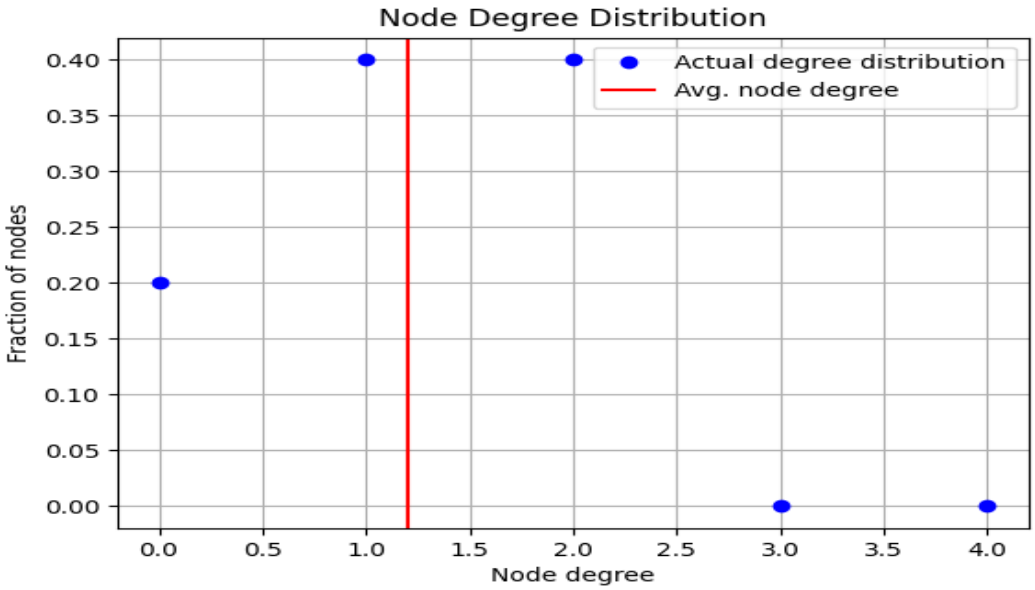
Coding Assignment 2:

Networks, Random Graphs and Percolation

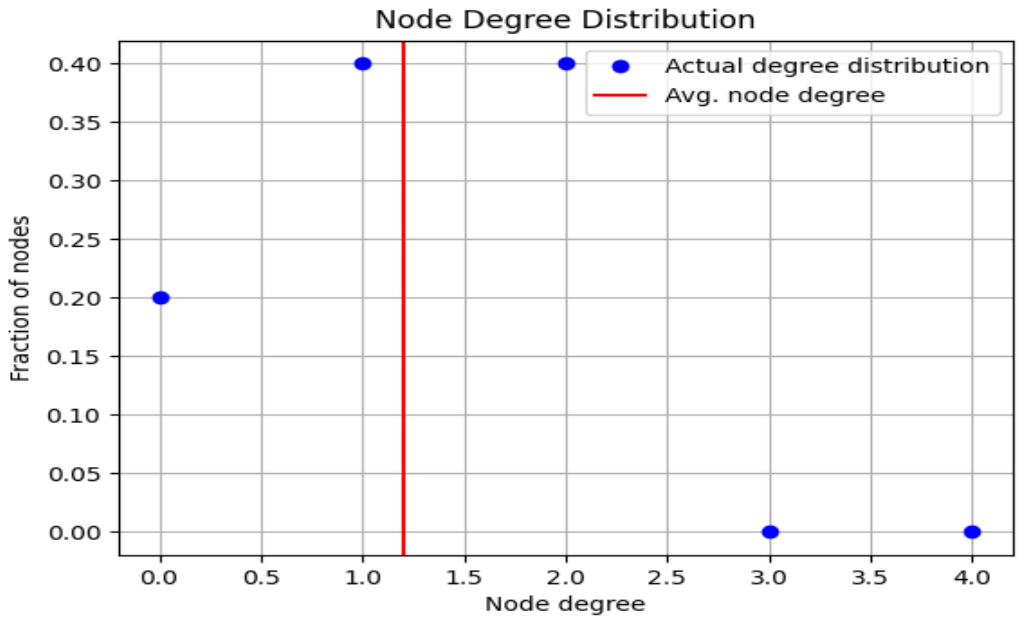
-Yukta Salunkhe
-112001052

Q1.

