NPTEL

NPTEL ONLINE CERTIFICATION COURSE

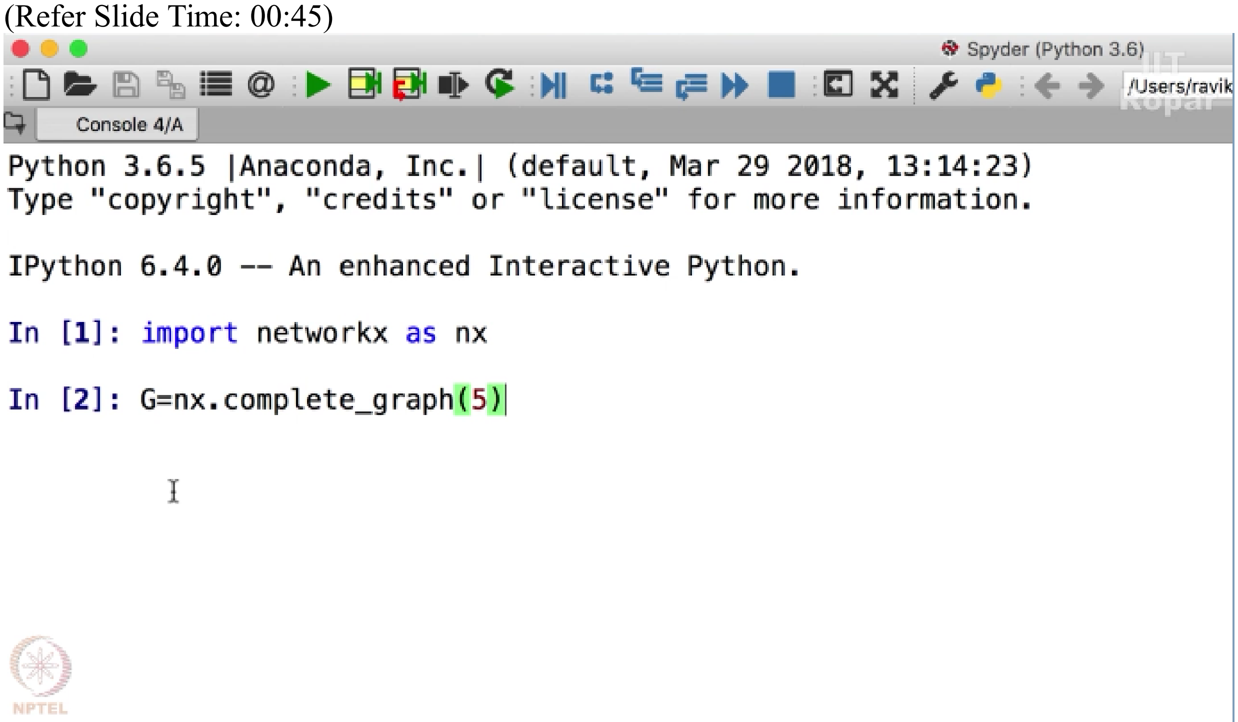
Discrete Mathematics
Graph Theory – 3 &
Generating Functions

