[SQUEAKING] [RUSTLING] [CLICKING]

PETER SHOR: But the first thing I want to do is explain the Cayley tree theorem. And so the question is, how many labelings of a tree on n vertices? And what I want to do is I want to give you the first few elements-- the first few numbers in this sequence and let you guess the answer.

So the first one is a tree on two vertices. And if you want to label it with 1 and 2, there's only really one way of doing it, which is that. And so that's n equals 2, n equals 3, 1, 2, 3, which is clearly the same as 1, 3, 2, because these trees-- actually, 2, 1, 3, and 3, 1, 2.

So there's one way of doing this, three ways of doing this. N equals 4. OK, this time I'm not going to list them all out because there are too many to count. But I'm going to list them in the form of the tree. So how many ways are they doing it? Are there labeling a tree which is a path? Can anyone tell me? Yeah?

AUDIENCE: Maybe like 4 choose 2 to choose the two middle ones. And then 2 times 2.

PETER SHOR: Yeah, 4 choose 2 times 2, which is 12, because you get to choose which two are the middle ones. And then there are two ways of completing that.

And how about a tree that looks like this? Wait, I drew one too many leaves on that. How many ways are there of doing this? 4, because the only thing that matters is the middle one. So 4 and this becomes 16.

And let's do n equals 5. And there's one of these-- oops, I drew an extra leaf on that one, too. Sorry. There are five of these. And there are 5 factorial over 2 ways, because there are 5 factorial ways of labeling the vertices from left to right. But this tree is symmetric in left and right, so you're double counting if you do that, which is 60.

And then there are again 5 factorial over 2 because you can label these 1, 3, 2, 4, 5. But you can swap 5 and 4 and get the same tree. So there's an extra factor of 2. And this is 60. And you add these up, you get 125. So does anybody see a pattern? Yeah.

AUDIENCE: Like you multiply by 3 then by 4 then by 5. Like I guess within the sequence.

PETER SHOR: Yeah. Well, I mean yes. That's essentially the right idea. This is 3 to the 1. This is 2 to the 0. This is 4 squared. And this one is 5 cubed. So the Cayley tree theorem says there are n to the n minus 2 ways of doing this. So that's the statement of the Cayley tree theorem.

And there are lots and lots of proofs. So one way you might think of proving it is to find a bijection between labeled trees and sequences of the numbers 1 through n of length n minus 2. And you can do it that way. But this bijection is really incredibly complicated and completely non-intuitive. So that's not really the right way to do it.

So what I'm going to do is try to explain a way of doing this by counting these certain objects twice, or certain object in two different ways. And how do we do this? So let T sub n be number of labeled trees on n vertices. And the first thing we're going to do is change the thing to number of labeled rooted trees on n vertices.

OK, so what do you do to get a rooted tree? Is a tree with a specified vertex, called the root. OK, so if T sub n is the number of labeled trees on n vertices, what is the number of labeled rooted trees on n vertices?

OK, well, what you do is you start with a tree. Let's do this tree, 1, 3, 2. And now you pick a vertex to call the root. So in this case, you can do this is the root. And 3, 1, 2, this is the root. Or you could do 2, 3, 1, and this is the root. So to go from a labeled tree to a rooted tree, labeled rooted tree, we just pick one vertex. And so there are r times T sub n of these.

And now what we're going to count-- what we count in two ways is labeled rooted trees with ordered edges. So let's pick the middle one there, 1, 3, 2. So we need to order the edges. And we can either order this one and then this one, and then this one and then this one. So I'm going to use letters to order the edges because otherwise we have too many numbers floating around and it gets confusing, so 3, 2, 1, A, B.

So how many labeled trees-- in terms of T sub n, the number of labeled trees, how many labeled rooted trees would ordered edges are there going to be? So there are n minus 1 edges. How many ways are there of ordering n minus 1 edges? n minus 1 factorial. Very good.

So this is T sub n times n times n minus 1 factorial. OK. So if we can count this in a way that tells us the answer is n to the n minus 2 times n times n minus 1 factorial, then we have proven the theorem.

OK. So the second way of counting, lemma, there are n to the n minus 2 n factorial labeled rooted trees with ordered edges. Good? So how do we prove this lemma?

