PROFESSOR: I'm going to spend most of time talking about chapters one, two, and three. A little bit talking about chapter four, because we've been doing so much with chapter four in the last couple of weeks that you probably remember that more.

OK. The basics, which we started out with, and which you should never forget, is that any time you develop a probability model, you've got to specify what the sample space is and what the probability measure on that sample space is. And in practice, and in almost everything we've talked about so far, there's really a basic countable set of random variables which determine everything else. In other words, when you find the joint probability distribution on that set of random variables, that tells you everything else of interest. And a sample point or a sample path on that set of random variables is in a collection of sample values, one sample value for each random variable.

It's very convenient, especially when you're in an exam and a little bit rushed, to confuse random variables with the sample values for the random variables. And that's fine. I just want to caution you again, and I've done this many times, that about half the mistakes that people make-- half of the conceptual mistakes that people make doing problems and doing quizzes are connected with getting confused at some point about what's a random variable and what's a sample value of that random variable. And you start thinking about sample values as just numbers. And I do that too. It's convenient for thinking about things. But you have to know that that's not the whole story.

Often, we have uncountable sets of random variables. Like in renewal processes, we have the counting renewal process, which typically has an uncountable set of random variables, a number of arrivals up to each time, t, where t is a continuous valued random variable. But in almost all of those cases, you can define things in terms of simpler sets of random variables, like the interarrival times, which are IID. Most of the processes we've talked about really have a pretty simple description if you look for the simplest description of them.

If you have a sequence of IID random variables-- which is what we have for Poisson and renewal processes, and what we have for Markov chains is not that much more complicated-- the laws of large numbers are useful to specify what the long term behavior is. The sample time average is, as we all know by now, is the sum of the random variables divided by n. So it's a sample average of these quantities.

The random variable, which has a main x bar, the expected value of x, that's almost obvious. You just take the expected value of s sub n, and it's n times the expected value of x divided by n, and you're done. And the variance, since these random variables are independent, you find that almost as easily. That has this very simple-minded distribution function.

Remember, we usually work with distribution functions in this class. And often, the exercises are much easier when you do them in terms of the distribution function than if you use formulas you remember from elementary courses, which are specialized to-- which are specialized to probability density and probability mass functions, and often have more special conditions on them than that.

But anyway, the distribution function starts to look like this. As n gets bigger, you notice that what's happening is that you get a distribution which is scrunching in this way, which is starting to look smoother. The jumps in it gets smaller. And you start out with this thing which is kind of crazy. And by time, n is even 50. You get something which almost looks like a-- I don't know how we tell the difference between those two things. I thought we could, but we can't. I certainly can't up there.

But anyway, the one that's tightest in is the one for n equals 50. And what these laws of large numbers all say in some sense is that this distribution function gets crunched in towards an impulse at the mean. And then they say other more specialized things about how this happens, about sample paths and all of that. But the idea is that this distribution function is heading towards a unit impulse.

The weak law of large numbers then says that if the expected value of the magnitude of x is less than infinity-- and usually when we talk about random variables having a mean, that's exactly what we mean. If that condition is not satisfied, then we usually say that the random variable doesn't have a mean. And you'll see that every time you look at anything in probability theory. When people say the mean exists, that's what they always mean.

And what the theorem says then is exactly what we were talking about before. The probability that the difference between s n over n, and the mean of x bar, the probability that it's greater than or equal to epsilon equals 0 in the limit. So it's saying that you put epsilon limits on that distribution function and let n get bigger and bigger, it goes to 1 and 0.

It says the probability of s n over n, less than or equal to x, approaches a unit step as n approaches infinity. This says this is the condition for convergence in probability. What we're saying is that that also means convergence and distribution function, and distribution for this case. And then we also, when we got to renewal processes, we talked about the strong law of large numbers. And that says that the expected value of x is finite. Then this limit approaches x on a sample path basis. In other words, for every sample path, except this set of probability 0, this condition holds true.

That doesn't seem like it's very different or very important for the time being. But when we started studying renewal processes, which is where we actually talked about this, we saw that in fact, it let us talk about this, which says that if you take any function of s n over n-- in other words, a function of a real value-- a function of a-- a real valued function of a-- a real valued function of a real value, yes. What you get is that same function applied to the mean here. And that's the thing which is so useful for renewal processes. And it's what usually makes the strong law of large numbers so much easier to use than the weak law. That's a plug for the strong law.

There are many extensions of the week love telling how fast the convergence is. One thing you should always remember about the central limit theorem, is it really tells you something about the weak law of large numbers. It tells you how fast that convergence is and what the convergence looks like. It says that if the variance of this underlying random variable is finite, then this limit here is equal to the normal distribution function, the Gaussian at variance 1 and mean 0.

And that becomes a little easier to see what it's saying if you look at it this way. It says probability that s n over n minus x bar-- namely the difference between the sum and the mean which it's converging to-- the probability that that's less than or equal to y sigma over square root of n is this normal Gaussian random variable. It says that as n gets bigger and bigger, this quantity here gets tighter and tighter. What it says in terms of the picture here, in terms of this picture, it says that as n gets bigger and bigger, this picture here scrunches down as 1 over the square root of n. And it also becomes Gaussian.

| it tells you exactly what kind of convergence you actually have here. Is not only saying that this does converge to a unit step. It says how it converges. And that's a nice thing, conceptually. You don't always need it in problems. But you need it for understanding what's going on. We're moving backwards, it seems.