NetworkX - Adjacency matrix

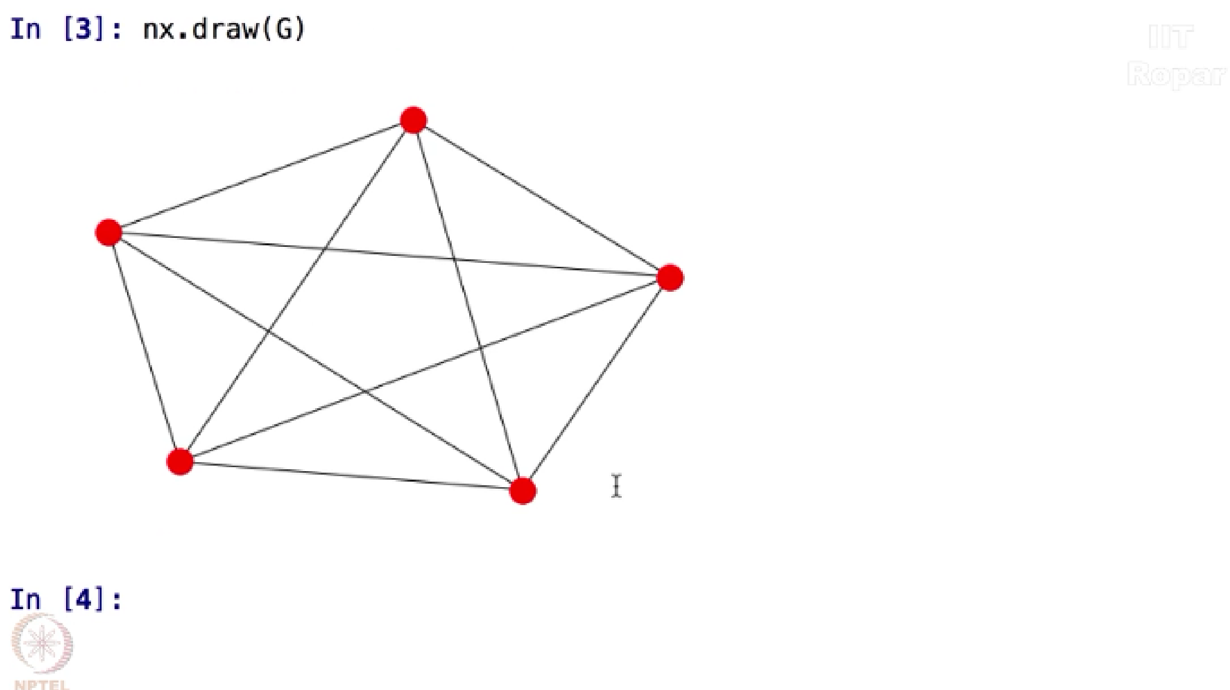
By
Prof. S.R.S Iyengar
Department of Computer Science
IIT Ropar

We will now be seeing how to create an adjacency matrix from a given graph, and vice-versa so if a graph is given how to create a adjacency matrix and from a matrix how to get a graph, right.

So now the first step is always importing NetworkX, so I have imported NetworkX here, the library has got loaded, what I am going to do next is creating a complete graph, so I have created, I'm going to create a complete graph on let's say 5 vertices, so I have named the graph as G.
(Refer Slide Time: 00:45)

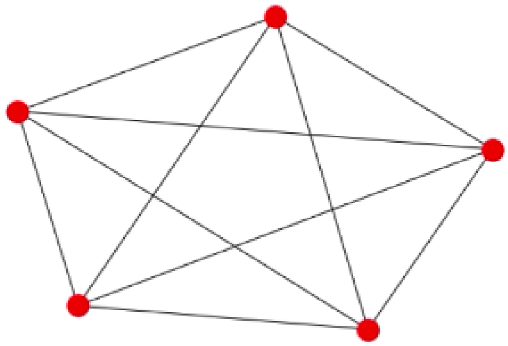


Now let us draw it nx.draw(G), so we have obtained the graph here,
(Refer Slide Time: 00:57)



do you see the graph? So we will now be creating an adjacency matrix for this graph, so before we go to create the adjacency matrix we have to import a new library called numpy, along with NetworkX numpy is also required, numpy is a library which we'll be using from now onwards for all the math part of this coding, right, so for all the math operations we will require the library numpy, so let me import numpy NUMPY as NP, so numpy has got loaded now.

Now I'll name the matrix as A, A = nx.to_numpy_matrix (G), so let us understand this command first, so I have created a graph initially and I have imported numpy now, so using numpy I can create a matrix, so I have named the matrix as A here, and the command as A = nx.to_numpy_matrix, right, so this will give me the matrix A, and for which graph is it going to create the matrix? For G which we have drawn.
(Refer Slide Time: 02:34)



```
In [4]: import numpy as np

In [5]: A=nx.to_numpy_matrix(G)
```

Now the matrix has got created by just typing A we can see the matrix like this, so do you see

(Refer Slide Time: 02:42)

```
In [4]: import numpy as np

In [5]: A=nx.to_numpy_matrix(G)

In [6]: A
Out[6]:
matrix([[0., 1., 1., 1., 1.],
        [1., 0., 1., 1., 1.],
        [1., 1., 0., 1., 1.],
        [1., 1., 1., 0., 1.],
        [1., 1., 1., 1., 0.]])

In [7]:
```

the matrix here 0, 1, 1, 1, so let me keep the graph and the matrix so that we can check, so observe it's a complete graph and hence the diagonals are 0 rest all are ones here, why is it so? Very obvious, it's a complete graph, every vertex here is connected to every other vertex and there is no loop on any vertex and hence only the diagonals are 0 and rest all are once.