What we will do-- proof method. Construct a labeled rooted tree, I'm going to say, via labeled rooted forests, by adding one edge at a time. So back here, we have this n minus 1 factorial, which is the order of the edges. And this order of the edges is just going to be the order that we add the edges to get to labeled rooted forests to get to the label rooted tree. And I'm going to do it by example. But-- well, basically because it's much easier to see by example than if I try to describe the process in words.

So we start with a forest. I never told you what a forest is. A forest is a collection of trees, which I'm going to say that this makes perfect sense. So start with a forest with n trees with one vertex each. 1, 2, 3, 4, 5, 6. Choose one of these trees. And what we're going to do is we're going to add it to a vertex and add it on top of a vertex in another tree. And let's see--

So let's take one of these trees and stick it on top of a vertex in another tree. So this gives us 1, 2, 4, 5, 3, 6. And we want to label this with the letter A because it was the first thing we did.

Now let's take a different tree and stick it on top of another vertex in a tree. And we get 1, 4, 5, 6, 2. And here this was 3. And this was a first step. And this was our second step.

And we keep on doing this. And maybe the next step is we pick this tree and stick it on top of this vertex. And we have 1, 4, 5, 6, 2, 3, and B, A, and C. And we keep on doing this until we get-- until we used up all the trees except one. And we have a labeled rooted tree with an ordering on the edges.

So how many ways are there of doing this? Oh gosh. This is one of these hard to erase chunks. OK. Well, how many configurations are there to start with?

Anybody? How many labeled-- how many labeled forests with n tree with one vertex each are there? I can see nobody's answering it because this question is too easy. There's only one of them. There are n vertices. And they have the labels 1 through n. And since order doesn't matter, there's one of them, so one after K steps have-- well, each step we have n-- each step we reduce one tree.

So we have n minus k trees. Take one node. Add a different tree. We take one node. We add a different tree on that node-- to that node.

So, for example, here, we chose node number 5. And we added the fourth tree to that node. So how many ways are there of doing this step? How many ways are there of choosing a node? Anybody? Yes. How many nodes are there? n.

And we can choose any one of them. So there are n ways of choosing the node. And now we have n minus k trees. And we want to choose a different tree to add on top of the node. So there are n minus k minus 1 trees we can add.

So there's n times n minus k minus 1 ways to do this. So number of-- OK, what was it? Labeled rooted trees-- labeled rooted trees with edge ordering is the product.

Well, so after k steps-- we want to start with how many ways there are after 0 steps. So 0 to-- well, there are n minus 1 steps altogether. So n minus 2, I believe. n, n minus k minus 1.

So this is equal to pi from k equals 0 to n minus 2 n times pi from k equals 0 to n minus 2 of n minus k minus 1, which is equal to-- well, there are n-- there are n minus 1 steps. And each of these, you have a factor of n. So this is just n times n minus 1. And this should be n minus 1 factorial, which-- if I haven't erased it-- yeah, we want to show that there are n to the n minus 2n factorial. But that's the same as n to the n minus 1 times n minus 1 factorial.

So this is the number of labeled rooted trees with an edge ordering. So this implies that T sub n times n times n minus 1 factorial-- so this is the number of labeled rooted trees on n node. And this is the number of edge orderings-- is equal to n to the n minus 1 times n minus 1 factorial, which shows you that T sub n equals minus 2 to the n-- n to the n minus 2, sorry, which is what we wanted to prove.

OK. Yes?

AUDIENCE: When you do the n times n minus k minus 1 thing is if you pick a node and then isn't the second one just picking a tree that's not one of the trees you picked?

PETER SHOR: That's right. Well, it's picking a tree-- no, because you can pick a tree that you've already picked. You just cannot pick a tree-- you cannot pick the tree that node is in. Yeah. So we pick a node here, and then we pick a different tree to put on top of it. And that's 5, 6, 2.

AUDIENCE: But doesn't it matter how you connect them with the other tree? Because you're just picking like another tree--

PETER SHOR: Oh, we're always going to put the root of this tree on top of this node. And you can see that the process can be undone because if we have A, B, C, the way we undo this step is we find the last edge. And we cut it. And we have a tree on top of that edge. And we have the tree that's left when you remove this edge.

