MA/CS 109 Lecture 3

Networks, Euler paths, and speaking carefully!

Last Time:

Last time we proved the statement:

Theorem 1: If a network has an Euler circuit, then the degree of every node must be an even number.

Sounds really sophisticated!

Vocabulary:

Network: Nodes (points) connected by edges (curves connecting nodes)

Path: list of edges where each edges ends at a node where the next edge begins

Circuit: a path that ends where it starts

Euler circuit: a circuit that uses each edge exactly once.

Degree of a node: number of edges that touch it

Theorem: a statement that has been proven

Application of the Theorem

To use the theorem, we “turned it around”

If a network has an Euler circuit, then ever node must have even degree.

Is the same as:

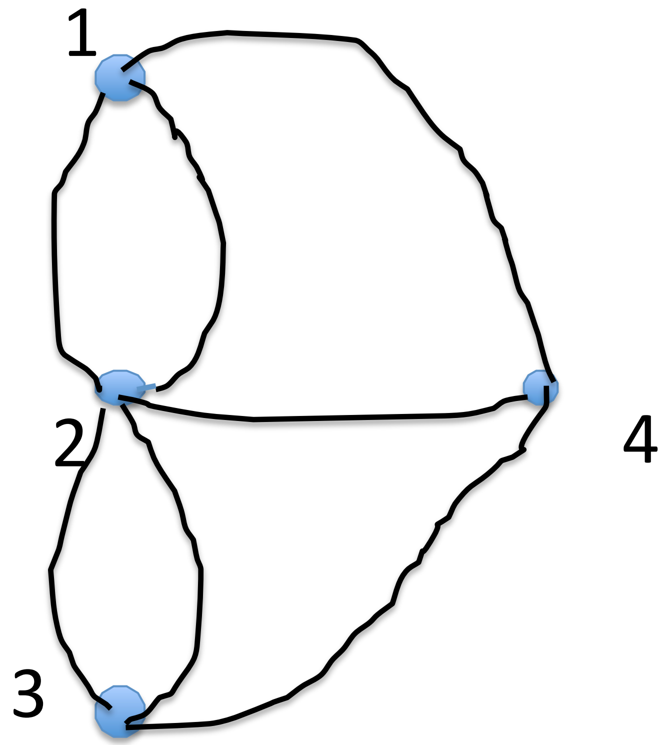
If a network has one or more nodes with odd degree then it does not have an Euler circuit.

Sounds very legalistic…

For good reason—when you are writing a contract and money is involved, you want to be sure of what you are saying!—so mathematical statements and legal statements sound alike because they must be clear and unambiguous.

Apply to Konigsberg:

The Konigsberg network can not have an Euler circuit because it has nodes with odd degree!!



Reached the

$$ Fame step…

Well Euler did OK—he made a good living as a "court" mathematician and his name is revered in mathematics. He is recognized as the founder of Network theory, a great luminary in number theory, differential equations, celestial mechanics and many other fields.

What next….

Modify the problem:

Could the people of Konigsberg have followed an Euler path?

Euler path: A path that uses every edge of the network exactly once. (Don’t care about start=end)

