The Towers of Hanoi
by Dr. Richard Larson

Hi. My name is Dick Larson. I'm a faculty member at MIT, Massachusetts Institute of Technology in Cambridge, Massachusetts. Welcome today.

We have a very interesting problem for you. It's called Towers of Hanoi. It's a famous problem that has applications in computer science, in mathematics, and, as we will see, in everyday life.

Now, this is not a new problem. Towers of Hanoi was proposed in 1883 by Edouard Lucas. You'll see on the table in front of me here I have three physical representations of the Towers of Hanoi-- what I call big, medium, and little, and we're going to talk about the rules of this game and what we're trying to do. Each of you should have a Towers of Hanoi in front of you.

Now, these three were made professionally, and they cost some money-- not too much, but some money-- to purchase, but you can make a Towers of Hanoi with just a minimal investment. It could be coins, four or five coins, on top of each other-- small ones and big ones-- or you could even make a Towers of Hanoi like this at home. About three days ago, I made this Towers of Hanoi in my home in Lexington, Massachusetts, and you might see a photograph or two of the production process, what's going on here.

Basically, these disks in the towers are nothing more than washers that I got from a local hardware store-- they just cost pennies apiece-- and the wooden dowels here that I cut here cost less than $1. This was a piece of wood that was left over. So, you can make your own Towers of Hanoi like this if you want.

OK. So what is the problem? What are we trying to solve today? Well, let's look at this.

What we're trying to do is move a set of disks like this-- we're going to call each of these things a disk-- and some have a smaller radius, and some have a larger radius. Our chore is to move this pile of disks from here-- we'll call this tower one, the medium one we'll call tower two, and this one on your far right we'll call tower three. We want to move these disks over to tower three, but we have two rules.

We have two rules we must obey: we cannot move two a time. We can only move one at a time. OK? Second rule is that any time we have a pile of disks, we must have them in ascending order, with the smallest one on top and the biggest one on the bottom. So, this is not allowed, but this is allowed. So, those are the two rules we have to obey, and let's see how we're going to do this.

Let's go over to the small one. Here we are with a small one, and we have a lot of disks. So, let's not start with the largest problem, let's start with the smallest problem. If we had to move this pile of disks-- one disk-- from tower one to tower three, all I do is pick it up, and put it over there. It took one move to do that, so that was easy.

Let's try it with two disks. Here we have two disks. How do I move this pile of two disks from tower one to tower three? That's easy-- I put this smaller one in a parking space, tower two. I put the larger one in its final destination, and then I take the smaller one off the parking space, put it over here, and solve it. That's solves the N = 2 problem.

So, now I have a challenge for you. We have solved the N = 1 and N = 2 problem. Let's go to N = 3.

So, in the tower front of you, whether it's coins or washers or a homemade tower, put three disks on tower one. Obey the rules: move them over to tower three, discuss the strategy with your neighbors, and record the number of moves it takes you to solve the problem. So, solve the tower-- the N = 3 problem with the Towers of Hanoi, and do this in your class for three, four, five minutes, and we'll see you soon.

Welcome back. Were you successful? Were you were able to solve the N = 3 disk problem? Did you talk to your neighbor, and did you keep count of the number of moves it took you to solve N = 3?

Let's remember what the rules are: the rules are you can only move one disk at a time, and any pile of disks on tower one, two, or three must be in ascending order with the smallest one on top and the biggest one on the bottom. So, I hope you remembered to obey those rules when you worked with your neighbor and yourself and you solved the N = 3 problem.

Now, let me try to do it over here. So, suppose I take the smallest one and I put it there, and I take the medium one and I park it over there. I have to move the smallest one again-- that's the only move I have. I put it over there. I'm not making much progress here.

If I put that over there, and this over there-- this doesn't seem to be working. Look-- what I've done is I've made the pile on the intermediate tower.

Now, did I tell you when I first learned of this problem, The Towers of Hanoi, several months ago, it actually took me several hours to figure out the solution and then, the beauty of the recursion that we're about to talk about? So, let's go back here, and see if we can do a better job at this.

So, I take the smallest one and park it over there on three. I take the medium one and park it on two, and then I take the smallest one and put it on top of medium. Then, I move just once the largest one from one to three, its final destination. I take the smallest one put it over there as a parking space like this, and like this. And lo and behold I've done it. It took me seven moves.

