NPTEL

NPTEL ONLINE CERTIFICATION COURSE

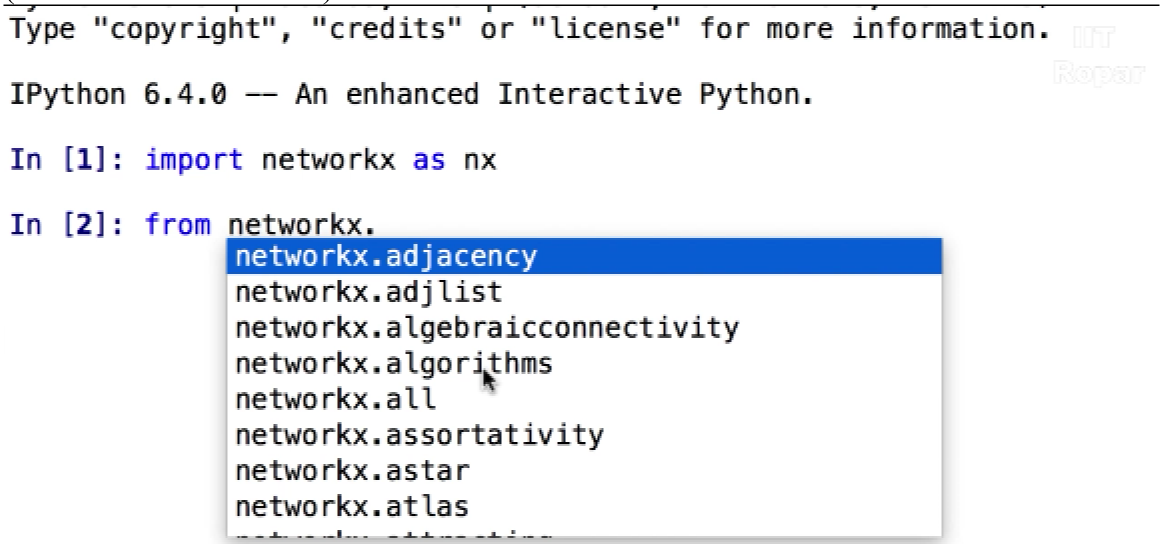
Discrete Mathematics
Graph Theory – 3 &
Generating Functions

NetworkX - Bipartite graphs

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

We are now going to learn more on bipartite graphs in Python, now as an always the first step is importing NetworkX as NX, right, now I'm going to create bipartite graphs, let me tell you how to do that first now bipartite is not present directly inside NetworkX, like how we did for Eulerian and all we cannot do it that way here, so what I'm going to do is from networkx.,

(Refer Slide Time: 00:49)

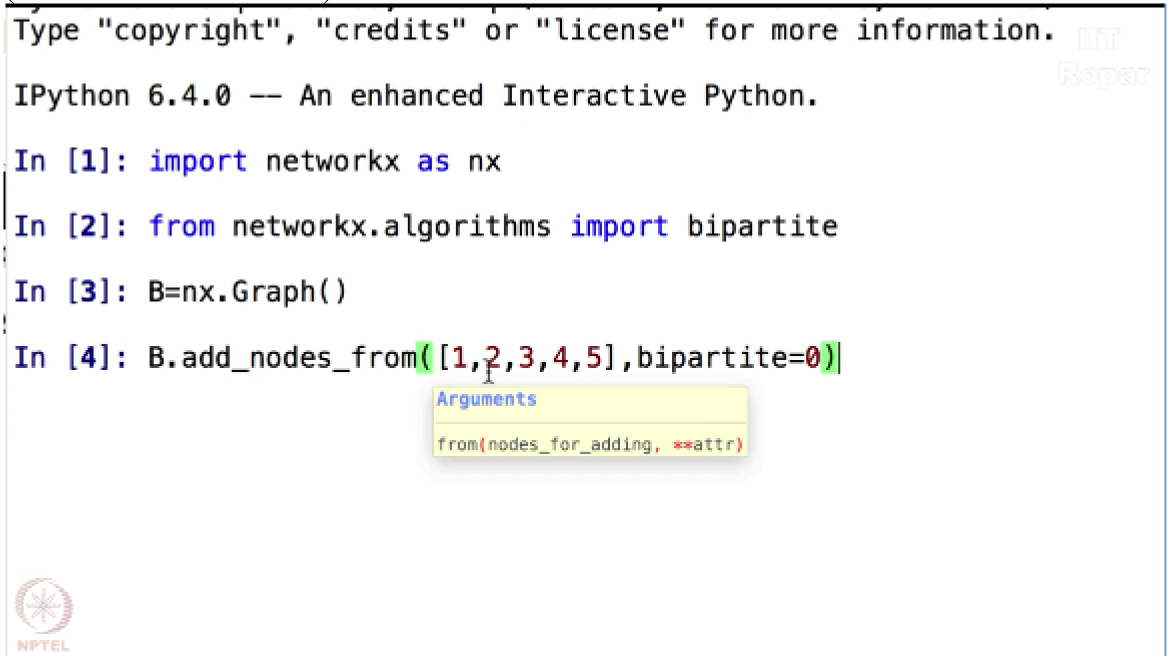


```
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: import networkx as nx

In [2]: from networkx.
```

