In [1]:
```
import random
import numpy as np
import matplotlib.pyplot as plt
```

The Barabási-Albert model for generation of scale-free networks

The two major drawbacks of the Erdős-Rényi model for creating random graphs are:

1. It assumes that the size of the network is constant, but in real world network the size of the network keeps growing continuously. For example, the WWW started with a single node, but each and every day new nodes are being added to the network and today WWW contains millions of nodes.
2. Nodes in random networks randomly choose their interaction nodes but, in real world, it happens that a newly added node tends to connect with the nodes having higher degree nodes or to more connected nodes. For example, if any new paper is published on a particular domain then it is highly likely that it will have citation of the most popular paper of that particular domain. This is called preferential attachment.

So, to overcome the drawbacks of growth and preferential attachment, Barabasi came up with a model that generates scale free network which behaves like real world network. In this question, I have implemented the model from scratch.No libraries has been used for implementing the Barabasi model.

In [2]:
```
#https://www.geeksforgeeks.org/building-an-undirected-graph-and-finding-shortest-path-using-dictionaries-in-python/
#Below function calculates shortest path between source and destination nodes
 using BFS
def BFS_for_finding_Shortest_path(source, destination,shortest_paths,adjacency
_list):
    visited = []
    #storing source node in a queue
    queue = [[source]]

    if source == destination:
        shortest_paths.append(0)
        return
    #checking while queue is not empty
    while queue:
        visited_path = queue.pop(0)
        vertex = visited_path[-1]
        if vertex not in visited:
            neighbouring_nodes = adjacency_list[vertex]

            for neighbour in neighbouring_nodes:
                new_visited_path = list(visited_path)
                new_visited_path.append(neighbour)
                queue.append(new_visited_path)

                # Condition to check if the neighbour node is the destination
 node
                if neighbour == destination:
                    shortest_paths.append(len(new_visited_path)-1)
                    return
            visited.append(vertex)
    print("Path does not exists")
    return
```

The below function calculates average clustering coefficient of network. Calculating clustering coefficient of a node n = (number of edges between neighbors of the node n)/ (maximum possible edges between neighbors of node n).

In [3]:
```
#calculating average clustering coefficients
def Clustering_Coefficient(adjacency_list,edge_list,n):
    clustering_coeff=[]
    for i in adjacency_list:
        neighbors=adjacency_list[i]
        count=0
        n_=len(neighbors)
        max_neigh_edges=(n_*(n_-1))/2
#     print(max_neigh_edges)
        for n1 in range(len(neighbors)):
            for n2 in range(n1+1,len(neighbors)):
                if (neighbors[n1],neighbors[n2]) in edge_list or (neighbors[n2
],neighbors[n1]) in edge_list:
                    count+=1
        if max_neigh_edges!=0:
            coeff=count/max_neigh_edges
            clustering_coeff.append(coeff)
    avg_cc=sum(clustering_coeff)/n
    return avg_cc
```

Implementing Barabasi-Albert Model

The following steps has been followed for each of the 100 instances:

An initial random graph has been generated with 5 nodes and 10 edges such that the graph is connected. In other words the degree of each node must be atleast 1.

A matrix is created for initial graph which is named init_graph and if nodes i and j are connected then init_graph[i][j] and init_graph[j][i] is marked as 1.

Now at each evolution step, one node and m edges(I have taken m=4) are added to the existing network till the total number of nodes reach a particular value(here value=100).

For each addition of node:

4.1 Compute the degree of all the pre exisiting nodes and calculate the probability Π(k) that a link of the new node connects to node i depends on the degree ki as $\Pi(ki)=ki/\sum jkj$

where ki is the degree of ith node and summation of Kj denoted the summation of all the degrees of the pre exisiting nodes. 4.2 Calculate the cummulative sum corresponding to each pre existing node.

4.3 Generate a random number between 0-1 and see in which range the random number belongs. For example, if the random number obtained is 0.33 and cummulative sum corresponding to each pre existing node are [0.1,0.2,0.4,0.8,1.0], then as 0.33 lies between 0.2-0.4, so connect the edge from new node to either node 2 or node 3(as 0.2 is in 2nd index and 0.4 is in 4th index)

4.4 Repeat step 4.3 till all the m edges are connected. Repeat step 4 till the total number of nodes in the network becomes 100.

