Coding Assignment-2 (Outputs)

Computational Methods & Applications

Name :- Zahir Khan. Roll no:- 112202010 -MTech CaM.

....

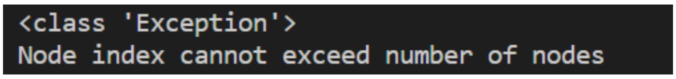
Note : Explanations of the codes are written in comments with the codes. This document is only for outputs.

Q.1) Output:

g = UndirectedGraph(10)

g.addNode(11)

Result:

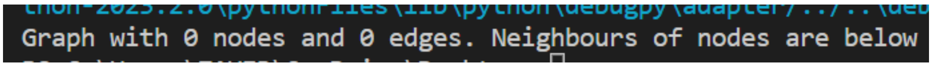


```
<class 'Exception'>
Node index cannot exceed number of nodes
```

g = UndirectedGraph()

print(g)

Result:

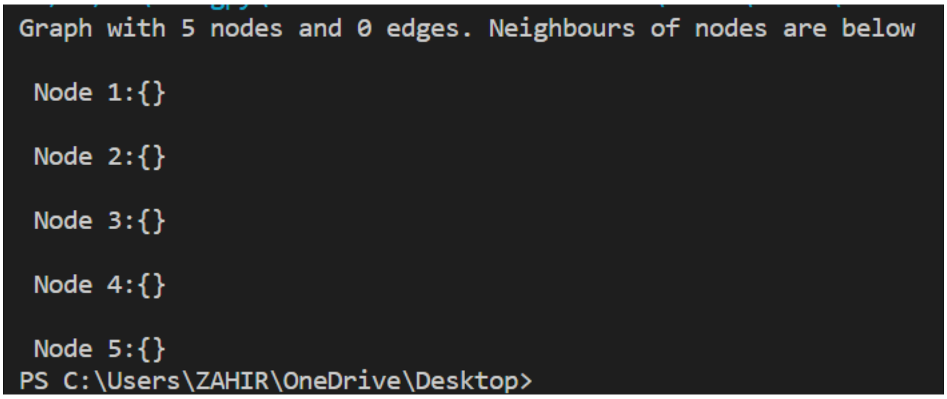


```
Graph with 0 nodes and 0 edges. Neighbours of nodes are below
```

g = UndirectedGraph(5)

print(g)

Result:



```
Graph with 5 nodes and 0 edges. Neighbours of nodes are below

 Node 1:{}

 Node 2:{}

 Node 3:{}

 Node 4:{}

 Node 5:{}
PS C:\Users\ZAHIR\OneDrive\Desktop>
```

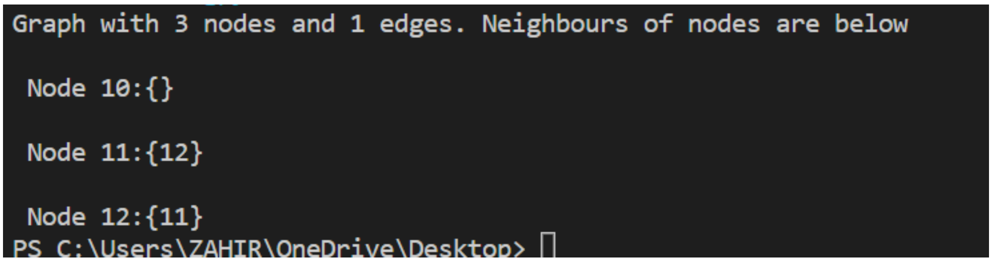
= UndirectedGraph()