do you see all these here, these are sub libraries, right, networkx.algorithms I am going to choose this from networkx.algorithms I am going to import bipartite, so in the sub library called algorithms bipartite is present.

Now so bipartite has got loaded, I am going to create this graph B let's say as nx.graph, oh it was caps so let me do it nx.Graph, this way, now I am going to add nodes to this graph from add nodes from this list 1, 2, 3, 4, and 5, right, so this is it, and I'm going to tell that bipartite is 0, so this means that this is the first set or the first partition clear,

(Refer Slide Time: 02:00)

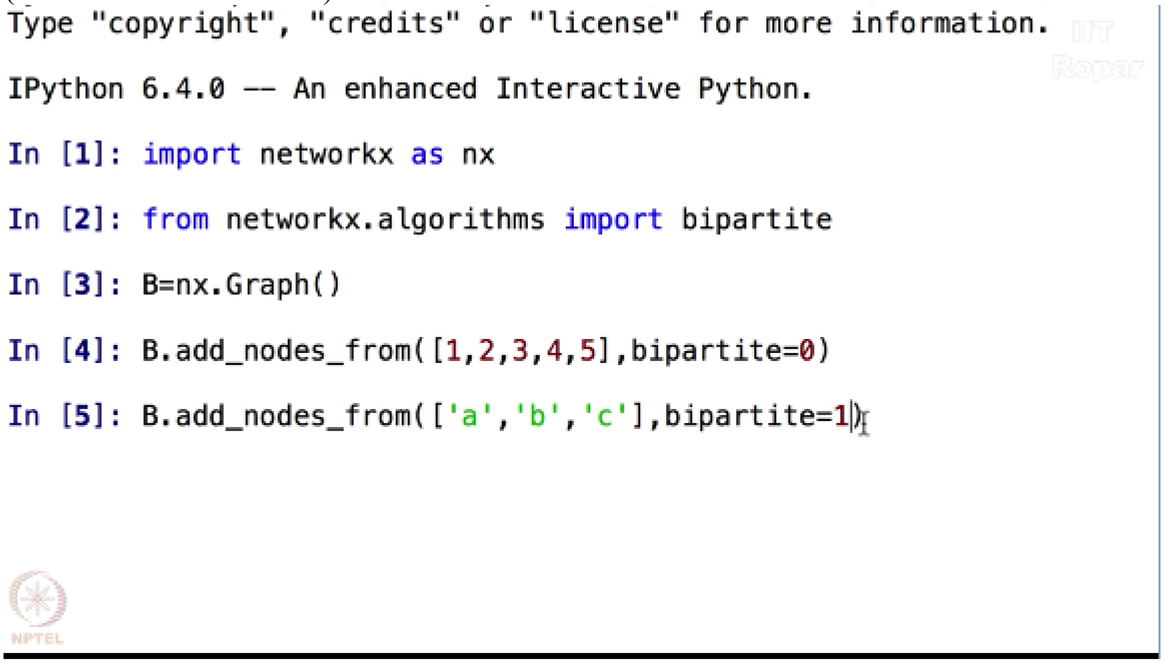


```
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: import networkx as nx

In [2]: from networkx.algorithms import bipartite

In [3]: B=nx.Graph()

In [4]: B.add_nodes_from([1,2,3,4,5],bipartite=0)
```

you see that in bipartite there are 2 sets, right, so this is going to be my first set.

