Transcript - Who Do You Know

So you were at the game?
Yeah, what a game.
Wow.
40,000?
40,000 in the stadium? And the stadium holds what, 50,000?
About 50,000, yeah.
Incredible. Wow. Oh hey, it's George! George! Hi!
Hi. Hi Jordan.
Good to see you.
How are you today?
Doing well.
What a gorgeous day out.
Fantastic. Hey, do you know Isam?
Hello, Isam.
Hi, George.
I'm George. Good to meet you.
Nice to meet you.
Isam's also in the faculty of engineering here at AUB.
Ah, is that right?
Where did you study, Isam?
UT Austin.
At UT. When was that?
It was between 2000 and 2005.
Hm. You know, I used to have an office mate of mine from graduate school there, who was there about that time. Do you happen to know Carl Haas?
Do I know Carl Haas? He's my advisor!
Wow.
Wow. What a small world. Imagine that. In a country the size of Lebanon, with about four million people, how many people does each person need to know to guarantee that any two randomly selected people would know one person in common? Think about that question for a while, and we've got to get back to the classroom.
That's right.
We'll meet you there!
Welcome to the classroom. So how many people would each person in Lebanon need to know to guarantee that two randomly selected people would share one friend in common? It's actually quite a large number. It's 2,000. So how was it that George and Isam would so quickly find a friend in common? That's most likely due to the network of friends that they both have. Let's look more at that connection. So we have George and Isam who both know a friend in common. We could say that they're connected by two links, or one person. It's a pretty direct connection, so it was easy for them to find. But in a larger network, might be a bit more difficult.
Let's imagine you're at a party, and you're interested in meeting someone that you do not know. The way to get to know them is to have one of their friends introduce you. So, how would you go about finding the person to introduce you in the least amount of time?
To help you play this game, your instructor is going to give you a piece of paper that says who you know in your classroom and who you would like to meet, who your target person is. You, then, need to find the right introductions following these rules. You cannot talk to anybody you do not know directly. Two, you cannot talk to anybody you have not been introduced to. Three, you can ask your direct connections for introductions. Four, you can only introduce a person to one of your direct connections, but you get to choose which one. And finally, be sure to write down the introductions that you need.
Here's a small example. So let's imagine there's a party. There's lots of people, but there's four people we're interested in, Rawad, Farah, Christopher, and Nelly. Rawad wants to meet Nelly. Rawad goes to one of the people that he knows directly, let's say Farah. Rawad tells Farah that he wants to meet Nelly. He then asks

who she can introduce him to. Farah does not know Nelly directly, so she decides to introduce Rawad to Christopher, and the process continues until Rawad knows Nelly. Now it's your turn. Let's see who can meet their target person in the least number of introductions possible. Have fun.

So, how many introductions did it take you to meet your target person? Did you do it as efficiently as possible? To start exploring this question, we need to introduce some mathematics in this problem. Social networking relies on a branch of mathematics known as graph theory. A graph is a structure that contains nodes and arcs. In the context of social networking, the nodes represent the people involved. So the four people you've been introduced to so far in this module are represented here by these four nodes, George, Carl, Isam, and Jordan.

The arcs represent whether two people know each other. So in this case, George and Carl know each other, Carl and Isam know each other, Jordan and Isam know each other, George and Jordan also know each other. So from the exercise you just did, can you, working together as a class, reconstruct the graph that represents the relationships in your classroom? Every one of you will be a node in this graph.

Now that you see the full graph, can you tell if you've achieved the least number of introductions to meet the target person when you were playing the game earlier?

So, did you find the least number of introductions needed by looking at the graph? Well, maybe you did because the graph of your class is probably pretty simple, but what if it was a graph of all the people in Lebanon, or all the people in the world? You really wouldn't be able to guarantee that you'd found the least number of introductions necessary. So for that, you need a strategy. The strategy we're going to tell you about is called Dijkstra's algorithm. Now, you hear me saying Dijkstra, you might wonder how it's spelled. It's spelled D-I-J-K-S-T-R-A.

So let's take a look at how this algorithm works. Suppose that we have the following graph. Well, now let's decide on two nodes in this graph, or in the context of social networking, two people. Let's choose myself at Node One, and George, at Node Seven. In this small graph, it was easy to retrace our steps and realize that it took three steps, or two introductions, to go from myself to George.

In a larger graph, this could be tricky. So we actually do need to track our steps as we move through the graph. To do this, we're going to label our nodes. Let's begin by labeling all of the nodes with a really large number, say infinity, and my node, the start node, with a number that's small, zero. Then, to remind myself where I started and that I visited my own node, I'll color it green. From there, we visit the closest neighbors, our direct connections, as before.

Arriving at our direct connections, we see that it took only one introduction to get there. We know these people one step away. So we change the label from infinity to one on both of these nodes. Now, looking at our graph, we have some nodes that are green and some that are blue. Amongst the blue nodes, we find two that are labeled One. We select one of them, say Node Three, color it green to remind ourselves we visited it already, and move forward to explore its neighbors. The only neighbor that doesn't have a label that is smaller or equal to one is Node Four. We then change the label at four to two. That's how many steps it took to get there. We follow two arcs, so we make the label two. The process then continues, updating labels as we move through the graph until we reach the end, and all of our nodes have turned green, and George's node is labeled with a three.

Well, that's easy enough. So now, using the graph of your class along with Dijkstra's algorithm, find the shortest path between yourself and your target person. To help you out with this, we'll leave the steps of Dijkstra's algorithm up on the screen.

So now that you've had a chance to try Dijkstra's algorithm on the graph of your class, how do you know what's the least number of introductions that you need to meet your target person? That's right, it's the distance label that appears on the node for your target person. Notice how easy it was to find that least number of introductions when you had the full graph of your class there, but how tricky it was when you were trying to meet people in the game we played in your classroom.

This is the idea behind websites like Facebook and LinkedIn. They maintain a full knowledge of the connections between users. They own the graph. As a result, they can provide you with information about the least number of introductions to meet a desired target.

It's also the idea behind a whole class of engineering problems, shortest path problems. What's the quickest way to go from your house your school, or from your house to the grocery store? What's the most efficient way to connect a new home to the electric grid or to the water system? So really, everybody meets a whole bunch of shortest path problems in their everyday life, and now you have an efficient way to solve them.

Hey, we'd better get back to the party if we want to meet people.

That's right.

Hi. I hope you enjoyed this module and will find it useful for your students. The real trick to making it work well, and especially the activity, is to have a graph that's suitable for your class, a good graph structure. And to help you with that, there's a library of graphs posted on the BLOSSOMS website, that's blossoms.mit.edu. And once you've selected a graph from that website that's appropriate for the size of your class, the other trick is to make sure that the target person you select for each student is at least two nodes away. That will ensure that they actually have to seek introductions and meet each other during the game. It'll be a lot more fun.

Also, if you enjoyed this module, you may want to consider one of the other related BLOSSOMS modules on the website. One of them is called, Taking Walks and Delivering Mail. It's an introduction to graph theory. Or Math Flu Games, which is a beautiful introduction to how epidemics can spread through networks. Thank you very much.

Thank you.