Applied Linear Algebra

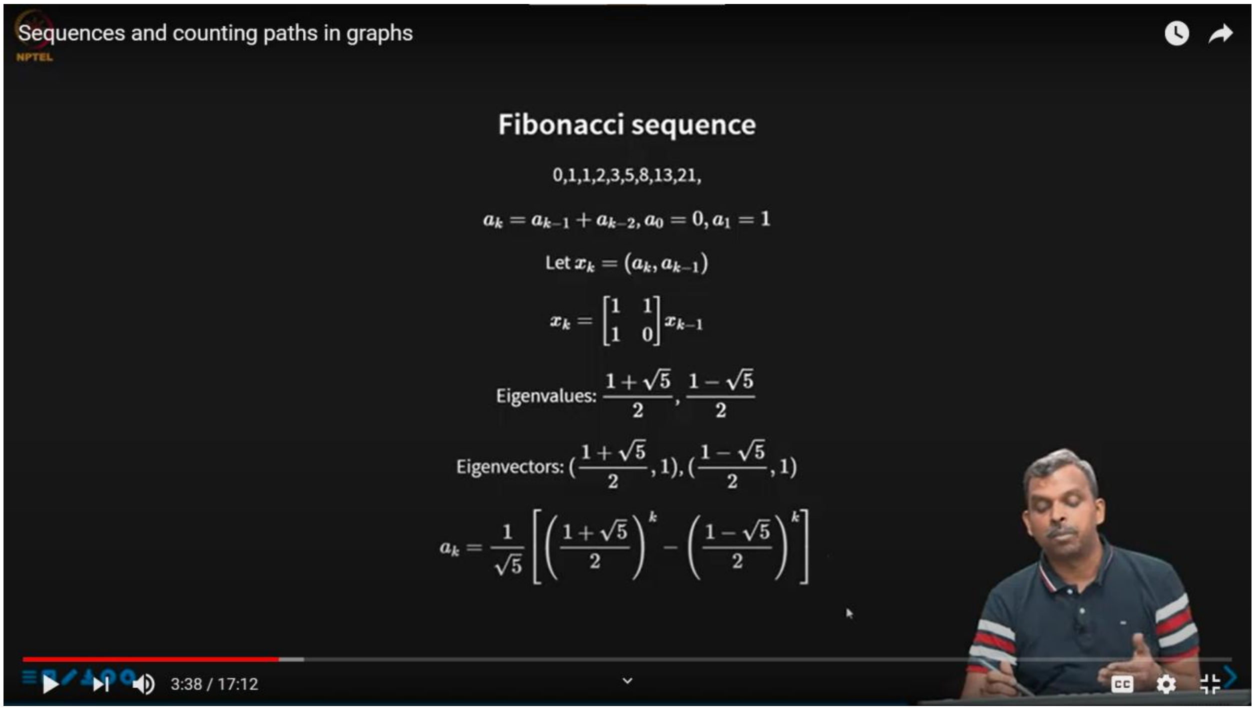
Prof. Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras

Week 06
Sequences and counting paths in graphs

Hello and welcome to this lecture on applications of eigenvalues and all that. So we are going to continue and look at more, a different type of application. So it is about some sort of counting, combinatorial. We'll use something called graphs and sequences and do some counting and things like that. Everything is tied to this evolution in discrete time, okay? So one step to the other if you can have a state, then multiply by a matrix to get the next state, you know what happens, right? Eigenvalues enter the picture. So it turns out in completely different looking applications also. Essentially what happens is this sort of recursion right? There is a state and the next state becomes a linear recursion, right? So linear relationship from this state gives you the next state. So this is used in so many other places and in something unlikely also. I mean, you may not have thought about it that way, but you can use eigenvalues for various different counting problems in the discrete setting, based on that, okay? So we're going to see a few quick applications in this short lecture on this topic, okay?

