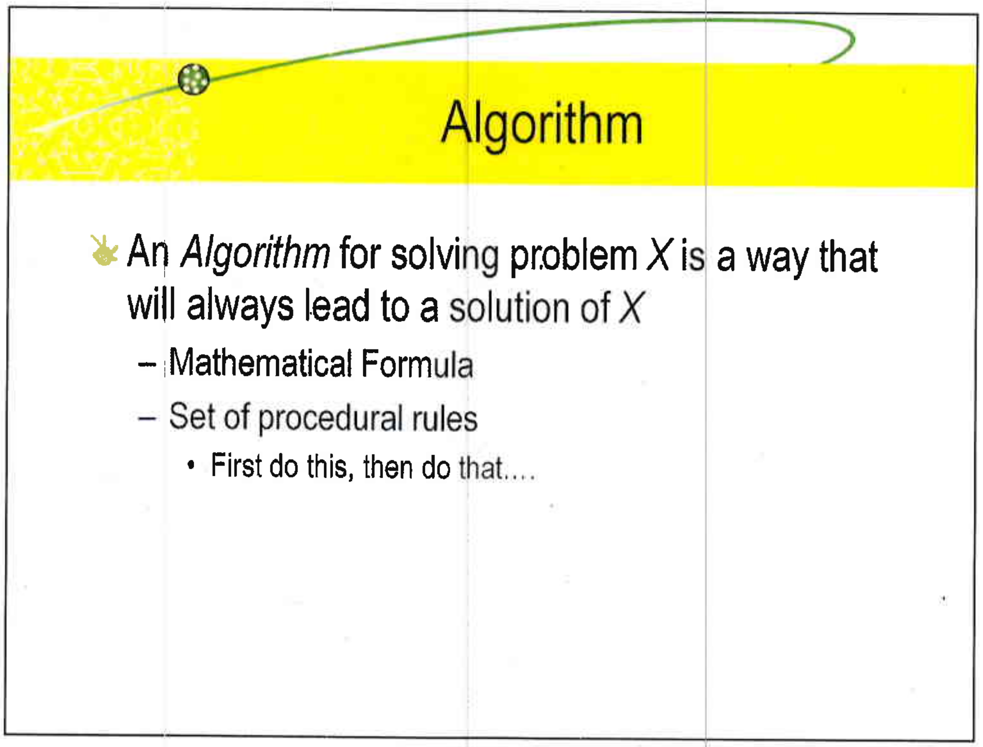
- A graph always has an even number of *odd* vertices

Another way to say it: odd vertices always come in pairs.



Existence proofs in math don't often help us find what exists. The theorem tells us it is there, but not how to find it.

To find Euler paths and circuits in very simple graphs, we can use trial and error. But for more complicated graphs, we need an *algorithm*.



An algorithm can either be a mathematical formula ($A = \pi r^2$ will *always* find the area of a circle) or it can be step-by-step instructions.

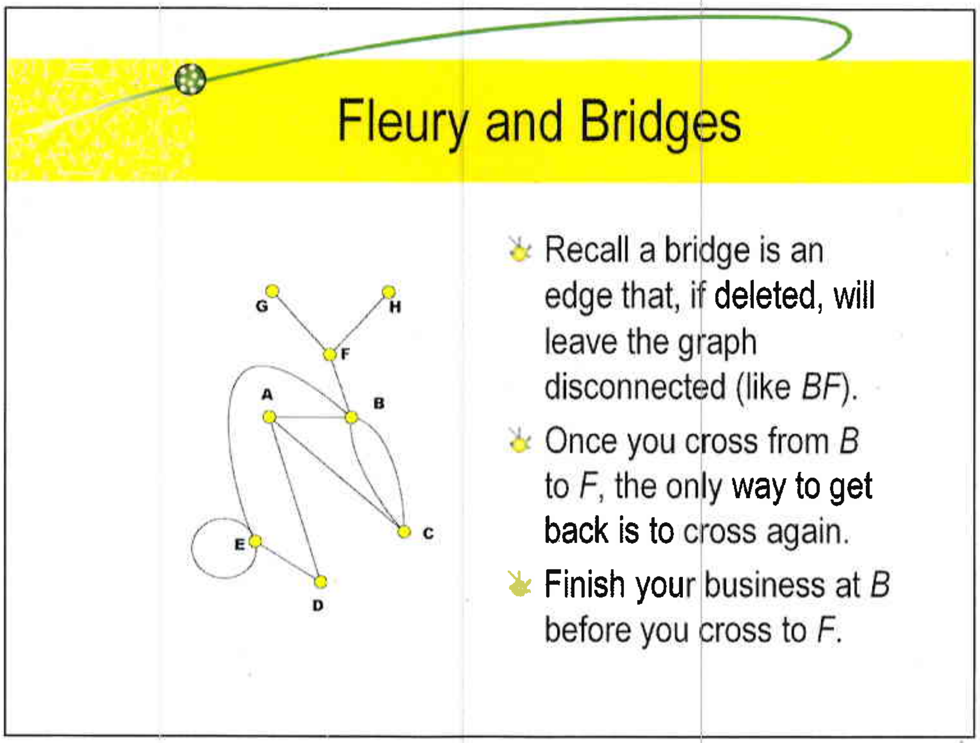
Fleury's Algorithm

- Allows us to find an Euler Circuit in a connected graph with no odd vertices, and
- Allows us to find an Euler Path in a connected graph with exactly two odd vertices.