The first time I did it, it might have taken me 11 or 12 or 13 moves, so I want to know in your class-- raise your hand if it you took you eight or more moves. Do I see some hands up? I think I see a few. How many of you solved it in seven moves? I see some more hands up over there-- good. How many did it in less than seven moves? No, I don't think that's true-- it's impossible, so seven is about the best we can do.

So, let's look at this. Let's look at the size of the solution as N the number of disks grows. Remember when we had one disk-- it just took one move from one to three. When we had two disks, it took three moves. Now, we see the best we can do with three is seven moves. Do you see some kind of pattern here in the number of moves we have to make to solve the problem as a function of the number of disks we have?

Think about this. What kind of pattern seems to be emerging here? Why don't you write some of these numbers down on a piece of paper, talk to your neighbor, talk to your teacher-- see if you can speculate what this pattern might be as we go forward to five disks, 10 disks, even 48 disks. Think about it, talk about it, we'll be back soon.

Welcome back. I bet you had some fun with that. If you look at the pattern one to three to seven, it's almost like two to four to eight. Two to four to eight is 2 raised to a power-- 2 to the 1, 2 squared, 2 cubed. So, if you speculated that this problem doubles in size virtually with every disk, you'd be right.

So, with $N = 4$, 2 to the fourth would be 16, and then it goes to 32, 64, 128, 256, 512, 1,024-- that's huge. Now, it's a little bit complicated because our numbers don't exactly equal those-- they equal those minus 1. So, instead of eight, we have seven. Instead of 16, we're speculating 15. Instead of 128, we're speculating 127. But 127 moves? That sounds impossible, doesn't it? That's very, very complicated-- 127 moves here.

So, that's pretty scary. So we need to come up with some systematic way of thinking about this problem where the size of the problem doesn't get in the way of our thinking, so let's think about this. Let's go back to the sequence of moves we had with the $N = 3$ problem.

Let's look at what we did here: small to right, medium to center, small to center, large to right, small to left, medium to right, small to right, done. Did you see any pattern there? How often is small moved? How often is the largest one moved? Remember, the largest one only was moved once, from its initial destination to its final destination.

What we want you to do is discuss this pattern with your neighbors, with your teacher, and see if you come up with a pattern there that can guide us into having to solve larger problems with the Towers of Hanoi and for larger values of N equals four, five, seventeen, whatever. Think about this, talk about the pattern to the neighbors, and we'll be back very soon.

Welcome back. Did you see a pattern? I can tell you're really very much involved now with the Tower of Hanoi's problem with your neighbors, with your teacher. You're really into it, and I think that's really good. Now, do you see a pattern that might help you for larger values of N, because we're going to see the beauty of recursion here. This segment and the next one or two segments, we're going to go about and talk about recursion.

So, let's go back and revisit for one more time the $N = 3$ problem. Look at how we did this. We moved small to three. We moved medium to two, and we moved small over on top of the medium.

What have we solved here? We have solved the $N = 2$ problem in three moves. Then we take the biggest one and move it once to its final destination-- that's a fourth move. Now, we take the smallest one, we put it on its parking space over here. We take the medium and put it over here, we put the smallest one over here-- three more moves-- 3 plus 1 plus 3 is seven.

So, now you see what we've done. In solving the $N = 3$ problem, we solved the $N = 2$ problem first, and then we moved the large disk to its final destination-- we moved it once. We solved the $N = 2$ problem again, so the n equals 3 problem-- its solution is really two applications of $N = 2$ problem plus 1 move of the big disk from its first home to its final home. So, that's the way to think about the $N = 3$ problem. It's really focusing on the $N = 2$ problem twice and doing one move in between.

Here's the challenge for you. Can you take that logic and solve the $N = 4$ problem? Talk to your neighbor, talk to your teacher, use the Towers of Hanoi that's in front of you, and try your best at this using this recursive logic. We're going to be back soon to talk about this.

Hi. I'm Dr. 4. Towers of Hanoi problem for $N = 4$? No problem. I can do it, and I don't even know that much math. What I do know how to do is delegate authority.

Now, here's my $N = 4$ problem for Towers of Hanoi. I'm going to solve it, even though I don't really know about how to do Towers of Hanoi problems. But, my secret is that I that to my right is Dr. 3. Dr. 3 is a whiz at solving the Towers of Hanoi $N = 3$ problems. So, what I'm going to do is I'm going to ask him to do most of the work for me.