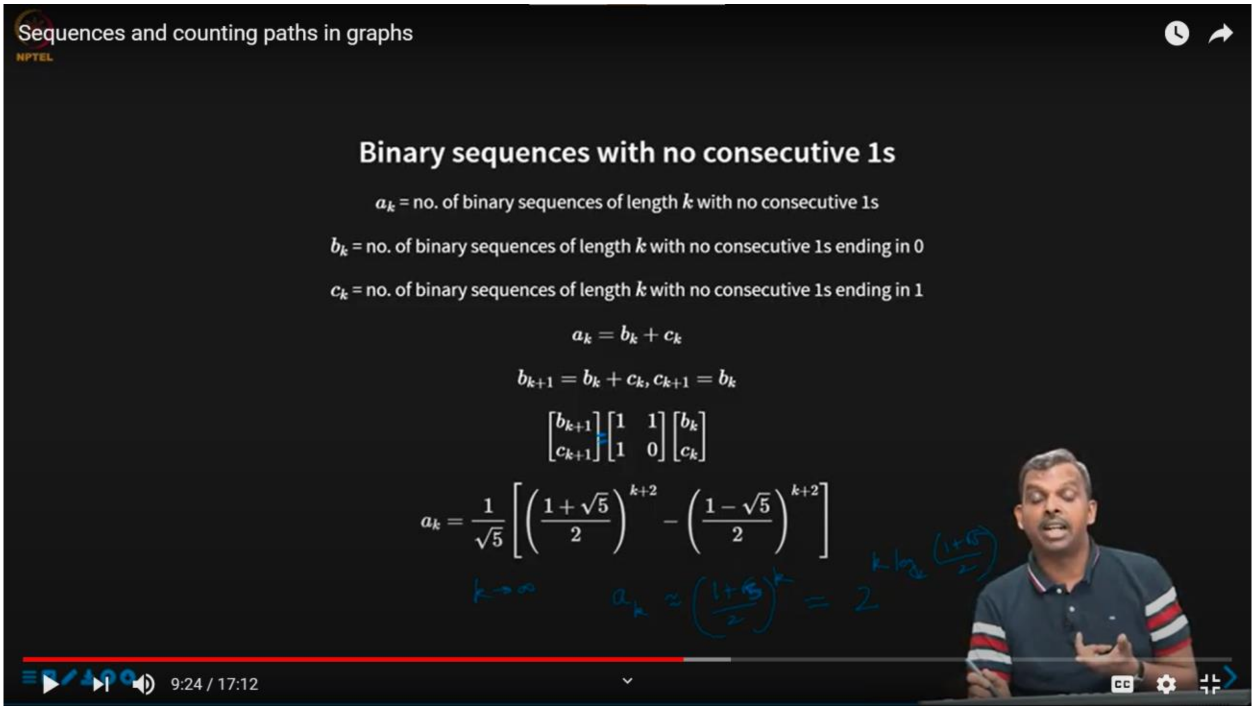
The recap is largely similar to before. So let's proceed. We'll start with one of the simplest applications out there, okay? So this fibonacci sequence, okay? Of course it continues, okay? It's common and well known, okay? So what happens here? You have this sequence. It starts with 0, 1. And the next value in the sequence is the sum of the previous two values, right? You have $a_k = a_{k-1} + a_{k-2}$. Hopefully I have done the addition correctly, okay? So you can keep proceeding like this, you will get the fibonacci sequence, okay? $a_k = a_{k-1} + a_{k-2}$, okay? So maybe you thought, you know, there is no eigenvalue here, but notice there is a recursion and the recursion is linear, okay? Anytime you have some recursive, you know, step like this and it's linear, you can imagine eigenvalues will enter the picture, okay? So when they do enter, you can describe this in eigenvalue, using eigen values also. So for that I need this notion of state. So I will define my state as $(a_k\ a_{k-1})$. The state at time k is $(a_k\ a_{k-1})$. So you can see why I need the a_{k-1} also, right? Because only when I have two things, I can find the evolution properly. So x_k you can write is simply this $[1\ 1;\ 1\ 0]$ times x_{k-1}, okay? So it's a very trivial thing to establish that this equation is true. So I have my linear, you know, single step time evolution for this sequence. And now I can use my eigenvalues, okay? So it's a very simple sort of problem here. But still, you know, eigenvalues enter the picture quite naturally. So you can find the eigenvalues for this matrix, you will get $(1 + \sqrt{5})/2$ and $(1 - \sqrt{5})/2$. You can find the eigenvectors $((1 + \sqrt{5})/2, 1), ((1 - \sqrt{5})/2, 1)$. Then you can express, you know, the initial state. What is the initial state? $(1\ 0)$, right? $(1\ 0)$ is the initial state. You can express the initial state in terms of the eigenvectors and then you will know how the evolution will happen, okay? This is what you do, right? So once you do that, you are done, isn't it? And if you do that you will get a formula for a_k which is this very famous formula for the fibonacci sequence, okay? So this is a very simple application but you again see that how anytime you have a linear equation, linear recursion, eigenvalues and eigenvectors naturally enter the picture, okay? So let us push to other types of applications of this form where maybe you do not think naturally about eigenvectors, eigenvalues but they will enter the picture, okay?