So now we have seen how to get a matrix from a graph, well we can also go backwards, how do we do that? We can create a graph from a matrix, now let me name the graph as H =

nx.from_numpy_matrix(A), so do you see what I have done here I've given a new name for the graph which I want from this matrix which is created,

(Refer Slide Time: 03:53)

```
In [5]: A=nx.to_numpy_matrix(G)

In [6]: A
Out[6]:
matrix([[0., 1., 1., 1., 1.],
        [1., 0., 1., 1., 1.],
        [1., 1., 0., 1., 1.],
        [1., 1., 1., 0., 1.],
        [1., 1., 1., 1., 0.]])

In [7]: H=nx.from_numpy_matrix(A)
```

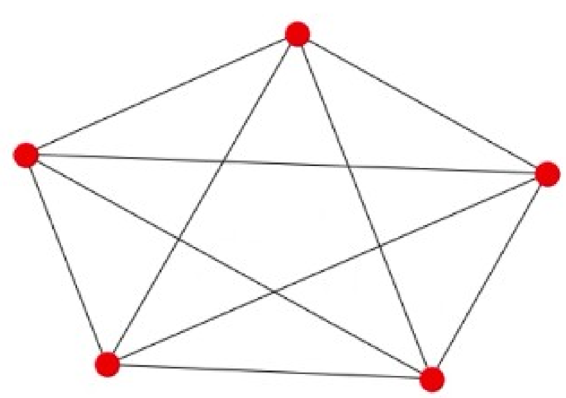
so from matrix and to matrix these are the commands, so from graph to matrix and matrix to graph, very simple to remember, so from numpy matrix and the matrix is given the name A and hence in brackets it is A.

Now the graph H has got created, now once I give nx.draw(H) the graph will be shown,

(Refer Slide Time: 04:23)

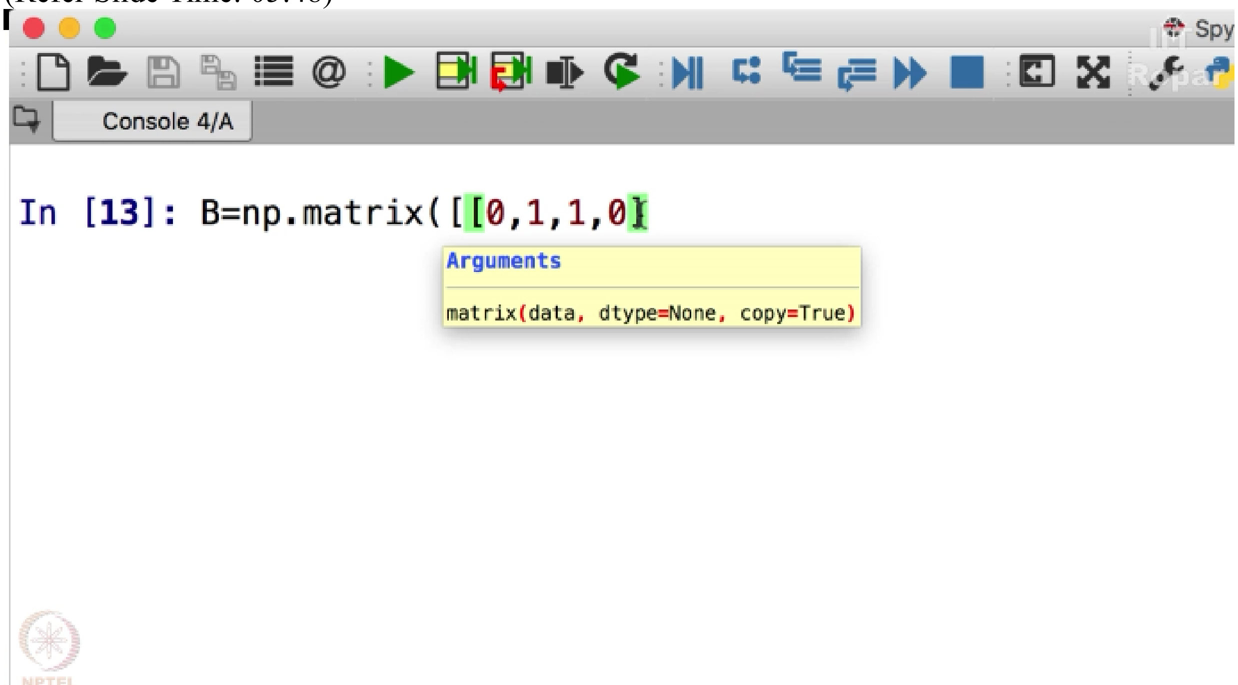
```
In [7]: H=nx.from_numpy_matrix(A)

In [8]: nx.draw(H)
```