Now, I have my Tower of Hanoi problem. I'm going to pass it over to my right, and I have this note, and I'm going to give it with my Towers of Hanoi problem. What my note says is that I'm asking Dr. 3 to solve the $N = 3$ problem and move the top three discs from the tower number one to the middle one on tower number two, and then when he's done to pass it back to me.

Alright. This is almost done. Now, what I have to do is move the really large disk from this tower to tower number three. All I have left to do is to delegate authority one more time. So, here's my second note, and I'm passing this problem along with the note back to Dr. 3. In this note, I'm asking him to move the three discs in the middle tower onto the tower number three.

And it's all done, and I didn't even have to do that much work.

Ooh, Dr. 5. Please do the N = 4 problem by placing the top four disks from tower number one to tower number two in the middle. What do I do now?

Welcome back, and thank you Dr. 4. Dr. 4's real name is Anna Teytelman. She's a doctoral student in operations research here at MIT, and we thank her for her participation here today.

Now, did you figure out what Dr. 4 is to do with the note that she got from Dr. 5? Remember, Dr. 4 only knows how to do two things-- she knows how to delegate authority down to Dr. 3, and she can move the largest disk in a pile of four from its original location to its final destination.

So, you've probably figured out that with the note that she gets from Dr. 5, she should do the same thing-- delegate authority twice-- and between delegating authority and writing notes at the same time-- and between those two things, move disk number four from its original location to its final location. So, if you did that, if you thought about that, that's absolutely correct.

Now, when Anna, who is also Dr. 4, hands back the partially completed Towers of Hanoi to Dr. 5, did you figure out what else is going to happen? Yes? Dr. 5, who also only knows how to delegate authority down to Dr. 4, is going to send Anna back another note with another partially completed Towers of Hanoi, and ask her to do the whole thing over again. She's got to write two more notes to Dr. 3, and that's the way the whole thing goes.

So, in general, you could be Dr. 17. Imagine, a pile of 17 of these disks, and you could be Dr. 17 and say, no problem, I can work with that, because all I know how to do is delegate authority. I'm going to write two notes. Each time I'm going to pass this to Dr. 16 here on my right-- my right, your left-- and in between getting those things back, and I'm going to do the largest disk and move it from its original location to its final destination. So, no problem, Dr. 17, Dr. 42, or whatever.

Still, the problem grows exponentially, and the amount of time it takes to solve this might grow very, very large. Certainly, larger than the amount of time you have in your classroom today. But, I have a challenge now for you to show that you really understand how to delegate authority in this recursive way. If you could have three volunteers in your class, and they sit next to each other, and one will be Dr. 3, one will be Dr. 4, and one will be Dr. 5, and solve the N = 5 disk Towers of Hanoi problem using the recursive logic we just had where we're delegating authority on down the line, passing responsibility and notes back and forth. Do you think in your class now you can solve the n equals 5 problem?

Now, N = 5-- 2 to the fifth, 32 minus 1 is 31. That's going to be a lot of moves, and I don't know how your class time is going today, so you may be a little bit stretched for time. So, if you can't do the whole thing, start it, be confident that you know to solve it, and then you might abbreviate if class time is short.

So, we need three volunteers, and maybe everyone in the class wants to do this and competitive, so you could have teams trying to do this. Do this in your class, talk to your teacher, work with your neighbors-- play Dr. 3, 4, or 5, and we'll see you soon.

Welcome come back, and congratulations. Great teamwork. I know that the three of you playing Dr. 5, 4, and 3 did a wonderful job. Maybe there were multiple teams of three of you, each playing Drs. 5, 4, and 3. The secret here is Dr. 5 and Dr. 4 continually reduce the problem to a problem previously solved, and Dr. 3 was the expert. He or she was expert in solving the N = 3 problem.

So, this method of reducing to a problem previously solved is called recursion, and we have a formal definition of recursion. Fasten your seat belts-- this is a little bit abstract, but we want to go and talk about it. So, we want to define. Recursion is a method of defining functions in which the function being defined is applied within its own definition.

That is a mouthful. Let's say it again slowly, and then we'll give an example. Recursion is a method of defining functions in which the function being defined is applied within its own definition.