So that undoes the step. So the whole process is, I want to say that there is a this function that takes you from trees on k-- n minus k nodes to trees-- or to forests of n minus k trees to forests of n minus k minus 1 trees is reversible. So it's a bijection. But, yeah, that was a good question.

And last year, we actually used this theorem for one of our term paper projects. This year we're not. I wanted to tell you this at the beginning of the class, but I forgot. So what we're going to do is tonight, assuming all goes well, I will put up the term paper projects and their assignments to students. And tomorrow in recitation, the recitation leaders will discuss them a little bit, although most of the recitation tomorrow is going to be devoted to writing. So the term paper projects are hopefully coming out tonight.

So now, we are going to switch topics and go to generating functions. OK. And generating functions are a very useful concept in mathematics. And you can use them to count things. I mean, there are proofs of the Cayley tree theorem that use generating functions. And there are proofs of probably nearly everything we've discussed in terms of counting that use generating functions, although often, the generating functions proofs are more difficult than the non-generating functions proofs.

On the other hand, there are cases where generating functions are really the only way to do it. And the other advantage of generating functions is that they're mechanical. So given, say, combinatorial identity and you look at it and you say, what things can I count in two different ways to prove it, or something else, well, it's-- you know, you have to use some creativity to find this. But if you have a generating combinatorial identity with generating functions, often it's fairly straightforward to prove it using generating functions, although it involves actual algebra. So it's maybe more difficult. So why am I?

Let a1-- well, a0, a1, a2, et cetera, be a sequence of numbers. The generating function associated with the sequence is F of x is equal to sum a sub i x to the i, i equals 0 to infinity.

So this is an algebraic function. And that means you can use all the tools of algebraic functions to manipulate it. And this, we will find, is going to be very useful.

Now, you can ask what does x mean here? And x doesn't really mean anything. It's just a tool for keeping track of what the i is on ai x to the i. So it's a way of keeping the i-th ai distinct from the j-th ai. So it usually doesn't really mean anything.

So the example-- well, one example-- is we'll take the sequence of binomial coefficients. So what is this? f sub n-- oh, right-- a sub n is equal to k choose n. So the generating function for this, F of x is equal to sum. And here n is what we're summing over. n equals 0 through infinity. And-- no, this is 0 if n is bigger than k. So that's sum n equals 0 through K, a k choose n. But let me write an x to the n, k n choose n, x to the n.

And you should know what this is. This is the binomial theorem. And it's just 1 plus x to the k. And one way of thinking about this is F of x equals 1 plus x, 1 plus x, and 1 plus x.

So for the k-th term-- no, for n-th term, I'm sorry, choose n x's, k minus n 1's. Good? So we need to choose whether we'll use the x from this one, the x from this one, the x from this one, et cetera. So we're choosing k x's-- we're choosing n x's out of k things, which is just n choose k-- or k choose n. I'm sorry. And so that's why a sub n is k choose n.

OK. So example 2, the a sub n is the probability of n heads when you flip a coin, a biased coin, k times. And we'll let p equal probability of heads. And q equals 1 minus p.

So A of x is equal to summation a sub n x to the n is equal to k choose n, p to the k, q to the n minus k times x to the k-- x to the n. I keep mixing up my k's and my n's. Let me apologize for that. There we go.

And this is equal to-- I forgot the sum here. This is equal to q plus px to the n, again, by the binomial theorem. So hopefully, I will have time to do a real example that shows you why generating functions are useful before the end of class. But if not, we'll get to it tomorrow, or not tomorrow, Thursday.

OK. I want to say a few more general things about generating functions. Let A, B be disjoint classes of objects. And C equals A union B, where U is disjoint union. So that just says that A intersection B is the empty set.

So I want to say the generating function for C of x. So what we're going to do is we want to express the generating function for C of x in terms of A of x and B of x is equal to A of-- well, it's going to be sum C sub n, x to the n. Well, C sub n is just A sub n plus B sub n is equal to sum A sub n plus B sub n, x to the n, which you can see is the generating function of A of x plus B of x.

So the next one, we want to talk about multiplication of generating functions. This is actually, first, pretty straightforward. And second, is not really that useful when manipulating generating functions. Multiplying generating functions is incredibly useful. And we'll do an example.

