PAGE RANK: HOW DOES GOOGLE WORK? 12

Alright guys so in the previous videos you had seen what is the directed graph? How can you work using networkx packages in python and how can you deal with directed graph in networkx also we have seen points distribution method, what is the procedure in which you play and how would you distribute your points, how would you share your points with your neighbours also we have seen a simulation in excel spread sheet that if you keep repeating for multiple iterations the values converge that is the difference between this iterations and previous operations will be either insignificant or there is no different at all that difference is highly negligible in that case we say the values converge. So we will now see simulation using python so will get started. The first step what you have to do is, you need to create a directed graph so let's do that. So how would you create a directed graph? For that we need networkx package let us import that first import networkx as nx so we have imported it so let us now go ahead with creating the directed graph, how would we do that? G is the graph object is equal to nx dot Di graph Di graph this is how you create a directed graph. So we have created it this is the empty directed graph so the next step is you should add the nodes so let us do it G dot add nodes from yes so will have to pass the list so let us say we need some ten nodes we have taken a some small example of three node graph in a demonstration now let us take a slightly bigger graph ten node graph may be you can say hundred node, thousand node graph you can just keep extending the same thing to larger number of nodes with the complex graph you can keep doing that. So let me now give the demo with the ten node graph so i am going to insert ten node from zero one two three till nine so using a for loop single line way as we have seen in the pre requisites portion of this lecture series we will be doing that so it is nothing but it just a i for i in range how many number of nodes you wanted you just need to give, i am going to do with the ten node graph so let me do it ten oh ten nodes from zero till nine had been added to the graph so the nodes have created so now we need to add some edges so let me say add edges so i need to add edges and get back to new graph with edges that is rently we just have nodes and no edges so i need to add edges and get new structure as graph G let me say G would be my new graph i have to add edges that's what i have to do. I have created a graph object i have added the nodes now i have to add the edges so undefined name add edges it says that means i have used this name but i have to define what this actually is so let me go ahead and do that let me say define add edges so first add edges what do you need? Between two nodes there will be an edge right so we need to get the list of we need to access the nodes frequently so will take the list of nodes so let me say nodes is equal to list of G dot nodes so we got the list of nodes now i need a source node let me say S for source for S from nodes and i need to pick a target node let me say T for target from nodes list i have picked it also i should see source and target must be two different nodes if it is a same node it will be a it will be like telling i like myself every one like them self so capturing this information is actually waste of space something like that so will not do that generally that's what we call as loop in graph theoretical terms so we don't prefer loops generally that's what we call as a simple graph we are going to construct a simple graph where there is no loops so we need to see that if these are two different nodes so i am going to perform a check here if s if not equal to p only then i will think if i have to add and edge or not so if i just like that keep adding an edge what happens this loop will run for all possible paths and what you get is a complete graph i don't want a complete graph i want graph to be randomly generated so randomly generating i have to go ahead with this package random so let me import it import random so i have imported it so i am going to randomly add an edge so how could i randomly add an edge so i can simulate it with a coin toss so i may say that i just pick two nodes two different nodes if they are two different nodes what i will do? I have picked a node s i have picked a node t source and target so s and t i have picked two nodes if they are two different nodes what they have to do, i will toss a coin if i get a head i will put an edge otherwise i won't put an edge if a toss an if you toss an coin what is the probability? What is the chance that you get the head? One by two right because head or tail right two possibilities one of the possibility is head which is your decided thing so one by two this is zero point five is the probability of your head so i am going to simulate the coin task if i get a head i will put an edge otherwise i won't put an edge so let me say i am going to toss a coin that is nothing but i am going to generate a random number that's what i am going to simulate so let me say r be my random number going to generate random dot random what does this do? It will generated a random number in the range zero and one so let me take zero point five that is the probability for my head right so i will say if this random number r is within zero point five i consider that i have got a head otherwise i will say i have got a tail this is how i am going to simulate the coin task process so if this r is within point five less than or equal to zero point five what do i do? I consider it as a head, so if there is a head yes i have to add an edge right so i will add an edge g got add an edge from source to target so i have added an edge so i will do this for all possible pair of nodes and whatever is the resulting graph i am returning the resulting graph i will return graph G after adding the edges based on coin toss so coin toss is a random process if you toss a coin you get a head or you get a tail so based on that if you draw an edge you will get an random graph every time so you can experiment what happens instead of having the faced structure you get a different structure every time and you experiment what happens whether there is convergence with or any structure you an experiment it and there will be convergence you need to run for sufficient iterations may be ten node graph so i suppose maximum some ten thousand iterations definitely there will be convergence so why do i say this concretely because there are some mathematical results which say that there will be convergence in any structure so just a intuition that i could give you is just a hint is same probability is a hint you can del deep into it if you are interested there are mathematical results there are proofs which states that definitely there will be convergence if you keep repeating this procedure so you can simulate it so for example i had taken as hundred hundred hundred each right initially all nodes where having hundred points that is how i had taken so i can instead of giving hundred points if there are some k people i can give one by k each the sum turns out to be one so this is one thing in probability right the individual probability is may be anything but the sum of all the probability will be sum up to one right so you consider it like that and you can think of it has probabilistic model and you can prove that its slightly rigorous the mathematical part but its not very difficult actually you can understand it all that you need to know is some amount of probability if you are good with probability you can look up to what the mathematical proof is and you will get to know that definitely whatever be the graph structure whatever be the initial configuration initial we are starting with equal even if you start with unequal configuration it will converge probably the number of iteration it takes to converge may differ but it will definitely converge so that is the mathematical result so you can experimentally verify this with different random graph you run it once you will get a graph you can verify whether there is convergence similarly you can run it again again again you can run it in you will get different graph every time you can verify whether there is convergence as per that mathematical result there should be convergence let us see what happens ok. so we have added the edges we have added the nodes we have added the edges now what should you do we should visualise the graph visualise the graph to plot the graph and visualise it what do i need? I need this package matplotlib so let me import it import matplotlib dot plyplot as plt shorter form so let me say matplotlib pyplot i have imported so what should i do? We should first draw the network so i will say nx dot draw the graph G and anything have a labels, labels are nothing but the entries of the node that is this is node one this is node two this is node five so i need the labels so i will say with labels equal to true i will save this so you will get the graph draw with the labels and now i have to show the drawn graph i will say plt dot show ok done so what we have done till now we have created a directed graph we have added a few nodes added some edges we have added edges randomly simulating a coin task and we are going to visualise the graph that's what we are done till now ok lets go ahead run this and see what happens. Let me run it see some graph got generated some network is there if i run some other one more time i will get another network another time get another network some other network some other network so every time you are getting a random network so on this you can run your distribution method and you can check whether there is convergence. As i have said there is a mathematical result its says there will be convergence irrespective of whatever is your initial distribution initially however you distribute the points and as well as how whatever is the structure for your graph irrespective of that you will have convergence there is a mathematical result let us verify that in the upcoming parts of the video. That's why we are generating random graph so if you run this iteration you will have a different output some other iteration you may have different output so we are basically experimenting it so for different structures how is the behaviour we are basically experimenting that's what we will be going to do. So will do this in the upcoming parts of the lecture.