g = g + 10 g

g = g + (11, 12)

```
print(g)
```

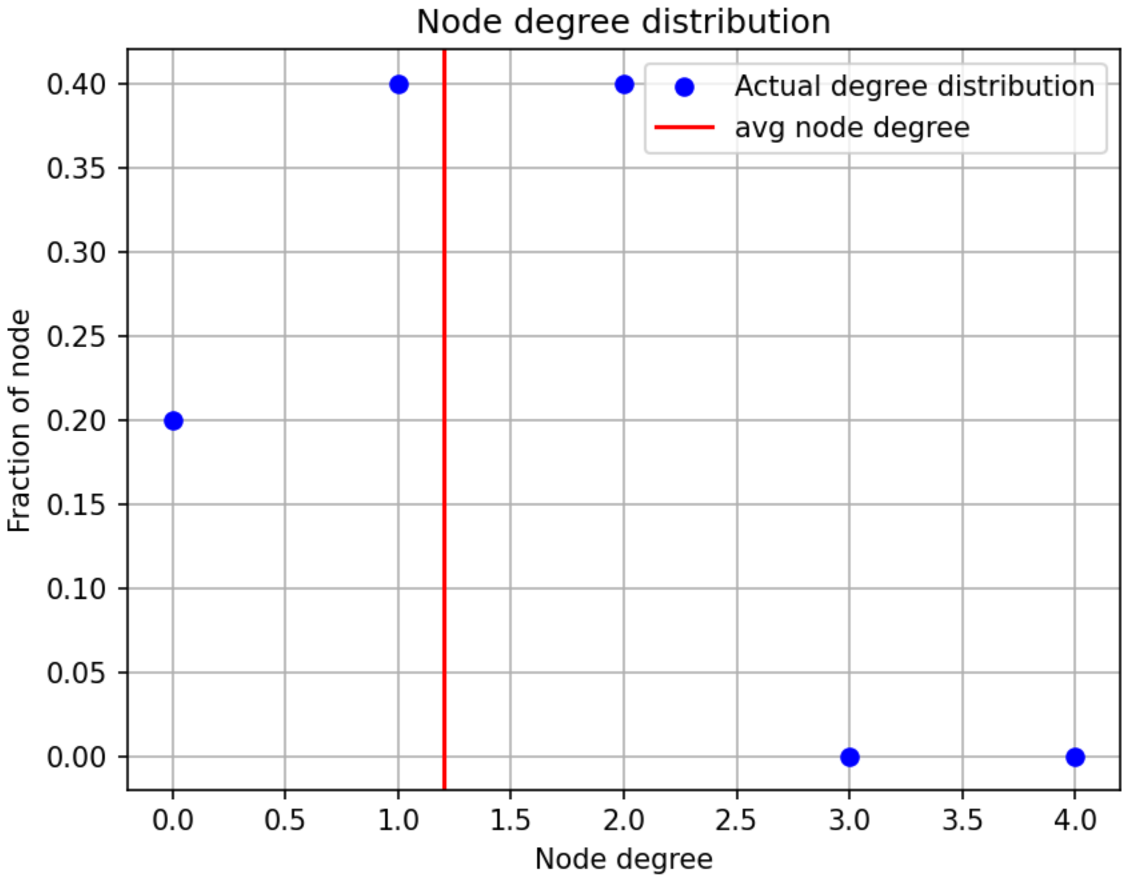
Result:



```
Graph with 3 nodes and 1 edges. Neighbours of nodes are below

 Node 10:{}

 Node 11:{12}

 Node 12:{11}
PS C:\Users\ZAHIR\OneDrive\Desktop>
```