After the whole network is created for one instance, its average clustering coefficient, average path length and degree distribution is plotted.

After all the 100 intances are completed, final degree distribution graph is plotted by taking mean and standard deviation.

In [4]:
```
final_degree_list=[] #stores average degrees for all 100 instances
final_avg_clust_coeff=[]  #stores average clustering coefficients for all 100
 instances
final_avg_path_length=[]  #stores average path length for all 100 instances
for instance in range(100):
    print("Instance= ", instance)
    nodes=5 #taking 5 nodes for creating initial network
    edges=10 #taking 10 edges for creating initial network
    e=0
    init_graph=np.zeros((nodes,nodes))
    while(e<edges):
        for i in range(nodes):
            #choosing a random node
            val=random.randint(0,nodes-1)
        #checking for self loop
            if val!=i:
                init_graph[i][val]=1
                init_graph[val][i]=1
                e=e+1
        # if all the edges are taken and attached then the loop will end
            if e==edges:
                break
    total_nodes=100 #total number of nodes considered for each instance
    new_edges=4 # number of edges that are added in each evolution stage
    while(nodes!=total_nodes): # loop for each evolution step till the network
contains 200 nodes
#     print(init_graph)
        degree_dict={}
#degree_dict contains degrees of every node
        c=0
        degrees=[]
        for i in init_graph:
#to calculate degree of each node counting the number of 1s in each row
            deg=list(i).count(1)
            degrees.append(deg)
            degree_dict[c]=deg
            c=c+1
#calculating summation of degrees of all the existing nodes
        summation_of_deg=sum(degree_dict.values())
        prob_dict={}
        prob_temp=[]
# each index of prob_dict contains the probabilty=(degree of ith node)/(summat
ion of degree of all the nodes) for each node
        for d in degree_dict:
            prob=degree_dict[d]/summation_of_deg
            prob_dict[d]=prob
            prob_temp.append(prob)
        cummulative_prob=[]
#cummulative_prob contains cummulative degrees of each existing node
        s=0
        for i in range(len(prob_temp)):
            s=s+prob_temp[i]
            cummulative_prob.append(s)
#     print(cummulative_prob)
        #adding a new node
        #to Add a new node I am adding one row and one column with values 0 in
```

```
the existing graph
        new_col = [0]*len(init_graph[0])
        init_graph = np.column_stack((init_graph, new_col))
        new_row = [0]*len(init_graph[0])
        init_graph = np.vstack ((init_graph, new_row))
        new_node_index=len(init_graph)-1
        count=0
#adding new edges to the network
        while(count<new_edges):
            new_index=0
            v=random.uniform(0,1)
            if v<cummulative_prob[0]:
                new_index=0
            else:
                for r in range(1,len(cummulative_prob)-1):
                #if random value lies between ith and (i+1)th index then edge
 will be added to (i+1)th node
                    if v>cummulative_prob[r] and v<cummulative_prob[r+1]:
                        new_index=r+1
#                     print(v,r+1,cummulative_prob[r+1])
                        break
            if(init_graph[new_node_index][new_index]!=1):
                init_graph[new_node_index][new_index]=1
                init_graph[new_index][new_node_index]=1
                count=count+1
#               new_edges+=1
        nodes=nodes+1
    edge_list=[]
# edge list contains all the edge pairs of the final network
    for i in range(len(init_graph)):
        for j in range(len(init_graph)):
            if(init_graph[i][j]==1):
