PROFESSOR: So today we're going to continue our course on the graph theory. It's going to be a mixture of all kinds of topics. We first start off with Euler tour, and then we get into directed graphs and cover the definitions. And we'll talk about a special type, which are called tournament graphs. And we'll do a whole bunch of proofs, and hopefully you will all contribute to make this work and think about how to do this.

So next week we will continue with graph theory, and we will discuss a very special type. We will use directed graphs in communication networks. And on Thursday, we'll actually use these special types of graphs that we'll talk about in a moment, DAGs.

So let's talk about Euler tour. Euler, a famous mathematician, he asked the question like-- he lived in Konigsberg. And there were seven bridges. He was wondering, can you traverse all the bridges exactly once. So he would start walking and try to cross all those bridges. And apparently this little islands in sort of the river as well, and so on. So how can you do that?

This is actually the birth of graph theory. And this particular problem is named after him. So what is an Euler tour? It's actually defined as a special walk. It's a walk that traverses every edge exactly once.

So it turns out the you can actually characterize these types of graphs. So we are talking here about undirected graph, so continuation of last time. So the edges that we consider have no direction. We'll come back to that in a moment. We will start defining those later. So Euler tour is a walk that traverses every edge exactly once. And at the same time, it's also a tour. So that means that it actually starts and finishes at the same vertex.

Now it turns out that undirected graphs, the graphs that we've been talking about, those that Euler tours can be easily characterized. And that's the theorem that we're going to prove next. The theorem states that actually, if a connected graph has an Euler tour, if and only if, every vertex of the graph has even degree. So that's a very nice and simple and straightforward characterization.

So how can we prove this? So let's have a look. So first of all, we have an if and only if statement. So we need to prove two directions. We need to proof that if I have a connected graph that has an Euler tour, I need to show that every vertex has even degree. And also the reverse-- if I have a graph for which every vertex has even degree, I need to show that it has an Euler tour. So the proof consists of two parts.

So let's first do this implication, where we assure that we have connected graph that has an Euler tour. So assume we have a graph. So we have G with the vertex at V, edge is at E, and it has an Euler tour.

So what does it mean? Well, it's really means that we can walk from some start vertex, V0. We can go all the way around to V1, and some more, and so on, to V, say, k minus 1. And then end vertex is the same as the start vertex. So we have a walk that goes all around the whole graph. So every edge in this walk are exactly the edges that are in the graph. And each edge in the graph is offered exactly once.

So what does it mean? So let's write this down actually. So since every edge is traversed once-- every edge in E is traversed once-- what can we conclude from that? Can we say something about, given such as walk of length k, what can we say about the number of edges, for example?

What can we say about the degree of the vertices? Because that's what you want to show, right? You want to show that every vertex has even degree. Does anyone know how to go ahead here? So we know that every single edge in E is referred ones within this whole walk. So what kind of properties can we derive?

AUDIENCE: [INAUDIBLE]

PROFESSOR: That's true.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Maybe someone else can pick up here. So every vertex that I have here, I will enter it. But I also leave it. So if I see a vertex in here, I can see at least two contributing edges that are coming from this.

AUDIENCE: Then you know that that note has at least degree 2, but it can't be more than 2 because otherwise that would be the endpoint.

PROFESSOR: Ah. But is it true though that-- so for example, this particular note here may repeat itself somewhere else. I only have a condition on the Euler tour that every edge occurs exactly ones. So how could I formally sort of continue to prove?

AUDIENCE: Well, it will intersect itself again, and you'll leave again and it will still have the even number.

PROFESSOR: Yeah. That's true. That's right. Any other additions?

AUDIENCE: You can define the number of edges that it will test by how many you enter it, so the number of edges that it has is twice the number of [INAUDIBLE].

PROFESSOR: Yeah. That's correct. [INAUDIBLE]

AUDIENCE: [INAUDIBLE] traverse once, then every time you enter a node you have to leave it. So you can say that you're never going to leave a node for a node that it already left for from there. And you're never going to enter from a node you already entered from or left to from there. So that's how you can say that you're only going to increment your degree by 2.

PROFESSOR: That's true. So what he's essentially saying is that every edge as I start to count here, in this particular tour, well they're all different. So they all counts towards a degree of the node. So everything that you have been saying is right on. What we can conclude here is that if you look at a vertex, say vertex U-- for example, somewhere over here-- well, it may repeat itself over here. And it has an incoming and outgoing edge, incoming and outgoing edge. And I may have it somewhere else, but say it's just at those two. Well, I know that all the edges are different.

