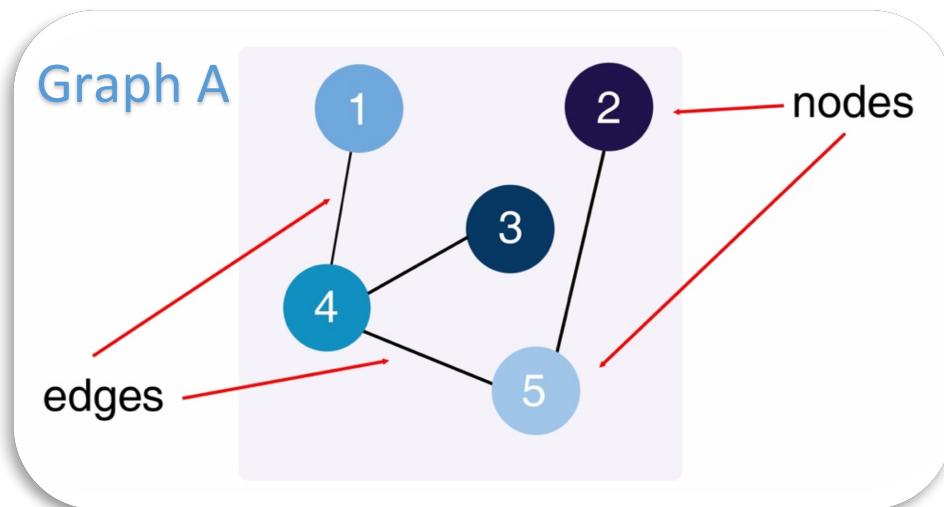




# What Are Networks?

**Networks** are a collection of points joined together by lines (Newman 2010). We can think of a network as a **graph**. The points on the graph are called **nodes** or **vertices**; the lines are called **edges**. A network can take many shapes, but the general form is something like this:



Graph A is an **undirected network**. That means that if there is an **edge** between two **nodes**, you can travel between them. For example, we can travel directly between node 1 and node 4, and back from node 4 to node 1. But to get from node 1 to node 3, we can't go straight there. We must go through node 4 first. This is called a **path**: in this case, the path is 1 – 4 – 3.

## Q: Why does this matter?

A: Networks are everywhere in real life. They help us to study objects and the relationship between them.

## EXAMPLE: Consider this picture of flights between different cities in Eastern USA.

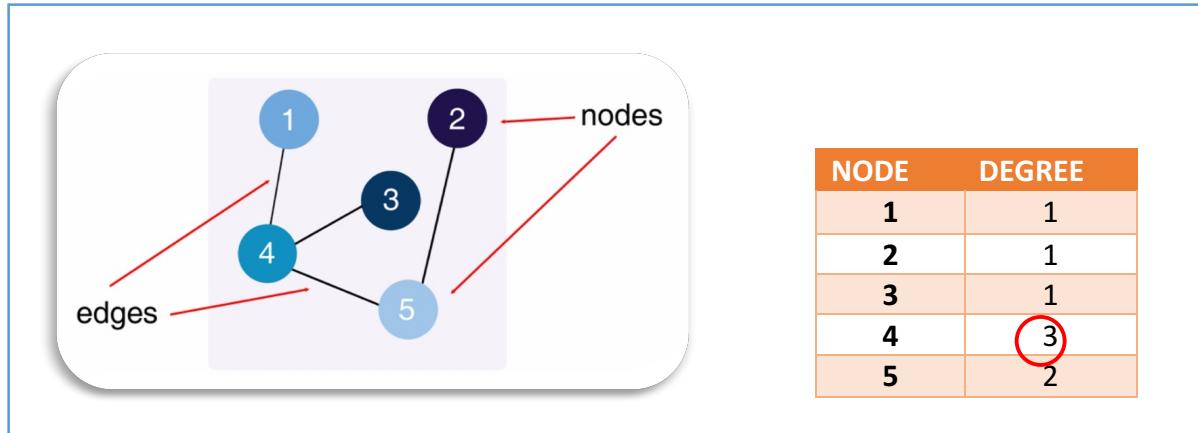


If you wanted to go from Indianapolis to Baltimore, and you could only take the flights pictured here, you would have to fly through Atlanta. Observe how much longer of a trip that is than flying directly from Indianapolis to Baltimore.