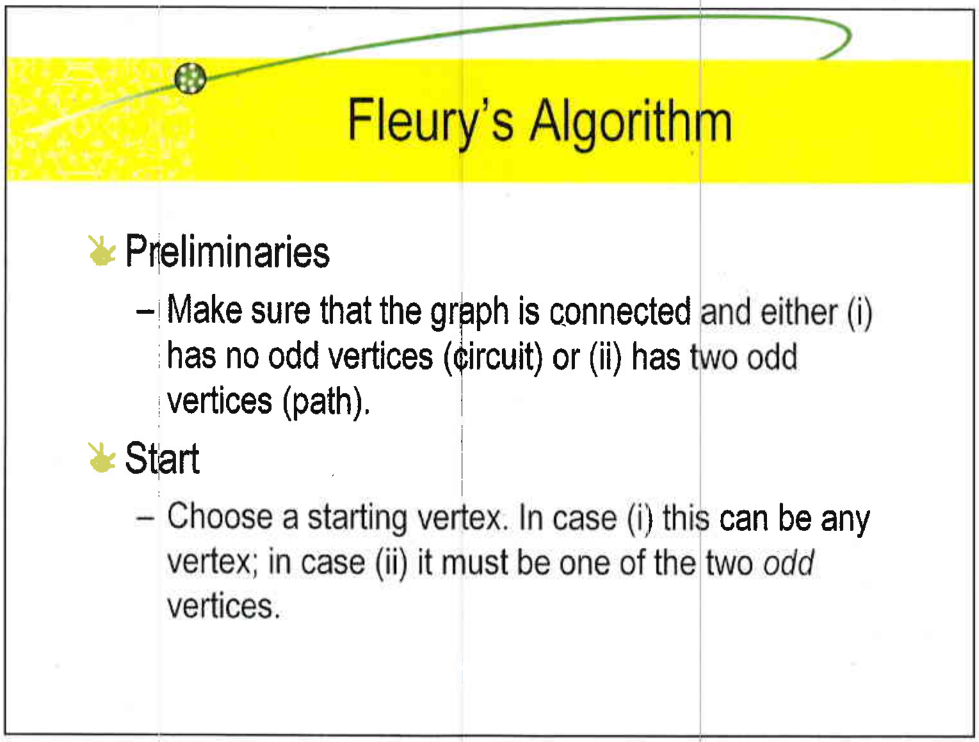
Don't burn your bridges behind you!

Fleury's algorithm will do the job for us. It can be summarized as Don't burn your bridges behind you.

Remind me: what is a bridge?



In short: recall a bridge is an edge that, if deleted, will leave the graph disconnected (like *BF*). Once you cross from *B* to *F*, the only way to get back is to cross again. Finish your business at *B* before you cross to *F*.

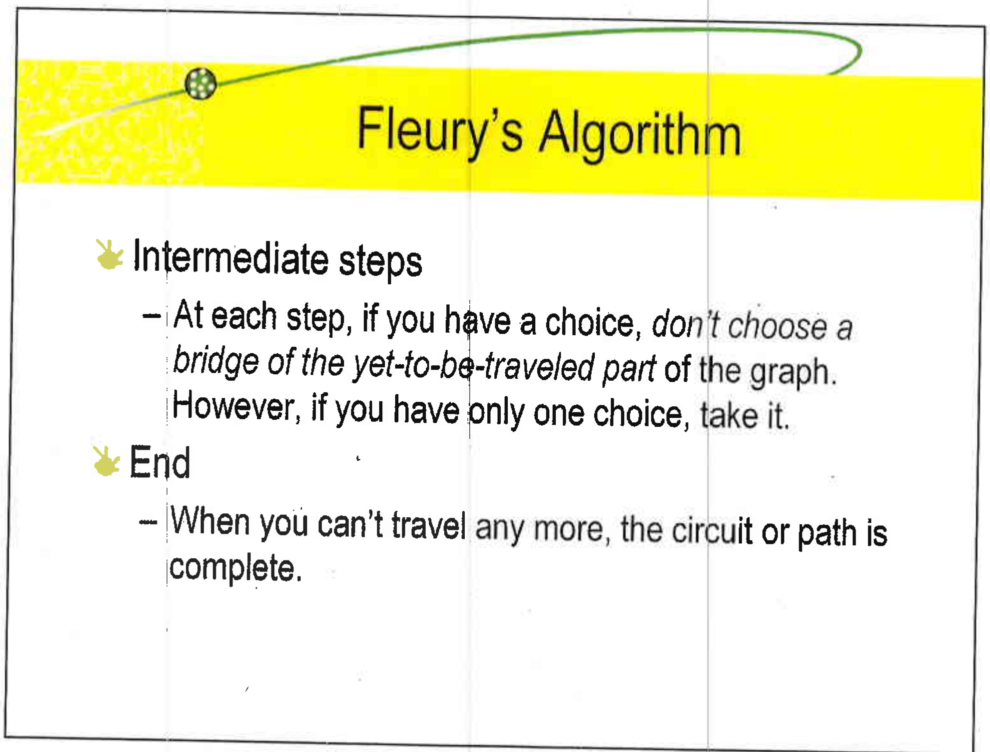


Preliminaries

Make sure that the graph is connected and either (i) has no odd vertices (circuit) or (ii) has two odd vertices (path). Why is this important?

Start

Choose a starting vertex. In case (i) this can be any vertex; in case (ii) it must be one of the two *odd* vertices. In (ii), why must it be an odd vertex?