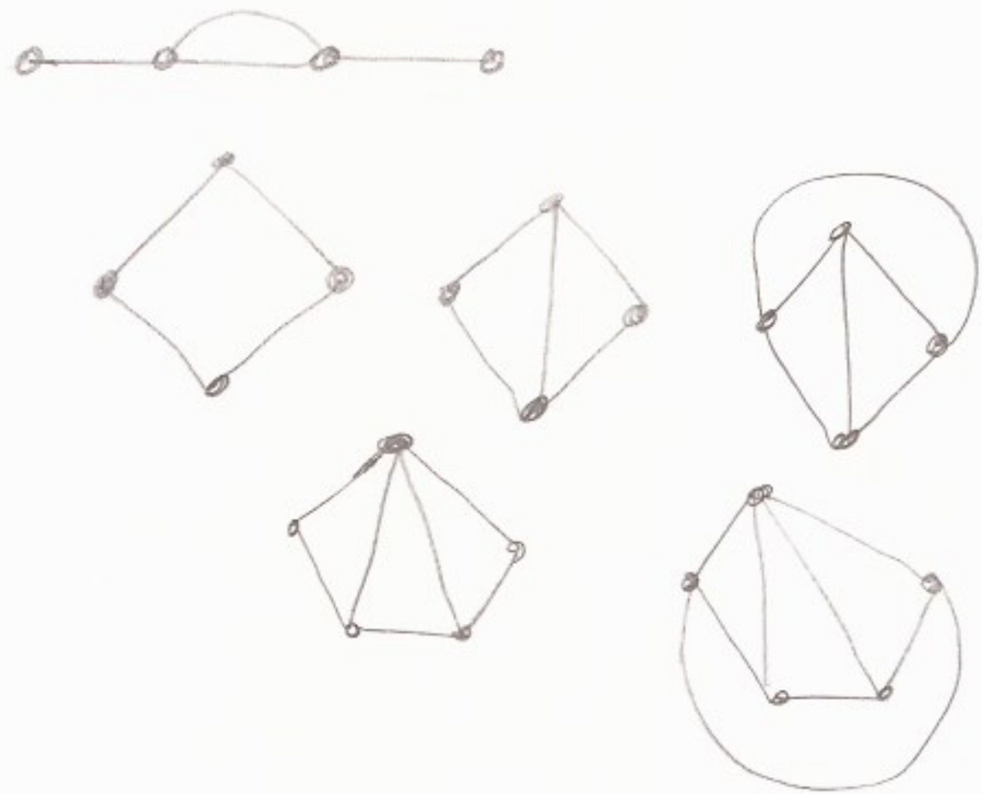
New, but closely related problem…

Same model: The definition of network, node, edge, path all stay the same—new idea is Euler path. Note, every Euler circuit is an Euler path, but Not the other way around…

But you saw that in discussion already--

Have an Euler Path?

Yes



Conjecture for Euler paths:

If a network has an Euler path, then …either all nodes have even degree or there are exactly two nodes with odd degree.

(if the network has an Euler circuit all nodes have even degree. If it has an Euler path that is not an Euler circuit then the there are 2 odd degree nodes and they are the start and end nodes.)

Proof

We could argue as we did for Euler circuits, but it is easier and more interesting to USE what we know about Euler circuits to prove the conjecture about Euler paths…

Theorems we have already proven are rock solid and can be used with complete confidence! This is one of the things that has allowed mathematics to progress so far. The shoulders of giants are as stable as the ground.

Proof

So we start with a network that has an Euler path.

If that Euler path is actually an Euler circuit, then we are done since we know that every node must have even degree! (The conjecture is true in this case).

What is left? If the Euler path in our network is not an Euler circuit, then it must start at one node and end at a different node….

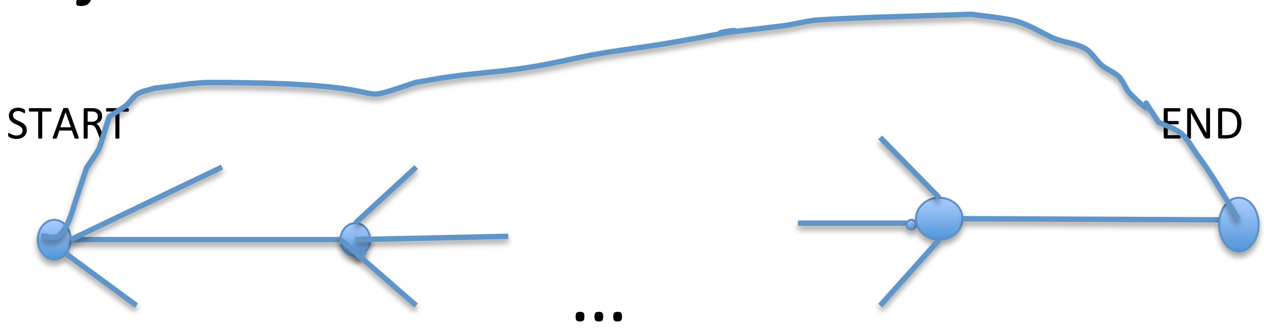
(This picture is just a schematic…there could be lots more edges attaching to the start and end

And the … could hide any network that has an Euler path.)