**What would happen if Atlanta airport was closed for bad weather?**

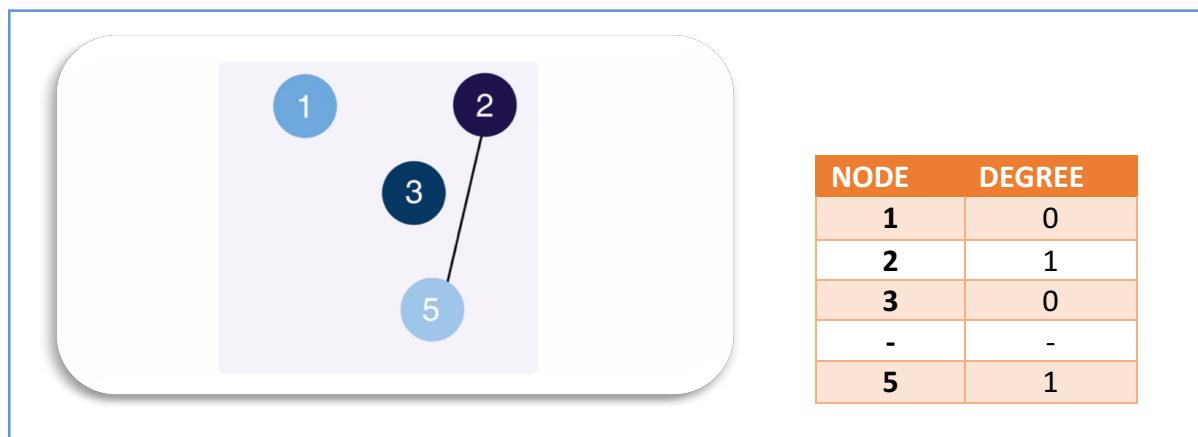


Let's look at Graph A again. We can reach any node on Graph A from any other node by following the edges. We can say that Graph A is **connected**. This is like the example above: we can reach any city from any other city, but we might have to travel through another city to get there.



The **degree of a node** is the number of edges connected to that node.

**Node 4 has the highest degree.** Node 4 in our graph is like Atlanta on the flight map, with a lot of connections coming in and going out. What happens if we remove node 4 from the network?

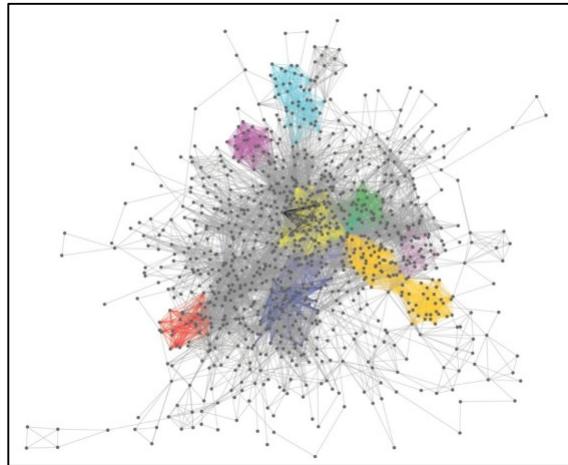


When we remove node 4, we must also remove any edges that go to 4. Now, you can only travel between node 2 and 5. **The other nodes are unreachable.**

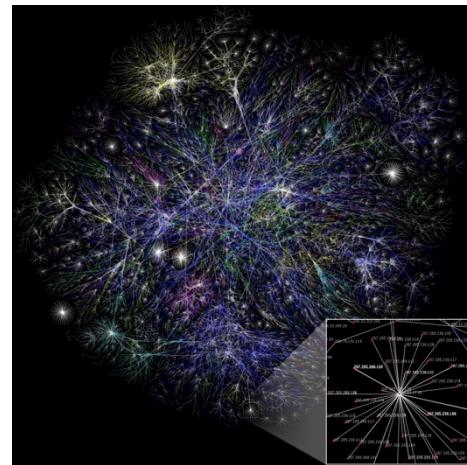
**That would mean a lot of people stranded at the airport!**