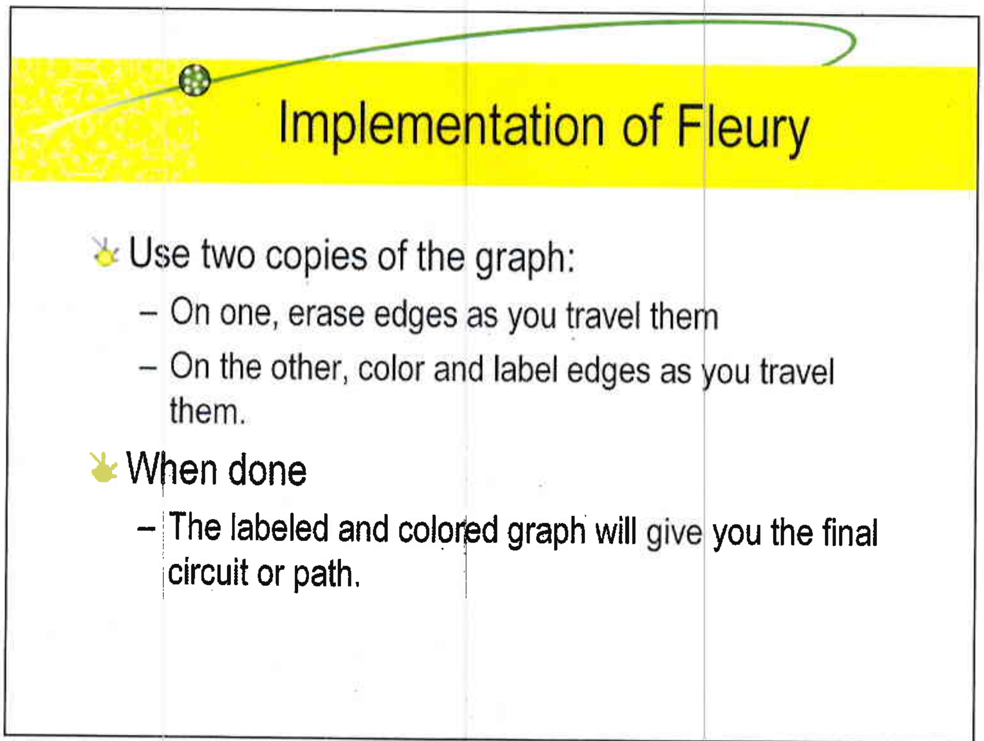
Intermediate steps

At each step, if you have a choice, *don't choose a bridge of the yet-to-be-traveled part* of the graph. However, if you have only one choice, take it.

End

When you can't travel any more, the circuit or path is complete.

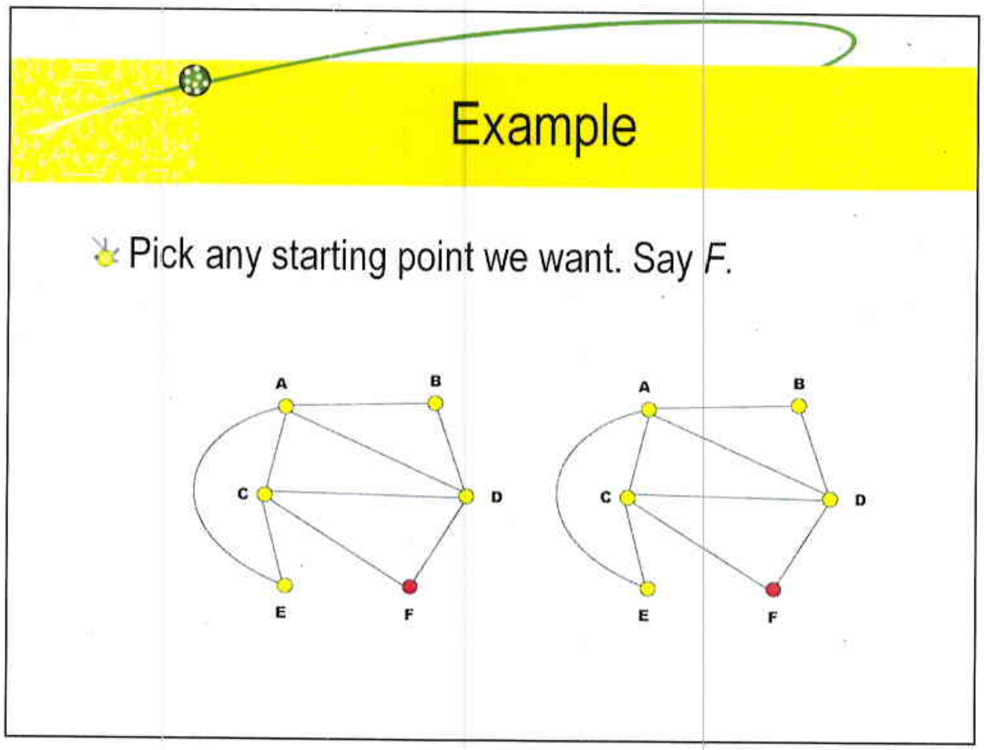
Note that we are always looking at the part of the graph we have *not yet traveled*. We can rephrase by saying "Don't cross a bridge until you are forced to do so."



Implementation is not part of the algorithm, but rather a handy way to do the bookkeepinmg.

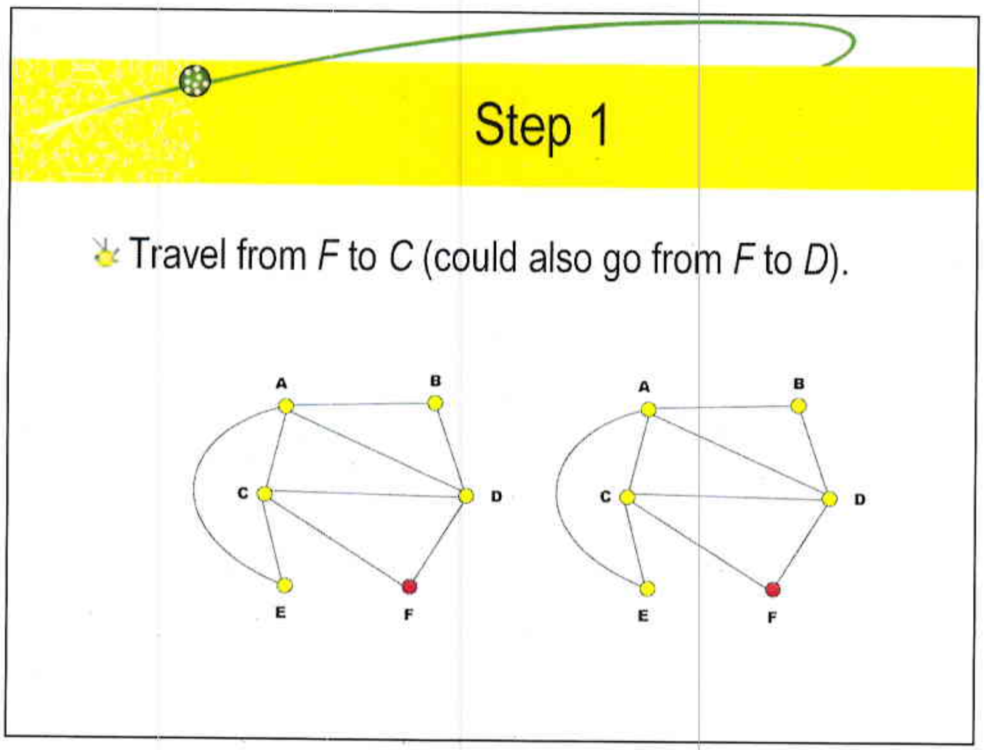
Copy 1: only shows what has not yet been traveled

Copy 2: records the steps we have taken. This could also be accomplished by listing edges as they are traveled.