Now, 1, 2, Poisson processes. We talked about arrival processes. You'd almost think that all processes are arrival processes at this point. But any time you start to think about that, think of a Markov chain. And a Markov chain is not an arrival process, ordinarily. Some of them can be viewed that way. But most of them can't.

An arrival processes is an increasing sequence of random variables. 0 less than s1, which is the time of the first arrival, s2, which is a time of the second arrival, and so forth. Interarrival times are x1 equals s1, and x i equals s i minus s i minus 1.

The picture, which you should have indelibly printed on the back of your brain someplace by this time, is this picture here. s1, s2, s3, are the times at which arrivals occur. These are random variables, so these arrivals come in at random times. x1, x2, x3 are the intervals between arrivals. And N of t is the number of arrivals that have occurred up until time t. So every time the t passes one of these arrival times, N of t pops up by one, pops up by one again, pops up by one again. The sample value pops up by one.

Arrival process can model arrivals to a queue, departures from a queue, locations of breaks in an oil line, an enormous number of things. It's not just arrivals we're talking about. It's all of these other things, also. But it's something laid out on a one-dimensional axis where things happen at various places on that one-dimensional axis. So that's the way to view it.

OK, same picture again. Process can be specified by the joint distribution of the arrival epochs or the interarrival times, and, in fact, of the counting process. If you see a sample path of the counting process, then from that you can determine the sample path of the arrival times and the sample path of the interarrival times. And since any set of these random variables specifies all three of these things, the three are all equivalent.

OK, we have this important condition here. And I always sort of forget this, but when these arrivals are highly delayed, when there's a long period of time between each arrival, what that says is the accounting process is getting small. So big interarrival times corresponds to a small value of N of t. And you can see that in the picture here. If you spread out these arrivals, you make s1 all the way out here. Then N of t doesn't become 1 until way out here. So N of t as a function of t is getting smaller as s sub n is getting larger.

S sub n is the minimum of the set of t, such that N of t is greater than or equal to N. Sounds like a unpleasantly complicated expression. If any of you can find a simpler way to say it than that, I would be absolutely delighted to hear it. But I don't think there is. I think the simpler way to say it is this picture here. And the picture says it. And you can sort of figure out all those logical statements from the picture, which is intuitively a lot clearer, I think.

So now, renewal processes is an arrival process with IID interarrival times. And a Poisson process is a renewal process where the interarrival random variables are exponential. So, Poisson process is a special case of renewal process. Why are these exponential interarrival arrival times so important? Well, it's because they're memoryless. And the memoryless property says that the probability that x is greater than t plus x is equal to the probability that it's greater than x times the probability that it's greater than t for all x and t greater than or equal to 0.

This makes better sense if you say it conditionally. The probability that x is greater than t plus x, given that it's greater than t, is the same as the probability that x is greater that-- capital X is greater than little x. This really gives you the memoryless property in a nutshell. It says if you're looking at this process as it evolves, and you see an arrival, and then you start looking for the next arrival, it says that no matter how long you've been looking, the distribution function, as the time to wait until the next arrival, is the same exponential random variable.

So you never gain anything by waiting. You might as well be impatient. But it doesn't do any good to be impatient. Doesn't to any good to wait. It doesn't do any good to not wait. No matter what you do, this damn thing always takes an exponential amount of time to occur. OK, that's what it means to be memoryless. And the exponential is the only memoryless random variable.

How about a geometric random variable? The geometric random variable is memoryless if you're only looking at integer times. Here we're talking about times on a continuum. That's what this says. Well, that's what this says. And if you look at discrete times, then a geometric random variable is memoryless also.

We're given a Poisson of rate lambda. The interval from any given t greater than 0 until the first arrival after t is a random variable. Let's call it z1. We already said that that random variable was exponential. And it's independent of all arrivals which occur before that starting time t. So looking at any starting time t, doesn't make any difference what has happened back here. That's not only the last arrival, but all the other arrivals. The time until the next arrival is exponential. The time until each arrival after that is exponential also, which says that if you look at this process starting at time t, it's a Poisson process again, where all the times have to be shifted, of course, but it's a Poisson process starting at time t.

The corresponding counting process, we can call it n tilde of t and tau, where tau is greater than or equal to t, where this is the number of arrivals in the original process up until time tau minus the number of arrivals up until time t. If you look at that difference, so many arrivals up until t, so many more up until time tau. You look at the difference between tau and t. The number of arrivals in that interval is the same Poisson distributing random variable again. So, it has the same distribution as N of tau minus t.

And that's called the stationary increment property. It says that no matter where you start a Poisson process, it always looks exactly the same. It says that if you wait for one hour and start then, it's exactly the same as what it was before. If we had Poisson processes in the world, it wouldn't do any good to travel on certain days rather than other days. It wouldn't do any good to leave to drive home at one hour rather than another hour. You'd have the same travel all the time. It's all equal. It would be an awful world if it were stationary.