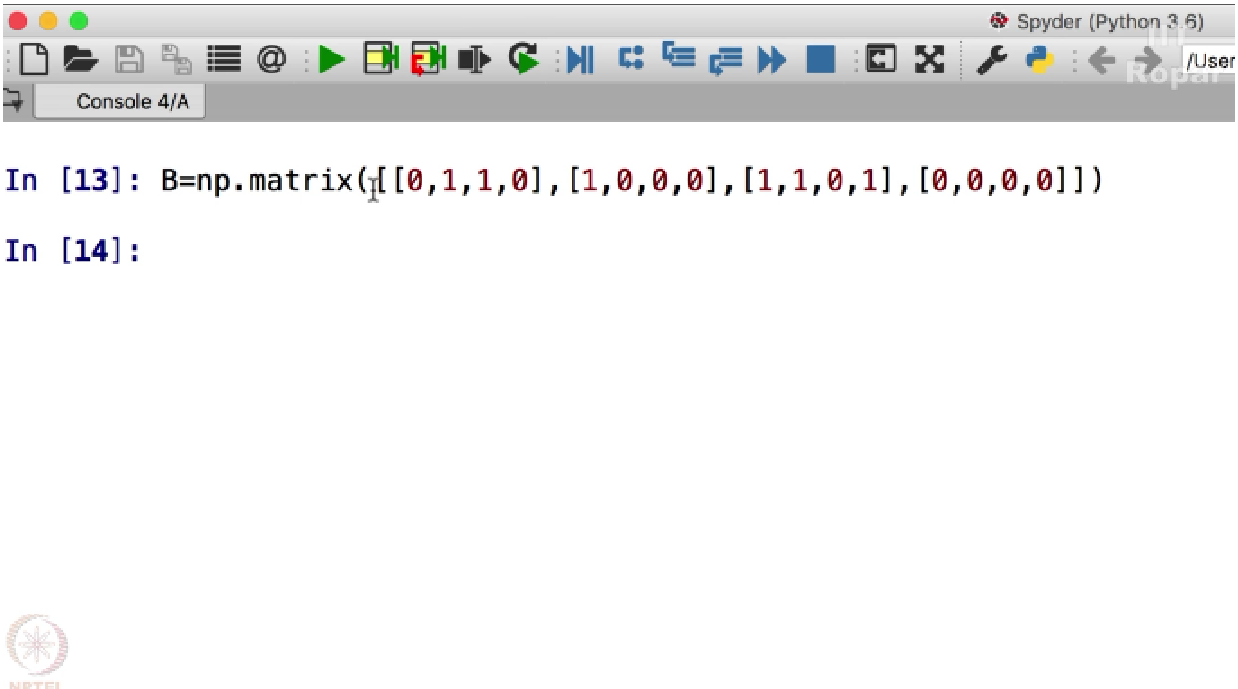


```
In [9]:
```

yes do you see that we have obtained the graph here corresponding to this matrix A, well we had used the same matrix and hence we have got back the same complete graph, now can you think for 2 minutes how do we do it by creating a matrix by ourselves, let us see if we have to construct a matrix and then construct the graph based on it how do we do that, let us see we have already imported numpy, let me give clear so that we can start a fresh, yes, so how do we construct a graph from a given random matrix? So what we are going to do first is create a matrix of our choice and then construct a graph based on it, so let me name the matrix as P, and do it as np.matrix, so I'm going to create a matrix like this and the syntax is parenthesis, and then closed bracket like this double brackets, please watch and then it is, I'm going to start giving a random inputs here as 0 and 1, so 0, 1, 1, 0, so this is going to be my first row.
(Refer Slide Time: 05:48)