Networks aren't just used to describe flight schedules though! Here are some more examples, both man-made and natural:



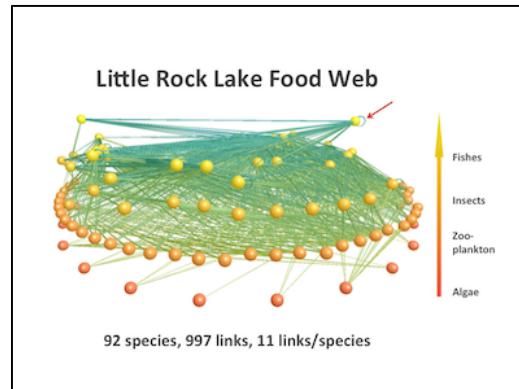
The metabolic network of a genus of green algae.  
(NYUAD)



The structure of the Internet.



A social network.  
(Aquila Style)



A food web showing who eats who in Little Rock Lake, Wisconsin. (Complexity Explorer)

## Let's get started!

### Walkthrough Overview

You will Recreate Graph A in NetLogo to learn the basics about networks.

We'll use the following concepts:

1. Breeds





2. Links
3. Labels
4. The network extension/network formatting

## Breeds

You can create a **breed** in NetLogo by defining it at the **top of your code, above any procedures**. Breeds can be named anything you want: animals, friends, cities, etc. We use

```
breed [dogs dog]
breed [cats cat]
breed [cows cow]
```

breeds to separate turtles into groups. By doing this, we can have separate groups of turtles perform different activities. On the left is an example of some breeds of animals.

Note that the plural form goes first and the singular goes second. Let's say we want to **ask** just the **dogs** to bark. We would use an ask command: `ask dogs`. That way the cats and cows won't bark.

*The breed members will be the nodes in the network.*

```
set shape "circle"
```

Change the shape of your agents from the default turtle to something else by including the `set shape` command when you create them. Available shapes can be found in the **Turtle Shapes Editor** under the **Tools** dropdown menu.

## Links

```
undirected-link-breed [edges edge]
```

**Links** are the connections between turtles, cows, cities, etc. They are the **edges** in the network. Links have their own special breed.

We'll be using the **undirected-link-breed** in this activity.

There's also a **directed-link-breed**, but you don't need to worry about it for now. Members of the undirected-link-breed can be anything that shows a connection between nodes: friendships, streets, etc.

## Labels

Labels allow us to give an identity to each member of a breed or a node. We can set its label directly when we create the agent, or later by **asking** it to set its label. If we use this method, the label must be in quotation marks, or it won't work:





```
set label "Chicago"
```

We can also **let** a variable be some value in quotation marks, then set the agent's label to the variable's value. To set the label **Chicago** from the variable **hub-city**:

```
let hub-city "Chicago"
...
set label hub-city
```

**NOTE:**

The **create** and **ask** code is not included here.

You can **ask** a specific turtle, node, city, etc. to do something by referencing its label:

```
ask cities with [ label = "Chicago" ]
```

This is really useful for keeping track of agents and telling them apart. In the next module, you'll learn how to set a bunch of labels at once, but for now we'll set them one at a time.

## The Network Extension/Network Formatting

```
extensions[ nw ]
```

**For each activity in this networks series, include this code at the very top of your file.** It will allow you to use special networks functions.