TestCase 4:

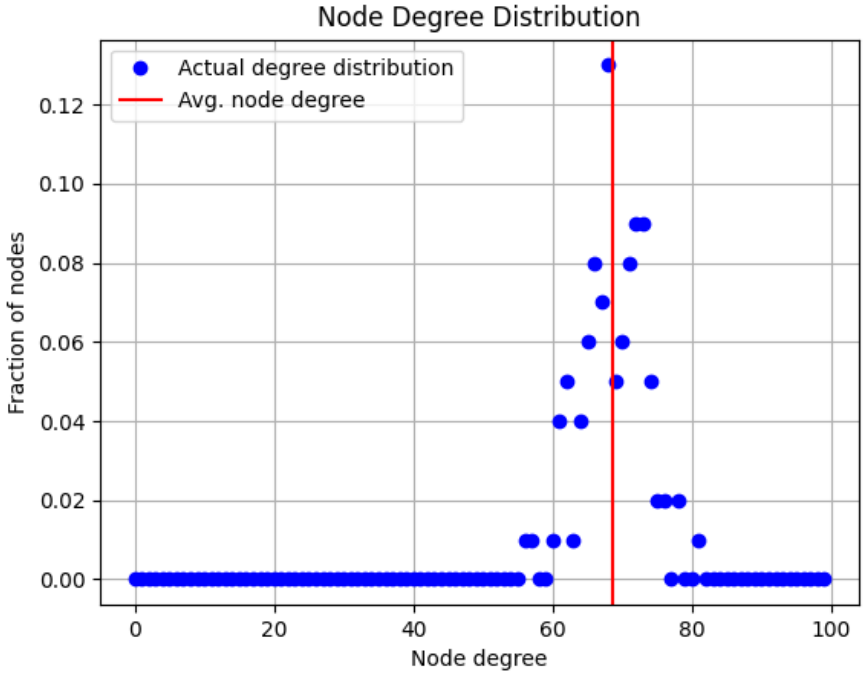


TestCase 5:

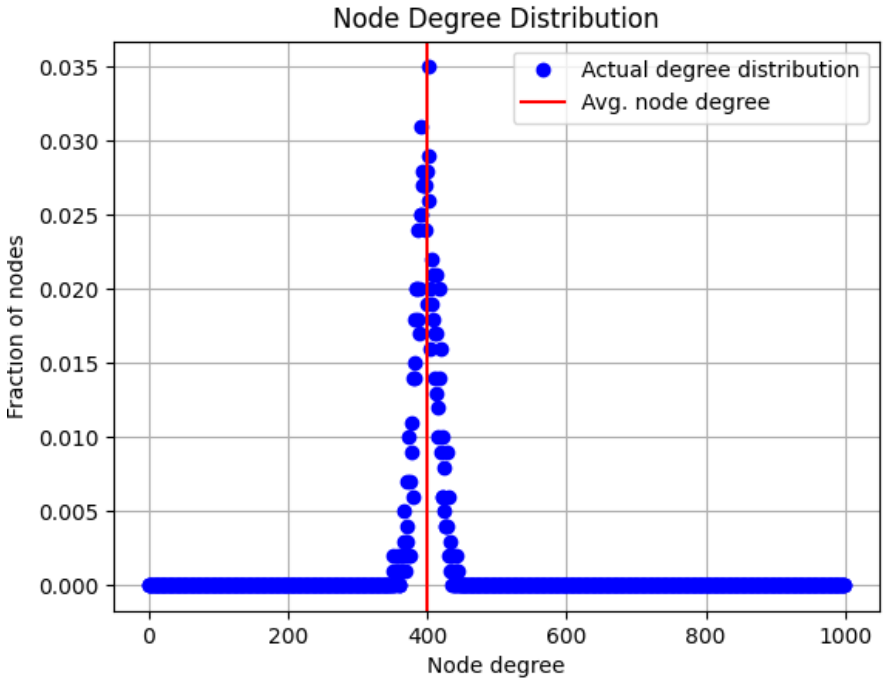


Q2.

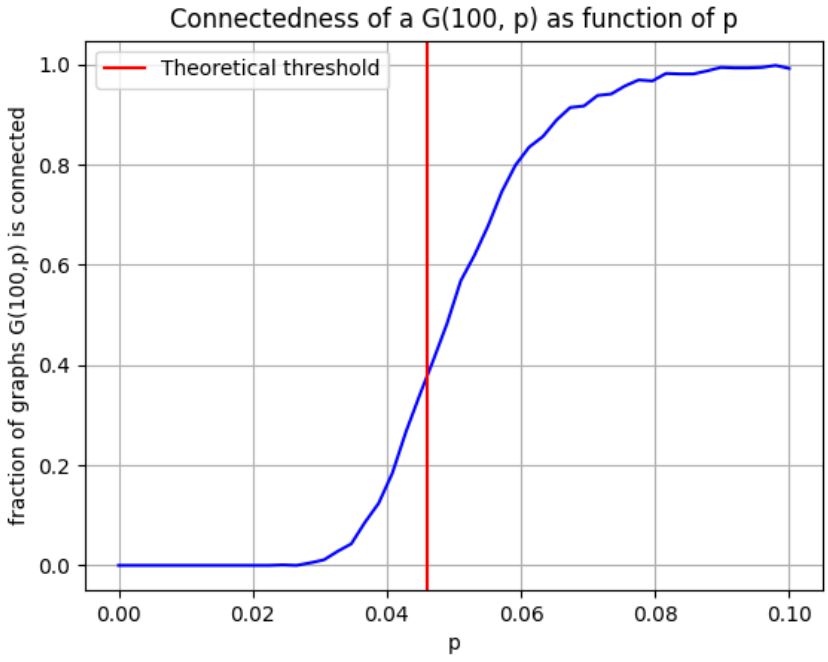
Test case 1:



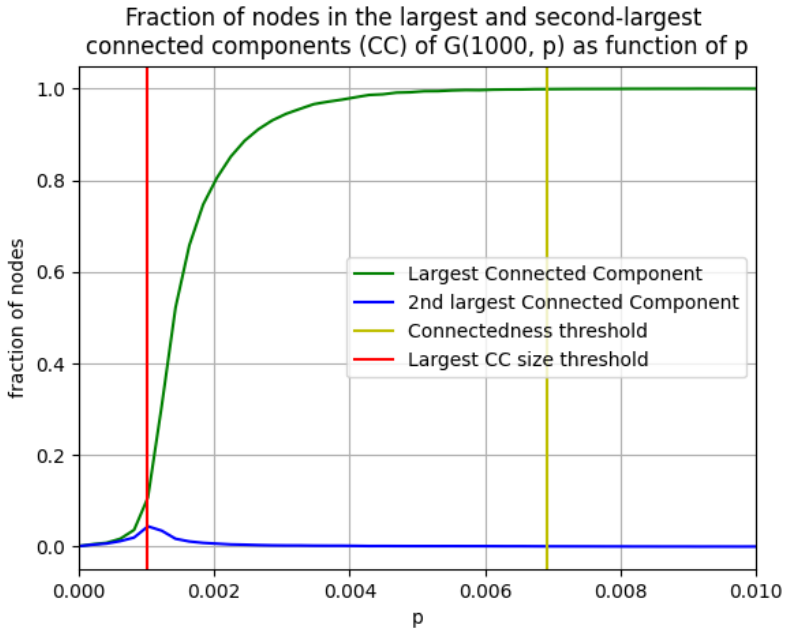
Test Case 2:


Node Degree Distribution
Actual degree distribution
Avg. node degree
Fraction of nodes
Node degree
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
200
400
600
800
1000

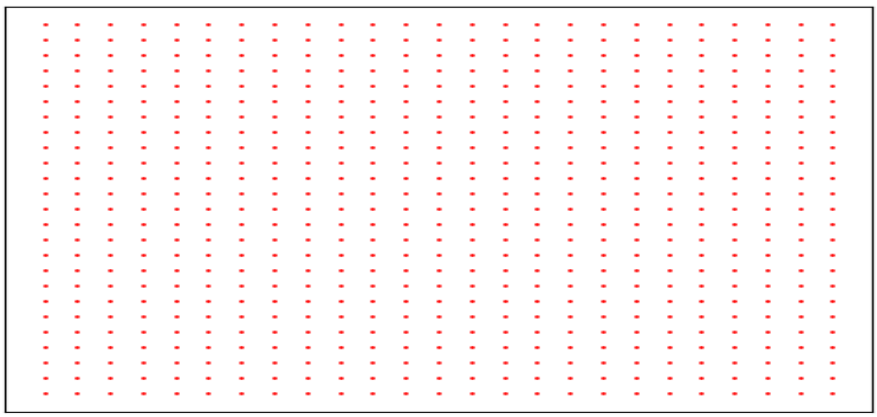
Q3.


Connectedness of a G(100, p) as function of p
Theoretical threshold
fraction of graphs G(100,p) is connected
p
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.02
0.04
0.06
0.08
0.10