Now, you might say what in the world does this mean? Well, let's take a simple function which you all know about-- N factorial. N factorial is defined to be equals to N times N minus 1 times N minus 2 dot dot dot dot times 3 times 2 times 1. You take the first N integers, you multiply them all together, and that gets N factorial.

So, a recursive definition N factorial is that N factorial equals N times N minus 1 factorial. You see, I have the function on both sides of the equation.

Why do I do that? Well, I reduced following the calculation of N factorial to a problem previously solved, because I'm going to assume that I've calculated N minus 1 factorial over here. I just have to take that, multiply it by N, I get N factorial. So, I could say N plus 1 factorial equals N plus 1 times N factorial, and continue onto larger values of N, and calculate N factorial that way. So, that's a simple example of recursion.

Our solution to the Towers of Hanoi problem is a slightly more complicated example of recursion, and that's basically what we're doing today. Now, this is an optional stop point, because you might be running out of time in your 50 minute or 60 minute class today, and that's OK. We have three more segments after this if you have time today, or maybe some other day you want to go into the other three segments, which are a little bit more formal, a little bit more mathematical, and we look at the exponential growth thing and prove that, for example.

But if this is a stop for some of you, I want to point to our website, which has additional resources, and you'll see a lot more information on the Towers of Hanoi there, including computer animation games that you can play. So, if you don't have one of these to work with, a physical one, you can play with animation games in color and these sorts of things-- play it as many times as you like, and have a lot more fun and gain a lot of proficiency in solving Towers of Hanoi.

Now, where's recursive thinking appearing in everyday life? Well, in American football, there's a football coach, here actually in New England near MIT here, who uses a fancy recursive method called dynamic programming-- you can look it up on Google, dynamic programming-- for figuring out what plays to call in a football game. We don't have time today to go into details, but there's a reference under additional resources, which provides those details for you.

Or, a simpler example is a group of people who are exploring a cave, and they go into the cave, and there are a lot of juncture points. They have to make a decision as to what segment of the cave they go into as they're exploring. Suppose they explore for four hours, and they go through 12 different segments. Well, they have to figure out how to get out at the end, so they put trail markers at each of the juncture points, showing once you get back to that point which cave segment you follow out to get out of the cave safely.

So, that's kind of recursive thinking. You know that you get to each juncture point, there will be trail markers, and you have confidence that when you get to the end of that, there'll be another trail marker to show you where to go that way. So, that's recursive thinking exploring caves.

So, a challenge problem for you now in the class discussion is in your class, can you find other examples of where recursive thinking might occur in your everyday life, or when you go out hiking or something, or playing sports? And for those of you who are leaving us today, thank you. We hope to see you again, and for those of you sticking with us, we'll see you soon in about five minutes.

Welcome back. For those of you who are continuing with us today, I hope you had a nice discussion about recursion in real life. Now, we're going to get a little bit more formal, if I can. Remember, we had the recursive equation for factorial, let's get the analogous and recursive equation for Towers of Hanoi. So, fasten your seat belts, because it's a little bit formal here.

So, what is the solution to the Towers of Hanoi problem? First of all, the idea of a solution-- sometimes we think of a solution to an equation, it's a number. The solution to Towers of Hanoi is not a number, it's a set of moves. For the $N = 3$ problem, it's a set of seven moves. That's as efficient as we can be with $N = 3$; $N = 4$, it's 15 moves.

So, let's have some notation. Let's have this notation solution Sol(4, 1, 3). You see on your screen right now Sol(4, 1, 3). What does that mean? That means it's the optimal set of moves for an $N = 4$ disk problem, where the disks originally are on tower one-- the leftmost tower-- and have to be moved following our two rules of movement to tower three, the rightmost tower. So, that's the solution: Sol(4, 1, 3), and we can write a recursive equation, and we know what that equation is, because we have seen how we can delegate authority on down the line.

So, here it is on your screen right now. So, the solution for the four problem of moving from tower one to tower three is the solution for the three problem, moving from one to tower two-- that's an intermediate move-- that clears the way to move the largest disk from its original location to its final location, the rightmost tower, tower three. So, that's the solution-- that's called Sol(1, 1, 3)., plus-- again, the solution for a three disc problem, moving from tower two, the intermediate tower, over to tower three.

So, when we say solution of (4, 1, 3), we're taking the four smallest disks, and figuring out the optimal set of moves to move those from tower one to tower three. There may be other larger disks under it-- maybe it's a six-disk problem, we're just focusing on the smallest four.