So I can clearly count this now. I can say that the degree of U must be equal to the number of times that U actually appears in this tour. So in the tour from V0 all the way to Vk minus 1. And then I have to multiply times 2.

And what I said, it's exactly what you have been saying, all the edges occur exactly once of the whole graph. So I just count the number of times I see U in here. I come the number of edges that go into it and leave from it. Well, that's twice times the number of times that U actually appears in this tour. So this implication is pretty straightforward because now we actually note that the degree of U is even. So that's great.

So let's talk about the other implication and see whether we can find a trick to make that happen. So what do we need to start off with? Well, now we have to assume that every vertex has even degree. So let's write this down. So say we have the graph. And for this graph we actually issue that the degree of every vertex V is even.

Well, what can we do? Well, this is sort of creative trick, so let me just continue here. So what we have is if you start to consider a walk, W, that touches, say, a lot of vertices in such a way that W is actually the longest walk. So that's often what you do in graph theory. You think about, say, the shortest path or whatever with a certain property, or the longest walk, or whatever with a certain property. So that's what you're doing here.

So let W be the longest walk. And well, we want to prove something about an Euler tour. An Euler tour has a very specific property that all the edges occur only once. So it makes sense to look at walks in which all the edges of which it walks are unique, are occurring only once. So we're interested in the longest walk that traverses no edge more than once. So that's important.

Now, let's think about this a little bit. So the first property that you may want to consider it-- you really want to show that W is going to be an Euler tour. That's what we're going to try to prove here. So the first thing we may want to show is that actually it goes all around. So V0 is equal to Vk. That will be great if we can do this.

So we want to show that this is true. So what would happen if this would not be the case? So suppose these two would not be equal to one another. Well, actually, I'm skipping a little bit ahead here in the proof, I notice. So this will be the conclusion of another observation. So let me start with the first observation.

So first of all, let us consider this end node over here, this end vertex, Vk. Now if there is an edge that is not covered in this walk-- and I can lengthen the walk, right? So if I have, say, another edge, which looks like Vk, an edge from Vk to some U, where this edge is not in the walk. Or the other one is not in the walk. Well, it doesn't matter in an indirect graph, of course.

So if this is not in the walk, then what can I do? I can just lengthen it, right? I can just say V0 goes to V1 goes to Vk goes to U. Now I have a longer walk. And that's a contradiction. So I know that all the edges in which Vk is participating are actually covered by the walk. So that's a great property. So let's write this down. So all edges that are incident to Vk are actually used-- traversed-- in the walk, W.

So now let's have a look whether we can show that it's a walk that goes all the way around. So let's try to prove this statement. So let me repeat this over here. So what we want to show is that Vk equals V0. So given this statement, do you have any idea how we could possibly prove this, also given what we have been doing over here?

Suppose this would not be the case, right? So suppose the start and the end node in this walk are not equal to one another. So what can we do here? So maybe there's some suggestions? I already had you once. Maybe someone else?

AUDIENCE: It has to attach to another vertex. And you know that all the degrees are even. So it means that Vk must attach to V0.

PROFESSOR: So what you're saying, first of all, is that if this would not be the case, then you know that the degrees must be even. But suppose this would not be the case. That would mean actually that the degree would be odd, right? OK. So let's think about that a little bit. So let's write this down.

So otherwise the Vk has odd degree. But the only node has odd degree in this particular walk, W. So that's what we do now. Because we really consider this walk. So we see that if I have V0 unequal to Vk and Vk maybe repeated in this walk a number of times, each time it is repeated it has an incoming and an outgoing edge. And all these edges are different because they do not occur more than once. So whenever Vk enters in this middle part over here-- it's partaking over here-- it gives an even contribution to the number of edges in this walk, and it has one extra incoming edge over here. So if V0 is not equal to Vk, we will have an odd degree of edges in W, in which Vk is participating.

So now we can go ahead and we can have a look over here because we just showed here that all the edges incident to Vk are actually used in W. So we can now conclude that this means that Vk also has odd degree in G because all the edges in G, in which Vk is participating, are actually used in the walk. So I hope that is clear.

So let's write this down. So we'll use the first statement over here, so by 1. We have that Vk has odd degree in G. And now we come to what you were saying. We have assumed over here that the degree is even. So this cannot be the case, so that's a contradiction. So we know that the otherwise situation cannot occur. So we know that Vk must be equal to V0. So that's proving that we actually have a walk that goes all around.

Now, we are not yet done. Because why would this be an Euler tour? For an Euler tour, we really want that every single edge that G has is actually used in the tour. So we need much more. So let's use this board.