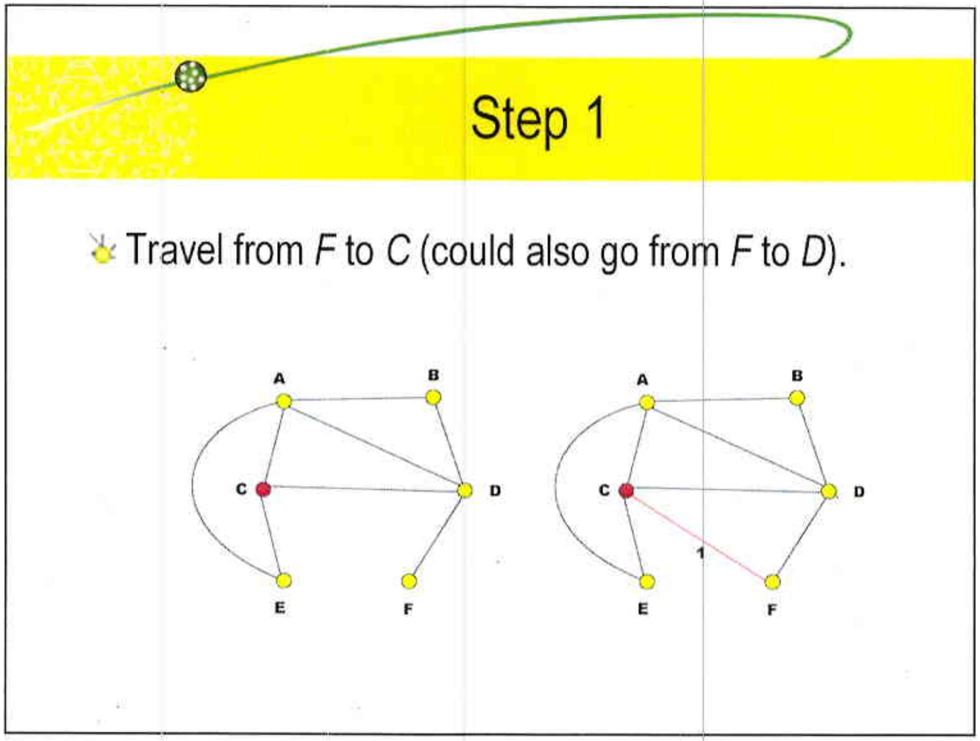


Let's work our way through an example.

Pick any starting point. Say *F*



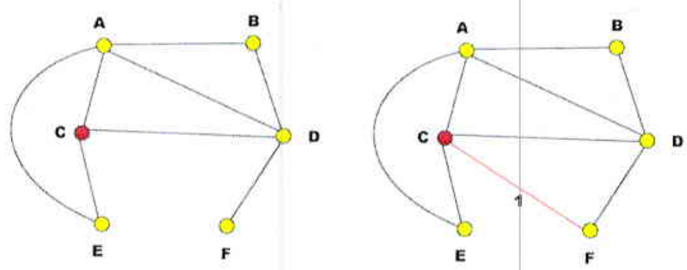
Let's go from F to C. So having decided that, we....



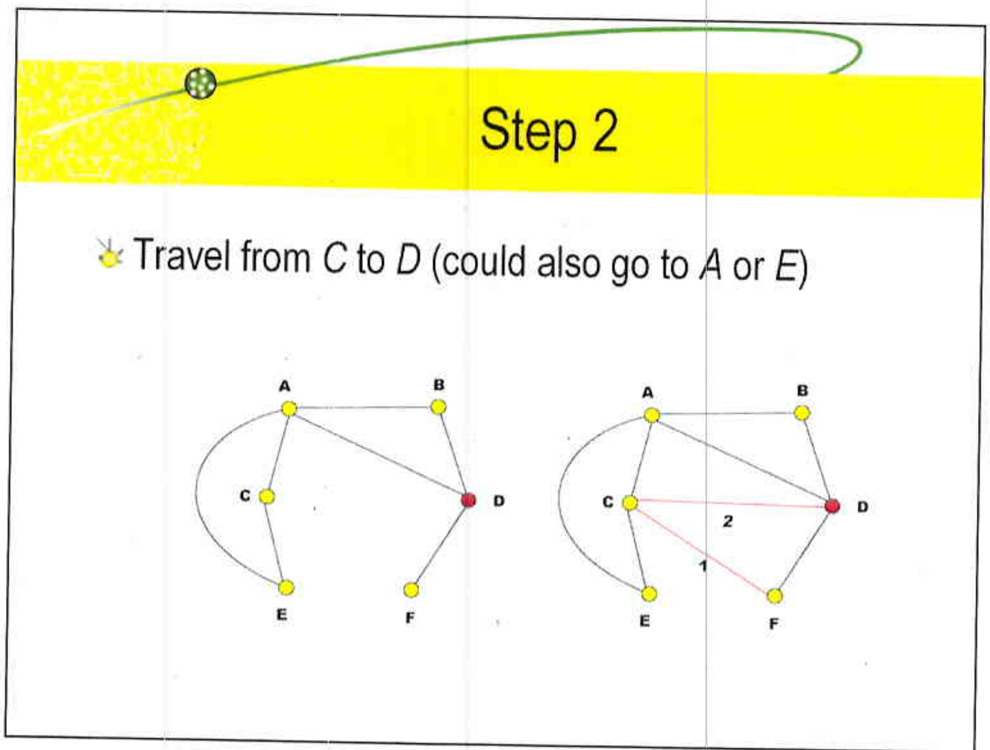
…erase FC on the left graph, and mark it as traveled on the right graph.