So, you see that equation on your screen. Now, what does plus mean in this case? We're saying solution of (3, 1, 2) plus solution of (1,1, 3) plus solution of (3, 2, 3). Well, the plus is not an additive sign like arithmetic. In set theory, it's the union of sets.

So, you see on your screen right now an image, for instance, for the solution of (3, 2, 1). We have seven moves, and then for the one disk problem, we have one move. Then again, we have seven moves. You sum those together: we have seven items plus 1 plus 7. Add those together, we have 15, and we know that that is the optimal number of moves for the four-disk problem.

So, that in equation form is the recursive equation for the Towers of Hanoi problem. Isn't that beautiful? Isn't that wonderful? So, you could say it on words.

If you think you understand this, I have a challenge for you. The challenge is write the recursive equation for N = 5. It shouldn't take too long-- write the recursive equation for N = 5. Talk to your neighbor if you have a question or problem about it, and we'll see you again in a few minutes.

Welcome back. Congratulations, because I know you did it for N = 5. It wasn't too complicated. I'm convinced now you could do it for N = 17, for any value of N. That's great.

Those of you who are interested in computer programming, using that recursive relationship, you could write a solution to the Towers of Hanoi problem with very few lines of code, and you just use recursion all the time. You might even find on the Web, and we have it on our additional resources, a video of a robot solving the Towers of Hanoi problem for a large number of disks. I'll bet that robot was programmed with a recursive relationship that you now know. So, congratulations.

We have something left over that we never proved. We speculated that the size of the problem grows exponentially. Basically, for every new disk that we add, we double the number of moves we have to make to solve the Tower of Hanoi problem, but we never proved it. So, let's do that, because it's going to be easy.

There's an equation involved. Let's have an operator called count-- C- O- U- N- T-- count, and we'll put count here, and solution here. So, count of solution of (3, 1, 3) would be the count of the number of moves we have to make to solve an N = 3 disk problem. We know that's seven for three disks. So, we have the count operator over the solution, and all we're doing is counting up the number of moves to do that particular solution.

We can take the recursive equation that we had last time, and put a count in front of each of the three terms on this side, in front of this term on this side. We then get-- and you see it on your screen right now-- the following equation that says Count for the N = 4 problem, moving all four disks from tower 1 to tower 3, equals the sum of three other counts. The first one is moving three disks from tower one to tower two, then moving the largest disk from tower one to tower three, and then moving the three disks that are parked temporarily on tower two over to tower three-- 7 plus 1 plus 7 is 15, 2 to the fourth minus 1. So, that's what we have for the count for 4, for N = 4 disks.

Suppose we hypothesize that the count for an arbitrary N number of disks is 2^N -1. So, if it were, let's say, N = 5-- 2, 4, 8, 16, 32, subtract 1, that would be 31 moves. Suppose we hypothesize that the count for the end of this problem is 2^N -1. Then, if we plug back in for the count for an (N + 1) problem, we see-- and you see the equation on your screen right now-- and we work out the three terms on the right hand side, which assumes that the count for N disk problem is 2^N -1. We see that-- aha!-- then the count for the N + 1 problem is 2^{N+1} -1. That's recursion-- that's iteration. If we know it's true for a small value of N, we know it's true for the next larger value of N, and then it's true for the next larger value of N, and so on and so on and so on.

Ladies and gentlemen, we have proved exponential growth. This kind of proof is called proof by induction. You may or may not have seen this before. Proof by induction deserves its own BLOSSOMS module. We don't have time today to do that, but I thought this is a nice way to show that the exponential hypothesis that we had way back, I think, in segment number two is, in fact, true, and we proved it. Congratulations to you.

Now, that was a little bit of-- you can unfasten your seat belts now, because I have a fun challenge for you. There is a myth, there is a legend, that in South Asia somewhere, there exists some monks in a monastery, or priests, and they have a very large-- much larger than this-- Towers of Hanoi problem. How large is it? 64 disks. They're very efficient-- these monks really are very efficient. They know how to make every move optimally, so they never make a mistake in terms of the moves and they work 24/7 on this. They move one disk every second-- one disk every second, every second of a minute, every minute of an hour, every hour of a day, every day of the week, as long is it takes.