Now I am going to again give B.add_nodes I am going to change this list now, I am going to give it something like this say A, B, say C, so this is going to be my second set, and I am going to name it as bipartite = 1,

(Refer Slide Time: 02:28)

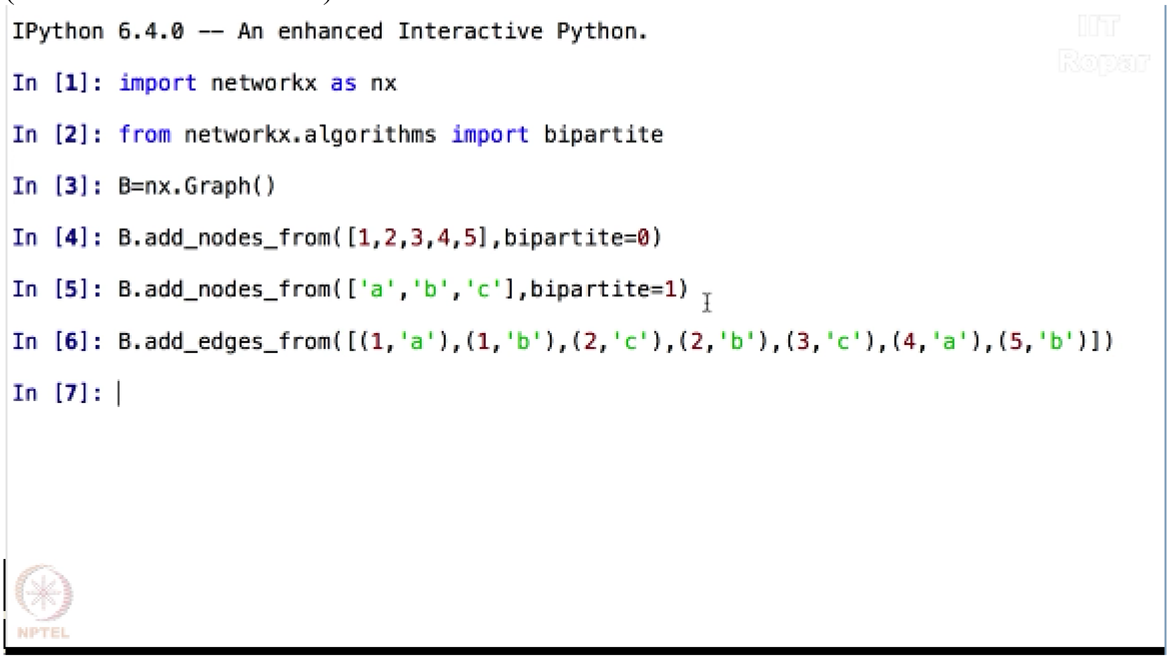


```
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: import networkx as nx

In [2]: from networkx.algorithms import bipartite

In [3]: B=nx.Graph()

In [4]: B.add_nodes_from([1,2,3,4,5],bipartite=0)

In [5]: B.add_nodes_from(['a','b','c'],bipartite=1)
```

so did you see what I just did, I’ve added nodes one set from this list, another set from this list, clear.