So suppose that W is not an Euler tour. We're going to use contradiction. But we already have shown this particular property, so we can use this. So suppose W is not an Euler tour. Well, we know that G is connected. So that's a good thing because that means that if it is not an Euler tour, there is an edge that is not used in W, in the walk, but it is incident to some vertex that is used in W.

So this is where the connectivity here comes in play. Now let us call this edge-- so let U-Vi be this particular edge. So U is not used in the walk-- well, it's maybe used in the walk, but the edge is not used in the walk. And Vi is part of the walk.

So what can we do here? So does anybody see what we could do here? Can we somehow get a contradiction? So when we started out here, we actually assumed that W is the longest walk in which the edges do not occur more than once. Can we create a longer walk by simply using this? Now, notice that we did prove that V0 goes all around to Vk, and so on. So can we find a longer walk that uses this extra edge? Any ideas?

What would happen if we, for example, started a walk with U? Let's see how that would work out. We can have U. We walk to Vi. Well, let's just simply start walking all around. So essentially we have Vi over here and U over here. So we could do is we can start walking like this and then end over here. And notice we have used one extra edge.

So we get a longer walk. So we go to Vi plus 1, and we go all the way up to Vk, which is equal to V0. And then we go up to V1, all the way up to Vi. This is a longer walk. And therefore we have a contradiction. And now the proof is finished because that means that this assumption was wrong. So W actually is an Euler tour. So now we have shown the existence of an Euler tour, and that was the other direction of the theorem.

So let's now start with a new topic on directed graphs. And later on, we will talk about how Hamiltonian paths. And these are different from an Euler tour in that in an Euler tour, every edge is used exactly once. In a Hamiltonian path, we will have that every vertex is used exactly once. So we'll do that a bit later.

So what are directed graphs? Directed graphs are graphs where we have edges that have a specific direction. So we can walk from one vertex to the other. Let me just depict an example.

For example, we could have something that looks like this. We have V1. We have V2, V3. We may have an edge that goes like this with an arrow. We use arrows in this case. I can go to V2 and V3. From V2 I may be able to go to V3, and from V3 to V2.

So this will be a directed graph, as an example. We also call these digraphs. And as you can see, we may have an edge that goes from V2 to V3 and also one that goes from V3 to V2. So these are two separate edges. So every edge has a direction. And we usually say that if we have an edge that points from V2 to, say, V3, then we call this the tail and over here we have the head. So this is some notation that you may use.

We also have now a different notion for the degree of a vertex because we have different types of edges, essentially. Take, for example, V2. You have incoming edges and outgoing edges. So that's why we're going to talk about an indegree and an outdegree. So, for example, the indegree of V2 is equal to, well, I have one, two incoming edges. And the outdegree of V2 is actually something different. It's just one outgoing edge. So this is equal to 1. So this is some notation.

So let's talk about walks. Let's figure out where we can enumerate walks in a directed graph and compute those. So how many walks do I have, say, of length k that go exactly from, say, V1 to V3. How can I compute this?

So I'd like to introduce adjacency matrices. You've probably seen them before. But let's go over this once more because induction is also important. So let's do the theorem. So the theorem is this. Suppose we have a graph. And suppose it has end nodes. And we have the vertices V1 up to Vn. And now we let the matrix that contains the entries aij denote the adjacency matrix for G.

And what does it mean? It's actually means that in this case, we say that aij is equal to 1 if we actually have an edge that goes from Vi to Vj-- so this is an edge-- and 0 if this is not the case. So this is the adjacency matrix.

And now we can state something about the number of directed walks in a directed graph. So it turns out that this is easily computed by taking powers of the adjacency matrix. Let aij with the superscript k be equal to the number of directed walks of length k that go from Vi to Vj. So you want to compute this and it turns out that-- wait a minute. So what did I do here? So actually let p kij be the number of directed walks of length k from phi i to phi j.

Then what you can show is if you look at matrix A to the power k, this actually is the matrix that contains all these numbers. So let me give a few examples. So let's take the matrix over here. Look at this ajc matrix, take a few powers and see what happens. So the matrix looks like this. If you just label the rows by, say, V1, V2, and V3. The columns by V1, V2, and V3. Well, V1 has an edge to V1. V1 has an edge to V2 and also one to V3. V2 has only one edge to V3, so we have zeros here. And we have this. So this is the matrix A.

For example, if you compute a squared. So how do we do this? How many of you know about matrix multiplication? Everybody knows? Well anyway, so let's assume that you do know actually. So that's pretty simple. So you take the column. You take the inners product with the first row, and so on. And you can easily compute this otherwise you may want to do it yourself later on. And you get this particular matrix. And, for example, a to the power of 3 is something very similar-- 1, 3, 3, 0, 0, 1, 0, 1, 0.