So, here's your challenge. How long will it take these monks to solve the N = 64 problem? Will it be one day? I want you to vote and share it with your teacher. One week, one month, one year, 10 years, 100 years, or longer? Vote first, let your teacher, put that up on the blackboard about your votes, and then do the calculation. See if you can estimate how long it would take these monks to solve the N = 64 problem, doing one disk move per second. This should be fun. See you soon in a few minutes

Welcome back. We're at our concluding segment for this BLOSSOMS module on Towers of Hanoi. I'm sure you had an interesting and fascinating discussion about the mystery of the monks with the 64 disks and their Towers of Hanoi. On your screen right now, you see the number of seconds it would take for those monks to solve this Tower of Hanoi problem, assuming they make one move per second, and they're very efficient, and never make an error.

Now, you probably didn't get to that exact number, because that takes a lot of calculation to do, but maybe you've approximated in certain ways. But, if you translate that into years-- and to do that, you have to figure out how many seconds there are in a year, and I know you how to do this-- 585 billion years. Do you know planet Earth is only 4.567 billion years old? I bet you know that, because we have a BLOSSOMS module on geologic time, and it talks about that.

So, those monks are going to be working for a long time, and obviously that's an intractable problem when N = 64, because of the exponential growth. It just shows you the power of exponential growth.

There's one last thing we should discuss, and that is suppose you yourself have to solve a Towers of Hanoi problem for, let's say, N =6, N = 7, and you don't have Dr. 4, Dr. 3, Dr. 2, all these doctors, next to you to do it. You have to pretend to be the doctors yourself, thinking this recursive way that we've framed and formulated and solved the problem.

We're not going to create a recipe for you, because if it's a recipe that you memorize, then you're going to forget the beauty of recursion. You have to pretend to be Dr. 3 and Dr. 5 and Dr. 7. You have to pretend to be those doctors when you're doing this. But, here's one of the key hints: what do you do on the first move? Where do you take the smallest guy, the smallest disk, and where do you put it?

Let's go and look again at the N = 3 problem. Here's the N = 1 problem. This is simple: we just take it from tower one to tower three, and we're done. Now, if we have the N = 2 problem, an even number of disks, our first move is to tower two, then to tower three, and then to tower three again, and we're done.

Suppose we have three. Our first move is to tower three, and then to tower two, then to tower two, and like this, and like this, and we're done. Our first move is to tower two if the number of disks is even, and our first move is to tower three if the number of disks is odd, so that's a hint on how to get started. But, we're not going to have a recipe for the whole thing. I want you to convince yourself if that's correct, and then you can think about, over time, over the next day or week, how you might do this for, let's say, an eight disk problem. If you're doing this yourself, thinking recursively.

You can work out the details. I know you'll have fun doing it. You can do it with your friends. So, thanks for your attention. We've had an interesting ride-- a couple of times you've fastened your seat belts, and you survived that, which is good. Going forward, may all of your complex problems be reduced to a sequence of previously solved ones. See you soon.

Hi. I'm Dick Larson. I'm, I guess, the architect of this BLOSSOMS module on Towers of Hanoi, and this is a video teacher's guide. So, if you're joining us here for the first time. Welcome, if you've looked at the module first, welcome here, too.

Basically what we have here is a highly interactive, problem-based experience for your students and yourself to learn about recursive thinking. This develops their critical thinking skills in them to look at a problem which is kind of complex but we can make it simpler by redefining it in terms of a sequence of simpler, previously solved problems.

We've designed it with no formal prerequisites at hand. It should be accessible to any high school math class which is college bound.

Now, you might say, well, don't they have to know anything? It would be nice if they have experienced set theory, but it's not mandatory. It'd be nice if they ever experienced exponential growth, but that's not mandatory, either. Even the beginning parts of the module could be accessible, I think, to middle school students, as well-- not the end parts. So, the end parts where we have mathematical proofs of recursion and exponential growth, probably those are at the junior or senior level for college bound math classes.

Now, it's important that every student have a Towers of Hanoi in front of them, and I doubt whether it's going to be as fancy as the ones we used here in this module. But, even if they just have a stack of five coins, each with a different radius, with the smallest one on top and the biggest one in the bottom in descending order. Or, it could be metal or rubber washers they get from a local hardware store. They could assemble their own Towers of Hanoi, put it on a piece of paper, and put three dots on the piece of paper as to where tower one, two, and three would go.

