PAGE RANK: HOW DOES GOOGLE WORK? 07

Welcome again all of you we have now come to the implementation part of random walk method so let us start. First of all as and always we will be importing some libraries so first of all let me import networkx import networkx as nx then will be importing random library we will also be importing matplotlib i order to visualise the graph so i will write import matplotlib dot pyplot as plt so i think now we are done with the importing libraries. So as and our previous programming screen cast i showed you how can you draw random graphs using networkx, so will be drawing that particular graph here so i will just write g is equal to nx dot gnp underscore random underscore graph and first parameter is number of nodes so i will take ten nodes here and i will also take the second parameter the second parameter is the probability of edge creation so i will take it has point five and then we also need a directed graph in page rank yes in very first video i explained that we need a directed graph for page rank so i will just write directed is equal to true since we need a directed graph. So next what you need to do, i will draw it and show you whether we are getting a directed graph or not. So nx dot draw g and plt dot show so let me run it. I will write random underscore walk dot py. As you can see we have a directed graph here we have a directed graph here we can see the arrows here but what i want more is i also want the labels of nodes to be visible so i just write with underscore labels is equal to true so that i can also get the labels on the particular node so let me run it again. So now we have labels visible also we have zero one till nine we have zero one till nine since we only passed ten has a parameter here. So ten is the total number of nodes here, so now we are done with drawing the graph here next what we need to do? As with the random walk method we will pick a node randomly from this set of nodes from zero to nine we will pick a node randomly so i will just take a variable here i will say x is equal to random dot choice. I hope that you know what random dot choice do it will pick a element randomly from a list so i need to pass the list here so i will write i for i in range what is the range here? G dot number of nodes, g dot number of nodes so this is the range here so we will pick a node randomly so i will just write a comment here x is the random source node since we have picked x here we also need to increment it counter but how will we store the counter? We will store the counter in a dictionary so what we need to do here? We need to initialise the dictionary here, i will use dict underscore counter is equal to curly braces this represent dictionary here. I also initialise the values of dictionary to zero so i will write for i in range g dot number number of nodes dict underscore counter i is equal to zero. Here what you need to do, g dot number of nodes in dict counter is equal to zero here. Since this is the method you need to pass you need to pass the round braces too. So i also need to implement the counter of source node so what is the source node here? We have x here dict counter x is equal to dict counter x plus one so we have incremented the counter too we have initialise the counter of each node as zero too. So for next what we need to do here, we need to look for the neighbours of x and then we need to traverse all of the graph this way. So i will be taking many numbers of iterations here so i will have a loop that will represent the iterations here so i will start with example for this graph and then what we need to do, we need to look for the neighbours of x. So i will just write i will take a list here list underscore n is equal to list of g dot neighbours of x. Now we need to check we also need to check here whether x is the sink or not here. If x is a sink then we need to select a node randomly from the given sets of nodes