                edge_list.append((i,j))
    adjacency_list={} #contains neighbors of each node
    it=0
    for row in init_graph:
        temp=[]
        for v in range(len(row)):
            if row[v]==1:
                temp.append(v)
        adjacency_list[it]=temp
        it+=1
    path_length=[] #stores path length for final network
    cc1=Clustering_Coefficient(adjacency_list,edge_list,total_nodes)
    print("average clustering coeff for instance = ",instance, " is: ", cc1)
    node_list=[]
    for a in adjacency_list:
        node_list.append(a)
    for i in range(len(node_list)):
        for j in range(i+1,len(node_list)):
            BFS_for_finding_Shortest_path(node_list[i], node_list[j],path_leng
th,adjacency_list)
    total_edges1=(total_nodes*(total_nodes-1))/2
    pl1=sum(path_length)/total_edges1
#     print("average path length for instance" ,instance," is: ", pl1)
    final_avg_path_length.append(pl1)
    final_avg_clust_coeff.append(cc1)
```

```
    degrees=[]
#degree corresponding to each node
    degree_dict={}
    c=0
    for i in init_graph:
        deg=list(i).count(1) #calculating degree of a node by counting 1s in i
ts correspnding row
        degrees.append(deg)
        degree_dict[c]=deg
        c=c+1
    print("Average degree of the network is :",sum(degrees)/total_nodes)
    degree_distribution1={}
    for degree in degrees:
        if degree not in degree_distribution1.keys():
            pro=degrees.count(degree)/total_nodes
            degree_distribution1[degree]=pro
    print("Maximum degree is: ",max(degrees))
    print("Minimum degree is: ",min(degrees))
    #plotting degree distribution
#     plot_x=list(degree_distribution1.keys())
#     plot_y=list(degree_distribution1.values())
    final_degree_list.append(degree_distribution1)
#     indexes=np.argsort(plot_x)
#     x,y=[],[]

#     for index in indexes:
#         x.append(plot_x[index])
#         y.append(plot_y[index])
#     plt.figure()
#     plt.title(" Degree distribution")
#     plt.xlabel("degree")
#     plt.ylabel("P(degree)")
#     plt.scatter(x, y,c='green',label='P(degree)')
#     plt.legend()
#     plt.show()
```

```
Instance=  0
average clustering coeff for instance =  0  is:  0.17247277715539258
Average degree of the network is : 7.74
Maximum degree is:  38
Minimum degree is:  3
Instance=  1
average clustering coeff for instance =  1  is:  0.19377449195064902
Average degree of the network is : 7.72
Maximum degree is:  42
Minimum degree is:  3
Instance=  2
average clustering coeff for instance =  2  is:  0.18957690166649566
Average degree of the network is : 7.74
Maximum degree is:  44
Minimum degree is:  3
Instance=  3
average clustering coeff for instance =  3  is:  0.20501762085004216
Average degree of the network is : 7.74
Maximum degree is:  43
Minimum degree is:  2
Instance=  4
average clustering coeff for instance =  4  is:  0.16803240402854208
Average degree of the network is : 7.76
Maximum degree is:  33
Minimum degree is:  3
Instance=  5
average clustering coeff for instance =  5  is:  0.13253847004911742
Average degree of the network is : 7.7
Maximum degree is:  25
Minimum degree is:  3
Instance=  6
average clustering coeff for instance =  6  is:  0.20404217822095005
Average degree of the network is : 7.7
Maximum degree is:  41
Minimum degree is:  1
Instance=  7
average clustering coeff for instance =  7  is:  0.1425736572620352
Average degree of the network is : 7.72
Maximum degree is:  32
Minimum degree is:  3
Instance=  8
average clustering coeff for instance =  8  is:  0.17301333552313308
Average degree of the network is : 7.72
Maximum degree is:  42
Minimum degree is:  3
Instance=  9
average clustering coeff for instance =  9  is:  0.18012177008904673
Average degree of the network is : 7.76
Maximum degree is:  34
Minimum degree is:  3