Now the next step is adding edges V.add_edges from I'm going to specify a list now again, so it's going to be something like this 1, A and then okay, there is a small trouble here, yeah, one more bracket, 1, E and then let's say 1, B, let's say 2, C okay, and then let's say 2, B and then 3, let's say C itself, and then 4, A, and then 5, B, so these are the edges in my graph,

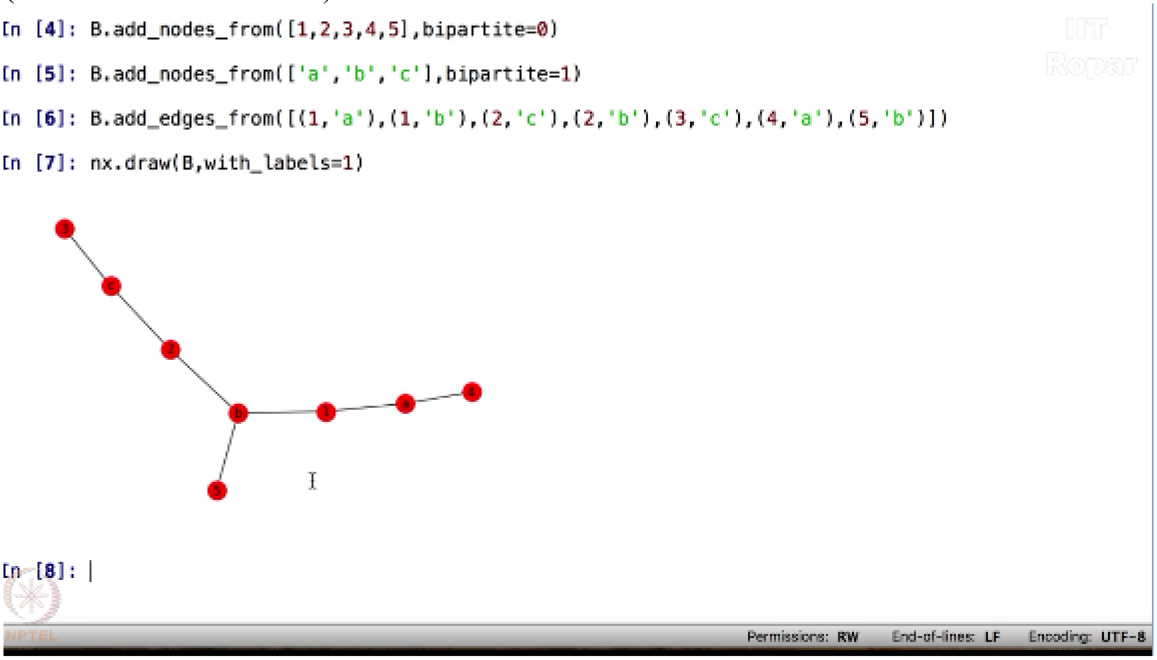
(Refer Slide Time: 03:44)



```
IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: import networkx as nx

In [2]: from networkx.algorithms import bipartite

In [3]: B=nx.Graph()

In [4]: B.add_nodes_from([1,2,3,4,5],bipartite=0)

In [5]: B.add_nodes_from(['a','b','c'],bipartite=1)

In [6]: B.add_edges_from([(1,'a'),(1,'b'),(2,'c'),(2,'b'),(3,'c'),(4,'a'),(5,'b')])

In [7]: |
```

and now I'm going to draw nx.draw(B) okay, we will specify the labels, with labels as true, so do you see the graph here,

(Refer Slide Time: 04:00)