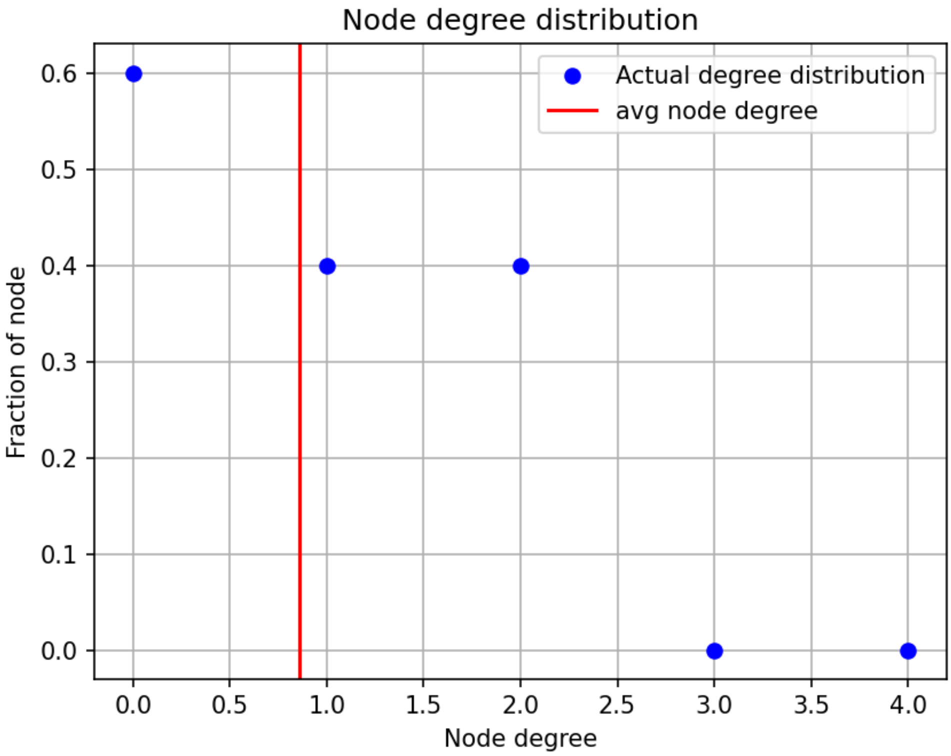
```
g = UndirectedGraph(5)

g = g + (1, 2)

g = g + (3, 4)

g = g + (1, 4)

g.plotDegDist()
```

Result:



```
g = UndirectedGraph()
g = g + 100 g = g + (1, 2)
g = g + (1, 100)
g = g + (100, 3)
g = g + 20
g.plotDegDist()
```

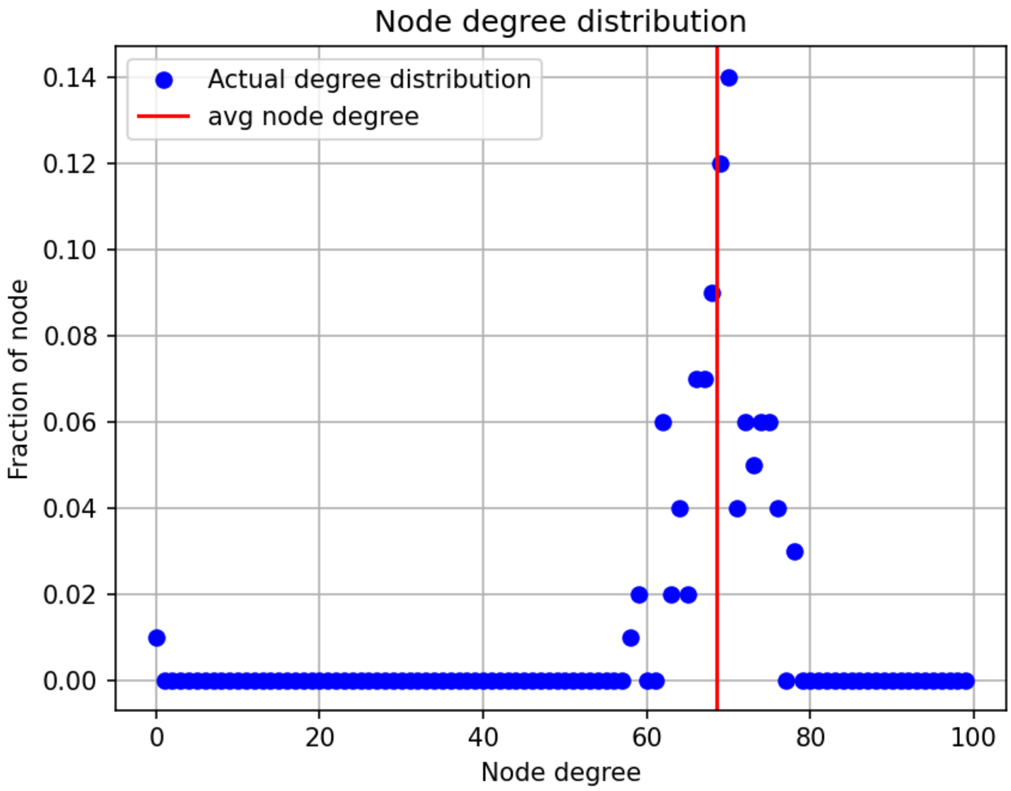
Result:



Q.2) Output:

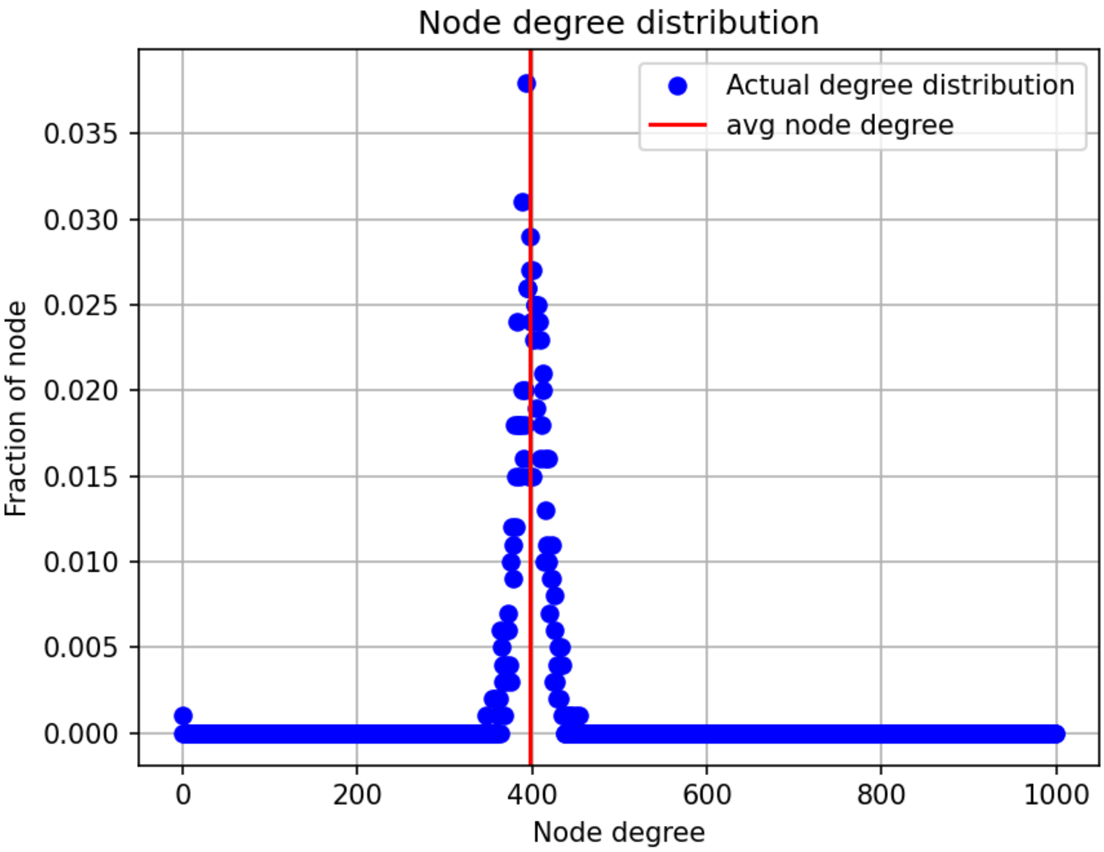
```
# The following code creates a G(100, 0.7) random graph and
 # plots its degree distribution
g = ERRandomGraph(100)
g.sample(0.7)
 g.plotDegDist()
```

Result:



```
# The following code creates a G(1000, 0.4) random graph and
 # plots its degree distribution
g = EERRandomGraph(1000)
 g.sample(0.4)
 g.plotDegDist()
```

Result:

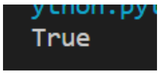


....

