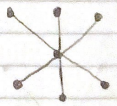
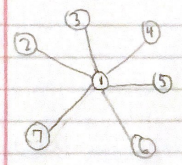
Question 1

a) Draw a star graph with N=7



b) let's call the central node as Node 1 and the rest of nodes as Node 2, 3, ... N. Show the above star graph's Adjacency Matrix

N=7 star Graph



adjacency Matrix for N=7 star Graph

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

c) Derive the Laplacian of this Graph.

$L = D - A$

D = Degree matrix, A = Aagacency Matrix, L = Graph laplacian

Degree Matrix

$$\begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

D - A = Laplacian Graph

$$\begin{bmatrix} 6 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

1) D.

```
"""d) Use your favorite programming language and write a code to compute the graph Laplacian based
on your answer of part b. [Hint: yes, this is about matrix subtraction. Do not use any software package that will directly generate the Laplacian for you.
"""
n = 7

A = [[0 for _ in range(n)] for _ in range(n)]


for i in range(1, n):
    A[i][0] = 1
    A[0][i] = 1

print("Adjacency Matrix A:")
for row in A:
    print(row)

D = [[0 for _ in range(n)] for _ in range(n)]

for i in range(n):
    D[i][i] = sum(A[i])

print("\nDegree Matrix D:")
for row in D:
    print(row)


L = [[0 for _ in range(n)] for _ in range(n)]

for i in range(n):
    for j in range(n):
        L[i][j] = D[i][j] - A[i][j]

print("\nLaplacian Matrix L:")
for row in L:
    print(row)
```

```
(base) brandontiong@brandons-MacBook-Air-2 pythonTextBookEx % python -u "/Users/
Adjacency Matrix A:
[0, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0]

Degree Matrix D:
[6, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 1]

Laplacian Matrix L:
[6, -1, -1, -1, -1, -1, -1]
[-1, 1, 0, 0, 0, 0, 0]
[-1, 0, 1, 0, 0, 0, 0]
[-1, 0, 0, 1, 0, 0, 0]
[-1, 0, 0, 0, 1, 0, 0]
[-1, 0, 0, 0, 0, 1, 0]
[-1, 0, 0, 0, 0, 0, 1]
(base) brandontiong@brandons-MacBook-Air-2 pythonTextBookEx %
```

2. Repeat Question 1 with N=100 and randomly add exactly w=10 more edges among the leave nodes.Obviously, you are not supposed to draw this graph by hand in part a) of this question, or even preassign the adjacency matrix as you might have done in Question 1.

Python code:

```
import random

N = 100
w = 10
total_nodes = N

A = [[0 for _ in range(total_nodes)] for _ in range(total_nodes)]

for i in range(1, N ):
    A[i][0] = 1
    A[0][i] = 1

edges_added = 0
while edges_added < w:
    i, j = random.randint(1, N), random.randint(1, N)


    if i != j and A[i][j] == 0:
        A[i][j] = 1
        A[j][i] = 1
        edges_added += 1


D = [[0 for _ in range(total_nodes)] for _ in range(total_nodes)]
for i in range(total_nodes):
    D[i][i] = sum(A[i])

# Compute Laplacian Matrix L (L = D - A)
L = [[0 for _ in range(total_nodes)] for _ in range(total_nodes)]

for i in range(total_nodes):
    for j in range(total_nodes):
        L[i][j] = D[i][j] - A[i][j]

#Print matrices (This part is optional, useful for verification)
print("Adjacency Matrix A:")
for row in A:
    print(row)

print("\nDegree Matrix D:")
for row in D:
    print(row)

print("\nLaplacian Matrix L:")
for row in L:
    print(row)
```

3. Write a program to compute the degree of each node in Question 2 and the average node degree of the entire graph. Again, please do not use any software package that allows you to call a function and compute node degrees directly. Write your own few lines of codes to accomplish this.

Python code:

```
import random

total_nodes = 100
additional_edges = 10


adj_matrix = [[0] * total_nodes for _ in range(total_nodes)]

for i in range(1, total_nodes):
    adj_matrix[i][0] = 1
    adj_matrix[0][i] = 1

edges_added = 0
while edges_added < additional_edges:
    node1, node2 = random.randint(1, total_nodes-1), random.randint(1, total_nodes-1)


    if node1 != node2 and adj_matrix[node1][node2] == 0:
        adj_matrix[node1][node2] = 1
        adj_matrix[node2][node1] = 1
        edges_added += 1


node_degrees = [sum(row) for row in adj_matrix]


average_degree = sum(node_degrees) / total_nodes

print(f"Node Degrees: {node_degrees}")
print(f"Average Node Degree: {average_degree}")
```

```
Users/brandontiong/Documents/pythonTextBookEx/AverageNod
eDegree.py"
Node Degrees: [99, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 3, 1, 2, 3, 1, 1,
 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1
, 1, 1, 1, 1, 2, 1]
Average Node Degree: 2.18
(base) brandontiong@brandons-MacBook-Air-2 pythonTextBookEx %
```