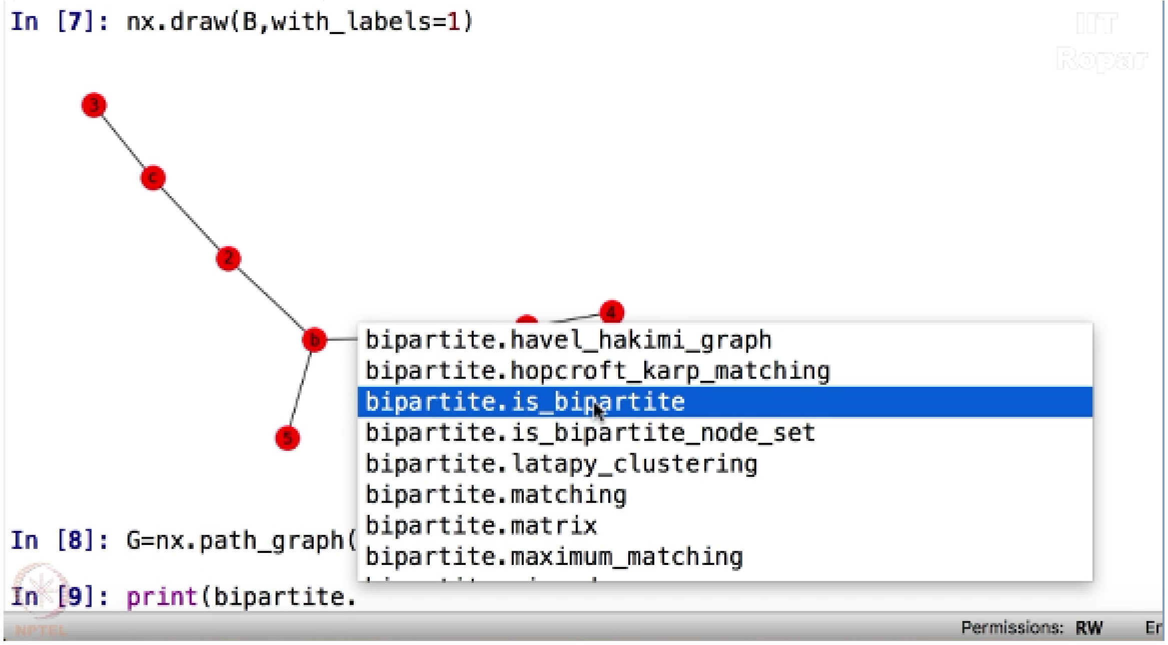


```
In [4]: B.add_nodes_from([1,2,3,4,5],bipartite=0)

In [5]: B.add_nodes_from(['a','b','c'],bipartite=1)

In [6]: B.add_edges_from([(1,'a'),(1,'b'),(2,'c'),(2,'b'),(3,'c'),(4,'a'),(5,'b')])

In [7]: nx.draw(B,with_labels=1)

In [8]: |
```

so this is a bipartite graph with the 2 partitions 1, 2, 3, 4, 5 and A, B, C right, if you probably write down properly as 2 sets like how we have showed in the videos you can verify this, then by that method.

Now let us check another graph let me say we are going to verify if the given graph is bipartite or not, how are we going to do that? Let's say G = nx.path_graph or let me say 4 nodes I have created this, so what am I going to print? Print bipartite., you see we can check various things here

(Refer Slide Time: 04:56)



```
In [7]: nx.draw(B,with_labels=1)

In [8]: G=nx.path_graph(

In [9]: print(bipartite.
```

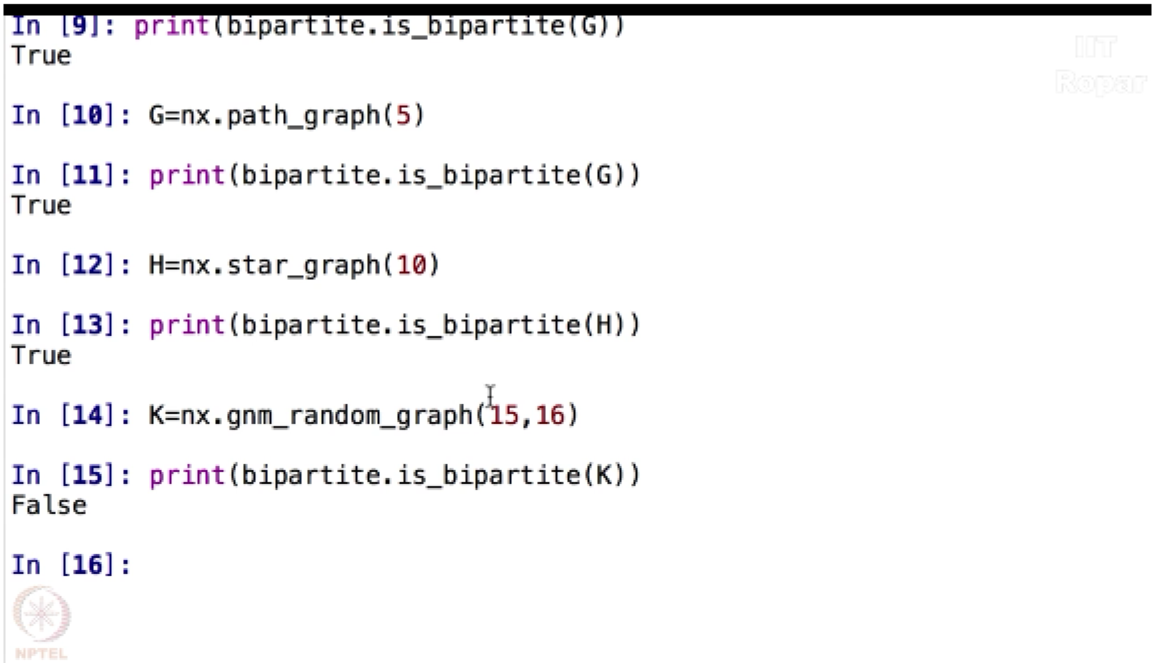
IIT Ropar
3
c
2
b
4
5
bipartite.havel_hakimi_graph
bipartite.hopcroft_karp_matching
bipartite.is_bipartite
bipartite.is_bipartite_node_set
bipartite.latapy_clustering
bipartite.matching
bipartite.matrix
bipartite.maximum_matching
Permissions: RW