(Refer Slide Time: 03:38)



Here is an example. Let us say I am interested in binary sequences. What are binary sequences? Binary means 0 or 1, okay? Every value is 0 or 1. And then I have a sequence of such values, okay? I am interested in binary sequences that do not have consecutive ones, okay? So if you have binary sequences of length 100 say, okay, how many total binary sequences are there of length 100, 2^{100}, all possible hundred, all possible sequences are there, 2^{100}. Out of this 2^{100}, how many do not have consecutive ones? Consecutive ones are not allowed. Every one should be followed by a zero, okay? How do you count that? How do you, how do you count that, okay? It seems a bit unnatural and what is the connection to eigenvalues, okay? So it turns out there is a connection. Once again this linear recursion type thing is very important, okay? So let us say a_k is the number of binary sequences of length k with no consecutive ones. I will do this little bit of trickery here. I will define two other numbers. One is b_k and another is c_k. What is b_k? Notice what I've done here. I'm introducing some sort of a state, okay? So you'll see how that works, number of binary sequences of length k with no consecutive ones, that's okay. But ending in zero, okay? The last bit is zero. So you can see that is important, no? You can imagine when it grows, what the last bit is will be important in the sequence. So number of sequences that I have which end in zero I am going to call b_k. I am going to call c_k as the number of binary sequences of length k with no consecutive ones but ending in one, okay? Once I have b_k and c_k, I can write $a_k = b_k + c_k$. That's easy enough, right? So just total. That's easy enough. But I can also write evolution for b_k and c_k, okay? $b_{k+1} = b_k + c_k$. Why is that? I can take every sequence of length k ending in 0, okay, and then add a zero to it, I will get another sequence which ends in zero, okay? Which is again valid. No consecutive ones. I can also take a sequence which ends in one and add a zero to it, I will get another length $(k + 1)$ sequence which will end in 0, okay? So $b_{k+1} = b_k + c_k$, okay? But what about c_{k+1}? c_{k+1}, if it has to end in 1, I cannot take anything from c_k, right? Because if it's already ending in 1, I cannot add one more 1 to it because that would violate the no consecutive ones principle. I have to take only previous sequences which end in zero and then add a 1 to it. So c_{k+1} becomes equal to b_k, okay?

(Refer Slide Time: 09:27)