Instance=  10
average clustering coeff for instance =  10  is:  0.20618566008743347
Average degree of the network is : 7.74
Maximum degree is:  36
Minimum degree is:  2
Instance=  11
average clustering coeff for instance =  11  is:  0.1640605961752842
```

```
Average degree of the network is : 7.76
Maximum degree is:  39
Minimum degree is:  4
Instance=  12
average clustering coeff for instance =  12  is:  0.2097860825499676
Average degree of the network is : 7.78
Maximum degree is:  44
Minimum degree is:  4
Instance=  13
average clustering coeff for instance =  13  is:  0.19813982677678074
Average degree of the network is : 7.74
Maximum degree is:  34
Minimum degree is:  3
Instance=  14
average clustering coeff for instance =  14  is:  0.18335810896568902
Average degree of the network is : 7.74
Maximum degree is:  33
Minimum degree is:  3
Instance=  15
average clustering coeff for instance =  15  is:  0.2097603828271111
Average degree of the network is : 7.72
Maximum degree is:  41
Minimum degree is:  2
Instance=  16
average clustering coeff for instance =  16  is:  0.16467948986294775
Average degree of the network is : 7.76
Maximum degree is:  30
Minimum degree is:  4
Instance=  17
average clustering coeff for instance =  17  is:  0.2133759421479547
Average degree of the network is : 7.7
Maximum degree is:  42
Minimum degree is:  2
Instance=  18
average clustering coeff for instance =  18  is:  0.21581655402115152
Average degree of the network is : 7.74
Maximum degree is:  36
Minimum degree is:  3
Instance=  19
average clustering coeff for instance =  19  is:  0.18075458279542456
Average degree of the network is : 7.74
Maximum degree is:  31
Minimum degree is:  2
Instance=  20
average clustering coeff for instance =  20  is:  0.18709792095975966
Average degree of the network is : 7.72
Maximum degree is:  38
Minimum degree is:  2
Instance=  21
average clustering coeff for instance =  21  is:  0.20792228201315602
Average degree of the network is : 7.74
Maximum degree is:  35
Minimum degree is:  3
Instance=  22
average clustering coeff for instance =  22  is:  0.19891295724536437
Average degree of the network is : 7.78
Maximum degree is:  37
```

```
Minimum degree is:  4
Instance=  23
average clustering coeff for instance =  23  is:  0.20731912352153373
Average degree of the network is : 7.72
Maximum degree is:  42
Minimum degree is:  4
Instance=  24
average clustering coeff for instance =  24  is:  0.2314566846610785
Average degree of the network is : 7.7
Maximum degree is:  37
Minimum degree is:  3
Instance=  25
average clustering coeff for instance =  25  is:  0.1580082764388558
Average degree of the network is : 7.72
Maximum degree is:  32
Minimum degree is:  2
Instance=  26
average clustering coeff for instance =  26  is:  0.2345595458562504
Average degree of the network is : 7.76
Maximum degree is:  49
Minimum degree is:  3
Instance=  27
average clustering coeff for instance =  27  is:  0.16445102838910877
Average degree of the network is : 7.74
Maximum degree is:  36
Minimum degree is:  3
Instance=  28
average clustering coeff for instance =  28  is:  0.1703994999059645
Average degree of the network is : 7.7
Maximum degree is:  37
Minimum degree is:  2
Instance=  29
average clustering coeff for instance =  29  is:  0.19586601466545034
Average degree of the network is : 7.7
Maximum degree is:  40
Minimum degree is:  3
Instance=  30
average clustering coeff for instance =  30  is:  0.16907047153249127
Average degree of the network is : 7.72
Maximum degree is:  30
Minimum degree is:  3
Instance=  31
average clustering coeff for instance =  31  is:  0.20365073082345791
Average degree of the network is : 7.7
Maximum degree is:  36
Minimum degree is:  2
Instance=  32
average clustering coeff for instance =  32  is:  0.21701566136099396