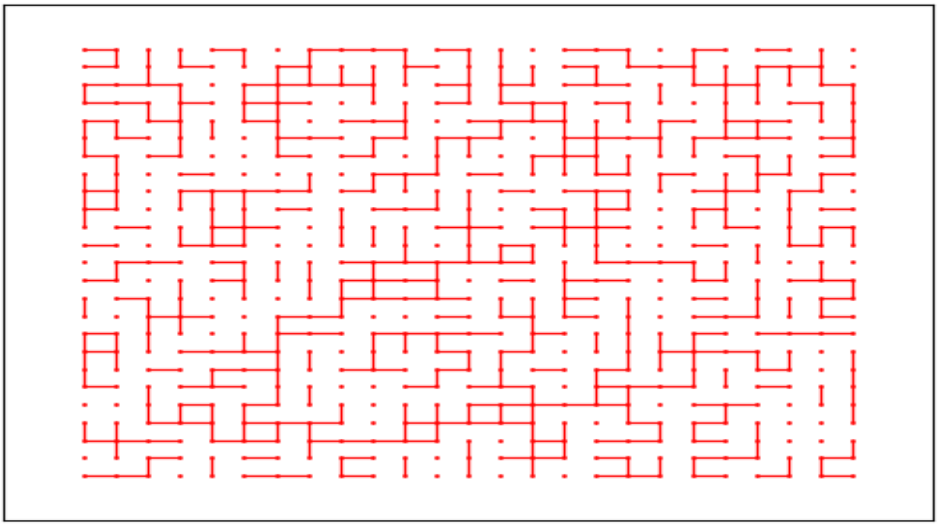
Q4.



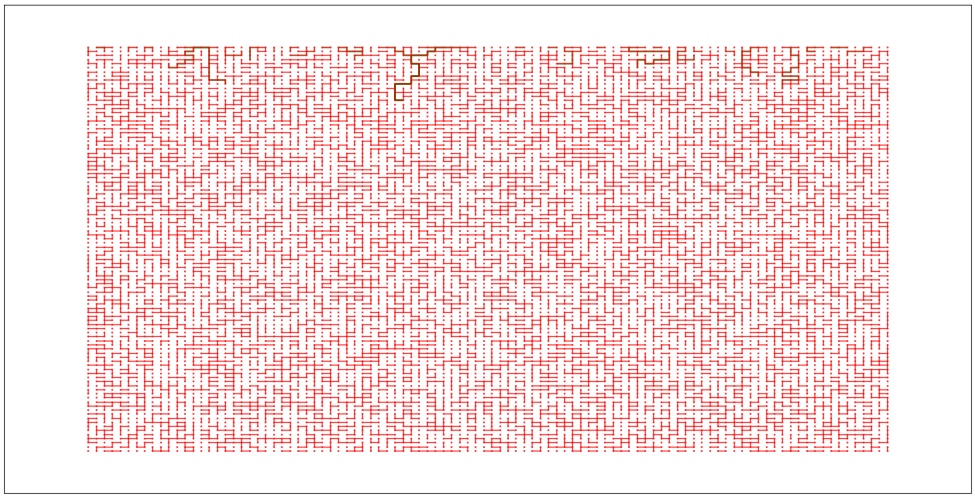
Q5. Test Case 1:



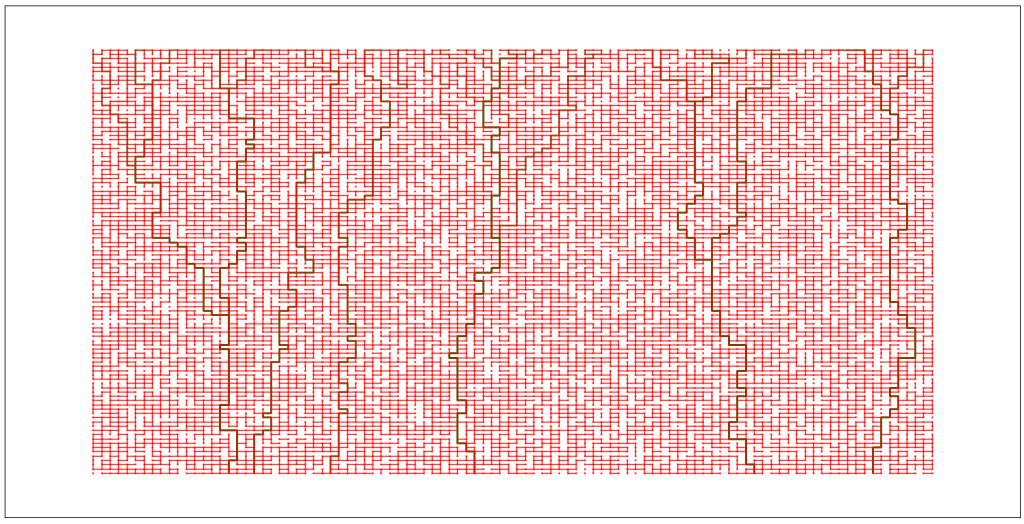
Test Case 2:



Test Case 4:



Test Case 5:



-> for 5th - showPaths question we can also use has_path inbuilt method of networkx package for showing path. For this we will have to iterate over all nodes of rows. From each start vertex get the path to the farthest node possible. Loop over till we find paths to the farthest node from the current start node. And then find the shortest path between the source and the farthest node possible.
-> I have implemented it using the queue and networkx packages. This is more optimized method and thus output takes lesser time

Q6.