If you create some nodes and edges without setting a specific x and y coordinate, they will bunch up in the middle and be impossible to see. Instead of manually placing each node with **setxy**, you can use one of NetLogo's built-in layout functions. Visit these links to the NetLogo dictionary to learn about different layouts:

Circle Layout: <http://ccl.northwestern.edu/netlogo/docs/dict/layout-circle.html>

Radial Layout: <http://ccl.northwestern.edu/netlogo/docs/dictionary.html - layout-radial>





Spring Layout: <http://ccl.northwestern.edu/netlogo/docs/dict/layout-spring.html>



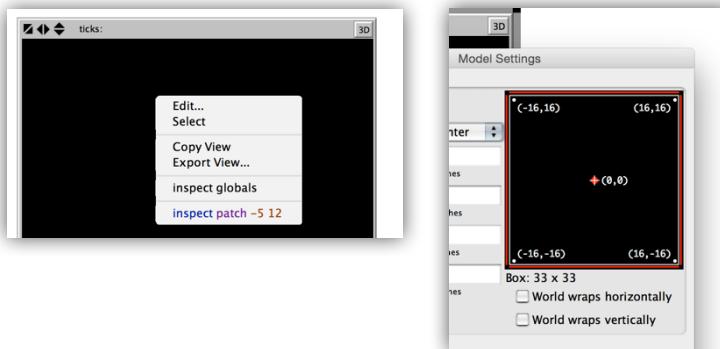
The many ways in which we are connected through networks. (startupist)

## Recreating Graph A

Create a new NetLogo file. Name it *yourlastname\_yourfirstname\_GraphA.nlogo*.

Turn off world-wrapping, or the edges between your nodes will go off the sides of the screen.

Go to the interface tab and right click on the simulation area. A contextual menu will pop up:



Select **Edit...** from this dropdown menu as pictured on the left.

On the next screen, uncheck the boxes for world wrapping as pictured on the right.

Click on the **code** tab, and let's start writing the program!

Include the network extension at the top of your file.

```
extensions [nw]
```

### Foundations

Underneath what you just wrote, create a breed of **nodes** and an undirected-link-breed of **edges**.



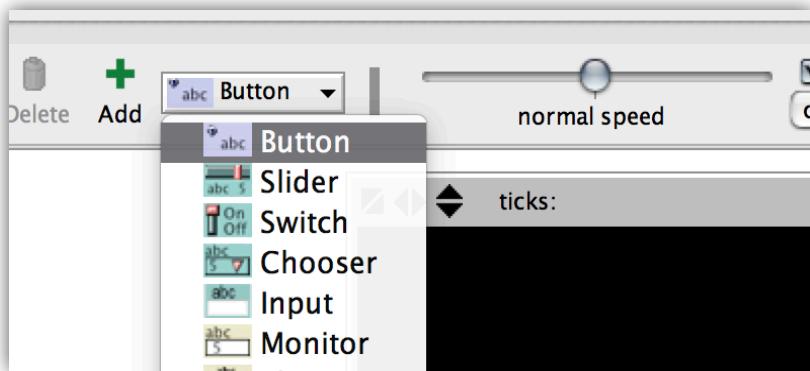


```
breed [nodes node]
undirected-link-breed [edges edge]
```

Next, we'll create a **setup** procedure to make the nodes. Enter the code pictured on the right into your program:

Now we're ready to try it out. Follow the three steps below to add a button for the **setup** procedure on the **interface** tab.

```
to setup
  clear-all
  create-nodes 1 [set label "1"]
  create-nodes 1 [set label "2"]
  create-nodes 1 [set label "3"]
  create-nodes 1 [set label "4"]
  create-nodes 1 [set label "5"]
  ask nodes [
    set shape "circle"
    set size 2
  ]
  layout-circle sort nodes 10
end
```



1) Select **Button** from the dropdown menu.