Average degree of the network is : 7.74
Maximum degree is:  35
Minimum degree is:  2
Instance=  33
average clustering coeff for instance =  33  is:  0.18191940670268839
Average degree of the network is : 7.74
Maximum degree is:  39
Minimum degree is:  3
Instance=  34
```

```
average clustering coeff for instance =  34  is:  0.17311767443083234
Average degree of the network is : 7.72
Maximum degree is:  36
Minimum degree is:  2
Instance=  35
average clustering coeff for instance =  35  is:  0.1294749452949868
Average degree of the network is : 7.72
Maximum degree is:  31
Minimum degree is:  4
Instance=  36
average clustering coeff for instance =  36  is:  0.18651525155719523
Average degree of the network is : 7.72
Maximum degree is:  28
Minimum degree is:  2
Instance=  37
average clustering coeff for instance =  37  is:  0.19409291348579635
Average degree of the network is : 7.72
Maximum degree is:  43
Minimum degree is:  3
Instance=  38
average clustering coeff for instance =  38  is:  0.23011920286159637
Average degree of the network is : 7.76
Maximum degree is:  40
Minimum degree is:  4
Instance=  39
average clustering coeff for instance =  39  is:  0.1785193119954932
Average degree of the network is : 7.78
Maximum degree is:  40
Minimum degree is:  4
Instance=  40
average clustering coeff for instance =  40  is:  0.1953494889463895
Average degree of the network is : 7.76
Maximum degree is:  33
Minimum degree is:  3
Instance=  41
average clustering coeff for instance =  41  is:  0.18213962645471518
Average degree of the network is : 7.72
Maximum degree is:  31
Minimum degree is:  3
Instance=  42
average clustering coeff for instance =  42  is:  0.18008060694440173
Average degree of the network is : 7.72
Maximum degree is:  32
Minimum degree is:  4
Instance=  43
average clustering coeff for instance =  43  is:  0.17047520306961036
Average degree of the network is : 7.76
Maximum degree is:  30
Minimum degree is:  4
Instance=  44
average clustering coeff for instance =  44  is:  0.1911282862267582
Average degree of the network is : 7.76
Maximum degree is:  40
Minimum degree is:  3
Instance=  45
average clustering coeff for instance =  45  is:  0.1784547927562633
Average degree of the network is : 7.76
```

```
Maximum degree is:  35
Minimum degree is:  3
Instance=  46
average clustering coeff for instance =  46  is:  0.17978408245050112
Average degree of the network is : 7.74
Maximum degree is:  34
Minimum degree is:  4
Instance=  47
average clustering coeff for instance =  47  is:  0.181312747272212
Average degree of the network is : 7.76
Maximum degree is:  35
Minimum degree is:  3
Instance=  48
average clustering coeff for instance =  48  is:  0.17319114962100668
Average degree of the network is : 7.74
Maximum degree is:  29
Minimum degree is:  3
Instance=  49
average clustering coeff for instance =  49  is:  0.16538135255644357
Average degree of the network is : 7.72
Maximum degree is:  29
Minimum degree is:  2
Instance=  50
average clustering coeff for instance =  50  is:  0.19055550487171527
Average degree of the network is : 7.72
Maximum degree is:  33
Minimum degree is:  2
Instance=  51
average clustering coeff for instance =  51  is:  0.18778285381716195
Average degree of the network is : 7.78
Maximum degree is:  34
Minimum degree is:  3
Instance=  52
average clustering coeff for instance =  52  is:  0.2304738899534583
Average degree of the network is : 7.74
Maximum degree is:  43
Minimum degree is:  3
Instance=  53
average clustering coeff for instance =  53  is:  0.16159695283104888
Average degree of the network is : 7.7
Maximum degree is:  28
Minimum degree is:  3
Instance=  54
average clustering coeff for instance =  54  is:  0.15969291335400562
Average degree of the network is : 7.74
Maximum degree is:  31
Minimum degree is:  3
Instance=  55