Well, it turns out that if you look at this graph and we, for example, want to know the number of walks of length 3 that go from V1 to, say, V2, let's see whether we can see those over here. Well, if I travel from V1 to V2 in three steps, I can go like this. I can go one, two, three. So that's one option. I can go one, and another one time, so it's a second step, and three. I can also go directly over here and go back here, and back over here.

So it turns out that's I can compute this. So how do we prove this? We'll use induction. And the steps are pretty straightforward.

So let's first define, because we want to prove that kth power of a is equal to that matrix up there, where all the entries represent the number of walks. So let's say that aijk, that this denote the i jth entry in a to the power k. And you want to show that this number is actually equal to pijk for all the entries.

So if you're going to use induction, it makes sense to assume that the theorem is true for k. So the induction hypothesis could be something like this. The theorem is true for k. And this is really the same as stating that all these entries for all i and j-- aij is equal to the pijk. So this is what you want to prove. And this is pretty straightforward because now what we can do is we can start to look at how we compute walks of length k, or k plus 1.

So let's first do the base case. That's how we always start. The base case is k equals 1. And we have essentially two options. So we want to prove this, so take an i and a j, whatever you want. So suppose we has an edge, Vi, that goes to Vj. Well in that case, we know that the number of walks of length 1, from i to j, is exactly 1 because there's this edge. So this is equal to 1. But this is also the definition of my adjacency matrix. So I can just write out here aij 1. So that's great. This case definitely works.

Now, if there is no edge of this type, then you know that in one step, we can never achieve Vj from Vi because there's no edge. So we know this is equal to 0. There's no such walk. And this is, by definition of the adjacency matrix, also equal to the aij. So this works. So the base case it's easy.

And induction step always starts by issuing pk. As you can see, these types of proofs always have the same structure. In this case, let me again assume pk. We want to prove pk plus 1. So what you want to know is what's the number of walks of length k plus 1. So how can we express those? pij k plus 1. How can we use this assumption over here? Do you have an idea so we can--

AUDIENCE: Any walk of length k plus 1 can be got by taking one of-- let's say from Vs to Vx. [INAUDIBLE] You go from Vs to V in k steps and then V to Vf in one step.

PROFESSOR: That's true. We could do that. So let's write it down. So what you're essentially saying is that you can enumerate all the walks by going of length k plus 1 by first going from i in, say, k steps, to whatever V, and then in one step to, say, Vi to V in k steps and then in one step to Vj. So let's write this down because V can be anything. So what to do is we have a sum over all the Vs such that-- well, we will use indices here let me do that a little bit differently. So let's say I have an h over here. So all the indices h such that Vh to Vj is actually an edge in G.

So then we can write out here that we go in k steps to Vh. So how many walks are there? Well, we can use the induction hypothesis now, right? So we can say you go from i to h in k steps. And then, well, we can use this particular edge to complete it to a k plus 1th walk from i to j.

So now what is this equal to? Can we simplify this sum? We can, right, because we know that there's an edge if and only if the adjacency matrix has a 1 in a particular position. So we could rewrite it and sum over all h from 1 to n. And write pij k times, and then aj-- oh, this should be an h. So we have the same over here. And then we have one edge from h to 2j.

So I only count this number over here if this is equal to 1, and that happens exactly if there's an edge. I do not count this if there's a 0 over here, that is as if there's no edge. But now we can use induction hypothesis because I know that these numbers are equal to the a's. So we rewrite this. And we see that we get aih k. So this is where we use the induction step. So it's like we assume pk over here. And I need to finish this formula. So we have this.

Now, by the definition of matrix multiplication, we actually see that this is equal to a k plus 1 ij to the ij-th entry in the k plus oneth power of the adjacency matrix. Here we have this represents the matrix of the kth power. This represents the matrix of a. So we multiply essentially a to the power k times a and get a to the power k plus 1. And that's what we see here. So this is the induction step, and so this proves the theorem up here.

So let's talk about a few more definitions concerning directed graphs. And then you go a step into a very special type of graph, which are the tournament graphs up here. One of the things that we have in undirected graphs, so where we have no directed edges, we talked last time a lot about cyclicity and stuff like that. And we talked about acyclic connected graphs. And as we categorized those, so we defined them as trees, and they have a very special structure.