The independent increment properties for counting process is that for all sequences of ordered times-- 0 less than t1 less than t2 up to t k-- the random variables n of t1-- and now we're talking about the number of arrivals between t1 and t2, the number of arrivals between n minus 1 and tn. These are all independent of each other. That's what this independent increment property says. And we see from what we've said about this memoryless property that the Poisson process does indeed have this independent increment property. Poisson processes have both the stationary and independent increment properties.

And this looks like an immediate consequence of that. It's not. Remember, we had to struggle with this for a bit. But it says plus Poisson processes can be defined by the stationary and independent increment properties, plus either the Poisson PMF for N of t, or this incremental property, the probability that N tilde of t and t plus delta, and the number of arrivals between t and t plus delta, the probability that that's 1 is equal to lambda times delta. In other words, this view of a Poisson process is the view that you get when you sort of forget about time. And you think of arrivals from outer space coming down and hitting on a line. And they hit on that line randomly. And each one of them is independent of every other one. And that's what you get if you wind up with a density of lambda arrivals per unit time. OK, we talked about all of that, of course.

The probability distributions-- there are many of them for a Poisson process. The Poisson process is remarkable in the sense that anything you want to find, there's generally a simple formula for it. If it's complicated, you're probably not looking at it the right way. So many things come out very, very simply.

The probability-- the joint probability distribution of all of the arrival times up until time N is an exponential just in the last one, which says that the intermediate arrival epochs are equally likely to be anywhere, just as long as they satisfy this ordering restriction, s1 less than s2. That's what this formula says. It says that the joint density of these arrival times doesn't depend on anything except the time of the last one.

But it does depend on the fact that they're [INAUDIBLE]. From that, you can find virtually everything else if you want to. That really is saying exactly the same thing as we were just saying a while ago. This is the viewpoint of looking at this line from outer space with arrivals coming in, coming in uniformly distributed over this line interval, and each of them independent of each other one. That's what you wind up saying.

This density, then, of the n-th arrival, if you just integrate all this stuff, you get the Erlang formula. Probability of arrival n in t to t plus delta is-- now this is the derivation that we went through before, going from Erlang to Poisson. You can go from Poisson to Erlang too, if you want to. But it's a little easier to go this way. The probability of arrival in t to t plus delta is the probability that n of t is equal to n minus 1 times lambda delta plus an o of delta, of course. And the probability that n of t is equal to n minus 1 from this formula here is going to be the density of when s sub n appears, divided by lambda. That's exactly what this formula here says. So that's just the Poisson distribution.

We've been through that derivation. It's almost a derivation worth remembering, because it just appears so often. As you've seen from the problem sets we've done, almost every problem you can dream of, dealing with Poisson processes, the easy way to do them comes from this property of combining and splitting Poisson processes. It says if n1 of t, n2 of t, up to n sub k of t are independent Poisson processes-- what do you mean by a process being independent of another process? Well, the process is specified by the interarrival times for that process. So what we're saying here is the interarrival times for the first process are independent of the interarrival times of the second process, independent of the interarrival times for the third process, and so forth.

Again, this is a view of someone from outer space, throwing darts onto a line. And if you have multiple people throwing darts on a line, but they're all equally distributed, all uniformly distributed over the line, this is exactly the model you get.

So we have two views here. The first one is to look at the arrival epochs that's generated from each process. And then combine all arrivals into one Poisson process. So we look at all these Poisson processes, and then take the sum of them, and we get a Poisson process.

The other way to look at it-- and going back and forth between these two views is the way you solve problems-- you look at the combined sequence of arrival epochs first. And then for each arrival that comes in, you think of an IID random variable independent of all the other random variables, which decides for each arrival which of the sub-processes it goes to. So there's this hidden process-- well, it's not hidden. You can see what it's doing from looking at all the sub-processes. And each arrival then is associated with the given sub-process, with the probability mass function lambda sub i over the sum of lambda sub j. So this is the workhorse of Poisson type queueing problems.

You study queuing theory, every page, you see this thing used. If you look at Kleinrock's books on queueing, they're very nice books because they cover so many different queueing situations. You find him using this on every page. And he never tells you that he's using it, but that's what he's doing. So that's a useful thing to know.

We then talked about conditional arrivals and order statistics. The conditional distribution of the N first arrivals-- namely, s sub 1 s sub 2 up to s sub n-- given the number of arrivals in N of t is just n factorial over t to the n. Again, it doesn't depend on where these arrivals are. It's just a function which is independent of each arrival. It's the same kind of conditioning we had before. It's n factorial divided by t to the n. Because of the fact that if you order these random variables, t1 less than t2 less than t3, and so forth, up until time t, and then you say how many different ways can I arrange a set of numbers, each between 0 and t so that we have different orderings of them. And you can choose any one of the N to be the first. You can choose any one of the remaining n minus 1 to be the second. And that's where this is n factorial comes from here. And that, again we've been over.