Step 2

- Travel from *C* to *D* (could also go to *A* or *E*)



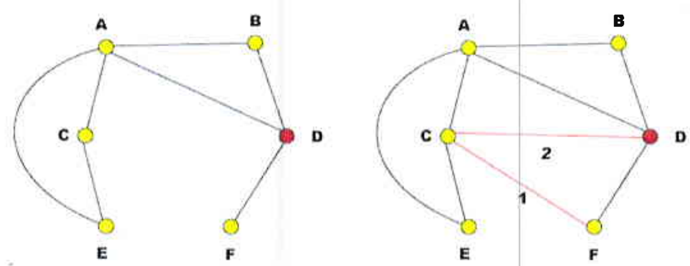
Now we go from C to D., so…



… erase CD on the left graph, and mark it on the right.

Step 3

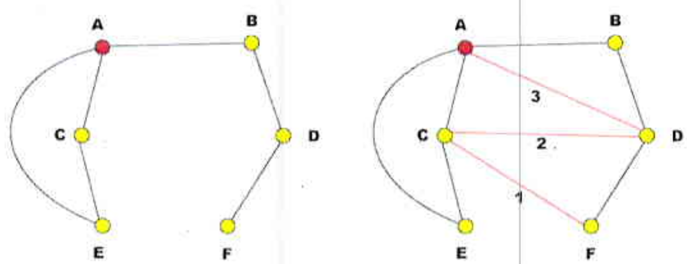
- Travel from *D* to *A* (Could also go to *B*, but not to *F* since *DF* is a bridge)



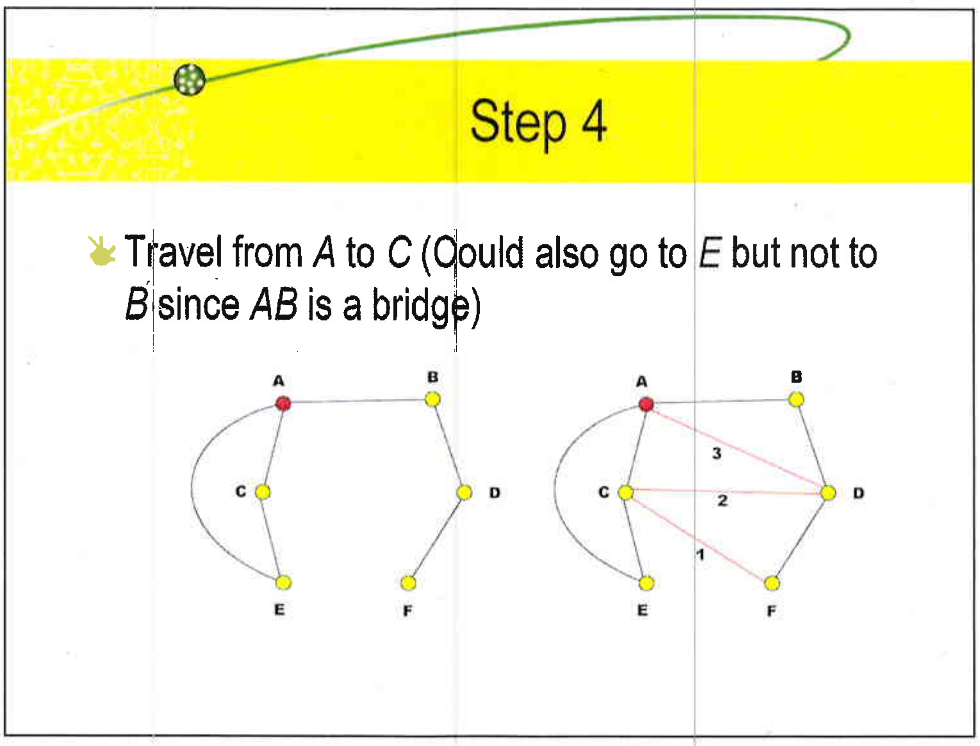
Look carefully at this. We are at D. We can go to A or B, but we cannot go to F since DF is a bridge, and we still have other choices. So, let's go to A. We …

Step 3

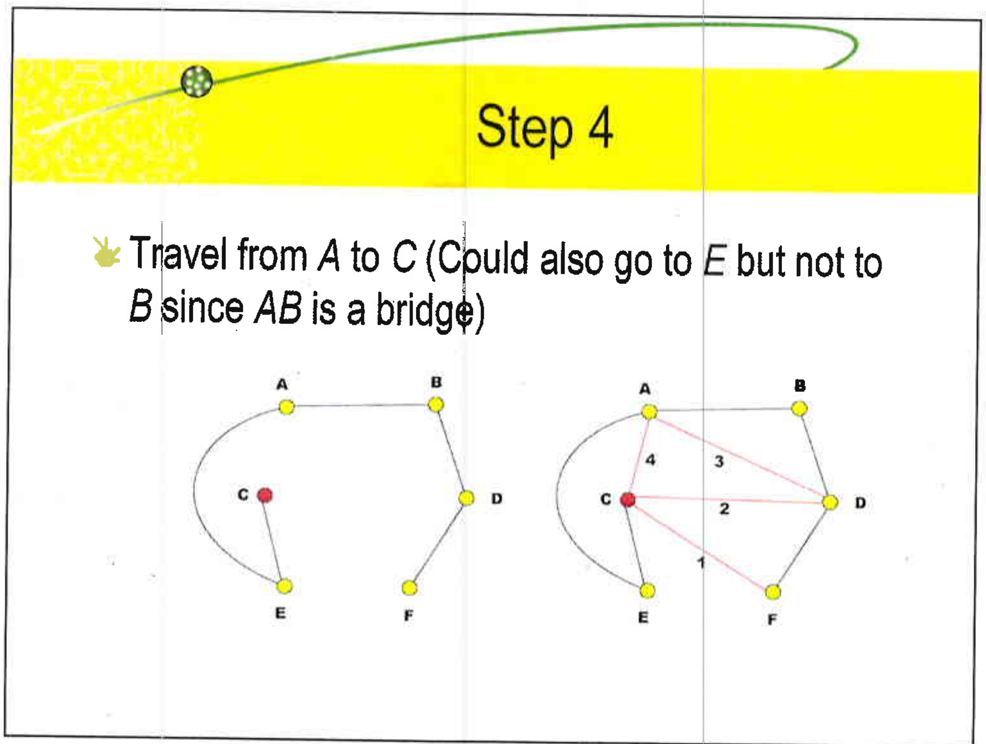
- Travel from *D* to *A* (Could also go to *B*, but not to *F* since *DF* is a bridge)