They could do it that way, but it is important that in your class you have at least one physical Tower of Hanoi, so that the students when they play Drs. 3, 4, and 5 can pass it back and forth to each other. Your class hopefully will have at least one. As you've seen, and as shown on the screen right now, it's me making my own Towers of Hanoi for less than $2 from the local hardware store-- I did this last weekend at my home.

That's the background. Now, let's go to some advice for you for each of the breaks.

First, for activity one: this is basically just so the students can get some familiarity with the problem. They work with the problem and get comfortable, they know our two rules, and they try to solve the N = 3 problem in a minimal number of moves. A few will get it-- I would say, most will not, and that's OK. It's just to get them familiarizing with the general problem. That's the focus, and you can help them with that focus in activity one.

For activity two, it's pretty simple. We see that for N = 1, there's one move. For N = 2, there are three moves. For N = 3, the best we can do, it seems, is seven moves. So, it's to get the students to think, how fast is this growing? Well, it's almost growing as 2^N. You as the teacher can guide them-- in fact, you might want to, once they get the hint and you give some hints, you might actually even show on the blackboard 2^N for N up to 7 or 8 or something like this. Then they'll see that gee, it looks like this grows as 2^N with a little bit of a tweak, and the tweak is minus 1. We don't prove it, but it gets them into the problem so that they understand that the problem grows-- basically almost doubles-- each time you add a disk.

Activity three continues a discussion of exponential growth. In that segment, I discuss the exponential growth, and say, it looks like that's our hypothesis. Let's hold on proving that till later, and that's OK, because we're getting into intuition now, and some knowledge of the problem. Then, basically, we ask them to think about patterns. If they mark down each of the seven moves they make in solving the N = 3 problem, they'll see things like certain patterns, like-- the smallest one is moved every other move, or the biggest one is moved just once from its initial location to its final destination, and the middle one is moved twice.

For them to think of those patterns and from those patterns, is there anything we can do to extend this more generally? I don't expect them to come up with the recursion idea. They'd be very, very, very bright if they did this, but at least they could start to see that there are some patterns here, and that's the objective of this break.

In activity four, we have basically shown for N = 3 that we really just solve the N = 2 problem twice, and in between each of those solutions, we move the biggest of the three disks from tower one to tower three. So, basically solving for N = 3 just involved solving N = 2 twice and plus making one move. We want them to think about how they might apply that to N = 4.

You can help them and guide them. Some of them will figure this out, and some of them will not-- we want them to talk with their neighbors and figure out strategies. Again, you can see this is experiential learning problem-based learning as they delve more deeply into it. They get more involved, they get more intuition, and they get more understanding of the problem.

Now, to activity five. This is after our guest appearance by Dr. 4, otherwise known as Anna Teytelman. This is the segment that probably is going to require your most detailed attentiveness, because Dr. 4 just asked a question to the audience, to your students, what do I do? I just got this note from Dr. 5. You have to encourage the students about how to think about this, and you encourage them and say, well, Dr. 4 only knows how to delegate authority. You might show the note again to them, the note that she got from Dr. 5. Basically, Dr. 5 is asking Dr. 4 to assemble the top four of the five disks onto tower number two.

So, Dr. 4 only knows how to do one thing, and that's delegate authority, so she has to write a note to Dr. 3 to say, initially, to take the top three of the five disks, and move them from tower one to tower three. She puts that note through, along with the Towers of Hanoi, and back it comes, and those three are assembled on the tower three. Then, Dr. 4 knows how to move the largest remaining disk on tower one, which is the next to the largest disk. She moves that onto tower two, and then she has to write another note that says reassemble the three disks that are on tower three, and put them back onto tower two. She writes that note, hands it to Dr. 3, and out it comes. Basically it sounds a little bit complicated, but it really is not, because all Dr. 4 is doing is delegating authority twice, and moving the one larger disk once-- that's all she knows how to do. Once she gets that second thing back from Dr. 3, she smiles, and hands the partially completed Towers of Hanoi back to Dr. 5.

Then you might say, well, what's going to happen next? What's going to happen next is there's going to be another note, and that thing is going to come back again, and, say, reassemble some things on the other tower. So, you might talk about that. Once the students understand this whole process, they'll understand the entire recursion. This is probably the most complex of the breaks, and will require your diligent attentiveness to these issues. The rest of the activities are a lot simpler.