So let's do the example first. And then we will do the formal explanation. So let's say we have a 6-sided dice and an 8-sided dice. 8-sided dice. And we want to know what is the probability of rolling, OK, let's call it n, when we add up the numbers from the two die. And I should specify that the 6-sided dice is numbered 1 through 6, and the 8-sided dice is numbered 1 through 8.

So let C sub n is equal to number of ways of rolling n. Claim-- C sub n is equal to-- or rather C sub n-- C of x is equal to sum C sub n x to the n is equal to x plus x squared plus x cubed plus-- plus x to the 6th, x plus x squared plus x cubed plus dot, dot, dot plus x to the 8th equals x plus 2x squared plus 3x cubed plus dot, dot, dot, plus 5x to the 5th plus 6x to the 6th, plus 7x to the-- plus, sorry, 6x to the 7th, plus 6x to the 8th plus 6x to the 9th plus 5x to the 10th, plus dot, dot, dot. Plus x to the 14th.

So how do we get that? So you can see from this that there are five ways of rolling a 10. You can roll a 6 and 4. Or you could roll a 5 and 5, a 4 and 6, a 3 and 7, and 2 and an 8. And this works because C sub n is equal to summation j equals 0. j equals 1 through n. OK. A sub j, B sub n minus j.

So if you want to get an n, the first place has to be j. And the second one has to be n minus j. And A j equals 1, if 1 is less than j is less than 6, 0 otherwise. And B sub j equals 1 if 1 is less than j is less than 8, and 0 otherwise.

Is this clear? I mean to get an n, you have to roll a j on the first dice and an n minus j on the second dice. And you can only roll a j on the first dice if j is between 1 and 6. And you can only roll it n minus j on the second die if j is between 1 and 8.

So if A of x is equal to summation, i equals 1 through 6 x to the i. And B of x is equal to summation i equals 1 through 8 x to the i. That's not a very good 8. Then C of x equals of A of x, B of x, because this is the formula for multiplying polynomials. So we have the generating function for the first die and the generating function for the second die. And we get the generating function for the sum of the two dies by just multiplying them.

And let's make this rigorous. Let A, B be two classes of objects. A of x and B of x be their generating functions. Then-- oh, well-- the Cartesian product, Product A cross B, is A, B, A and A, B and B. And we want to claim that size of A, B equals size of A plus size of B. Then there are the Cartesian product, A cross B, has generating function A of x, B of x equals C of x.

Proof-- OK, C sub n equals summation k equals 0 through n, a sub k, b sub n minus k. So we want to look at the things of size n in the Cartesian product. Well, they're the things of size k in the first generating function and times the things of size n minus k in the second generating function. And we want to sum this over all k.

And now, we just need to check that summation C sub n x to the n is equal to summation k equals 0 through n, a sub k, b n minus k x to the n is equal to sum k equals 0-- or rather, n equals 0 to infinity. I forgot the second sum here-- n equals 0 to infinity, k equals 0 to n, which is just sum k equals 0 to infinity, a sub k, x to the k, summation j equals 0 to infinity, b sub j, x to the j, because when you multiply these, you group the things that have the same k plus j equals n. And that gives you this term, sum k equals 0 to n, A sub k, B sub n minus k. Yeah.

AUDIENCE: Sorry. What does size mean?

PETER SHOR: Size is just an arbitrary function on these objects that correspond to the n here in summation a sub k, x to the k. So when you have generating functions, you're grouping objects into things that go with x to the k. And those are what we're calling size.

AUDIENCE: So is it the power of x?

PETER SHOR: It's the power of x that is associated with the object. And I think we should look at an actual example. So this is a simple example.

Example-- OK, suppose you have one penny-- no, 6 pennies, 1 nickel, 2 dimes. How many different ways can you make n cents?

Claim-- the generating function, let's call it C of x is equal to sum c n, x to the n. So we'll let c sub n be the number of different ways, cn be number of ways to make n cents with 6 pennies, a nickel, and 2 dimes.

C of x is equal to 1 plus x plus x squared plus x cubed plus x 4th plus x 5th plus x 6th, 1 plus x to the fifth, and 1 plus x to the 10th plus x to the 20th. OK, so when you multiply these things together-- let's look at the x to the 20-- let's look at the x to the, I'm thinking, x to the 21st term. So it can be got from x to the 6th times x to the 5th times x to the 10th.