so I am going to check is bipartite now, so print bipartite.is_bipartite, so what are we going to check? If this graph G is bipartite or not, it says that G is bipartite, so a path graph on 4 nodes is bipartite.

Now let me say G path graph on let's say 5 vertices, let us check if this is bipartite or let me call it as H here, okay, G itself fine, so let it be G now, this is also bipartite on 5 vertices.

Now let me consider another graph say H = nx.star_graph on let's say 10 nodes, right, let us check, print bipartite is by H it is now, this is again bipartite, so bipartite.is_bipartite is the command where we verify if a graph is bipartite or not, let me take a random graph let me say K = nx.gnm_random_graph on let me say 15 nodes, and say 16 edges right, now let us check if K is bipartite, so how are we going to do it? Print bipartite.is_bipartite(K) it says it is not bipartite.

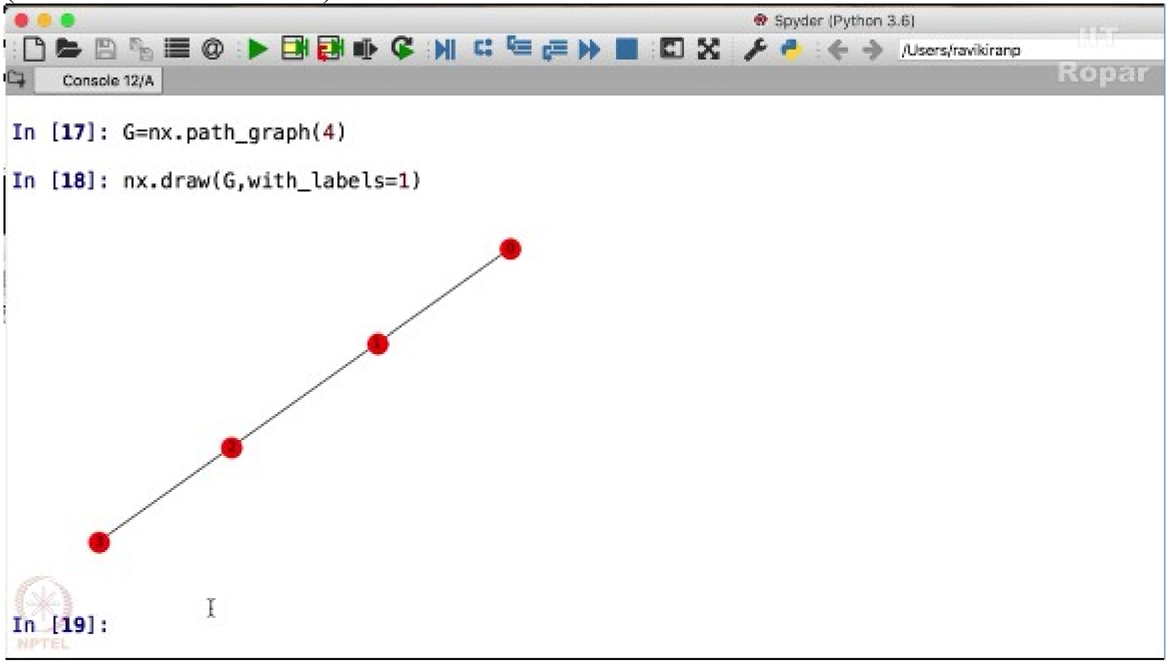
(Refer Slide Time: 06:41)



```
In [9]: print(bipartite.is_bipartite(G))
True

In [10]: G=nx.path_graph(5)

In [11]: print(bipartite.is_bipartite(G))
True

In [12]: H=nx.star_graph(10)

In [13]: print(bipartite.is_bipartite(H))
True

In [14]: K=nx.gnm_random_graph(15,16)

In [15]: print(bipartite.is_bipartite(K))
False

In [16]:
```