Basically for activity six, you ask for three student volunteers if there's a lot of excitement in the class, maybe you'll get nine student volunteers, in which case, you create three teams-- team one, two, or a, b, and c, whatever. In each of the teams, there's a Dr. 3, Dr. 4 and Dr. 5. You start off the whole process with Dr. 5, and she or he delegates authority down to 4, 4 delegates authority down to 3, and 3, we assume, knows how to solve an N = 3 problem efficiently in seven moves, so there's no need for a Dr. 2 or Dr. 1. If the students don't like that, you can put in a Dr. 2 and move on down the line-- that's OK.

But, basically, then, the idea is to see if these three students, following our recursive ideas, can solve the N = 5 problem. It depends on how you're doing in your class with regard to time and timing-- they may not be able to have time to do all 31 moves. Maybe they'll just do the first 10 or 11 or 12 moves to show that they know what's going on, and that would be fine, and that's the end of that segment, that break.

Activity seven is where we formally introduce what we recursion is, so we give the formal definition. It's a mouthful: I repeat it once or twice, and then we give an example of N!. I challenge the class, and you might want to ask them about it and say, well, if you could take 10 of you in the class to do factorials—10! recursively-- how would you do it? You might even want them to do it during this break. They line up, and each one will take the number given to them by the adjacent one next to them, and multiply it by their standing in the line.

So, for instance, if I am number seven in the line, I'm going to take whatever number is given to me by number six, multiply it by 7, and pass it on to the person to my left, who's position number eight, and that's the way the students recursively, with 10 students participating, and can create and calculate 10!. If they could do this in this break, they would really demonstrate they understand experiential recursive thinking.

At the end of this break, you may decide that this is the conclusion for the first class, because probably by the time you get to this point in your class, a 50 or 60 minute class might be over. The final three segments are also mathematically more involved. So, I would suggest that in a first class you stop here, unless you have a 90 minute or 120 minute class and you want to go through the whole thing. So, you might want to visit the next three segments tomorrow, or next week, or some later time.

In activity eight, the break should be really short, because we showed and argued the validity of the recursive relationship for the solution to the Towers of Hanoi for the N = 4 problem. We just ask the students to write down themselves the solution to the N = 5 problem. It's basically the same equation, but you replace the 4 with a 5, and you make equivalent changes on the right hand side of the equation. If they know what they're doing, they'll be able to do this, definitely within a minute or so. You might get some questions from them, be able to answer them, and we come back from that break.

Activity nine involves two things. It's perhaps their first exposure to proof by induction where we actually prove the exponential growth and the number of moves they have to make as this problem increases in size. So, it's a good exercise for them to understand the power of exponential growth, and that's one aspect of what's going on here. Whether or not they've seen proof by induction before, it's OK, because we try to do it in a gentle way, where we don't invoke a lot of theory and formulations.

After that, they get this fun problem with the 64 disks with these monks or priests in a monastery. I don't expect them to come out with that exact number—$2^{64} - 1$ -- that we showed on the screen, but here's a hint. 2^{10} is 1,024, so you could tell the students they can approximate this. We want to get an order of magnitude, so how about approximating 2^{10} as 1,000. So, instead of 1,024 the exact, it's 1,000, and then you can approximate 2^{64} as 1,000 times 1,000 several times and then 2^4 at the end-- you get an answer that's very, very close to the exact one by approximating. The key is for them to see it's hundreds of billions of years, which hopefully should be a big surprise for them.

OK. The number of segments is 10, so at the end of segment 10, there's no formal activity that we talk about. But, hopefully, if we're successful at that time, the students will be abuzz with excitement. They will have learned how to approach the Towers of Hanoi problem in a recursive way, hoping that they can solve it in various recursive approaches that we've talked about. They will have seen the power of exponential growth, they will have seen proof by induction for the first time, they will have seen applications of recursion in everyday life, and other mathematical situations like N! Maybe they've done this experientially in their class.

So, if you have any time left by that time of day, it would be great to guide the students in a discussion about what they think went right, what they think went wrong, are they still confused about a few things, and how they see that recursion could occur in their own lives.

We hope you've enjoyed this. Thank you for selecting this segment, and for this BLOSSOMS module. If you have any questions or issues, please feel free to email me--rclarson@mit.edu. I hope to hear from you, and have fun with this module. Thank you.