We know something about Euler circuits—how can we use this to prove our Euler path conjecture??

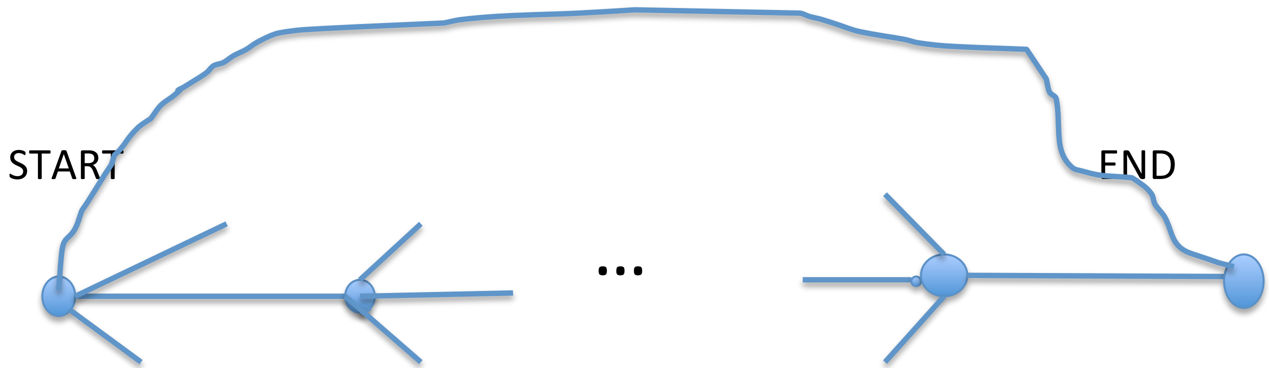
Clever idea—

We know something about Euler circuits—how can we use this to prove our Euler path conjecture??



Clever idea—Add one more edge from connecting the "end" node back to the "start" node.

Wow…Now our network has an Euler circuit!



Follow the Euler path, then do the added edge. This is an Euler circuit…every edge is used exactly once and it ends where it starts

BUT, now we know from our Theorem that the degree of every node of the new network MUST be even! (We are SURE of this, Theorem 1 is proven!)

Now, remove the added edge. By removing the added edge, we reduced the degree of the start and end by one.

Since the degree of every node is even when the added edge is there, removing the added edge leaves exactly two nodes with odd degree!

And that is what we wanted to show!

Success-Fame-$$$

<u>Theorem 2</u>: If a network has an Euler path, either every node has even degree (in the case the Euler path is an Euler circuit) or, there are exactly two nodes with odd degree and the rest have even degree.

ONLY THE BEGINNING…

As we said—these statements only go “one way”. If we have a network with an Euler circuit or Euler path, we can say something about the degrees of the nodes.

But we don’t know the “other way”—if the degree of every node is even, is there an Euler circuit? If there are exactly 2 nodes with odd degree, is there an Euler path?

To these statements we have to say NO COMMENT.

(Turns out…)

It turns out that there are Theorems which say
"If every node has even degree and the network is strongly connected, then there is an Euler circuit. If there are exactly two nodes with odd degree and the network is strongly connected, then there is an Euler path."

BUT these statements require proof…so you shouldn't believe them yet (I am over 30!)

Related question:

If we know a network has an Euler circuit—how do we find it? This is the sort of question a computer scientist would ask—what is the algorithm?

We can ask for other sorts of paths...a path that visits every node once, or a path that visits every node using the fewest number of edges...

Why a particular page comes up first in your Google search (…ok not the “ad” pages) is a question about networks.

Prof Snyder may return to these later…for now, we’ll move on to other ideas in mathematics …

BIG PICTURE: Our goal is to think about the process of doing mathematics and what it means to prove a precise mathematical statement.

Next: Return to the happy days of your youth…

The thing in mathematics that almost everybody is good at is counting…we learn very early how to count.

And we count things…pencils, straws, toys, trees…

When we get older we count mostly one thing—

Money.

But the idea that a set of things has a number associated to it is a pretty deep idea. Three apples and three oranges are very different, but they both share the property of “threeness”.