… erase DA on the left graph, and mark it as traveled on the right.



From A we cannot go to B. Why? Now let's go from A to C. So,

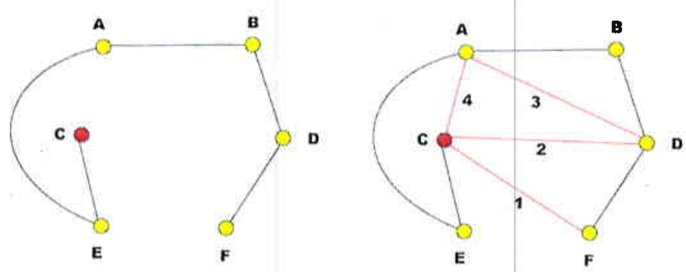


… erase AC on the left, and mark it as traveled on the right.

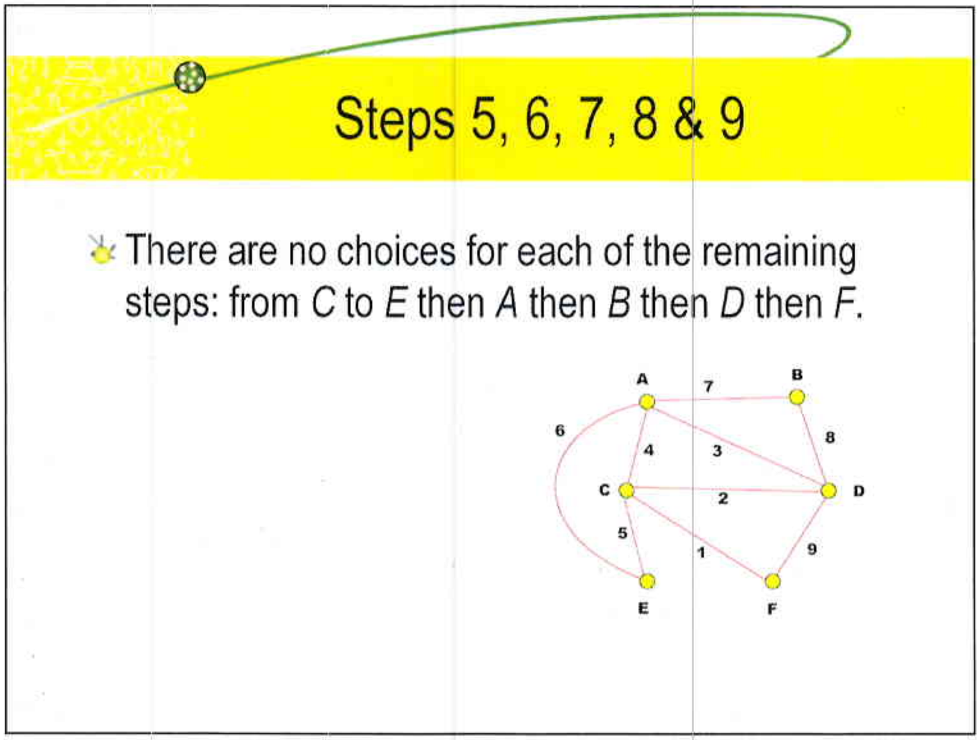
⌘

Steps 5, 6, 7, 8 & 9

- There are no choices for each of the remaining steps: from *C* to *E* then *A* then *B* then *D* then *F*.



At this point we have nothing but a sequence of bridges left: C-E-A-B-D-F. So we can traverse them in order…



… and we find ourselves back at our starting point. We have found an Euler Circuit for this graph:

F-C-D-A-C-E-A-B-D-F

Eulerizing Graphs

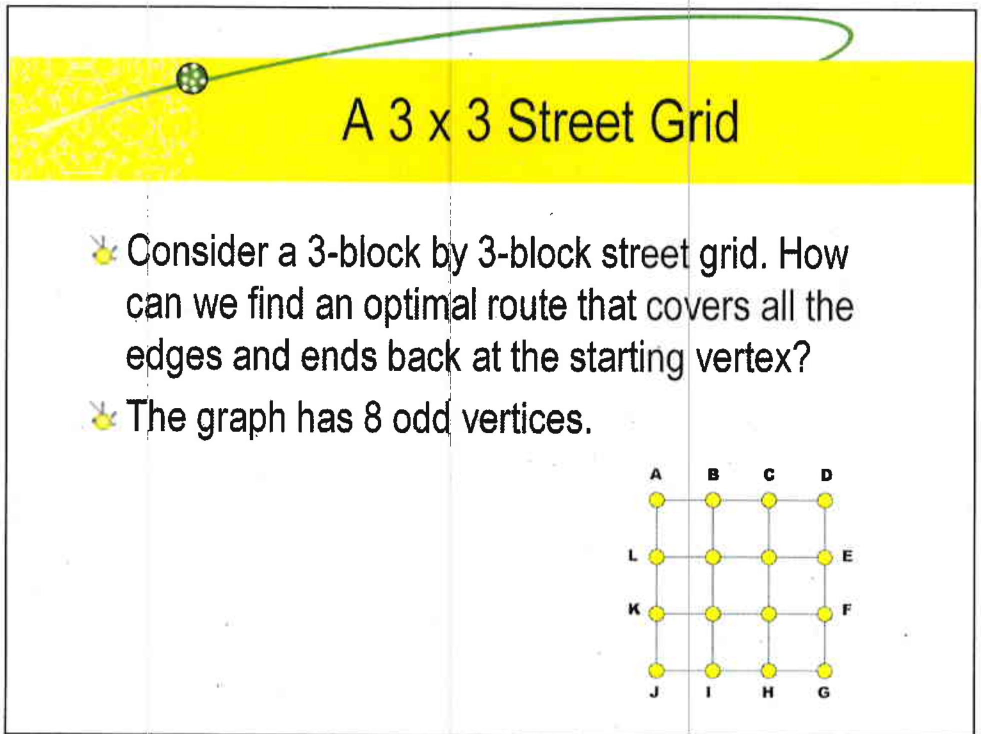
- Finally!
- We begin to answer some of the routing problems from the beginning of the chapter.
- Common thread: find optimal exhaustive routes.
- Exhaustive route: cover each and every edge at least once
- Optimal: minimize deadhead edges (retracing).

