PAGE RANK: HOW DOES GOOGLE WORK? 14

Alright guys so in the previous videos we had seen what is a directed graph how can we work in networkx and then we had seen what is points distribution method, we had simulated it with small network with just three nodes in excel sheet now we had started with the programming part. We had created a network we had visualised it we are creating a random network simulating coin tosses, we had visualised that network then we had initialised the points equally some seed values we had initialised all the nodes we are getting equal points first at the first stage then we have to keep distributing under keep distributing we need to now define how to distribute the points actually so we will do that now. So let me define it define distribute points on a given graph on a given point configuration so let me distribute so as and always we need the list of nodes to operate so let me get the list of nodes list of G dot nodes alright so before starting initially every node has given something to its neighbours so it will have nothing so let me say new points are initially zero right so i need to assign tit so let me say or i in range length of nodes that is how many number of nodes are present length of nodes in n o d e s ok so for that we need a list as well right so let me say new points so let me create a list new points let me say new points new points to append zero why zero? Because the node has given everything to his neighbours and it has nothing. That is the state then we will that's how we had even introduced in the slides right so the node will give its to neighbours that's it the next stage we considered what is it getting from? If everyone is giving, someone has to get so what is that it's getting is the next consideration so first the giving part so this is the giving part so we have to keep appending the zero value because it has given everything it has nothing so this is your new point so now let us say for n, n for node let me say n in nodes i take a node n and say out for out edges so we have to see the out edges and we have to share it with the outward edges right so i am taking the list of out edges list of G dot out edges and you need to give the node id it is n basically so i gave it so how will this appear? List of tuppples an edges represented by a tupples source comma target something like that list of tupples we will get so if this is an empty list what happens, it need not share anything with anyone it will just keep getting whatever it was having previously so we will handle that case here if length of this list out is equal to zero meaning that it has no outgoing edges then it will just keep getting the same points what it had got in the previous iteration right so i should say the new points for this particular node n is nothing bit new points of that node n previously plus whatever is the points it had in the previous iteration previous value of points whatever was the initial configuration will again retain so it will it had got some for example say it had got some ten points again it will get some ten points it may not share with any one even that there is no outgoing edges if there are outgoing edges you need to share, if there are no outgoing edges it will again get some ten points, if ten iterations again ten points it keeps accumulating so that accusation thing we have handled here right? ok so if this is not the case its having some outgoing edges then it has to share some points so let me say share it has to share how much? How should it share? It has to share equally among all of its neighbours, how many neighbours are present? That is given by length of this out so how many outgoing edges? It has to share it equally so whatever is the points it has been having you have to divide it equally among all the outgoing edges so that is the procedure so we have got the value of share and so this is the value of share so this

much it has to give the whole it has to give, it has to give to the target nodes in the outgoing edges list right so that we will see so for the items in this outgoing list. What is formats of the items in the outgoing list? Source comma target right so this is of the format source comma target that is how the items in the outgoing list are for each of this in outgoing list your source will be the same list n right target differs so we will take the target value and we will work on it so the new points of target target is nothing but whatever it has earlier plus what is the share it is receiving from this node n so why do we having this new points plus share why do we have this? Because for example we had one node where it will half from i guess from node C half from node A and fully from node B so it had different shares so when you run the loop with node A you would have given half when you again run it with node B what happens is, if you just say it is equal to share the previous share whatever it has got from node A that value will get destroy. We need to retain that as well as this that is why we are adding it ok so this is how the new points calculated ok once everything is done what you do? You return your new points right so this is how the distributions occurs just let us see once this part so we get the list of nodes we have new points initially it is zero given that we have all given whatever they have nothing so given that they have given something has to be someone has to be there to get so each node will get some share as well so we are going to see the getting part now. What happens? We are taking a node n and then we are seeing the outgoing edges so to the outgoing edges only it would have shared so the outgoing edges the target nodes will get the share will get the new points if there is no outgoing edge what happens is? The nodes will keep accumulating the same points in this iterations as well, so that is why i am adding the same points again no if that is not the case there are outgoing edges it has to share it so it will equally share the points it ahs with all the neighbours so this share has to be given to the neighbours so it is of the form source and target that is how you get the list of edges so to all the target nodes you have to increase this share you have to give this share to the target nodes that is what we are doing the target nodes will get the value of share along with that whatever it has got and why we are doing this addition because a node may get different share from different nodes to account that we are doing this addition so once you have computed it for all the nodes you can reform the list of new points. So that is how single time distribution occurs so this is distributing just for one iteration. So in this keep distributing function you are repeatedly calling the distribution functionality and you keep repeating the process again and again till the user stops alright i hope you are now clear with the procedure may be you can take a pause and you can think what we have done in the previous parts what we have simulated in the excel sheets, what is the procedure we have explained, you think you give a pause you will understand it really well. So ok let me clear the screen here actually i will do it fresh ok so this screen is clear here so let me run it let me run so i have got some values initially it was hundred now i have got something hundred and thirty five, hundred and three nine, ninety eight something i have got it so it's asking me if i have to stop or i have to continue, i want to continue so let me press some other key i prefer enter key it has continued continue continue continue continue continue continue like keep continuing after some time did you see here the values here as well as here there is no much difference i guess it has almost converged so there is convergence so i can up here lets press hash yeah it has stopped so as you could see we had some graph structure you can see some random graph structure is here so i had done distribution once and this is a result of distribution once and it is asking me

if i want to continue or stop i had continued for lot of iterations and at the end i found there is convergence right? see the values here its one thirty six point eight nine eight five, one thirty six point eight nine eight five there is no much difference so it has started converging so once it converges after that whatever be the number of times you repeat it the same values will remain so given that i had hit the convergence i stop that's what i have done now so this is how we have checked the values of points after repeatedly distributing it now we have to run the nodes based on the points it has accumulated so for example forty five is the minimum value so this node gets the least rank and one thirty six is the highest value so this node gets the highest rank something like that so i have to rank the nodes so basically you can considered as marks so initially i am giving hundred marks to people and there are playing the game and distributing it and at the end whoever has got maximum marks whoever got minimum marks i have to sort the nodes i have to compare the result with the actual page rank procedure that is implemented in networkx i have to compare that result and so that we will be doing in the next part of the video.