Q.3) Output:

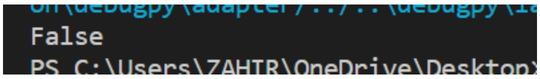
```
g = UndirectedGraph(5)
 g = g + (1, 2)
 g = g + (2, 3)
 g = g + (3, 4)
 g = g + (3, 5)
print(g.isConnected())
```

Result:



```
True
```

```
g = UndirectedGraph(5)
g = g + (1, 2)
 g = g + (2, 3)
 g = g + (3, 5)
print(g.isConnected())
```

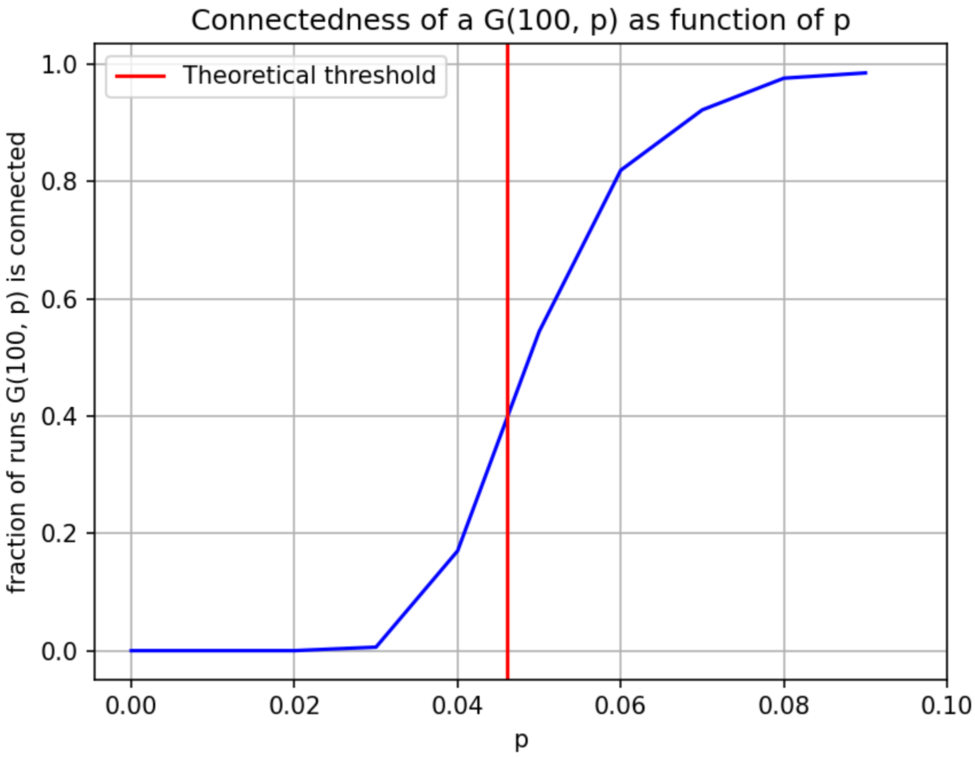
Result: 

```
on\debugpy\adapter/../..\debugpy\la
False
PS C:\Users\ZAHIR\OneDrive\Desktop>
```

```
g = UndirectedGraph(5)
g = g + (1, 2)
g = g + (2, 3)
 g = g + (3, 5)
print(g.isConnected())
print(g)
```

Result:

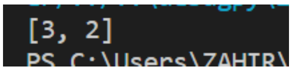
> *“Erd˝os-R´enyi random graph G(100, p) is almost surely connected only if p > ln 100 100 .”*



Q.4) Output:

```
g = UndirectedGraph(6)
g = g + (1, 2)
g = g + (3, 4)
g = g + (6, 4)
print(g.oneTwoComponentSizes())
```

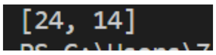
Result:



```
[3, 2]
PS C:\Users\ZAHIR\
```

```
g = RandomGraph(100)
g.sample(0.01)
print(g.oneTwoComponentSizes())
```

Result:



```
[24, 14]
PS C:\Users\Z
```