So what would happen here, if we look at a directed graph, and we wonder, what does it mean to be connected. Can we really talk about that? What does that mean? For example, if I look at this particular example graph over here, I can say, well, V1 has an edge that points towards V1 itself, and towards V2, and towards V3. So maybe I would call this graph connected.

But if I look at V2, I only see an edge that goes from V2 to V3, and not to V2. So maybe I do not call it connected. So that's why we define a stronger notion for a digraph. So a digraph, G, VE is called strongly connected if we know that for every pair-- so if for all vertices U and V in the vertex set-- there exists a directed path that starts in U and ends up in V in G. So this is what we would call strongly connected.

So now in undirected graphs, we had connectivity and then we said, well, if a connected, undirected graph has no cycles. We have trees and they have special properties and all that. So what about this over here? Suppose you have an acyclic strongly connected digraph. What does that look like? So let's give an example. It's not completely clear what kind of structure it has. So for example, I may have a graph that looks like this, for example. Say this is an acyclic graph. but It does not have at all a tree structure.

So actually, the type of graph the we have here is called a directed acyclic graph. As you can see, there are no cycles because I only go forward, essentially. I can never go backward in this particular way that I depicted the graph. So this is an example for directed acyclic graph. But it doesn't look like a tree at all.

So it's worth to define it separately, and we will use this on Thursday when we'll talk about partial orderings. And it turns out that, as you can see here-- well, at that case, a directed acyclic graph has really nice properties, and one of them is that you can order these vertices in such a way that you go from, say, left to right in a directed fashion. And that will lead to partial ordering. So that's something that we will talk about as well.

So what's the definition? A directed graph is called a directed acyclic graph, and we appreciate this by DAG. We call them DAGs. They're used everywhere, actually. If it does not contain any directed cycles. So these kinds of graphs are used in scheduling and optimization, and we will use them next week in the lecture on partial orderings. Now we have done a lot of definitions concerning directed graphs.

So now let's talk about these very special ones, tournament graph, and see whether we can prove a few really nice theorems about them. Let's see whether we can figure that out together. So what is a tournament graph?

In a tournament graph, I have a bunch of vertices. And essentially we want to represent like a tournament. So every vertex, say, represents a team. And a team can play against another team, and beat them or lose against the other team. So we want to use the directed edges to indicate who is winning from home. And such types of graphs have very special property.

So let me first depict one. So for example, we have E goes to A, goes through B, incoming edge from C, one coming from B. And over here we have this directed edge. We have this one. We have this one. We have this one. Let me see [INAUDIBLE] don't make any mistakes here. And this one. And over here is another one.

So what do we see in this graph? We see that either say, team U, beats team V. And that means that we have a directed edge from U pointing at V. Or it's other way around. V actually beats U, and we have a directed edge from V to U.

Let's have a look at this graph and see how this works. So for example, we have that B is beating E, and E is beating A, and so on. So let's have a look. Maybe we can figure out who's the best player in here.

So this is sort of a general question if who would like to answer. Maybe you cannot answer this. So let's have an example. Has anybody seen an example where we start with A, then we may beat another vertex, and maybe another vertex, and so on, until we have covered all the different vertices.

Do you see a path that works like that? And that could gives us an ordering on who is the best player, like the one at the top, like A, is able to, for example, beat B. And B is, for example, able to beat D and this one, E, and this one, C. So you would say, OK, that's great. Now I know that this one is the strongest player.

But there's a little problem here, right? Because I can produce many such paths. And actually, if I look at C, then C beats A as well. So that's kind of weird. So wait a minute. We have that there's a directed edge from C to A. It's like teams beats one another. And it's not very clear how we can talk about a best player.

Well, we would have a best player if one player sort of wins from everybody else. But there's many examples here. So let's look at another walk. For example, C can go to B, to D, and then to E, and then to A. So there are many possibilities here.

So this leads us to a concept. And we call that's a directed Hamiltonian path. And we're going to show that, in a tournament graph, you can always find such a directed Hamiltonian path.

So what's a Hamiltonian path? This is actually an example of it. There's a walk that goes around the graph and visits every vertex exactly once. So we're going to prove that a tournament graph has this beautiful property.

So let's first write out a definition of this. A directed Hamiltonian path is a directed walk that visits every vertex exactly once. So as I said already, here we have such an example. We can go from A to B, to D to E, to C. Maybe there are even other examples. I did not actually see them. Maybe you can have a look at them as well. So maybe there's something that starts with B going to E maybe. That's a very different direction, like this. There will be one as well, and so forth.

So what's the theorem that you want to prove? The theorem in that you want to show that every tournament graph actually contains such a directed Hamiltonian path. So let's have a look at how we can prove this.