The probability that s1 is greater than tau, given that they're interarrivals in the overall interval t, comes from just looking at N uniformly distributed random variables between 0 and t. And then what do you do with those uniformly distributed random variables? Well, you ask the question, what's the probability that all of them occur after time tau? And that's just t minus tau divided by t raised to the n-th power.

And see, all of these formulas just come from particular viewpoints about what's going on. You have a number of viewpoints. One of them is throwing darts at a line. One of them is having exponential interarrival times. One of them is these uniform interarrivals. It's only a very small number of tricks. And you just use them in various combinations. So the joint distribution of s1 to s n, given N of t equals n, is the same as the joint distribution of N uniform random variables after they've been ordered.

So let's go on to finite state Markov chains. Seems like we're covering an enormous amount of material in this course. And I think we are. But as I'm trying to say, as we go along, it's all-- I mean, everything follows from a relatively small set of principles. Of course, it's harder to understand the small set of principles and how to apply them than it is to understand all the details. But that's-- but on the other hand, if you understand the principles, then all those details, including the ones we haven't talked about, are easy to deal with.

An integer-time stochastic process-- x1, x2, x3, blah, blah, blah-- is a Markov chain if for all n, namely the number of them that we're looking at-- well-- for all n, i, j, k, l, and so forth, the probability that the n-th of these random variables is equal to j, given what all of the others are-- and these are not ordered now. I mean, in a Markov chain, nothing is ordered. We're not talking about an arrival process. We're just talking about a frog jumping around on lily pads, if you arrange the lily pads in a linear way, if these are random variables.

The probability that the n-th location is equal to j, given that the previous locations are i, k, back to m, is just some probability p sub i j, a conditional probability of j given i. In other words, once if you're looking at what happens at time n, once you know what happened at time n minus 1, everything else is of no concern. This process evolves by having a history of only one time unit, a little like the Poisson process. The Poisson process evolves by being totally independent of the past. Here, you put a little dependence in the past. But the dependence is only to look at the last thing that happened, and nothing before the last time that happened.

So p sub i j depends only on i and j. And the initial probability mass function is arbitrary. Markov chain is finite-state if the sample space for each x i, as a finite set S. And the sample space S is usually taken to be integers 1 up to M.

In all these formulas we write, we're always summing from one to M. And the reason for that is we've assumed the states are 1, 2, 3, up to M. Sometimes it's more convenient to think of different state spaces. But all the formulas we use are based on this state space here.

Markov up chain is completely described by these transition probabilities plus the initial probabilities. If you want to write down the probability of what x is this some time N given what was at some time 0, all you have to do is trace all the paths from 0 out to N, add up the probabilities of all of those paths, and that tells you the probability you want. All probabilities and be calculated just from knowing what these transition probabilities are.

Note that when we're dealing with Poisson processes, we defined everything in terms of how many-- how many variables are there in defining a Poisson process? How many things do you have to specify before I know exactly what Poisson process I'm talking about? Only the Poisson rate. Only one parameter is necessary for a Poisson process.

For a finite-state Markov process, you need a lot more. What you need is all of these values, p sub i j. If you sum p sub i j over j, you have to get 1. So that removes one of them. But as soon as you specify that transition matrix, you've specified everything. So there's nothing more to know about the Poisson process. There's only all these gruesome derivations that we go through. But everything is initially determined.

Set of transition probabilities is usually viewed as the Markov chain. And the initial probabilities are usually viewed as just a parameter that we deal with. In other words, we-- in other words, what we study is the particular Markov chain, whether it's recurrent, whether it's transient, whatever it is. How you break it up into classes, all of that stuff only depends on these transition probabilities and doesn't depend on where you start.

Now, a finite-state Markov chain can be described either as a directed graph or as a matrix. I hope you've seen by this time that some things are easier to look at if you look at things in terms of a graph. Some things are easier to look at if you look at something like this matrix. And some problems can be solved by inspection, if you draw a graph of it. Some can be solved almost by inspection if you look at the matrix. If you're doing things by computer, usually computers deal with matrices more easily than with graphs.

If you're dealing with a Markov chain with 100,000 states, you're not going to look at the graph and determine very much from it, because it's typically going to be fairly complicated-- unless it has some very simple structure. And sometimes that simple structure is determined. If it's something where you can only-- where you have the states numbered from 1 to 100,000, and you can only go from state i to state i plus 1, or from state i to i plus 1, or i minus 1, then it becomes very simple. And you like to look at it as a graph again. But ordinarily, you don't like to do that.

But the nice thing about this graph is that it tells you very simply and visually which transition probabilities are zero, and which transition probabilities are non-zero. And that's the thing that specifies which states are recurrent, which states are transient, and all of that. All of that kind of elementary analysis about a Markov chain all comes from looking at this graph and seeing whether you can get from one state to another state by some process.

So let's move on from that. Talk about the classification of states. We started out with the idea of a walk and a path and a cycle. I'm not sure these terms are uniform throughout the field. But a walk is an ordered string of nodes, like i0, i1, up to i n. You can have repeated elements here, but you need a directed arc from i sub n minus 1 to i sub m. Like for example, in this stupid Markov chain here-- I mean, when you're drawing things is LaTeX, it's kind of hard to draw those nice little curves there. And because of that, when you once draw a Markov chain, you never want to change it. And that's why these nodes have a very small set of Markov chains in them. It's just to save me some work, drawing and drawing these diagrams.