Now let me clear the screen, we will now be seeing if we can check if a particular vertex or if true particular vertices are in a set or not, what I mean by that? If a graph is given and if we see that the graph is bipartite then we can check if particular nodes are in a partition or not, in one particular partition you'll be able to understand once I do it.

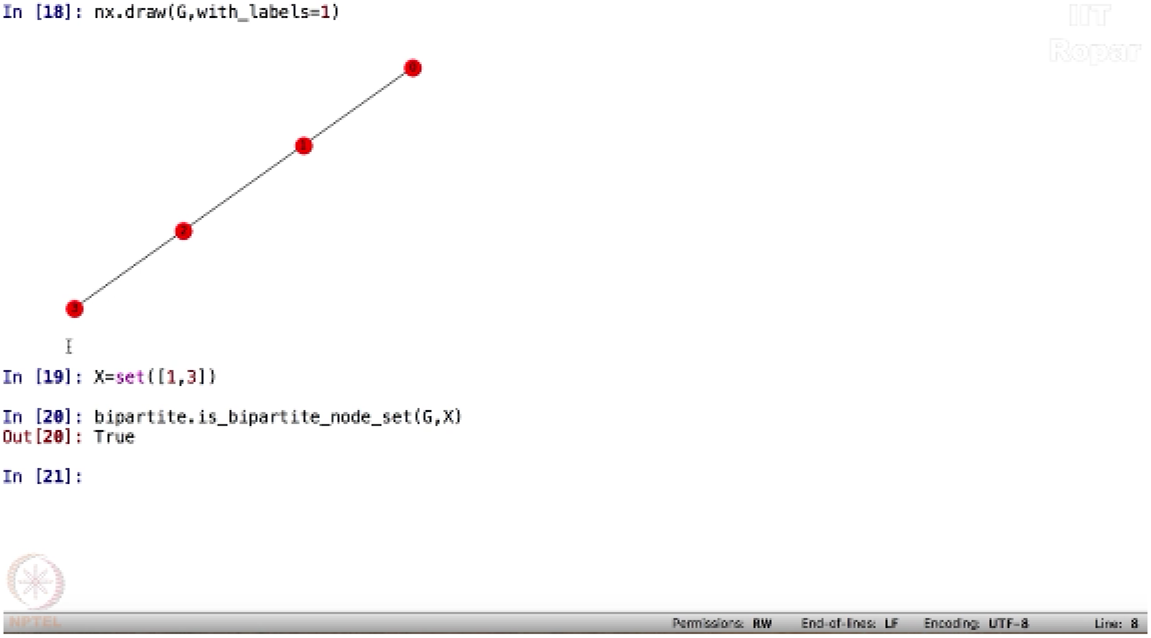
Now let me say create a graph G = nx.path_graph on let's say 4 vertices, so the graph has got created, now let me draw that nx.draw(G, with_labels) as true, so this is the graph,
(Refer Slide Time: 07:41)



```
In [17]: G=nx.path_graph(4)

In [18]: nx.draw(G,with_labels=1)

In [19]:
```