What kind of proof technique are we going to use? Do you have an idea? Well, usually we use induction so that's what we're going to do here as well. But what kind of induction hypothesis can we do? So what would we induct on, you think? Someone else? Maybe someone up there?

AUDIENCE: The number of nodes.

PROFESSOR: The number of nodes. So we use induction on the number of nodes. And why would that be of interest? So let's have a look at how we can think of that. So we start thinking about such a problem, this really sort of-- one parameter here that's the number of nodes in a tournament graph. I also have edges. But if I think about edges, then, the edges is always directly related to the number of nodes in a tournament graph. So I really have just that one parameter.

So it makes sense to use induction on number of nodes. So induct on n, where Pn is going to be that every-- and essentially, the theorem holds true for a tournament graph on n nodes-- so every tournament graph on n nodes actually contains a directed Hamiltonian path.

So this is semi induction hypothesis. So when you think about that, well, I feel pretty confident because if I look at the base case-- that's how we always start-- well, n equals 1. If n equals 1, I have just a single vertex. There's no edges. Everything is fine because the single edge is a directed Hamiltonian path.

So this is great. So this works. So what about inductive step? Now with inductive step, we always perceived in the same way. We start to assume that Pn is true.

Essentially, the theorem holds for a tournament graph on n vertices. Actually, let me keep this over here. So now let's just think about how we can prove this induction step. I need to prove P of n plus 1. So how do I start? I have to start with a tournament graph on n plus 1 vertices. And somehow I got to be able to use this property because that's what I assume. And the property only holds for a tournament graph on end points. So what will be a really good strategy to do here to sort of proceed our proof?

So let me first write out what we want to do here. Maybe you can think about how to advance here. So we have shown Pn. Now we want to actually prove something about tournament graphs on n plus 1 nodes, so let's consider one.

Consider a tournament graph on n plus 1 nodes. Now how can I use my induction hypothesis? So I start with this, and I want to use something that talks about the tournament graph on n nodes. So how could I proceed here? Any suggestions? So what do you usually do, right? If I have like an n plus 1 nodes, I have to somehow look at least maybe there exists a subgraph in this bigger graph. So this is really how you always think about these types of proofs or also other problems.

So there must be some kind of subgraph that already has this property. Well, let's take out one node and see what happens because then we have one node less and maybe we will be able to apply our induction step. So let's take out one node V.

And what can we say about remaining graph if we take out one node? For example, if I take out the node E over here and I just look at all the rest, I can still see that for all the other nodes either, say, U beats V or V beats U. So I still have an edge in one direction between each two nodes.

So actually I still have a tournament graph, so that's great. So this gives a tournament graph. So essentially, so far, we really haven't done anything creative or anything that we had to make a big leap in order to prove this theorem.

We started out with, if you want to prove something like this you have to really look at the number of vertices. And then we start to write down this stuff over here that makes total sense. And then we are going to figure out where we can use this induction step. And so we just take out one node. And yes, it is a tournament graph on n nodes. So this is very systematic. That's what I try to get at here.

So by the induction step because by the induction hypothesis, we know now that we actually have a directed Hamiltonian path. So let V1 to V2 to Vn be such a path. So now that we can use this, we apply it and we get a path. Now what do we want to do? We want to show that we can create a new path, which is also a directed Hamiltonian path, but now one that also includes V in the bigger graph. If we can do that, we are done. So now we have to start really looking at how we can make that happen.

In order to do this, we have to see how we can somehow plug V, the vertex that we have removed, in this path over here. If you can do this, we are in really good shape. So far, we haven't used at all-- so there's also something that you can look at if you start solving these types of problems-- we haven't used at all the property that the tournament graph has, which I just wiped out. So let's figure out what we can do here. We should be able to use something.

So of course, we have a few simple cases. For example, if V has a directed edge into V1, and I have a Hamiltonian path that goes from V to V1 to V2 all the way to Vn, and then I cover all the vertices exactly once. And I will have a direct Hamiltonian path. So this is great. So this is definitely easy. So this is case one.

In case two, suppose that V1 has a directed edge to V. And So now we have a little problem because somehow there is an edge like this to V, but they cannot go like this. This is not a Hamiltonian path. I need to have a directed walk that covers all the vertices exactly once. So now we have a little problem.

