PAGE RANK: HOW DOES GOOGLE WORK? 09

Alright guys so in the previous videos you had been to introduced to the new data structure of the direct graph that is variant of the graph data structure called the directed graph where the direction of edges matter so let us get into the networkx the package for graphs and let us see what are the support it provides to deal with directed graph data structure. So let's get into the python syntax and other cases let's get started. So as the first step the package that supports the graph data structure operations is networkx that will be importing will do that, import networkx as nx so this is just a short name given that the networkx is a lengthier name so i am using a shorter name nx so i have imported it. So how did you create a undirected graph? If you want to create a new undirected graph, you would say let me say in directed graph so let me say u, u is nothing but you create a nx dot graph this would create a undirected graph right this was used to create a undirected graph in case you want to create a directed graph how would we do that? Let's see let me say G is my graph i will create a nx dot DI graph is the command see observe that d is in caps and g is in caps di graphs is another terminology use to say directed graph. Directed graph di graph is a sort of acronym if you consider it as. DI graph d and g has to be in caps please observe this DI graph so DI graph directed graph object has been created so if you would want to know what are the nodes present in the graph that's let us see g dot nodes, nodes see its empty node view is abstract data type it is using but we are comfortable with the list representation so given that it is using a different representation in python two if you are following the output of g dot nodes will be a list data type but in python three they have a different representation so we need to convert this into list representation so that it is easier for our programming purposes. So currently it is empty so there are no nodes so let us add a few nodes something like this G dot add nodes from add nodes from you have a thing so you need to pass a list as a parameter so a list where you enter the list id's. Ok for example abcd whatever you want you want to give it as a list let me use numbers itself for the nodes. I see i want to create some three nodes zero one and two so i can say zero one and two what if i want to do for hundred nodes? Thousand nodes? If i want to increase, is it possible for me to manually enter all the numbers, no right so in that case what we can do is we can use for loop here and we can do it to demonstrate that even for this case as well i will use a for all and demonstrate for i in range how many nodes you wanted you just need to give that. Let me say i want some five nodes so i have given five so this is there so i want to add node number i for i in range five that is you iterate over the number line till you reach five so from zero it starts zero one two three four it will stop so all those whatever is hte values that i has been taking i had said zero one two three four you append that to the list that is what this particular line means. This is one line way by which you can add a collections of nodes if you want to name them in the numbered fashion only so you can added it this way so please observe this g dot add nodes from you need to pass a list given that i want to number it sequentially i can use a for loop and get it done instead of enumerating those numbers manually this is how you have added the nodes now let me see what are the nodes now. See as i have said zero one two three four has been added so as i have said this representation the tupple sort of representation has not very friendly for us. List is friendly for us we can iterate over the list easily so we will convert it into a list. List of G dot nodes so yes see it has been converted into a list format. Zero one two three four has been

converted list format so what are the edges present in the graph. G dot edges you would see out edge view it says so in an undirected graph you can just say edges but in a directed graph if you just say edges it by default is taking out edge that is the outgoing edges so we need to mention in edges and out edges G dot out edges there are no outgoing edges and G dot in edges for incoming edges there are no incoming edges and as this is the different data type we wanted to be in a list format so we will be typecasting it into list bit given that currently we don't have any edges so let us add a few edges and again will do it. So let me say G dot add edge you need to give the source and the target source is the starting point of your edge and target is the end pint of your edge. So if i say one comma two and edge it is a directed edge from node one to node two will be created, will be added into a network. So now if i see G dot edges so one comma two is the only edge i have so if i say in edges it's also one comma two in case of out edges its one comma two so basically it doesn't matter if you just say edges where this in edges and out edges terminology matters is if you give a particular node and say what are the incoming edges to this node. Or what are the outgoing edges from this node in that case this thing actually matters so let me say G dot out edges of let me say one let me save this see one comma two where as if i give G dot in edges of one its empty because there is no incoming edge to node one currently this is how this works so let me add few more edges and let us see add edge from zero comma three let us say let us say two comma three, let us say that this be a bidirectional edge three comma two as well let us add three comma four, four comma one please note that the order matters. You are giving the source the starting point as the first element and the target as the end point as the second element this order matters. Please note this as like in undirected graph you cannot give in any order please note four comma doesn't mean that one comma four is present that is if there is an edge form node four to node one node one to node four there is an edge that doesn't, you cannot guarantee that there will be an edge. So this is the specially of the directed graph as you could see now if i say as i have said list will converted into list given that it is easier for us G dot out edges of any node say for example let me say node two so this is an outgoing edge, let me say node three these are the outgoing edges, so from node three there are two outgoing edges one towards node two and one towards node four there are two outgoing edges so you have a list of tupples an edge is denoted by a tupple. List of tupples this is how you have the output of this particular function out edges. Please note this, this will be used in our programming screen cast, this functionality out edges we will be using it also if you want to find the incoming edges you can say in edges these are the incoming edges from zero you have incoming edge towards three and from two you have an incoming edge towards three these are the incoming edges so this is how you can see the outgoing edges and incoming edges. We convert it into list format because that is easier for us. Alright guys so we have seen some pre requisites some functionalities that are provided in networkx package to work with directed graph and what is so important about directed graph is that you have to again and again keep in mind is edges are directed an edge from node A to node B doesn't mean that you have path from node B to node A as well. That is direct edge from node B to node A as well exist this is not guarantee that is if there is an edge from node A to node B there is no guarantee that you will have an edge from node B to node A this is key point from directed graph please do keep this in mind and will see about the algorithm of points distribution

method how do we do that and what is the procedure to be followed will see that in the upcoming parts.