And we can also get it from x times x to the 20th. And those are the only two ways you can get it. So there are two ways to make change-- two ways to pay $0.21. So here the size is just the number of cents.

OK, let's do something a little bit more complicated. How many ways are there to pay, let's say, n cents in an arbitrary number of pennies, nickels, dimes. Well, I want to claim that it's 1 plus x plus x squared plus x cubed plus dot, dot, dot, times 1 plus x to the 5th plus x to the 10th plus x to the 15th, plus dot, dot, dot, times 1 plus x to the 10th, plus x to the 20th, plus x to the 30th, plus dot, dot, dot, which is equal to 1 over 1 minus x, 1 over 1 minus x to the 5th, times 1 over 1 minus x to the 10th. So cn x to the n is equal to that.

OK, so now I want to give an example for how you would actually use a generating function. I mean, we have not gotten to the end of our overall description of generating functions. But I want to give you an example first of how you would actually calculate something with them. And I want to claim this is really pretty neat.

What is sum k equals 0 through n, k n choose k. So the first thing is you can do it by counting something two different ways. And I want to do it with generating functions first. And if I have any time at the end of the class, I'll check it by using the technique of counting in two different ways. But I'm not sure I will.

So what is it? k equals 0 through n, n choose k times x to the k is equal to-- does anybody want to tell me what this is? Yeah.

AUDIENCE: Is it n times 2 to the n minus 1?

PETER SHOR: No, because we want this as a function of something with x's in it.

AUDIENCE: 1 plus x to the n.

PETER SHOR: Yeah, 1 plus x to the n, that's right. 1 plus x to the n. Excellent. OK. Now-- and this is-- we should call this F of x.

Now, how can we get this out of that? Well, let's differentiate. d/dx F of x is equal to sum k equals 0 through n, k n choose k, x to the k minus 1. Right? OK.

And now, let's differentiate this. Well, that's just n times 1 plus x to the n minus 1. And now let's substitute x equals 1. Well, in this, we get sum k equals 0 through n, k n choose k. And in this, we get n times 2 to the n minus 1. OK. We have our answer.

And I want to say that we kind of skipped a step here. Generating function, for k n choose k is sum k n choose k, x to the k. k equals 0 through n. And what we calculated here was just k n choose k, x to the k minus 1. So this is actually n x 1 minus x to the n minus 1.

But since x, if we substitute in x equals 1, it doesn't matter if we have x to the k or x to the k minus 1 here, or if we have this extra-- sorry, plus-- or if we have this extra x here. It will-- if x is 1, they're the same.

So to multiply the n-th term in our-- to multiply the k-th term in a generating function by k, we just differentiate and multiply by x. And this is a very useful trick, playing with generating functions.

So how would you do this with counting two different ways? Anybody? Yes.

AUDIENCE: Is that-- I'm just like, what's the question is it the line above it or the--

PETER SHOR: You mean this?

AUDIENCE: Yeah.

PETER SHOR: It's k n choose k--

AUDIENCE: Oh, you ask like what's like-- sorry, I want to ask, what question are you answering?

PETER SHOR: Equals n to the n minus 1. Well, we're answering what is summation k n choose k.

AUDIENCE: I mean like you just asked, counting two ways, like which--

PETER SHOR: Oh, we want to answer this question by counting in two different ways.

AUDIENCE: So it's like you have a set of n elements and you pick k elements out of it. And then from the k elements you pick one special element?

PETER SHOR: Yeah. Exactly. So we have a set of n elements. Choose k elements. Choose one of the k elements. So we have a special element. And then we have a subset of k and a subset of n minus k minus 1.

And sum over k, we just have a special element of k n choose k. We have a special element and a subset of the remaining subset of n minus 1 other elements. And you can pick the special element n ways.

Other elements-- or the subset, sorry-- the subset of other elements is 2 to the n minus 1 ways. And this gives n times 2 to the n minus 1, which is what we wanted in the first place. OK, so that's two different ways to solve this problem.

And I want to say that so far we're only doing things with generating functions that we could have done more easily some other way. But there are things that you can do with generating functions that are much harder to do other ways. And we will eventually get to those when we start talking about Fibonacci numbers.

And I think the next thing I want to do is too long to fit into 3 minutes. So I will let you go and see everyone on Thursday.