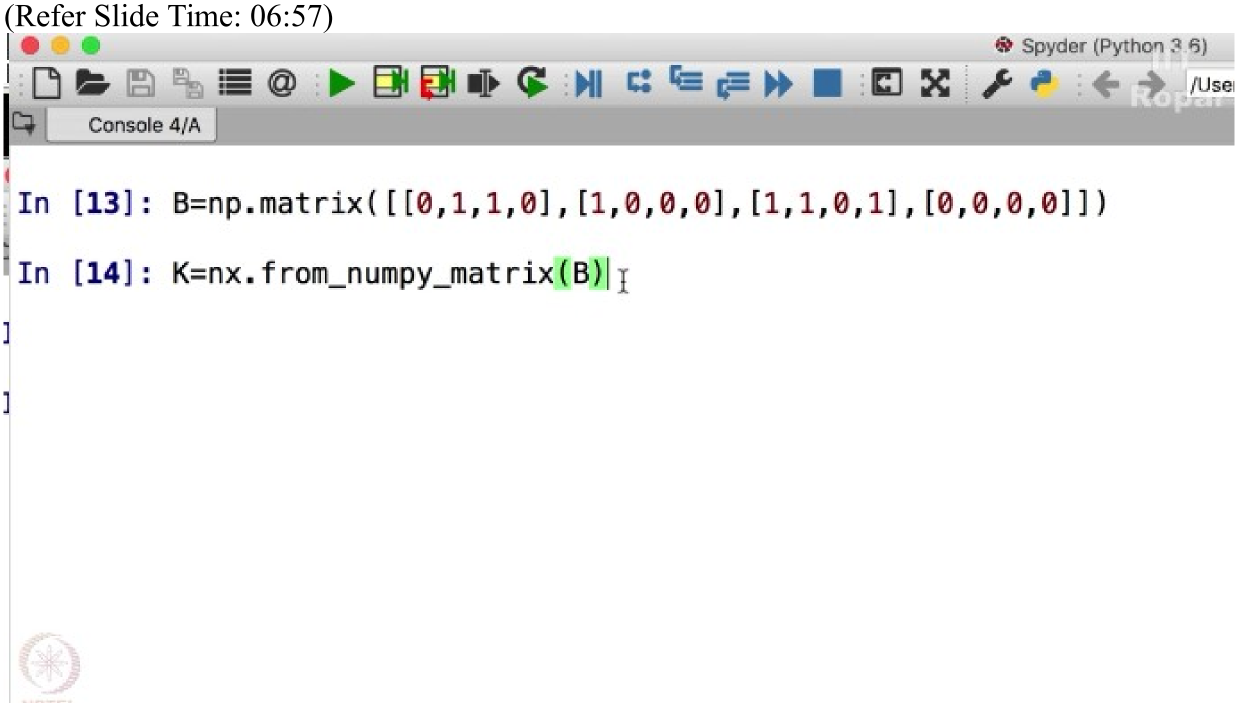


And then we have 1, 0, 0, 0, this will be my second row, or in the matrix and then 1, 1, 0, 1, this is the third row, and then the last one I'll just keep 0, 0, 0, 0, right, and so I close the parenthesis for the bracket for this one, and then at last the parenthesis, yes, so do you see that the matrix has got created now with the rows specified by us.
(Refer Slide Time: 06:22)