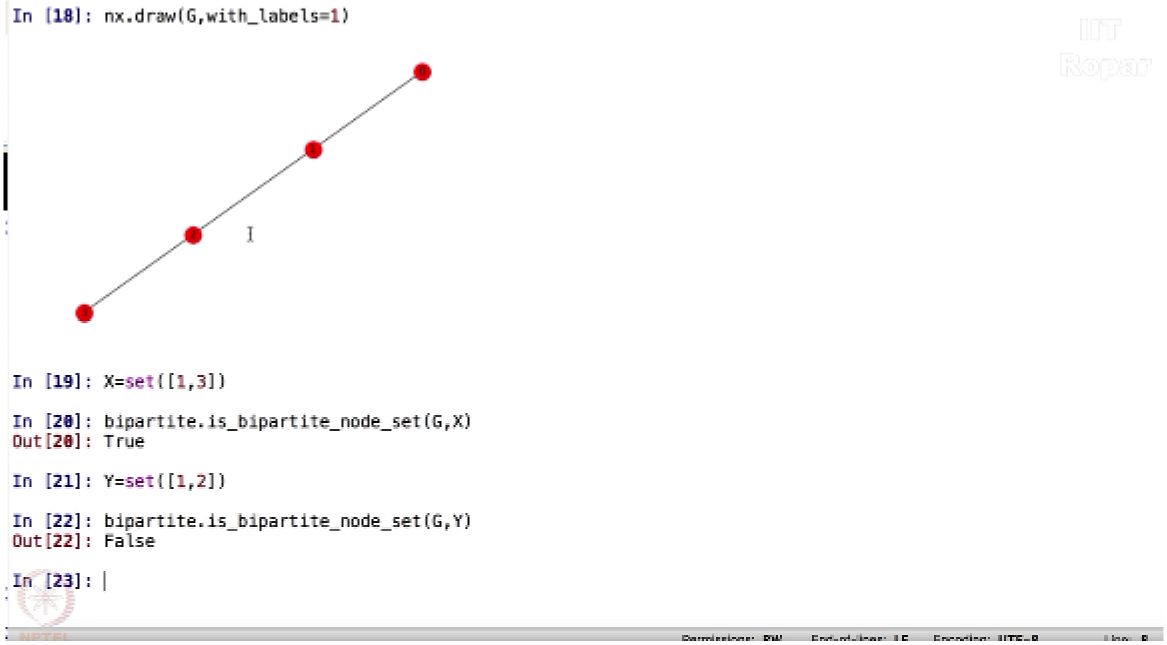
now observe what I'm going to do, let me take this set X as set 1, 3, if you observe here 1 and 3, these two vertices are not connected hence they will fall into one part I had said, right, so we are going to verify that now bipartite.is_bipartite_node_set, and of which graph? This graph G

and the set X, so now this is going to give me the answer if these two nodes 1 and 3 are in a bipartite set or not, see it says true, so 1 and 3 fall into one partite set.
(Refer Slide Time: 08:40)



```
In [18]: nx.draw(G,with_labels=1)

In [19]: X=set([1,3])

In [20]: bipartite.is_bipartite_node_set(G,X)
Out[20]: True

In [21]:
```

Now I'm going to go ahead and create this set, let me say 1, 2, okay and I'm going to check if this set forms a partite set or not, G, Y, see it says false because 1 and 2 are connected and hence these two cannot fall into the same partite set,
(Refer Slide Time: 09:09)



```
In [18]: nx.draw(G,with_labels=1)

In [19]: X=set([1,3])

In [20]: bipartite.is_bipartite_node_set(G,X)
Out[20]: True

In [21]: Y=set([1,2])

In [22]: bipartite.is_bipartite_node_set(G,Y)
Out[22]: False

In [23]: |
```

I hope with this it was clear to you how we create a bipartite graph, how we can verify if a graph is bipartite or not, and how to check if particular vertices fall into one partite set or not.

IIT MADRAS PRODUCTION

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

www.nptel.iitm.ac.in