Woo – Hoo! We made it.

We begin to answer some of the routing problems from the beginning of the chapter. The common thread: find optimal exhaustive routes. An exhaustive route is one that covers each and every edge at least once. Optimal means minimizing deadhead edges.

A little history: deadheading is an old railroad term which refers to pulling empty cars to move them somewhere else. Actually it derives from Roman times, where a free admission ticket to the theater were small ivory skulls. Thus a non-paying audience member was known as a deadhead.

In any case, we use it to refer to retracing an edge.

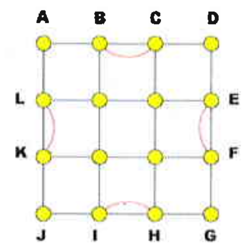


We can work our way up to solving some of our initial problems. Let's look at this street grid. We want to cover each block once.

We notice before we begin that there are 8 odd vertices. Where are they (B C E F H I K L).

A 3 x 3 Street Grid

- When we allow 4 edges to be retraced, then all vertices are even, and we know there is an Euler Circuit of the grid. This is called *Eulerization* of the graph.
- Can you find the Euler circuit through the grid? Hint: use Fleury.

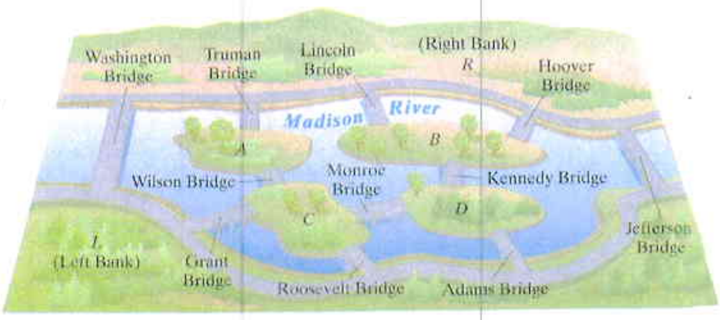


When we allow 4 edges to be retraced (equivalent to adding multiple edges there), then all vertices are even, and we know there is an Euler Circuit of the graph.

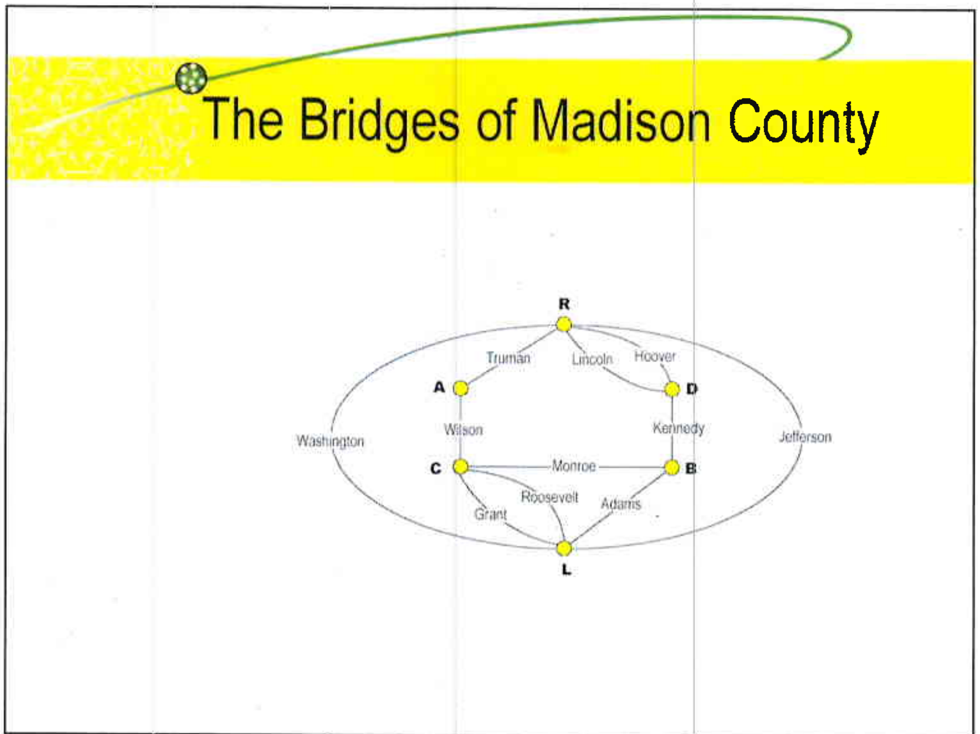
Important: You may not add new edges, only duplicate existing edges.

Exercise: use Fleury to find an Euler circuit of this graph.

The Bridges of Madison County



Remember the photographer that needs to cross every bridge at least once? At a cost of \$25 per crossing?



Here's the graph. There are 4 odd vertices. Where are they?