average clustering coeff for instance =  55  is:  0.21232218626072835
Average degree of the network is : 7.72
Maximum degree is:  45
Minimum degree is:  2
Instance=  56
average clustering coeff for instance =  56  is:  0.15764085256221827
Average degree of the network is : 7.76
Maximum degree is:  36
Minimum degree is:  3
```

```
Instance=  57
average clustering coeff for instance =  57  is:  0.13379582186516326
Average degree of the network is : 7.72
Maximum degree is:  29
Minimum degree is:  3
Instance=  58
average clustering coeff for instance =  58  is:  0.23925434495511372
Average degree of the network is : 7.74
Maximum degree is:  37
Minimum degree is:  2
Instance=  59
average clustering coeff for instance =  59  is:  0.17798959100468392
Average degree of the network is : 7.74
Maximum degree is:  35
Minimum degree is:  4
Instance=  60
average clustering coeff for instance =  60  is:  0.16976112379150787
Average degree of the network is : 7.76
Maximum degree is:  29
Minimum degree is:  2
Instance=  61
average clustering coeff for instance =  61  is:  0.19742703289142916
Average degree of the network is : 7.76
Maximum degree is:  39
Minimum degree is:  3
Instance=  62
average clustering coeff for instance =  62  is:  0.22609846082867857
Average degree of the network is : 7.74
Maximum degree is:  40
Minimum degree is:  3
Instance=  63
average clustering coeff for instance =  63  is:  0.1955399092279363
Average degree of the network is : 7.74
Maximum degree is:  37
Minimum degree is:  3
Instance=  64
average clustering coeff for instance =  64  is:  0.16894219421938486
Average degree of the network is : 7.74
Maximum degree is:  28
Minimum degree is:  3
Instance=  65
average clustering coeff for instance =  65  is:  0.14851268963234898
Average degree of the network is : 7.72
Maximum degree is:  35
Minimum degree is:  2
Instance=  66
average clustering coeff for instance =  66  is:  0.1848587998165152
Average degree of the network is : 7.74
Maximum degree is:  30
Minimum degree is:  3
Instance=  67
average clustering coeff for instance =  67  is:  0.18570682267728686
Average degree of the network is : 7.72
Maximum degree is:  36
Minimum degree is:  3
Instance=  68
average clustering coeff for instance =  68  is:  0.17625437174163477
```

```
Average degree of the network is : 7.74
Maximum degree is:  42
Minimum degree is:  2
Instance=  69
average clustering coeff for instance =  69  is:  0.18120630100893254
Average degree of the network is : 7.74
Maximum degree is:  48
Minimum degree is:  3
Instance=  70
average clustering coeff for instance =  70  is:  0.1786696731066902
Average degree of the network is : 7.7
Maximum degree is:  34
Minimum degree is:  2
Instance=  71
average clustering coeff for instance =  71  is:  0.19157522511975375
Average degree of the network is : 7.76
Maximum degree is:  48
Minimum degree is:  4
Instance=  72
average clustering coeff for instance =  72  is:  0.19481495494804082
Average degree of the network is : 7.72
Maximum degree is:  41
Minimum degree is:  2
Instance=  73
average clustering coeff for instance =  73  is:  0.16796151097497844
Average degree of the network is : 7.72
Maximum degree is:  34
Minimum degree is:  3
Instance=  74
average clustering coeff for instance =  74  is:  0.23245798117549593
Average degree of the network is : 7.76
Maximum degree is:  41
Minimum degree is:  3
Instance=  75
average clustering coeff for instance =  75  is:  0.21470014288177708
Average degree of the network is : 7.72
Maximum degree is:  37
Minimum degree is:  2
Instance=  76
average clustering coeff for instance =  76  is:  0.222100823153501
Average degree of the network is : 7.76
Maximum degree is:  42
Minimum degree is:  2
Instance=  77
average clustering coeff for instance =  77  is:  0.19179626904177574