So now we have to start really thinking about how we can solve this. So are there any suggestions to make this happen? So let's think a little bit about this. So somehow if I start thinking about this, I would like to plug V somewhere in this sequence. That will be like a pretty obvious way to-- go ahead.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah. That's true. So for example, I may have that for example, V2 beats V as well. But suppose that I have V3 over here, and this one beats V3. Then, as you say, I could sort of plug V in here in this sequence and I may have a longer sequence. Or if this is not the case, then maybe the next one, V4, may have the property and I can plug V in here. So essentially what I want to show is that I can plug V somewhere in the middle of two of these Vi's.

And the property that you were using is actually that you said, well, I know that V2 either beats V, or V beats V2. So that's the property of the tournament graph that we're going to use here in order to prove this. So yes, that's a great observation.

So the way you formulated it, we can maybe use induction, for example, to prove this. We can sort of go recursively through this until you find the right spot. Maybe they can immediately precisely indicate such a spot. So how do we usually do that? If we are thinking about other theorems that we have tried to prove.

AUDIENCE: What's the smallest value of i such that V beats Vi?

PROFESSOR: OK. So what's the smallest value of i where V beats Vi. So usually we have words in our mind like largest or smallest, et cetera, and sort of an extreme precision. And then we like to find out that we can make it happen. And then we say, well, if something goes wrong, we violate that smallest condition. So let's do that here. So let's indicate a specific spot.

Let's considered the smallest i such that V beats Vi. Well, let's have a look how this would work. So this is a little bit of a different proof than what I had but this should work fine as well. So let's just see how it works.

So we have V1. First of all, we notice that, of course, if i equals 1, which that cannot be the case, so we know that i is larger than 1. So there's really somewhere a Vi minus 1-- we have to check that, right? If that exists, this index is not equal to zero-- that goes to Vi and then goes all the way up to Vn.

So now we say we can use this, that V actually beats Vi. Now what's about-- maybe someone else can help me here-- I would like to have that Vi minus 1 beats V. That would be fantastic because then I have a path that goes from V1 all the way up here, goes here, goes there, and all the way up to here. So we have a directed Hamiltonian path that covers all the vertices exactly once, and that's what you want to prove. But why would there be an edge that goes this way? So how do we reason about this? Someone else?

AUDIENCE: Because Vi is the smallest number that V beats, then anything smaller than i must have beat V.

PROFESSOR: Exactly. So if V would beat Vi minus 1, that would contradict the smallest property over here. So that's a contradiction. Now we use a property of the tournament graph. So now I know that Vi minus 1 must beat V. So I really have an edge over here. So that works. So that's the end of the proof.

Now another version could have been in which we would do something similar like this. But we could also have used, say, the largest i-- just something that you may want to look at. You can also use the largest i such that Vi beats V. We have a completely symmetrical argument here, but you could use this solution as well. So I'm just trying to sketch here the way of thinking that you may want to consider in these types of problems.

So why would this work, by the way? Well, we have the same kind of argument like this. We plug V right after Vi. We know there's an edge from V to Vi plus 1. Why? If it's not the case, there will be a large index, i, that contradicts our assumption that we have the largest i already. It's a tournament graph, so we know that there's an edge from V to Vi plus 1. And we get a directed Hamiltonian path as well. So you may want to look at that as well.

So this is about tournament graphs. So let's talk about an interesting tournament graph with a funny game. And this is actually a chicken tournament, like the chickens here represent the vertices and they are pecking one another, but in a certain rule that defines a chicken to be the king chicken. So let's see how that works. So that's a great application of graph theory.

So what do we have? We have that either a chicken, U, pecks a chicken, V. And we said that U has a direct edge to V, so we're actually defining a tournament graph here. Or we have a chicken, V, that pecks a chicken, U, and we get V has a direct edge to U. So we have a tournament graph.

But now we define something new. We say that U virtually pecks V if one of the two conditions holds-- one of these two conditions-- either U, of course, pecks V. That's great. He's in good shape. Or there exists another chicken, W, such that U actually pecks W and W, in turn, pecks V. So this is very special kind of first relationship. So we are wondering now can we now define something-- is there a question?

AUDIENCE: [INAUDIBLE] in between? To be virtually pecked, is one chicken in between U and the other one?

PROFESSOR: Well, there can be multiple chickens in between here. I have several friends who help me out pecking someone else. So when we were looking at these tournament graphs, we were wondering, can we really indicate a winning player? Well in this case, if you start to talk about virtual pecking, we look at the pecking order. Then we can actually define something like a chicken king. So let me write that down. Let me first explain what I mean here. And I give an example of a graph.

So a chicken that is able to virtually peck everyone else, we will call a king. So chicken that virtual pecks every other chicken is called a king chicken. So let's give an example of a graph.