from the given set of nodes present in the graph otherwise we need to otherwise we need to select a node randomly from the neighbours of x. I repeat if the given node is the same we need to select a node randomly from the nodes present in the graph otherwise we need to select a node randomly from the list of neighbours of x. So i will just write if length of list underscore n is equal to is equal to zero here then that means this is the sink here it has no neighbours. So i will write then x will be random dot choice again we have to select the neighbours we have to select x from g dot number of nodes. I write i for i in range g dot number of nodes. So we are also done with it, we have handled the case if x is the sink else what we will do? Will select a node randomly from the list of n so will say x is equal to random dot choice list of n list underscore n. Then we also increment the counter of the particular node so i will write dict of counter of x is equal to dict of counter x plus one. i also increment the counter here too dict of counter of x is equal to dict of counter of x plus one so i think we have ended all the cases, i will just write some comments here if x is a sink choose a node randomly from neighbours of x so i repeat this loop this particular loop first of all we are looking for the neighbours of the source node, our source node is x here if it has no source node that means if it is a sink then we will choose a node randomly from the given set of nodes of present in the graph. Otherwise we will choose a node randomly from the list of neighbours of given node ok so we have handled the cases here and we have also incremented the counter of our particular node if that particular node is traversed in the random walk method so now we are done with it so you must be thinking how can you verify that this particular random walk method reflects the exact page rank values well we have networkx to our rescue, we have a built in function in networkx that tells us what are the page rank values so i will be using that particular function and we will be verifying that the values returned by a networkx function and a values returned by our random walk method are same or not? So let us do that, what is the method used here? Page rank so i will just use p is equal to nx dot page rank this is the method that will be using and we will be passing our g graph here. So nx dot page rank is the method that will tell us the exact page rank values and i will print p here and i will also print the dict of counter here so i have to verify whether this values same or not. So i think we are done with it, done with the program let me run it. So we have the values here we have zero as the first node, yes zero then one then two then three four five let me see we have we haven't sorted this values, we must sort this values so that we can get the clear idea the ranking of node given by the random walk method as well as the page rank method are same or not. This will be difficult for us for looking at these values so i will just sort this values here so i will just write, i will sort this dictionaries the dictionary p as well as the dictionary dict count. How can you sort the dictionaries based on the values it is very much do able in python so i will take two list two dictionaries here first one sorted p that is equal to sorted i will write p dot items and key is equal to item operator dot item getter one if oyu pass one here it will sort based on the values if you pass zero here and it will sort on the basis of the items on the basis of keys as we know in dict we have a key we have a value so it is your choice whether you want to sort the dictionary on the basis of the keys or on the basis of values since we have to sort the dictionary based on the page rank values so we will be passing one here and then again i will write sorted random walk is equal to sorted here we have dict counter as the dictionary dict counter dot items and we also will write key is equal to operator item getter one. Here i will print sorted p and here i will print sorted underscore

rw sorted underscore rw since we are using the operator library here we need to import operator two so i will just write import operator i hope we are now done with it and let me try to run it. We have the values here we have zero here and we have two here we have one here we have zero here so as you can see these values are not matching with the exact values so what you can do here is? We can always increase the number of iterations let me increase the number of iterations suppose i increased it to ten thousand let me run it again let me see we have two here by then we have eight here then we have one here we have one and eight here, here it is two eight one nine here it is two one eight nine and there are some discrepancies in these values also so let me again increase the counter suppose i increase it one lack, let me run it again so we have four here four here one here one here then again we have eight six here we have six eight here then we have two here then we have seven here three five zero nine three five zero nine so these values are matching to the exact values but there is still a discrepancy we have encountered that is here we have eight six here we have six eight so i will again increase the counter let me increase one zero here again so let me run it again so we have five nine eight we have five nine eight then four then six then three then one then zero seven two perfect! We have exact page rank values so you can see random walk method is effective to find the page rank values but you have to re iterate it many number of times so it depends on the size of the graph too you have to re iterate it again and again in order to get the exact page rank values the correct page rank values. so i hope that this method was correct to you guys i will explain you this method again. So first of all what we did we took a random graph a gnp random graph to start with and then we choose a node randomly we choose the source node randomly from the given set of nodes where we called it as x then we also initialise the counter of each node to zero and we incremented the counter of x since this is the source node and then we took a for loop in which we will re iterate many number of times this random walk method. First of all i took n in which we have the neighbours of x, list of n represents the neighbour of x and then we checked whether it is a sink or not? If it is a sink then we need to choose a node randomly from the nodes of the graph otherwise we need to choose a node randomly from the neighbours of x and side by side we are also incrementing the counter of the particular node if we traverse that particular node so increment the counter here of this node x and here also. So every time the node x is changing and this particular loop is iterating for many number of times and then we took the page rank method given by networkx so as to check whether page rank values given by the random walk method and the exact page rank values are same or not. And we sorted these two dictionaries based on the values so that we can get a clear idea what is happening here then we printing these two dictionaries so in this way you can find page rank using the random walk method. So my only suggestion is to iterate this method many number of times only then you can get the correct page rank values. i hope this programming screen cast was useful to you all, Happy learning.