Average degree of the network is : 7.72
Maximum degree is:  40
Minimum degree is:  3
Instance=  78
average clustering coeff for instance =  78  is:  0.1899578521283205
Average degree of the network is : 7.74
Maximum degree is:  32
Minimum degree is:  2
Instance=  79
average clustering coeff for instance =  79  is:  0.21120701964112776
Average degree of the network is : 7.74
Maximum degree is:  37
```

```
Minimum degree is:  2
Instance=  80
average clustering coeff for instance =  80  is:  0.17214939651466282
Average degree of the network is : 7.74
Maximum degree is:  35
Minimum degree is:  2
Instance=  81
average clustering coeff for instance =  81  is:  0.19017869479421118
Average degree of the network is : 7.72
Maximum degree is:  36
Minimum degree is:  2
Instance=  82
average clustering coeff for instance =  82  is:  0.16095090406316387
Average degree of the network is : 7.72
Maximum degree is:  32
Minimum degree is:  2
Instance=  83
average clustering coeff for instance =  83  is:  0.19788415147326283
Average degree of the network is : 7.76
Maximum degree is:  41
Minimum degree is:  4
Instance=  84
average clustering coeff for instance =  84  is:  0.15701490405094515
Average degree of the network is : 7.74
Maximum degree is:  33
Minimum degree is:  2
Instance=  85
average clustering coeff for instance =  85  is:  0.21196244698091615
Average degree of the network is : 7.74
Maximum degree is:  36
Minimum degree is:  4
Instance=  86
average clustering coeff for instance =  86  is:  0.17842694235426668
Average degree of the network is : 7.72
Maximum degree is:  30
Minimum degree is:  3
Instance=  87
average clustering coeff for instance =  87  is:  0.1776699687191677
Average degree of the network is : 7.76
Maximum degree is:  37
Minimum degree is:  3
Instance=  88
average clustering coeff for instance =  88  is:  0.17548179607767028
Average degree of the network is : 7.76
Maximum degree is:  35
Minimum degree is:  3
Instance=  89
average clustering coeff for instance =  89  is:  0.22610938816795595
Average degree of the network is : 7.72
Maximum degree is:  40
Minimum degree is:  2
Instance=  90
average clustering coeff for instance =  90  is:  0.1779483703205745
Average degree of the network is : 7.72
Maximum degree is:  36
Minimum degree is:  3
Instance=  91
```

```
average clustering coeff for instance =  91  is:  0.2091090069374175
Average degree of the network is : 7.7
Maximum degree is:  38
Minimum degree is:  2
Instance=  92
average clustering coeff for instance =  92  is:  0.16071025843078604
Average degree of the network is : 7.74
Maximum degree is:  38
Minimum degree is:  3
Instance=  93
average clustering coeff for instance =  93  is:  0.19565800465610722
Average degree of the network is : 7.74
Maximum degree is:  34
Minimum degree is:  3
Instance=  94
average clustering coeff for instance =  94  is:  0.23043194702098876
Average degree of the network is : 7.72
Maximum degree is:  45
Minimum degree is:  2
Instance=  95
average clustering coeff for instance =  95  is:  0.19163473280916815
Average degree of the network is : 7.74
Maximum degree is:  40
Minimum degree is:  2
Instance=  96
average clustering coeff for instance =  96  is:  0.1589610994546028
Average degree of the network is : 7.76
Maximum degree is:  34
Minimum degree is:  3
Instance=  97
average clustering coeff for instance =  97  is:  0.1758365232243527
Average degree of the network is : 7.76
Maximum degree is:  30
Minimum degree is:  4
Instance=  98
average clustering coeff for instance =  98  is:  0.1769673944370339
Average degree of the network is : 7.74
Maximum degree is:  39
Minimum degree is:  3
Instance=  99
average clustering coeff for instance =  99  is:  0.22121248645819946
Average degree of the network is : 7.74
Maximum degree is:  38
Minimum degree is:  2
```