So, for example, suppose I have four chickens that know how to pick one another in this order. So who in the pecking party here is going to be king? Do you see some solutions here? So take, for example, this one. So this one pecks this one. Because we talk about virtually pecking, it can also peck this one over here. It does, right? It pecks this one and this one helps out, and can peck both this one and this one. That's cool. So this one is king.

And this one, actually-- let's have a look. It pecks this one this one, and it virtually also pecked this one. Yay. He has a friend over here that is doing that for him. So this one is actually also a king. So you can have multiple king chickens in here. What about this one? The same story-- pecks this one, pecks this one, and virtually-- wait a minute. The one on the left-- oh yeah, over here. This, this. So this one is king as well.

Now what about this one? Well, it can peck this one. And then in one more step from here, because there's only one outgoing edge, it can virtually peck this one, but not this one. So this one is definitely the loser of the four. So he's not the king.

So now what we want to prove this a theorem sort of trans-identify one of the chickens that we know for sure is going to be king. So you can have multiple kings. But maybe there's one chicken that, from our intuition, we may feel is definitely going to be king. So what will be a good intuition? So we're talking here about virtual pecking.

We have this definition. So what kind of node in a tournament graph essentially would be-- can we know for sure that it's a king? Do you have an intuition for a theorem that you may want to prove? So that's often what we do in mathematics. We have some kind of funny new structure, and then we want to find out whether we can prove interesting properties about it. So we start to search for actual nice properties and theorems.

So in this case, it makes sense that the vertex that has the most outgoing edges may be always king. Can we prove this? So that's what we're going to do. So that's the theorem. And let's see whether we can do this in an elegant way.

So the theorem is that even though there are multiple kings-- as indicated in that particular example, that may happen-- but I certainly know that the chicken that has the highest outdegree is definitely a king. And the way we're going to prove this is by contradiction. Let's assume that's not the case. That must be really, really weird.

If you are the one who has the largest outdegree, it means that you can directly, just by yourself, peck the most others. So suppose you're not king. So by contradiction, first of all, let U have the highest outdegree. And we want to show that this U is king. So let's assume the contrary. So let's suppose that U is not king. So what does that mean? So let's look at a definition over there and see what it means that U is not king.

So that means that both those conditions are violated because if one of those two holds, I know there must be one vertex, V, such that U does not virtually peck V. So I know that. So let's see what that implies. So I know that there must be a V such that U does not virtually peck

V. So maybe can help me out. What does that mean? It means that both these conditions are not true.

So let's look at the first. So U pecks V. If that's not true, and we're in the tournament graph, we know that V must peck U. So we have this. and we also know that the other condition, the second one over here, does not hold.

So what's the negation of this second condition? Maybe you can help me out. So we have here, there exists a W such that U pecks W and W pecks V. So how do we negate that logical expression?

AUDIENCE: [INAUDIBLE]

PROFESSOR: For all W, what's over there is not through true. So can we formulate that a little bit better? So it's not true that U pecks W and W pecks V. So that means that either U pecks W is not true, or W pecks V is not true. So let's write that down. Let's just write it all out. So not U pecks W or not W pecks V. Well, how can we rewrite this? Well, it's a tournament graph, so we know that W pecks U. Or this particular condition, which is V pecks W.

So what do I have here? Is this going in the right direction? Well, I want to prove something about-- if I use contradiction and I suppose that U is not the king, I've assumed that U has the highest outdegree. So I want to show that I somehow violated. So somehow I'm able to construct some vertex, V. And by negating that U is not a king it seems that this vertex, V, makes a really good candidate to show that there's a higher degree outdegree than U. So let's see whether we can do this.

We can rewrite this logical expression once more. We can also say that, well, if U pecks W-- so this is not true-- then it must be true that this one holds because if that's not the case, then this condition is not true. So if U pecks W, then this is not true. So then it must be the case that that is true. So V pecks W. So now let's have a look at V. We noticed that for all outgoing edges from U, there exists a similar outgoing edge for V. But V has one more outgoing edge. V, actually, is an outgoing edge to U.

So what do we see here? That the outdegree of V is actually at least the outdegree of U-- which is this particular condition over here that we show here, for all the W we have that this is true-- plus and we have an extra one. It's this one. Oh, but now we have a contradiction because we said that U has the highest degree. But it turns out that you have constructed one that has a higher outdegree. So that's a contradiction. That means that our original assumption is actually wrong. So suppose that U is not the king was a wrong assumption, and U must be king. So that's the end of this proof.

This is the end of this lecture. So see you tomorrow at recitation and next week we will continue with communication graphs and partial orderings.