

PROFESSOR: So connectivity is more than just an all or nothing affair. We can talk about how connected a graph is. So let's begin with two vertices. Two vertices are said to be k -edge connected if they remain connected if you remove fewer than k edges from the graph.

Let's look at an example. So here's a graph, and let's focus on those two vertices that I've highlighted in magenta. They are 1-edge connected because they're connected, and if you remove one edge they become disconnected. So they're 1-edge connected, but they're not 2-edge connected. In particular, if I delete that edge, then they no longer is a path between the two magenta vertices.

Here's an example of two vertices, these two green vertices, that are 2-edge connected. That means that I can remove any number of edges less than two, which is to say one edge, and they'll stay connected. But if I do remove two vertices, they become disconnected. So with those two edges judiciously selected, there will no longer be a path between the two green vertices.

For a 3-edge connected example, we can look at these two vertices highlighted in purple and discover that we can remove any two edges, and they're still going to be stuck together. You can kind of see that because they're in a cycle here and in a cycle there. And in order to break them, you're going to have to break both cycles, which means two edges to break two cycles.

And if you remove three, on the other hand, you can separate them. So if I cut these three edges-- might have been more obvious to cut that one, but I'm doing it this way to make it interesting-- if I cut these three edges, then in fact I wind up with no path between those two vertices. So they are 3-edge connected but not 4-edge connected. Meaning you can remove two, but you can't remove three and keep them connected.

For a whole graph, a whole graph is k -edge connected if every two vertices are k -edge connected. And the point of this degree of connectivity is to measure-- one motivation of it is if you think of this graph as indicating a communication network, where vertices are centers that are sending and receiving data, and an edge corresponds to having some kind of a channel or cable between adjacent centers.

Then connectivity is a measure of how many of your channels or cables can go down and still have the property that every center of information can communicate with every other center of

information. So how many connections can fail without cutting off communication?

Here's an example of a graph that globally-- this was the graph that we saw before that has some vertices that are 1-edge connected, others that are 2, others that are 3. But the graph as a whole is only 1-edge connected because there's this one edge that can be removed that splits it in two.

This graph, on the other hand, a modification of it, is 2-edge connected. Now I can remove any one edge, and no matter what single edge I remove, this will remain a connected graph, which makes it 2-edge connected.

Now, there's a corresponding definition of vertex connected, meaning it's k -vertex connected if you can remove any number of vertices up to k and it will stay connected. If it's k -vertex connected, it's certainly k -edge connected, but not conversely. Here's a graph that is 2-edge vertex connected but 1-vertex connected. If you remove one vertex, it breaks. If you remove that vertex, it breaks. But on the other hand, it needs two edges to be cut in order to split the graph in two.

The complete graph on n vertices is-- to cut it up, it requires cutting n minus 1 edges and, in fact, n minus 1 vertices to break it up because everything is connected to everything else. So K_n is as connected a graph on n vertices as you can possibly have. But of course, it's got a lot of edges. It's got $\binom{n}{2}$ edges, or about n times n minus 1 over 2 edges.

Another interesting structure is the n -dimensional hypercube. So the square is a 2-dimensional, and the cube, the ordinary cube is the H_3 , has eight vertices and looks like a cube. And in general, the way we define H_n is that its vertices are the binary strings of length n .

And two vertices are adjacent if they differ in one bit. They're the same except that in one coordinate one of them has a 1 and the other has a 0. In that case, you can make the edges adjacent. It turns out that the n -dimensional hypergraph is n -vertex connected. And this is worked out in a homework or a class problem.

So to summarize-- well, actually before we summarize, there's a basic theorem that we're not going to prove. It's just a little bit too much of a challenge for us to get into. It would take maybe a couple of lectures-- is to prove Menger's theorem, which says that a graph is k -vertex connected if and only if you can connect any two vertices with completely non-overlapping

different paths k of them.

So if two vertices are k -connected, then there are completely separate, k separate paths connecting them, which means, of course, that you have to cut each of the paths in order to separate them. And a similar theorem goes for edge connected.

So to summarize about some graphs that we know about, given that connectivity measures fault tolerance in a network, we're interested in, well, how many edges did it take to achieve this level of connectivity? So the number of edges is kind of a measure of the cost. And the complete graph on n vertices is $n - 1$ connected, but it has about $n^2 / 2$ edges.

The hypercube. Now, if we're going to be talking about graphs of size n to be uniform, the hypercube on H_n has 2^n vertices. So if we want an n -vertex version, we're talking about $h \approx \log_2 n$, which has n vertices. And then it has about $n \log_2 n / 2$ edges. So it's got fewer edges than the complete graph but significantly less, exponentially less connectivity.

The grid is simply like the x, y axis or the x, y plane with points at integers so that every node is connected to four adjacent nodes-- one up, one down, one right, and one left. Now of course, when you get to the edge of it, if you make it finite, the ones on the edge are only going to be 2-connected. So you have to wrap it around in the shape of a donut to really make it work with a finite graph. But if you do that, then the thing is 4-connected-- 4-edge connected and only has $2n$ edges.

And then a cycle is 2-connected. A cycle of size n is 2-connected with n vertices. And finally, trees, which we'll talk about next time, are the smallest possible connected graphs. They are 1-connected. Indeed everything in a tree is connected, and they only have $n - 1$ edges. But if you remove one edge, it splits. And that's sort of the summary of the k -connectivity story.