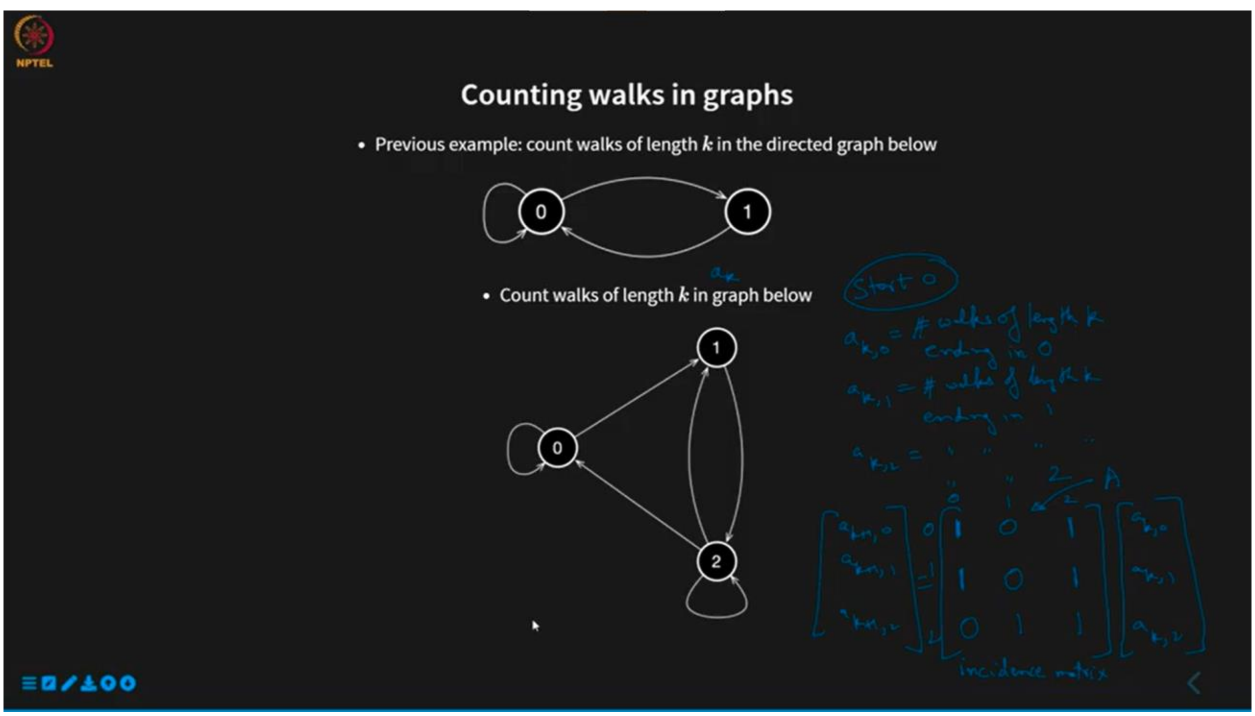


So I forgot the equal to sign here, apologies for that. There is an equal to, sorry about that. Hopefully I will fix it in the slides. So $(b_{k+1}\ c_{k+1})$ is equal to, look at the same matrix showing up here [1 1; 1 0] times $(b_k\ c_k)$, okay? So this is a nice relationship. I have a state evolution and once I have a state evolution, once I know how to start off, you know, I mean for $k = 1$ and all it's easy to find out $(b_1\ c_1)$, you know where it starts, okay? Then I can do the full evolution and to find a_k I simply have to add up these two things, okay? So I'll skip the details here and you can quickly show that this is true also for a_k, okay? So let me draw this equal to here, okay? So you have this simple relationship for a_k, okay? So this is nice, isn't it? So to count the number of, you know, binary sequences of length k, you have a very simple formula. In fact, you know, if you look at what might happen here, you know? So $((\sqrt{5} - 1)/2$, okay? So this if you look at the absolute value of this guy, $((\sqrt{5} - 1)/2$ is actually less than 1, okay? So when k increases, the second term will go off to 0. $((\sqrt{5} + 1)/2$ is actually greater than 1, so only this term will survive. So as k blows up, a_k becomes something like this, okay? So one can write based on this as $k \rightarrow \infty$, $a_k \approx ((1 + (\sqrt{5})/2)$^$k$, okay? So of course there is all constants and all that, doesn't matter. And if you want to rewrite it a little bit... This will become... $\sqrt{5}$ I am sorry. This will become, you know, $2^{klog_2\left(\frac{1+\sqrt{5}}{2}\right)}$, okay? So this gives you a good picture, right? So a_k is the number of length k binary sequences, okay, with no consecutive ones. What is the total number of length k binary sequences? It's 2^k. If you want no consecutive ones, what will happen? It's $2^{klog_2\left(\frac{1+\sqrt{5}}{2}\right)}$. So this number I don't know what it is, I think it's some 0.69 or something like that. So you will get a good feel for how this number grows, okay? As k becomes larger and larger, okay? So that's, so this kind of expression is very useful to understand what happens to these sequences, okay? So once again a very, you know, an application which looks like there is no eigenvalue there. But there is a linear recursion hidden there and eigenvalues once again wonderfully show up and control the asymptotic behavior in a very nice way.

Okay. So I am going to give you a picture to understand the previous one, okay? So the previous counting that we did of number of binary sequences of, you know, without consecutive ones can be thought of as a walk in a graph. So now what is a graph? A graph is basically nodes connected with edges, okay? So here is a graph, very simple graph. There are two nodes. One node is zero corresponding to the node bit zero in the binary sequence. And the other node is one corresponding to bit one in the binary sequence, okay? So now notice what do I mean by walk in the graph? So at every node, when you, when you are walking in the graph, at every node you will be, you will start at some node, let us say you start at node 0, okay? You have a choice to pick one of the outgoing edges and walk along it, okay? So out of node 0, I could either pick this edge and walk to node 1 or I could pick this edge and stay in node 0 itself, okay? So that is called a walk in the graph. So once you come to node 1, what happens? There is only one outgoing edge. So I have to walk back to 0, okay? Is that okay? I have to walk back to 0. So this, if I count the number of walks of length k in this graph, I will be counting the number of binary sequences of length k with no consecutive ones, okay? So counting walks in graphs is what I was doing in the previous example.