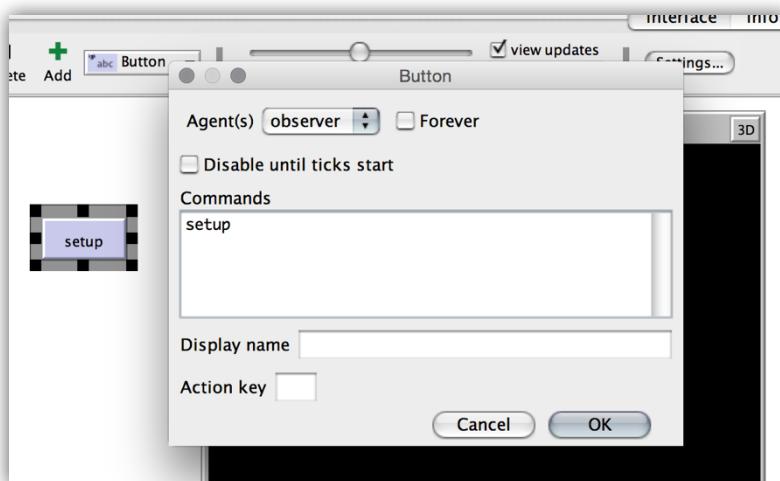
2) Click Add.



3) Enter setup in the pop-up box that appears. The text also appears in your new button.

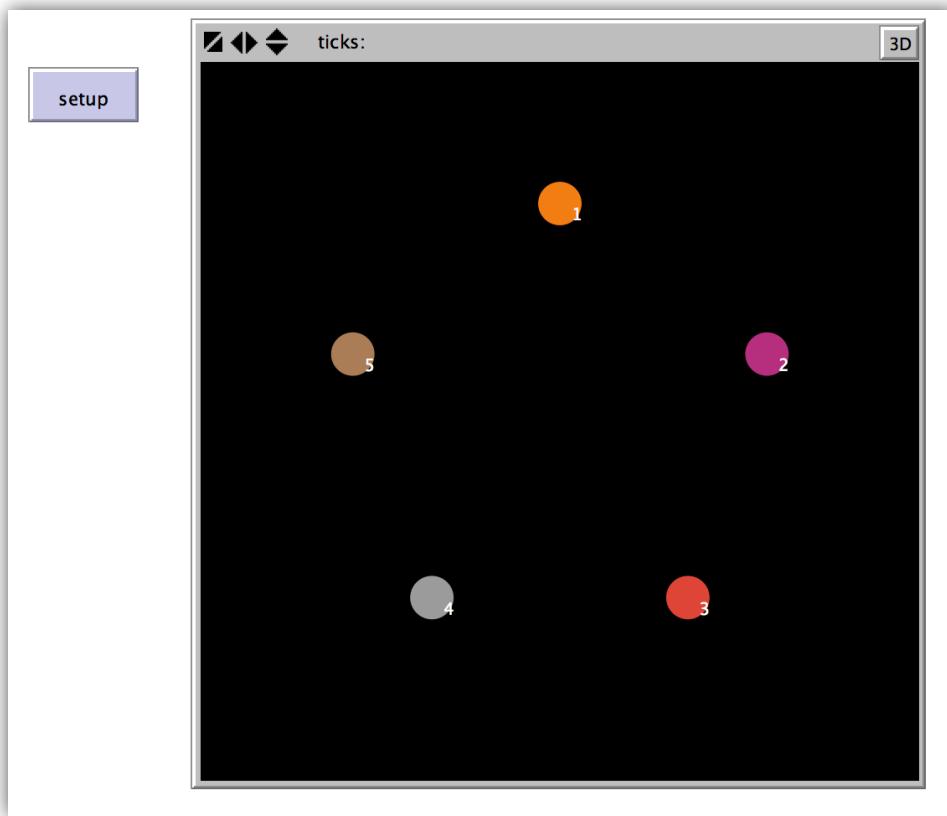
Click the button to try it out!

**NOTE:** If you need to move or resize the button, right-click it and choose **select**. To change the values, click **Edit**.