An example of a walk, as you start in 4, you take the self loop, go back to 4 at time 2. Then you go to state 1 at time 3. Then you go to state 2 at time 4. Then you go to stage 3, time 5. And back to state 2 at time 6.

You have repeated nodes there. You have repeated nodes separated here. Another example of a walk is 4, 1, 2, 3. Example of a path, the path can't have any repeated nodes. We'd like to look at paths, because if you're going to be able to get from one node to another node, and there's some walk that goes all around the place and gets to that final node, there's also path that goes there. If you look at the walk, you just leave that all the cycles along the way, and you get to the n. And a cycle, of course, which I didn't define, is something which starts at one node, goes through a path, and then finally comes back to the same node that it started at. And it doesn't make any difference for the cycle 2, 3, 2 whether you call it 2, 3, 2 or 3, 2, 3. That's the same cycle, and it's not even worth distinguishing between those two ideas. OK That's that.

If there's a path from-- where did I-- node j is accessible from i, which we abbreviate as i has a path to j. If there's a walk from i to j, which means that p sup i j to the n-- this is the transition probability, the probability that x sub n is equal to j, given that x sub 0 is equal to i. And we use this all the time. If this is greater than zero for some n greater than 0. In other words, j is accessible from i if there's a path from i that goes to j.

And trivially, if i go to j, and there's a path from j to k, then there has to be a path from i to k. If you've ever tried to make up a mapping program to find how to get from here to there, this is one of the most useful things you use. If there's a way to get here to there, and a way to get from here to there, then there's a way to get from here all the way to the end. And if you look up what most of these map programs do, you see that they overuse this enormously and they wind up taking you from here to there by some bizarre path just because it happens to go through some intermediate node on the way.

So two nodes communicate-- i double arrow j-- if j is accessible from i, and if i is accessible from j. That means there's a path from i to j, and another path from j back to i, if you shorten them as much as you can. There's a cycle. It starts at i, goes through j, and comes back to i again. I didn't say that quite right, so delete that from what you've just heard.

A class C of states as a non-empty set, such that i and j communicate for each i j in this class. But i does not communicate with j for each i and C-- for i and C and j, not in C.

The convenient way to think about this-- and I should have stated this as a theorem in the notes, because it's-- I think it's something that we all use without even thinking about it. It says that the entire set of states, or the entire set of nodes in a graph, is partitioned into classes. The class C, containing, is i in union with all of the j's that communicate with i. So if you want to find this partition, you start out with an arbitrary node, you find all of the other nodes that it communicates with, and you find them by picking them one at a time. You pick all of the nodes for which p sub i j is greater than 0. Then you pick-- and p sub j i is great-- well-- blah.

If you want to find the set of nodes that are accessible from i, you start out looking at i. You look at all the states which are accessible from i in one step. Then you look at all the steps, all of the states, which you can access from any one of those. Those are the states which are accessible in two states-- in two steps, then in three steps, and so forth. So you find all the nodes that are accessible from node i. And then you turn around and do it the other way. And presto, you have all of these classes of states all very simply.

For finite-state change, the state i is transient if there's a j in S such that i goes into j, but j does not go into i. In other words, if I'm a state i, and I can get to you, but you can't get back to me, then I'm transient. Because the way Markov chains work, we keep going from one step to the next step to the next step to the next step. And if I keep returning to myself, then eventually I'm going to go to you. And once I go to you, I'll never get back again. So because of that, these transient states are states where eventually you leave them and you never get back again.

As soon as we start talking about countable state Markov chains, you'll see that this definition doesn't work anymore. You can-- it is very possible to just wander away in a countable state Markov chain, and you never get back again that way. After you wander away too far, the probability of getting back gets smaller and smaller. You keep getting further and further away. The probability of returning gets smaller and smaller, so that you have transience that way also.

But here, the situation is simpler for a finite-state Markov chain. And you can define transience if there's a j in S such that i goes into j, but j doesn't go into i. If i's not transient, then it's recurrent. Usually you define recurrence first and transience later, but it's a little simpler this way.

All states in a class are transient, or all are recurrent, and a finite-state Markov chain contains at least one recurrent class. You did that in your homework. And you were surprised at how complicated it was to do it. I hope that after you wrote down a proof of this, you stopped and thought about what you were actually proving, which intuitively is something very, very simple. It's just looking at all of the transient classes. Starting at one transient class, you find if there's another-- if there's another state you can get to from OK i which is also transient, and then you find if there's another state you get to from there which is also transient. And eventually, you have to come to a state from which you can't go to some other state, from which you can't get back.

That was explaining it almost as badly as the problem statement explained it. And I hope that after you did the problem, even if you can't explain it to someone, you have an understanding of why it's true. It shouldn't be surprising after you do that. So the finite-state Markov chain contains at least one recurrent class.

