function [ ans ] = disto( m )

%The function takes an input matrix and outputs the standard normal

%distrobution values of the lambda inputs. It also outputs the mean of this

%distrobution and the probability that it is manmade.

lambda = eig(m)

e = 2.718281828;

p = 3.141592653589793238;

part1 = (1/((3\*e)/(4))\*(sqrt(2\*p)));

part2 = e.^((-(lambda.^2))/((4.\*(e.^2))/(8)));

ans = part1 \* part2;

avg\_dis = mean(ans)

prob\_man = (4.\*e - avg\_dis)/(4.\*e)

end

Adjacency Matrix of ARPANET

0 1 0 0 0 1 0 0 0 0 0 0 0

1 0 1 0 0 0 1 0 0 0 0 0 0

0 1 0 1 0 0 0 0 0 0 0 1 0

0 0 1 0 1 0 0 0 0 0 0 0 0

0 0 0 1 0 1 0 0 0 0 0 0 0

1 0 0 0 1 0 0 0 0 0 0 0 0

0 1 0 0 0 0 0 1 0 0 0 0 1

0 0 0 0 0 0 1 0 1 1 1 0 0

0 0 0 0 0 0 0 1 0 1 0 0 0

0 0 0 0 0 0 0 1 1 0 1 1 0

0 0 0 0 0 0 0 1 0 1 0 0 0

0 0 1 0 0 0 0 0 0 1 0 0 1

0 0 0 0 0 0 1 0 0 0 0 1 0

Degree Matrix of ARPANET

2 0 0 0 0 0 0 0 0 0 0 0 0

0 3 0 0 0 0 0 0 0 0 0 0 0

0 0 3 0 0 0 0 0 0 0 0 0 0

0 0 0 2 0 0 0 0 0 0 0 0 0

0 0 0 0 2 0 0 0 0 0 0 0 0

0 0 0 0 0 2 0 0 0 0 0 0 0

0 0 0 0 0 0 3 0 0 0 0 0 0

0 0 0 0 0 0 0 4 0 0 0 0 0

0 0 0 0 0 0 0 0 2 0 0 0 0

0 0 0 0 0 0 0 0 0 4 0 0 0

0 0 0 0 0 0 0 0 0 0 2 0 0

0 0 0 0 0 0 0 0 0 0 0 3 0

0 0 0 0 0 0 0 0 0 0 0 0 2

Laplacian Matrix of ARPANET

2 -1 0 0 0 -1 0 0 0 0 0 0 0

-1 3 -1 0 0 0 -1 0 0 0 0 0 0

0 -1 3 -1 0 0 0 0 0 0 0 -1 0

0 0 -1 2 -1 0 0 0 0 0 0 0 0

0 0 0 -1 2 -1 0 0 0 0 0 0 0

-1 0 0 0 -1 2 0 0 0 0 0 0 0

0 -1 0 0 0 0 3 -1 0 0 0 0 -1

0 0 0 0 0 0 -1 4 -1 -1 -1 0 0

0 0 0 0 0 0 0 -1 2 -1 0 0 0

0 0 0 0 0 0 0 -1 -1 4 -1 -1 0

0 0 0 0 0 0 0 -1 0 -1 2 0 0

0 0 -1 0 0 0 0 0 0 -1 0 3 -1

0 0 0 0 0 0 -1 0 0 0 0 -1 2

disto(ARPANET\_lap)

avg\_dis = 0.4465

prob\_man = 0.9589

ans =

1.2295

1.2048

0.9788

0.9076

0.5785

0.4164

0.3209

0.0965

0.0479

0.0188

0.0032

0.0010

0.0003

-----------------------------------------------------------------------------

function [ rand\_lap ] = randZeros( n )

%Computers a random Laplacian Matrix and displays the corresponding

%adjacency matrix and degree matrix.

%Creates the random adjancency matrix, M.

M = round(rand(n));

for i = 1:n

M(i,i) = 0;

end

for i = 1:n

for j = 1:n

M(i,j) = M(j,i);

end

end

%Assigns adjacency matrix M to the variable rand\_adj and displays it as an

%output.

rand\_adj = M

%Creates the values that will be in the random degree matrix and the

%inserts them into the random degree matrix.

deg\_val = sum(M);

rand\_deg = zeros(n);

for i = 1:n

rand\_deg(i,i) = deg\_val(i);

end

%Displays random degree matrix.

rand\_deg = rand\_deg

%Computes the random Laplacian Matrix and outputs it as 'ans'.

rand\_lap = rand\_deg - rand\_adj;

end