In [5]:
```
print("Average Clustering Coefficient of all 100 instances is: ", np.mean(fina
l_avg_clust_coeff))
print("Average path length of all 100 instances is: ", np.mean(final_avg_path_
length))
```

```
Average Clustering Coefficient of all 100 instances is:  0.18750819559433374
Average path length of all 100 instances is:  2.3510141414141414
```

In [39]:
```
scaled_mean_degree={}
for i in range(50):
    temp=[]
    for dict1 in final_degree_list:
        if i in dict1:
            temp.append(dict1[i])
    if len(temp)!=0:
        scaled_mean_degree[i]=np.mean(temp)
    else:
        scaled_mean_degree[i]=0
```

In [40]:
```
scaled_std_deviation={}
for i in range(50):
    temp=[]
    for dict1 in final_degree_list:
        if i in dict1:
            temp.append(dict1[i])
    if len(temp)!=0:
        scaled_std_deviation[i]=np.std(temp)
    else:
        scaled_std_deviation[i]=0
```

Plotting degree distribution

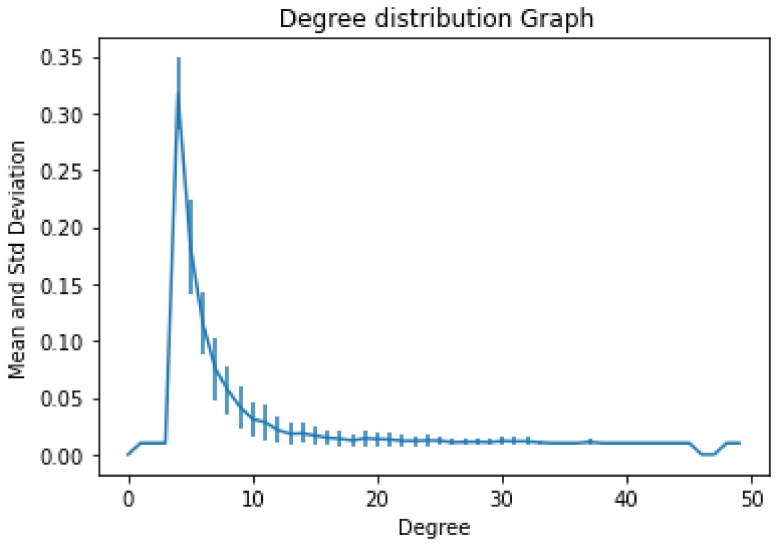
We can see from the graph that the model follows power law distribution.This happen due to rich gets richer phenomena. In other words, when a new node is added to a network, it is more likely to connect to a high-degree node than an obscure, low-degree node. The rich get richer and the hubs get hubbier.

Most of the nodes have smaller degree and only few nodes(called hubs) will have very high degree due to preferetial attachment. This is why we are getting power law degree distribution.

In [41]:
```
# example data
x=[]
for i in range(50):
    x.append(i)
xval = scaled_mean_degree.values()
yval = scaled_std_deviation.values()

plt.errorbar(x, scaled_mean_degree.values(), yerr = scaled_std_deviation.value
s() )
plt.title(" Degree distribution Graph")
plt.xlabel("Degree")
plt.ylabel("Mean and Std Deviation")

plt.show()
```



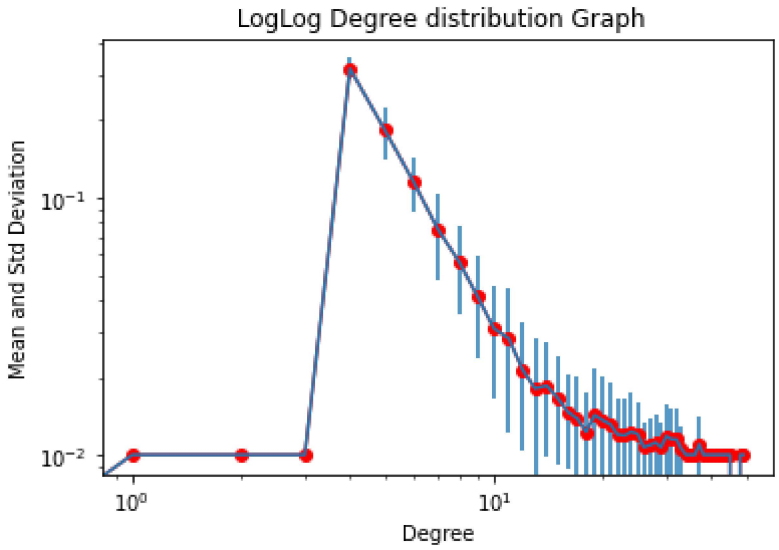
Plotting log scale

In [42]:
```
x=[]
for i in range(50):
    x.append(i)
xval1 = scaled_mean_degree.values()
xval=[]

for i in xval1:
    xval.append(i)
yval = scaled_std_deviation.values()
plt.loglog(x,xval  ,'ro-')

plt.errorbar(x, scaled_mean_degree.values(), yerr = scaled_std_deviation.value
s() )
plt.title(" LogLog Degree distribution Graph")
plt.xlabel("Degree")
plt.ylabel("Mean and Std Deviation")

# plt.xscale("loglog")
plt.show()
```



In [ ]:
```

```