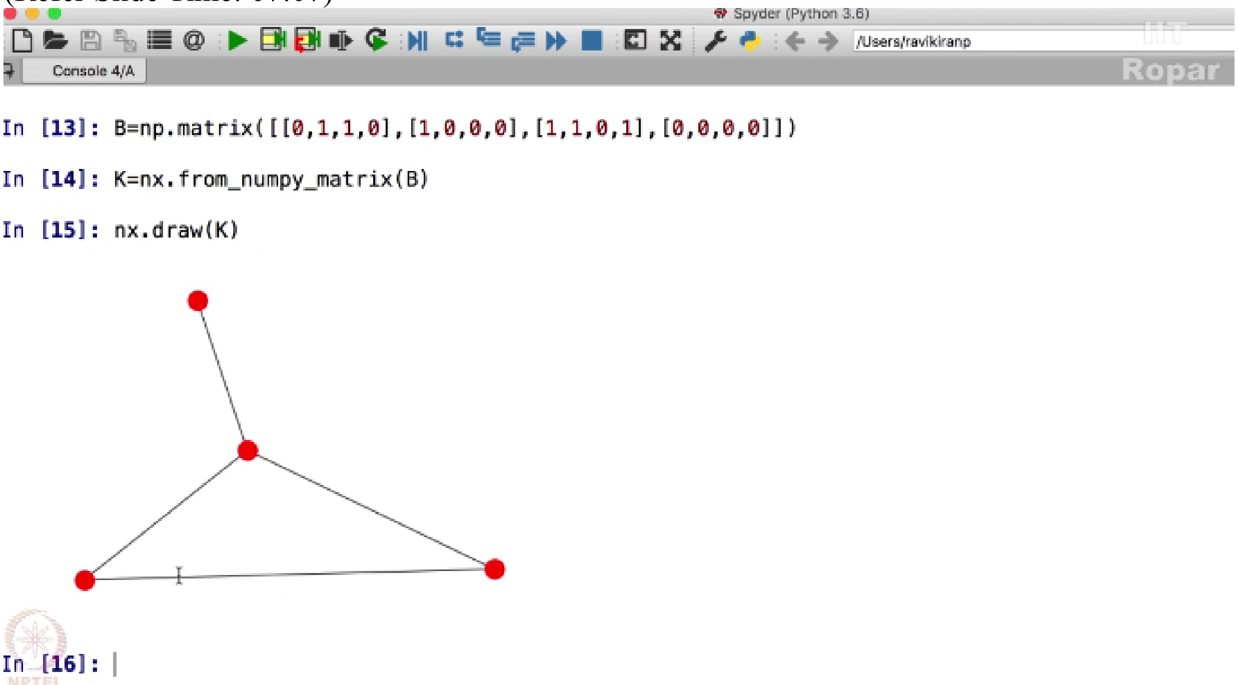
Now the graph will be created now by giving the command K = nx, so we have to go from matrix to the graph, so what will be the command? Nx.from_numpy_matrix, so what is the name of the matrix given B and hence nx.from_numpy_matrix, so if it is from matrix to graph it is from numpy matrix, if it's from graph to matrix it is to numpy matrix.

(Refer Slide Time: 06:57)



Now the graph has got created, let us try it nx.draw(K), K is the name of the graph,

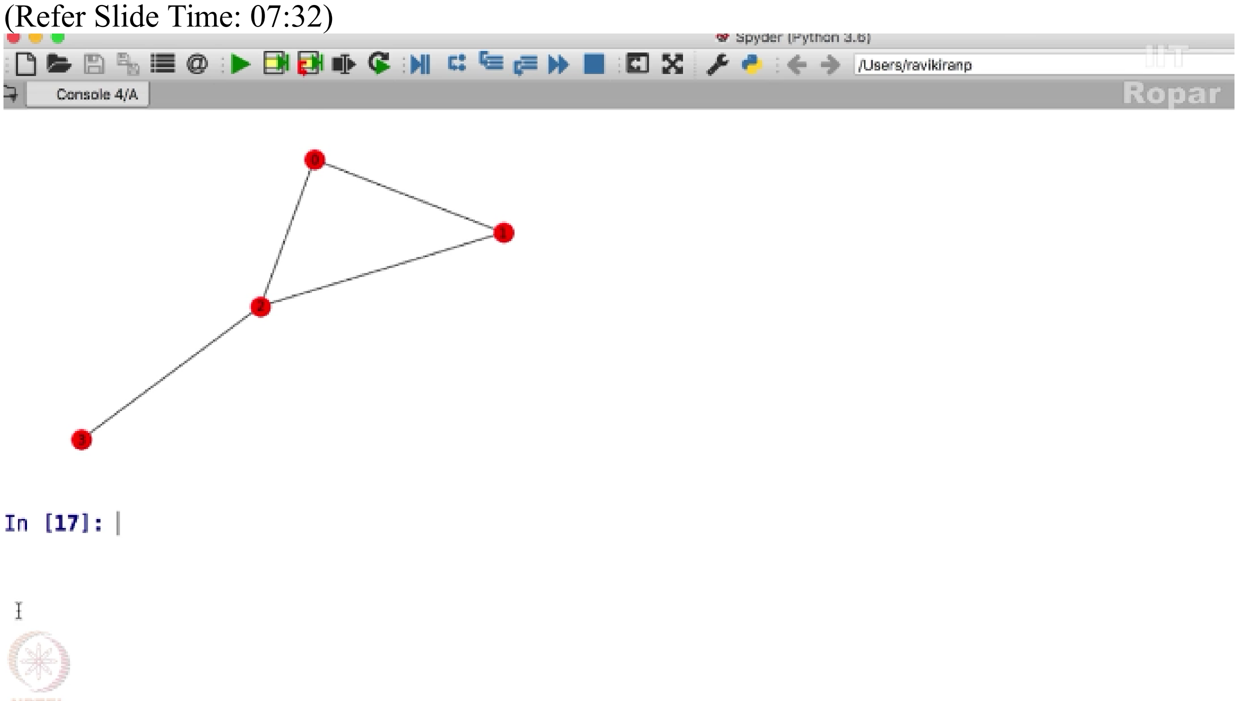
(Refer Slide Time: 07:07)



so do you observe here that the graph has got created with 4 vertices and edges as specified in the adjacency matrix,