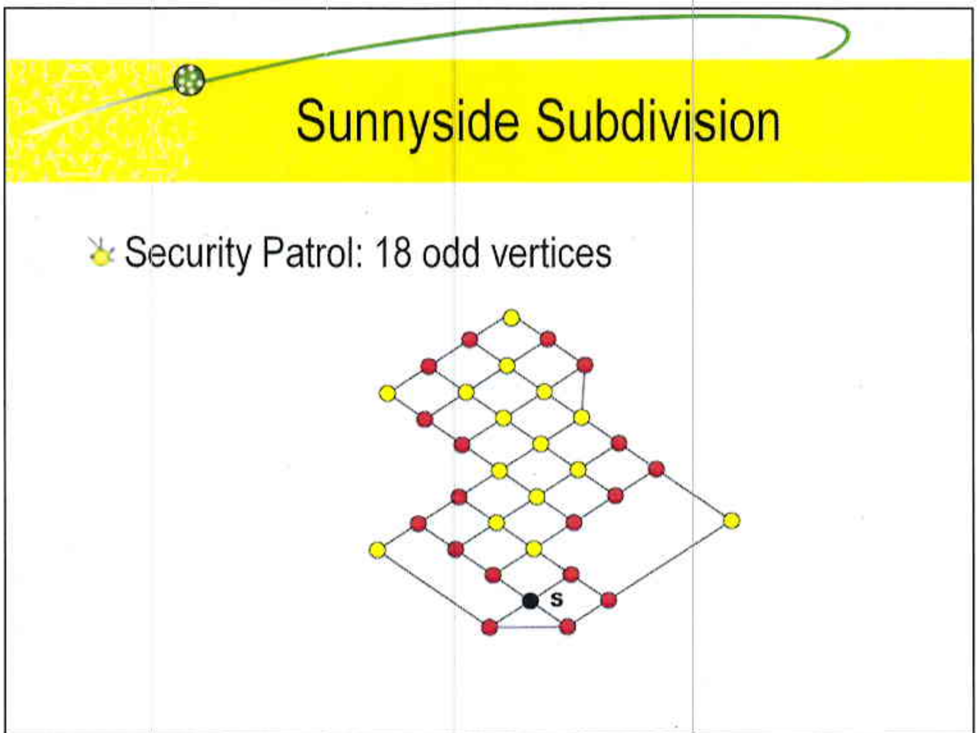
Possible scenarios (note that 11 bridges means a base cost of $275)

1. The photographer needs to start and end in the same place. This requires an optimal Eulerization of the graph—for example, recrossing Hoover and Adams bridges for a total cost of $325.

2. Start and end anywhere. This requires an optimal semi-Eulerization. For example, to start at D and end at R, it is sufficient to recross Adams. Cost $300.

3. Must start at D and end at L. We need a semi-Eulerization where D and L remain odd, and R & B become even. To do this, we must add two edges. Cost $325.

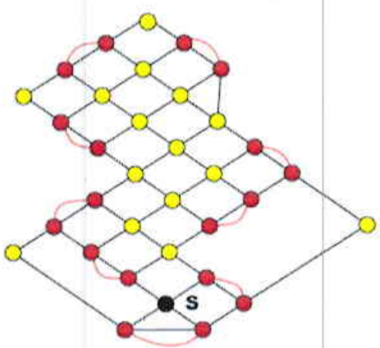
4. ✂



Luckily, odd vertices (marked in red) pair up nicely. This is not always possible. Anyway, we can Eulerize by allowing nine deadheads

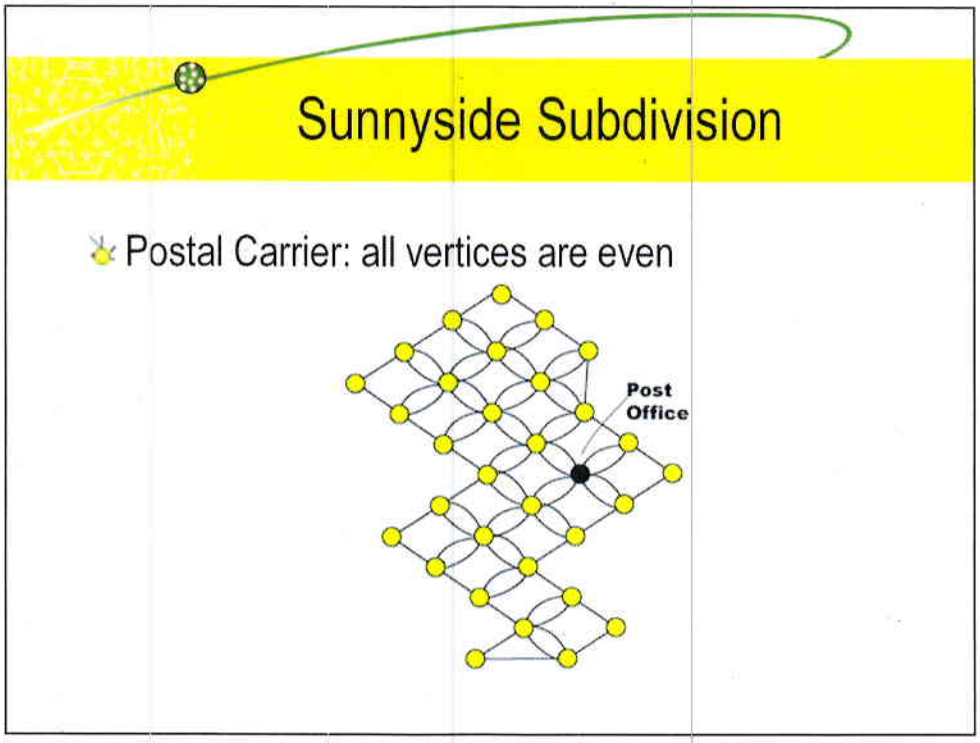
Sunnyside Subdivision

- Security Patrol Eulerized: 9 deadhead blocks



Now all the vertices are even, and an Euler circuit exists.

How could we find it?



Finally, the lucky postal carrier finds that all the vertices are even, so there is an Euler circuit with no deadheads.