OK, the period of a state i as the greatest common denominator of n, such that p i n is greater than 0. Again, a very complicated definition for a simple kind of idea. Namely, you start out in a state i. You look at all of the times at which you can get back to state i again. If you find it that set of times has a period in it, namely, if every sequences of states is a multiple of some d, then you know that the state is periodic if d is greater than 1. And what you have to do is to find the largest such number. And that's the period of the state.

All states in the same class have the same period. A recurring class with period d greater than one can be partitioned into sub-class-- this is the best way of looking at periodic classes of states. If you have a periodic class of states, then you can always separate it into d sub-classes. And in such a set of sub-classes, transitions from S1 and the states in S1 only go to S2. Transitions from states in S2 only go to

S3. Up to, transitions from S d only go back to S1. They have to go someplace, so they go back to S1.

So as you cycle around, it takes d steps to cycle from 1 back to 1 again. It takes d steps to cycle from 2 back to 2 again. So you can see the structure of the Markov chain and why, in fact, it does have to be-- why that class has to be periodic.

An ergodic class is a recurrent aperiodic class. In other words, it's a class where the period is equal to 1, which means there really isn't any period. A Markov chain with only one class is ergodic if the class is ergodic. And the big theorem here-- I mean, this is probably the most important theorem about finite-state Markov chains. You have an ergodic, finite-state Markov chain. Then the limit as n goes to infinity of the probability of arriving in state j after n steps, given that you started in state i, is just some function of j.

In other words, when n gets very large, it doesn't depend on how large M is. It stays the same. It becomes independent of n. It doesn't depend on where you started. No matter where you start in a finite-state ergodic Markov chain. After a very long time, the probability of being in a state j is independent of where you started, and it's independent of how long you've been running.

So that's a very strong kind of-- it's a very strong kind of limit theorem. It's very much like the law of large numbers and all of these other things. I'm going to talk a little bit at the end about what that relationship really is. Except what it says is, after a long time, you're in steady state, which is why it's called the steady state theorem. Yes?

AUDIENCE: Could you define the steady states for periodic changes [INAUDIBLE]?

PROFESSOR: I try to avoid doing that because you have steady state probabilities. The steady state probabilities that you have are, you take-- is if you have these sub-classes. Then you wind up with a steady state within each sub-class. If you assign a probability of the probability in the sub-class, divided by d, then you get what is the steady state probability. If you start out in that steady state, then you're in each sub-class with probability 1 over d. And you shift to the next sub-class and you're still in steady state, because you have a probability, 1 over d, of being in each of those sub-classes to start with. You shift and you're still in one of the sub-classes with probability 1 over d. So there still is a steady state in that sense, but there's not a steady state in any nice sense.

So anyway, that's the way it is. But you see, if you understand this theorem for ergodic finite state and Markov chains, and then you understand about periodic change and this set of sub-classes, you can see within each sub-class, if you look at-- if you look at-- if you look at time 0, time d, time 2d, times 3d and 4d, then whatever state you start in, you're going to be in the same class after d steps, the same class after 2d steps. You're going to have a transition matrix over d steps. And this theorem still applies to these sub-classes over periods of d. So the hard part of it is proving this. After you prove this, then you see that the same thing happens over each sub-class after that.

That's a pretty major theorem. It's difficult to prove. A sub-step is to show that for an ergodic M state Markov chain, the probability of being in state j at time n, given that you're in state i at time 0, is positive for all i j, and all n greater than M minus 1 squared plus 1.

It's very surprising that you have to go this many states-- this many steps before you get to the point that all these transition probabilities are positive. You look at this particular kind of Markov chain in the homework, and I hope what you found out from it was that if you start, say, in state two, then at the next time, you have to be in 3. Next time, you have to be in 4, you have to be in 5, you have to be in 6. In other words, the size of the set that you can be in after one step is just 1. One possible state here, one possible state here, one possible state here.

The next step, you're in either 1 or 2, and as you travel around, the size of the set of states you can be in at these different steps, is 2, until you get all the way around again. And then there's a way to get-- when you get to state 6 again, the set of states enlarges. So finally you get up to a set of states, which is up to M minus 1.

And that's why you get the M minus 1 squared here, plus 1.

And this is the only Markov chain there is. You can have as many states going around here as you want to. But you have to have this structure at the end, where there's one special state and one way of circumventing it, which means there's one cycle of size M minus 1, and one cycle of size M. And that's the only way you can get it. And that's the only Markov chain that meets this bound with equality. In all other cases, you get this property much earlier. And often, you get it after just a linear amount of time.

The other part of this major theorem that you reach steady state says, let P be greater than 0. In other words, let all the transition probabilities be positive. And then define some quantity alpha as a minimum of the transition probabilities. And then the theorem says, for all states j and all n greater than or equal to 1, the maximum over the initial states minus the minimum over the initial states of P sub i j to the n plus-- first step, that difference is less than or equal to the difference a the n-th step, times 1 minus 2 alpha. Now 1 minus 2 alpha is as a positive number. And this says that this maximum minus minimum is 1 minus 2 alpha to the n, which says that the limit of the maximizing term is equal to the limit of the minimizing term. And what does that say? It says that everything in the middle gets squeezed together. And it says exactly what we want it to say, that the limit of P sub l j to the n is independent of l, after n gets very large. Because the maximum and the minimum get very close to each other.

