In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:
```
import pandas as pd
import numpy as np
import pickle
import networkx as nx
```

In [4]:
```
path = "/content/drive/My Drive/"
project = "2_TwitterFollowGraph"
```

In [5]:
```
def degree_values(edgelist,undirected= True):
  in_degree_nodes = {}
  out_degree_nodes = {}
  nodes = set([])

  for edge in edgelist:
    if edge[1] in in_degree_nodes:
      in_degree_nodes[edge[1]].add(edge[0])
    else:
      in_degree_nodes[edge[1]] = set([edge[0]])

    if edge[0] in out_degree_nodes:
      out_degree_nodes[edge[0]].add(edge[1])
    else:
      out_degree_nodes[edge[0]] = set([edge[1]])

    nodes.add(edge[0])
    nodes.add(edge[1])

  in_degree = {}
  out_degree = {}

  for node in in_degree_nodes:
    in_degree[node] = len(in_degree_nodes[node])

  for node in out_degree_nodes:
    out_degree[node] = len(out_degree_nodes[node])

  if undirected:
    node_degree = {}
    for node in nodes:
      node_degree[node] = in_degree.get(node,0) + out_degree.get(node,0)
    return node_degree,nodes

  return in_degree,out_degree,nodes
```

In [6]:
```
def degree_distribution(degree,nodes):
  degree_dist = {}
  N = len(nodes)
  total = 0
  for node in degree:
    if degree[node] in degree_dist:
      degree_dist[degree[node]] += 1/N
    else:
      degree_dist[degree[node]] = 1/N
    total += 1/N
  degree_dist[0] = 1-total

  degree_dist = sorted(degree_dist.items(),key=lambda x:x[0])
  degree = [value[0] for value in degree_dist]
  node_count = [value[1] for value in degree_dist]

  cummulative_count = np.cumsum(node_count[::-1])[::-1]
  return degree,node_count,cummulative_count
```

In [7]:
```
def generate_pickle(edgelist,name,undirected=True):
  name = path+project+"/PickleFiles/Degree_Distribution/"+name+"/"
  if undirected:
    degree,nodes = degree_values(edgelist)
    degree,node_count,cummulative_count = degree_distribution(degree,nodes)

    graph = nx.Graph(edgelist)
    subgraph_nodes = max(nx.connected_components(graph),key=len)
    largest_connected_component = graph.subgraph(subgraph_nodes)

    degree_gaint,nodes_gaint = degree_values(largest_connected_component.edges())
    degree_gaint,node_count_gaint,cummulative_count_gaint = degree_distribution(degree_gaint,nodes_gaint)

    distribution_dict = dict(zip(degree,node_count))
    cummulative_dict = dict(zip(degree,cummulative_count))

    distribution_dict_gaint = dict(zip(degree_gaint,node_count_gaint))
    cummulative_dict_gaint = dict(zip(degree_gaint,cummulative_count_gaint))

    with open(name+"degree_distribution", 'wb') as file_name:
      pickle.dump(distribution_dict,file_name)

    with open(name+"cummulative_degree_distribution", 'wb') as file_name:
      pickle.dump(cummulative_dict,file_name)

    with open(name+"degree_distribution_giant", 'wb') as file_name:
      pickle.dump(distribution_dict_gaint,file_name)

    with open(name+"cummulative_degree_distribution_giant", 'wb') as file_name:
      pickle.dump(cummulative_dict_gaint,file_name)

  else:
    in_degree,out_degree,nodes = degree_values(edgelist,False)
    in_degree,node_count,cummulative_count = degree_distribution(in_degree,nodes)
    in_distribution_dict = dict(zip(in_degree,node_count))
    in_cummulative_dict = dict(zip(in_degree,cummulative_count))

    graph = nx.DiGraph(edgelist)
    subgraph_nodes_weakly = max(nx.weakly_connected_components(graph),key=len)
    largest_weakly_connected_component = graph.subgraph(subgraph_nodes_weakly)
```

```
out_degree,node_count,cummulative_count = degree_distribution(out_degree,nodes)
out_distribution_dict = dict(zip(out_degree,node_count))
out_cummulative_dict = dict(zip(out_degree,cummulative_count))

in_degree_weak,out_degree_weak,nodes_weak = degree_values(largest_weakly_connected_component.edges(),False)
in_degree_weak,node_count_weak,cummulative_count_weak = degree_distribution(in_degree_weak,nodes_weak)
in_distribution_dict_weak = dict(zip(in_degree_weak,node_count_weak))
in_cummulative_dict_weak = dict(zip(in_degree_weak,cummulative_count_weak))

out_degree_weak,node_count_weak,cummulative_count_weak = degree_distribution(out_degree_weak,nodes_weak)
out_distribution_dict_weak = dict(zip(out_degree_weak,node_count_weak))
out_cummulative_dict_weak = dict(zip(out_degree_weak,cummulative_count_weak))


with open(name+"indegree_distribution", 'wb') as file_name:
  pickle.dump(in_distribution_dict,file_name)

with open(name+"cummulative_indegree_distribution", 'wb') as file_name:
  pickle.dump(in_cummulative_dict,file_name)

with open(name+"outdegree_distribution", 'wb') as file_name:
  pickle.dump(out_distribution_dict,file_name)

with open(name+"cummulative_outdegree_distribution", 'wb') as file_name:
  pickle.dump(out_cummulative_dict,file_name)

with open(name+"indegree_distribution_giant", 'wb') as file_name:
  pickle.dump(in_distribution_dict_weak,file_name)

with open(name+"cummulative_indegree_distribution_giant", 'wb') as file_name:
  pickle.dump(in_cummulative_dict_weak,file_name)

with open(name+"outdegree_distribution_giant", 'wb') as file_name:
  pickle.dump(out_distribution_dict_weak,file_name)

with open(name+"cummulative_outdegree_distribution_giant", 'wb') as file_name:
  pickle.dump(out_cummulative_dict_weak,file_name)
```

In [8]:
```
df = pd.read_csv(path+project+"/Datasets/EU-Email/email-Eu-core.txt",sep = ' ')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Email",False)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Twitch/musae_ENGB_edges.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Twitch_ENGB",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Twitch/musae_FR_edges.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Twitch_FR",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Facebook/athletes_edges.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Facebook_Athletes",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Facebook/politician_edges.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Facebook_Politician",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Facebook/public_figure_edges.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Facebook_Public_Figure",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Linkedin/soc-linkedin.csv",sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Facebook",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Facebook/musae_facebook_edges.csv",sep = ' ')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Linkedin",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Twitter/twitter-final.csv")
df.columns = ["Destination","Source"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Twitter",False)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Flickr/soc-flickr.txt",sep = ' ')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Flickr",True)
```

In [ ]:
```
df = pd.read_csv(path+project+"/Datasets/Twitter-Ego/twitter-ego.csv,sep = ',')
df.columns = ["Source","Destination"]
edgelist = df[["Source","Destination"]].values.tolist()
generate_pickle(edgelist,"Twitter_Ego",False)
```