So maybe this graph was too simple. Maybe we want to, you know, make the graph a little bit more complicated. Here is another graph, okay? There are three nodes now. 0, 1 and 2. And I might be interested in counting walks in this graph. So why would you want to do that? There are lots of applications quite often. But this is interesting. But let me not go into great detail here. So let us just say I am interested in counting walks of length k in this graph below, okay? So how does the walk behave? Means I could be, I could start at zero, let's say. I can either stay at 0 or go to 1. If I stay at 0, the same thing continues. But if I go to 1, I have to pick up this guy and go to 2, okay? So at 2, now I have a choice, okay? All possible choices are there. I could go off to 0 or I could go off to 1 or I could stay in 2 itself, okay? So this is what happens in this graph. So how do you count walks in this graph? So what you do is this. Again the same trick, okay? So supposing walks of length k in this graph, supposing this is a_k. I will define a_{k0}, okay? This is number of walks of length k. Length of a walk is basically number of edges that you traversed, okay, ending in 0, okay? So let's say you assume you started at 0, okay? You can change the starting also. And then a_k, you say you start at zero and you are interested in a_k. a_k is the number of walks of length k starting at zero. Let us say that's what you are interested in. So a_{k0} will be the number of walks of length k ending in 0, node 0. You define also a_{k1} which is number of walks of length k ending in 1. Likewise you define a_{k2}, okay? So I will put a ditto here. Ending in 2, okay? So now you can write $(a_{(k+1)0}\ a_{(k+1)1}\ a_{(k+1)2})$ as a 3×3 matrix here times $(a_{k0}\ a_{k1}\ a_{k2})$, okay? Think about what you would put here if you want to end at 0 and if you had previously ended at 0. Yes you can have a possibility. There is one there. If you had previously ended at 1, there is no way you can go to 0, right? So this would be 0. If you had previously ended at 2? Yes, it's possible, okay? Now what about $a_{(k+1)1}$? If you previously ended at 0? Yes it's possible. If you previously ended at 1, no chance. If you previously ended at 2 also it is possible. Is that okay? You have that. And now what about $a_{(k+1)2}$, okay? If you previously ended at 0, you cannot get 2. If you previously ended at 1, yes it is possible. This is also possible, okay? So hopefully I got this right.

(Refer Slide Time: 16:16)



So notice what this matrix is. You could call it A, okay? And this defines your linear recursion and the eigenvalues of this matrix determine what is going to happen, okay? So now notice what this matrix is, okay? This matrix is an interesting matrix. It has 3 columns and 3 rows. You can imagine that every row corresponds to 0, 1, 2. Every column also corresponds to 0, 1, 2 node by node, okay? And what does this 1 represent? If there is an edge from 0 to 0, I put a 1 here. And here what do I do if there is an edge from 1 to 0? I put a 1 there. There is no edge from 1 to 0. So I put a 0 here. If there is an edge from 2 to 0, I put a 1 here. Yes that is there. So here there is an edge from 0 to 1, I put a 1 here. If there is an edge from 1 to 1, I put a, there is no edge so I put a 0 here. There is an edge from 2 to 1, I put a 1 here. So this matrix sort of captures the graph. It's called in, in some places it is called some sort of an incidence matrix, okay? And it captures the connections in the graph, okay? So this matrix captures very cleanly for you the connections in this graph and that connection also gives you the evolution that is needed when you want to count walks in a graph, okay? So you go here and I am not going to give you the eigenvalues, eigenvectors of this A. There is a very powerful theorem which says if you have a matrix with positive values then there is the largest absolute value, absolute value of the eigenvalue is actually positive. And, you know, all those kinds of wonderful results are there. I'm not going to talk in detail about it. But you can go ahead and find the eigenvalues, eigenvectors. See what happens and you will know even if it is non diagonalizable, right? So you will know what happened to A^k as k becomes really, really large, okay?

So hopefully this lecture... So this is the last slide. I am going to stop as far as this lecture is concerned. Hopefully this lecture gave you a feel for how, you know, eigenvalues and this linear recursion idea, okay, shows up in counting problems of various types in graphs and how, you know, matrices are naturally associated with graphs. Notice how this incidence matrix became naturally associated with this graph, particularly with respect to counting problems of this type, okay? So this is a very powerful idea. In a large area of mathematics and many applications people use this connection between graphs which express some relationship to matrices. And then eigenvalues of those matrices, you know, control the linear recursions or counting or walking that happens on the graph and you get some nice results in that fashion, okay? So I will leave you to work out additional details of this. But this is the conclusion of the lecture. Thank you very much.