We also showed that [? our ?] approaches that limit exponentially. That's what this says. The exponent here is just this alpha, determined in that way. And the theorem for ergodic Markov chains then follows by just looking at successive h steps in the Markov chain when h is large enough so that all these transition probabilities are positive. So you go out far enough that all the transition probabilities are positive. And then you look at repetitions of that, and apply this theorem. And suddenly you have this general theorem, which is what we wanted.

An ergodic unichain is a Markov up chain with one ergodic recurring class, plus perhaps a set of transient states. And most of the things we talk about in this course are for unichains, usually ergodic unichains, because if you have multiple recurrent classes, it just makes a mess. You wind up in this recurrent class, or this recurrent class. And aside from the question of which one you get to, you don't much care about it.

And the theorem here is for an ergodic finite-state unichain. The limit of P sub i j to the n probability of being in state j at time n, given that you're in state i at time 0, is equal to pi sub j. In other words, this limit here exists for all i j. And the limit is independent of i. And it's independent of n as n gets big enough.

And then also, we can choose this so that this set of probabilities here satisfies this, what's called the steady state condition, the sum of pi i times P sub i j is equal to pi j. In other words, if you start out in steady state, and you look at the probabilities of being in the different states at the next time unit, this is the probability of being in state j at time n plus 1, if this is the probability of being in state i at time n. So that condition gets satisfied. That condition is satisfied. You just stay in steady state forever. And pi i has to be positive for a recurrent i, and pi i is equal to 0 otherwise.

So this is just a generalization of the ergodic theorem. And this is not what people refer to as the ergodic theorem, which is a much more general theorem than this. This is the ergodic theorem for the case of finite state Markov chains. You can restate this in matrix form as the limit of the matrix P to the n-th power.

What I didn't mention here and what I probably didn't mention enough in the notes is that P sub i j-- but also, if you take the matrix P times P time P, n times, namely, you take the matrix, P to the n. This says the P sub i j is the i j element.

I'm sure all of you know that by now, because you've been using it all the time. And what this says here-- what we've said before is that every row of this matrix, P to the n, is the same. Every row is equal to pi. P to the n tends to a matrix which is pi 1, pi 2, up to pi sub n. Pi 1, pi 2, up to pi sub n. Pi 1, pi 2, up to pi sub n. And the easiest way to express this is the vector e times pi, where e is transposed.

In other words, if you take a column matrix, column 1, 1, 1, 1, 1, and you multiply this by a row vector, pi 1 times pi sub n, what you get is, for this first row multiplied by this, this gives you-- well, in fact, if you multiply this out, this is what you get. And if you've never gone through the trouble of seeing that this multiplication leads to this, please do it, because it's important to notice that correspondence.

We got specific results by looking at the eigenvalues and eigenvectors of stochastic matrices. And a stochastic matrix is the matrix of a Markov chain. So some of these things are sort of obvious. Lambda is an eigenvalue of P, if and only if P minus lambda i is singular. This set of relationships is not obvious. This is obvious linear algebra. This is something that when you study eigenvalues and eigenvectors in linear algebra, you recognize that this is a summary of a lot of things.

If and only if this determinant is equal to 0, which is true if and only if there's some vector nu for which P times nu equals lambda times nu for nu unequal to 0. And if and only if pi P equals lambda pi for some pi unequal to 0. In other words, if this determinant is equal to 0, it means that the matrix P minus lambda i is singular. If the matrix is singular, there has to be some solution to this equation here. There has to be some solution to this left eigenvector equation.

Now, once you see this, you notice that e is always a right eigenvector of P. Every stochastic matrix in the world has the property that e is a right eigenvector of it. Why is that? Because all of the rows of a stochastic matrix sum to 1. If you start off in state i, the sum of the possible states you can be at in the next step is equal to 1. You have to go somewhere.

So e is always a right eigenvector of P with eigenvalue 1. Since e is also is a right eigenvector of P with eigenvalue 1, we go up here. We look at these if and only if statements. We see, then, P must be singular. And then pi times P equals lambda pi.

So no matter how many recurrent classes we have, no matter what periodicity we have in each of them, there's always a solution to pi times P equals pi. There's always at least one steady state vector.

This determinant has an M-th degree polynomial in lambda. M-th degree polynomials have M roots. They aren't necessarily distinct. The multiplicity of an eigenvalue is the number roots of that value. And the multiplicity of lambda equals 1. How many different roots are there which have lambda equals 1?

Well it turns out to be just the number of recurrent classes that you have. If you have a bunch of recurrent classes, within each recurring class, there's a solution to pi P equals pi, which is non-zero only one that recurrent class. Namely, you take this huge Markov chain and you say, I don't care about any of this except this one recurrent class. If we look at this one recurrent class, and solve for the steady state probability in that one recurrent class, then we get an eigenvector which is non-zero on that class, 0 everywhere else, that has an eigenvalue 1. And for every other recurrent class, we get the same situation.