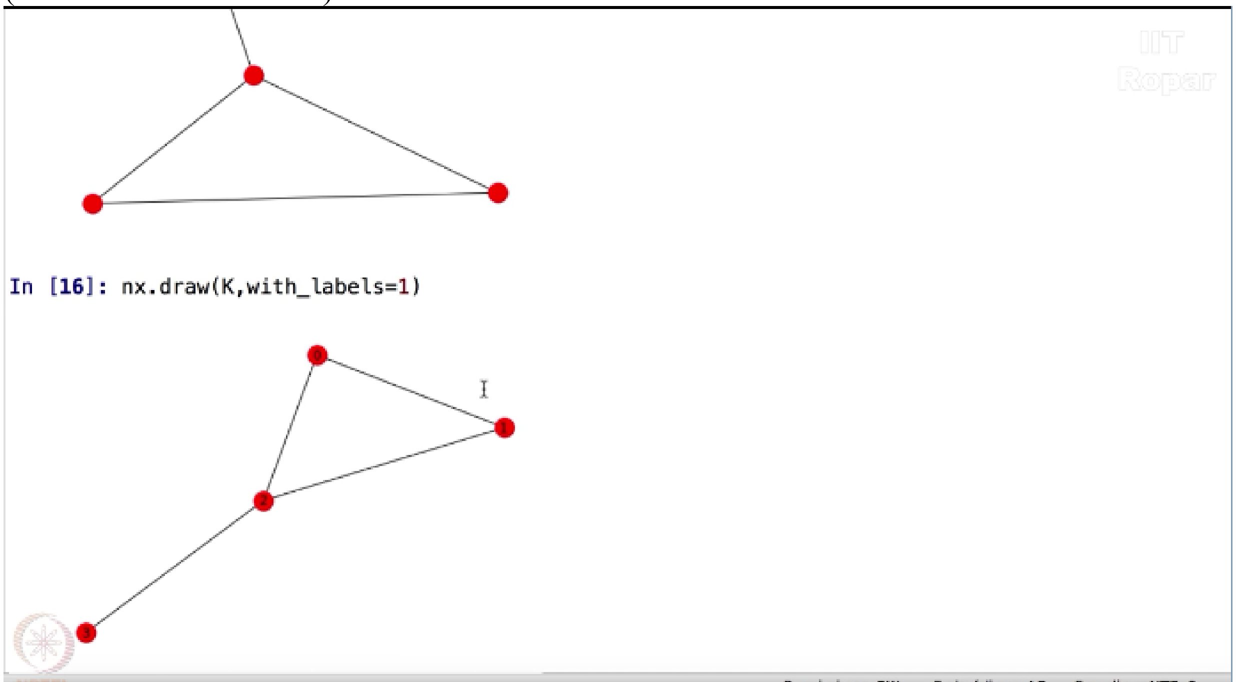
Now let us take nx.draw(K) with labels equals true, let us check,

(Refer Slide Time: 07:32)



so do you see we have obtained the graph with labels here, but let me tell you something do you see that the graph which we have obtained from the matrix specified here, it is not a symmetric matrix you can probably write it down another notebook as a matrix and you can see that it is not symmetric actually but NetworkX is very flexible in terms of drawing the graph in such a way that let me explain, in case the matrix is not symmetric NetworkX converts the non-symmetric matrix to a symmetric matrix, and then draws the graph, in what sense? In case there is an edge between 2, 3 vertices, and in case you have a 0 in 3 2 in the matrix specified here it will not consider 3 2, it will just consider 2 3 and go ahead and put an edge between the vertex 2 and the vertex 3, this holds for every pair of vertices, observe here,

(Refer Slide Time: 08:48)



by comparing the matrix with this graph you will be able to understand it well, you must probably write down the matrix on your notebook and check with this graph here, so the labels are there, check it with the matrix and you will be able to understand it better.

IIT MADRAS PRODUCTION

Founded by
Department of Higher Education
Ministry of Human Resources Development
Government of India

www.nptel.iitm.ac.in