Just straight forward counting is something we all know how to do, but it can get very tedious when the number of things is very large…and counting lots of things is when things get interesting.

Toy Counting Problem

Let's do a counting problem that is "real" but not nearly as huge as many encountered every day…

I have to schedule the mathematics and statistics TA's to classes…there are 50 TA's and 50 classes that get TA's—one TA to each class. Of course, I want to assign the best person for each class (math TA's to math classes, stat TA's to stat classes…) and experience for each class..

Solution:

Solving this problem is easy, right??

Just list all the possible ways to assign the 50 people and then go through and pick the best arrangement…

How many possible arrangements—

Well, let's put the classes in order—class 1, class 2, class 3, …, class 50.

How many ways are there to assign a TA to class 1? Well there are 50 TA's so 50 ways.

Now let's do class 2—For each assignment of TA to class 1, there are 49 TA's left, so for each way of assigning class 1, there are 49 ways to assign class 2.

So how many for class 1 and class 2? Hmmm, let's number the TA's as well—TA 1, TA 2, … to TA 50. Now we can make a list.

TA 1 to class 1, TA 2 to class 2

TA 2 to class 1, TA 1 to class 2

TA 1 to class 1, TA 3 to class 2

TA 3 to class 1, TA 1 to class 2

TA 2 to class 1, TA 3 to class 2

(AARGH!)---need a more systematic way to even list all the ways to assign classes 1 and 2!

A surprising rescue…

For just this simple problem, we can use… geometry! Make a square listing TA's across the top for class 1 and TA's across the side for class 2

	TA 1	**TA 2**	**………..**	**TA 50**
TA 1	Not allowed	2 to cl 1 1 to cl 2	…..	50 to cl 1 1 to cl 2
TA 2	1 to cl 1 2 to cl 2	Not allowed	…….	1 to cl 1 50 to cl 2
……..	…	…..	Not Allowed	………..
TA 50	1 to cl 1 50 to cl2	2 to cl 1 50 to cl 2	………	Not allowed

There are 50 rows and each of the rows has 49 possible assignments for the TA's…so the total number of allowed ways to assign TA's to the class 1 and class 2 is the number of filled in boxes which is 50 x 49 = 2450.

(A rectangle that is 50 units by 49 units has area 50 x 49=2450.) This idea is called the "Multiplication Principle". If you are doing 2 things, the total number of ways to do them is the number of ways to do the first MULTIPLIED BY the number of ways to do the second.

That’s a lot of ways – and that is only the first 2 classes. What about class 3?

Well, for any way of assigning classes 1 and 2 (that is, any of the 2450 ways above) there are 48 ways to assign class 3 (48 TA’s left)…

So, using the multiplication principle again, there are

$$2450 \times 48 = 117{,}600$$

Ways to assign TA’s to classes 1, 2 and 3…

Keep going! Same idea over and over again—the number of ways to assign TA’s to all 50 classes is

50 x 49 x 48 x 47 x 46 x ..... x 3 x 2 x 1

Approximately

30,414,093,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000

So…No, I will not write out all possible arrangements and choose the best.

You encounter problems like this every day…you are choosing a flight to LA for winter break or vacation. A travel site says it will check all possible flights Boston to LA and tell you the one that is cheapest….

They are lying there are 1,000 flights per day from Logan to… ~500 different cities and from each of these you could continue your trip to LA…the airport you stop in first might be even bigger than Logan. The cheapest flight (say made up of Southwest Airlines $59 fares) could be 7 or 8 stops…The number of possible routes is gigantic…it would take even the worlds fastest computer millennia to check them all one at a time.

Three interesting questions from this example

1. How do we find the cheapest airline fare, or shortest driving route, etc., when there are (way) too many possible ways to check one at a time? (Sort of problem Computer Scientists love!).
2. How can we efficiently count other types of things? ("Combinatorics" problem in mathematics.)

3. The number of possible assignments of TA's gets bigger the more TA's and classes we have (for 100 TA's there are

9332622000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
00)

How fast is this growth? Are there problems where the count grows faster? (yes) and do they occur in nature? (definitely!)

A Zoo of Functions