In [1]:
```
import os

root_dir = os.path.dirname(os.path.abspath('PageRank_IMDB.jpynb'))

content_dir = os.path.join(root_dir, "content/")
if not os.path.isdir(content_dir):
    os.mkdir(content_dir)

kaggle_dir = os.path.join(root_dir, ".kaggle/")
if not os.path.isdir(kaggle_dir):
    os.mkdir(kaggle_dir)

variables_dir = os.path.join(content_dir, "variables/")
if not os.path.isdir(variables_dir):
    os.mkdir(variables_dir)
```

In [2]:
```
import gc
import json
import zipfile
import pickle
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import sys

def getsize(obj):
    print('{:.2f} MB'.format(sys.getsizeof(obj)/(2**20)))
```

Loading variables

In [3]:
```
with open(os.path.join(variables_dir, 'page_rank.pkl'), 'rb') as inpt:
    page_rank = pickle.load(inpt)
```

In [4]:
```
with open(os.path.join(variables_dir, 'person.pkl'), 'rb') as inpt:
    person = pickle.load(inpt)
```

In [5]:
```
with open(os.path.join(variables_dir, 'connection_matrix.pkl'), 'rb') as inpt:
    connection_matrix = pickle.load(inpt)
```

In [7]:
```
movie_useful_cols = ['tconst', 'titleType', 'primaryTitle', 'startYear', 'genres']
subset_m = None
movie = pd.read_csv(os.path.join(content_dir, "title.basics.tsv/title.basics.tsv"), sep="\t", usecols=movie_useful_cols,
                    nrows=subset_m)
```

```
/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3444: DtypeWarning: Columns (5) have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [8]:
```
subset_mr = None
movie_ratings = pd.read_csv(os.path.join(content_dir, "title.ratings.tsv/title.ratings.tsv"), sep="\t",
                    nrows=subset_mr)
```

In [9]:
```
movie_with_ratings = movie_ratings.merge(movie, on='tconst', how='right')
```

In [10]:
```
with open(os.path.join(variables_dir, 'movie_person.pkl'), 'rb') as inpt:
    movie_person = pickle.load(inpt)
```

Data visualization

In [11]:
```
rank = pd.DataFrame(page_rank, index=range(len(page_rank)), columns=['pageRank'])
rank['rank'] = rank['pageRank'].rank(method='first', ascending=False)
```

In [12]:
```
person_rank = person.join(rank, how='right')
```

In [13]:
```
person_rank[person_rank['rank'] <= 10].sort_values(by='rank')
```

Out[13]:

	nconst	primaryName	primaryProfession	pageRank	rank
31453	nm0090367	Christian Bobet	actor,writer	0.000060	1.0
443	nm0000465	Ron Jeremy	actor,director,writer	0.000040	2.0
589	nm0000616	Eric Roberts	actor,producer,soundtrack	0.000039	3.0
59297	nm0180569	Sergio Corona	actor,composer,producer	0.000034	4.0
637262	nm2187104	Malikha Mallette	actress	0.000033	5.0
18140	nm0048389	Dee Bradley Baker	actor,soundtrack,miscellaneous	0.000033	6.0
120092	nm0392056	Steve Holmes	actor,director,producer	0.000030	7.0
84351	nm0263582	Erik Everhard	actor,director,producer	0.000029	8.0
197959	nm0676182	Mr. Pete	actor,director,camera_department	0.000029	9.0
763	nm0000799	Edward Asner	actor,soundtrack,producer	0.000029	10.0

How many actors did the top-ranked actor starred with?

In [14]:
```
top_index = person_rank[person_rank['rank'] == 1].index[0]
for (b, a, p) in connection_matrix:
    if a == top_index:
        break

1/p
```

Out[14]:
```
1250.0
```

In [15]:
```
set([1,2]) in [set([2,1]), set([1,3])]
```

Out[15]:
```
True
```

Which actors in the top 10 starred together?

In [16]:
```
top10_index = list(person_rank[person_rank['rank'] <= 10].index)
duplicates = list()
for (b, a, p) in connection_matrix:
    if b in top10_index and a in top10_index:
        if set([b,a]) not in duplicates:
            print(f'{person_rank.iloc[b,1]} and {person_rank.iloc[a,1]}')
            duplicates.append(set([b,a]))
```

```
Eric Roberts and Ron Jeremy
Steve Holmes and Ron Jeremy
Erik Everhard and Ron Jeremy
Edward Asner and Eric Roberts
Dee Bradley Baker and Edward Asner
Mr. Pete and Erik Everhard
Steve Holmes and Erik Everhard
Mr. Pete and Steve Holmes
```

What are the top-rated movies starring the top-ranked actors?

In [17]:
```
def get_toprated(r):
    movies_r = list(movie_person[movie_person['nconst'] == person_rank[person_rank['rank'] == r]['nconst'].values[0]]['tconst'].values)
    movies_r = movie_with_ratings[movie_with_ratings['tconst'].isin(movies_r)]
    toprated = movies_r.loc[movies_r['averageRating'].idxmax(skipna = True),:]

    return toprated
```

In [18]:
```
for r in range(1,11):
    who = person_rank[person_rank['rank'] == r].iloc[0,1]
    toprated = get_toprated(r)
    print(f'"{toprated["primaryTitle"]}" was the top-rated movie starring {who}, with {toprated["averageRating"]}/10 stars.')
```

```
"Petits secrets entre voisins" was the top-rated movie starring Christian Bobet, with 5.7/10 stars.
"Porn Academy" was the top-rated movie starring Ron Jeremy, with 9.3/10 stars.
"Orange Kid" was the top-rated movie starring Eric Roberts, with 9.4/10 stars.
"Accidente del destino" was the top-rated movie starring Sergio Corona, with 9.3/10 stars.
"Deadly Divorce" was the top-rated movie starring Malikha Mallette, with 9.6/10 stars.
"Episode XCIII" was the top-rated movie starring Dee Bradley Baker, with 9.8/10 stars.
"Private Black Label 27: Private Love Story" was the top-rated movie starring Steve Holmes, with 9.2/10 stars.
"3-Way Fever" was the top-rated movie starring Erik Everhard, with 9.4/10 stars.
"Phat Ass Tits 4" was the top-rated movie starring Mr. Pete, with 9.6/10 stars.
"Spear-Carriers" was the top-rated movie starring Edward Asner, with 9.6/10 stars.
```