So the multiplicity of lambda equals 1 is equal to the number of recurrent classes. If you didn't get that proof on the fly, it gets proved in the notes. And if you don't get the proof, just remember that that's the way it is.

For the special case where all M eigenvalues are distinct, the right eigenvectors are linearly independent. You remember that proof we went through that all of the left eigenvectors and all the right eigenvectors are all orthonormal to each other, or you can make them all orthonormal to each other? That says that if the right eigenvectors are linearly independent, you can represent them as the columns of an invertible matrix U. Then P times U is equal to U times lambda.

What does this equations say? You split it up into a bunch of equations. P times U and we look at it as nu 1, nu 2, nu sub [? n ?]. I guess better put the superscripts on it. If I take the matrix U and just view it as M different columns, then what this is saying is that this is equal to-- nu 1, nu 2, nu M, times lambda 1, lambda 2, up to lambda M.

Now you multiply this out, and what do you get? You get nu 1 times lambda 1. You get a nu 2 times lambda 2 for the second column, nu M times lambda M for the last column, and here you get P times nu 1 is equal to a nu 1 times lambda 1, and so forth.

So all this vector equation says is the same thing that these n M individual eigenvector equations say. It's just a more compact way of saying the same thing. And if these eigenvectors span this space, then this set of eigenvectors are linearly independent of each other. And when you look at the set of them, this matrix here has to have an inverse. So you can also express this as P equals this vector-- this matrix of right eigenvectors times the diagonal matrix lambda, times the inverse of this matrix. Matrix U to the minus 1 turns out to have rows equal to the left eigenvectors. That's because these eigenvectors-- that's because the right eigenvectors and the left eigenvectors are orthogonal to each other.

When we then split up this matrix into a sum of M different matrices, each matrix having only one-- and so forth. Then what you get-- here's this-- this nice equation here, which says that if all the eigenvalues are distinct, then you can always represent a stochastic matrix as the sum of lambda i times nu to the i times pi to the i. More importantly, if you take this equation here and look at P to the n, P to the n is U times lambda times U to the minus 1, times U times lambda times U to the minus 1, blah, blah, blah forever. Each U to the minus 1 cancels out with the following U. And you wind up with P to the n equals U times lambda to the n, U to the minus 1. Which says that P to the n is just a sum here. It's the sum of the eigenvalues to the n-th power times these pairs of eigenvectors here.

So this is a general decomposition for P to the n. What we're interested in is what happens as n gets large. If we have a unit chain, we already know what happens as n gets large. We know that as n gets large, we wind up with just 1 times this eigenvector e times this eigenvector pi. Which says that all of the other eigenvalues have to go to 0, which says that the magnitude of these other eigenvalues are less than 1. So they're all going away.

So the facts here are that all eigenvalues lambda have to satisfy the magnitude of lambda is less than or equal to 1. That's what I just argued. For each recurrent class C, there's one lambda equals 1, with a left side and vector equals the steady state on that recurrent class and 0 elsewhere. The right eigenvector nu satisfies the limit as n goes to infinity. So the probability that x sub n is in this recurring class, given that x sub 0 is equal to 0, is equal to the i-th component of that right eigenvector.

In other words, if you have a Markov chain which has several recurrent classes, and you want to find out what the probability is, starting in the transient state, of going to one of those classes, this is what tells you that answer. This says that the probability that you go to a particular recurrent class C, given that you start off in a particular transient state i, is whatever that right eigenvector turns out to be. And you can solve that right eigenvector problem just as an M by M set of linear equations. So you can find the probabilities of going through each transient state just by solving that set of linear equations and finding those eigenvector equations.

For each recurrent periodic class of period d, there are d eigenvalues equally spaced on the unit circle. There are no other eigenvalues with lambda equals 1-- with a magnitude of lambda equals 1. In other words, for each recurrent class, you get one eigenvalue that's equal to 1. If that recurrent class is periodic, you get a bunch of other eigenvalues put around the unit circle. And those are all the eigenvalues there are.

Oh my God. It's-- I thought I was talking quickly. But anyway, if the eigenvectors don't span the space, then P to the n is equal to U times this Jordan reform, U to the minus 1, where J is a Jordan form.

What you saw in the homework when you looked at the-- when you looked at the Markov chain-- OK. This is one recurrent class with this one node in it. These two nodes are both transient. If you look at how long it takes to get from here over to there, those transition probabilities do not correspond to this equation here. Instead, P sub 1 2-- P sub 2 3, the way I've drawn it here. P sub 1 3 is n times this eigenvalue, which is 1/2 in this case. And it doesn't correspond to this, which is why you need a Jordan form.

I said that Jordan forms are excessively ugly. Jordan forms are really very classy and nice ways of dealing with a problem which is very ugly. So don't blame Jordan. Jordan simplified things for us. So that's roughly as far as we went with Markov chains.

Renewal processes, we don't have to review them because you're already immediately familiar with them. I will do one thing next time with renewal classes and Markov chains, which is to explain to you why the expected amount of time to get from one state back to itself is equal to 1 over pi-- 1 over pi sub i. You did that in the homework. And it was an awful way to do it. And there's a nice way to do it. I'll talk about that next time.