Your screen should look like this:



## Making the Connection

Now, let's create some edges between the nodes. Go back to the **code** tab. Below your **setup** procedure, create a procedure called **make-edges**, as shown below:

```
to make-edges
  ask nodes with [label = "1"] [create-edge-with one-of nodes with [label = "4"]]
  ask nodes with [label = "2"] [create-edge-with one-of nodes with [label = "5"]]
  ask nodes with [label = "3"] [create-edge-with one-of nodes with [label = "4"]]
  ask nodes with [label = "4"] [create-edge-with one-of nodes with [label = "5"]]
end
```

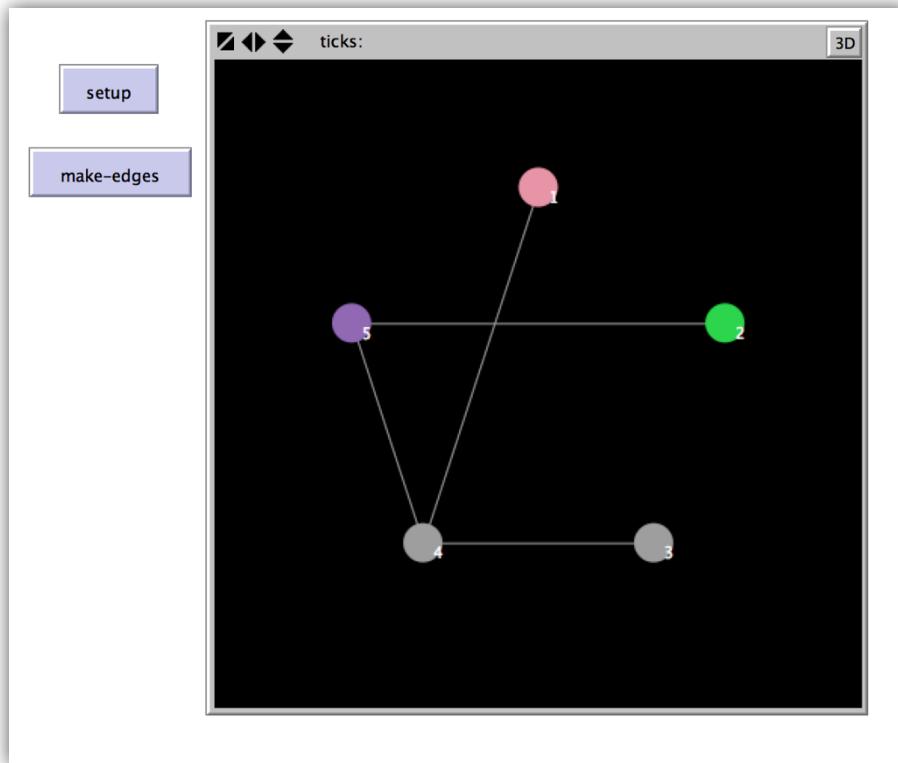
Create another button on the interface tab the same way as you did for the **setup** button. Name this button **make-edges**.

Click **setup** and then **make-edges**.





# A network appears!



That's the basics for networks.

Using this walkthrough as a base, you can model any type of network you choose!

## Things to Try

- Change the breeds of the nodes and edges to simulate another kind of connection. For example, a simulation of a social network would have people as the nodes and friendship as the edges.
- Go to Tools → Turtle Shapes Editor. Choose a shape for your nodes to fit your simulation, or make your own shapes!
- Add and remove nodes and edges. How does the network change? How do the paths between nodes change? If you wanted to stop communication on the network, which edges are the most important? What does that mean in the real world?



## Turning in Your Project

### Pre-submission checklist

- Is your name, date, and project title at the top of your file in comments?
- Is your code organized and formatted to course standards?
- Does it work without errors?
- Is your file named correctly?

If you answered YES to all of these questions, **congrats!** You are ready to turn in your project.

You will be given instructions on how to hand in your assignment in class.

### Rubric

<b>Grading Rubric: Intro to Networks (10 points)</b>	
Points	Task to be completed
<b>2</b>	<b>Clarity and professionalism:</b> Your file is named and commented appropriately. <b>[1 point]</b> The code is formatted and is readable. <b>[1 point]</b>
<b>8</b>	<b>Activity 1:</b> The setup and make-edges buttons create a graph that has the same structure as Graph A. <b>[8 points]</b>