Social Networks HW2

- 4006213069 - AliAsghar Zare
- 4006213175 - MohammadJavad Nemati ---

Importing the neccecary libraries

```
import numpy as np
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
```

Graph Info Function

```
def graph_info(graph, graph_name):
    print(f"The {graph_name}'s Graph has:")

    number_of_nodes = len(graph.nodes())
    print(f"{number_of_nodes} nodes")

    print("&")

    number_of_edges = len(graph.edges())
    print(f"{number_of_edges} edges")

    if(dolphins.is_directed()):
        print("And it's Directional")
    else:
        print("And it's not Directional")

    graph_radius = nx.radius(graph)
    print(f"The {graph_name} Graph has a Radius of {graph_radius}")

    graph_diameter = nx.diameter(graph)
    print(f"The {graph_name} Graph has a Diameter of
{graph_diameter}")
```

Centrality Functions

```
def plot_centrality_table(graph, sort_by='node'):
    centrality_measures = {
        "Degree Centrality": nx.degree_centrality(graph),
        "Betweenness Centrality": nx.betweenness_centrality(graph),
        "Closeness Centrality": nx.closeness_centrality(graph),
        "Eigenvector Centrality": nx.eigenvector_centrality(graph,
max_iter=1000),
        "PageRank": nx.pagerank(graph)
    }
    centrality_df = pd.DataFrame(centrality_measures).round(4)

    if sort_by == 'node':
        centrality_df = centrality_df.sort_index(ascending=True)
    else:
        centrality_df = centrality_df.sort_values(by=sort_by,
ascending=False)

    fig, ax = plt.subplots(figsize=(12, 6))
    ax.axis('tight')
    ax.axis('off')

    table = ax.table(cellText=centrality_df.values,
                     colLabels=centrality_df.columns,
                     rowLabels=centrality_df.index,
                     cellLoc='center',
                     loc='center')

    table.auto_set_font_size(False)
    table.set_fontsize(10)

table.auto_set_column_width(col=list(range(len(centrality_df.columns))
))
    table.scale(1.2, 1.2)

    fig.patch.set_facecolor('white')
    plt.show()
```

Executing the functions on our Datasets

Dolphins

```
dolphins = nx.read_edgelist("dolphins/dolphins.edges", nodetype=int)
graph_info(dolphins, "Dolphins")
plot_centrality_table(dolphins)
```

```
The Dolphins's Graph has:
62 nodes
&
159 edges
And it's not Directional
The Dolphins Graph has a Radius of 5
The Dolphins Graph has a Diameter of 8
```

	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	PageRank
0	0.0984	0.0191	0.3466	0.1285	0.017
1	0.1311	0.2133	0.372	0.0421	0.0247
2	0.0656	0.0091	0.2824	0.0398	0.0133
3	0.0492	0.0024	0.3081	0.0793	0.0096
4	0.0164	0.0	0.249	0.0293	0.0051
5	0.0656	0.0044	0.2383	0.0066	0.0144
6	0.0984	0.0294	0.2675	0.0122	0.0201
7	0.082	0.1182	0.3653	0.0429	0.0156
8	0.0984	0.0224	0.3631	0.1431	0.0171
9	0.1148	0.0209	0.2521	0.0122	0.0235
10	0.082	0.0161	0.3128	0.0753	0.0151
11	0.0164	0.0	0.249	0.0293	0.0051
12	0.0164	0.0	0.2687	0.0391	0.0048
13	0.1311	0.0528	0.2711	0.015	0.0262
14	0.1967	0.062	0.3765	0.3158	0.0321
15	0.1148	0.0333	0.3389	0.1642	0.0199
16	0.0984	0.0033	0.3297	0.208	0.0166
17	0.1475	0.1143	0.3096	0.0175	0.0317
18	0.1148	0.0149	0.337	0.2025	0.0194
19	0.0656	0.0133	0.3161	0.0207	0.0129
20	0.1475	0.1026	0.391	0.1845	0.0246
21	0.0984	0.0127	0.3333	0.2073	0.0169
22	0.0164	0.0	0.2374	0.0024	0.0054
23	0.0492	0.0422	0.3333	0.0874	0.0099
24	0.0984	0.0074	0.3128	0.1932	0.0169
25	0.0492	0.0016	0.2552	0.006	0.0115
26	0.0492	0.0044	0.2773	0.0089	0.0112
27	0.082	0.0292	0.3161	0.0163	0.0171
28	0.082	0.0668	0.3653	0.0682	0.0148
29	0.1475	0.0655	0.3228	0.2118	0.0265
30	0.082	0.0331	0.3228	0.0408	0.0153
31	0.0164	0.0	0.2374	0.0024	0.0054
32	0.0492	0.0328	0.2163	0.0039	0.0133
33	0.1639	0.0572	0.3653	0.2811	0.0284
34	0.082	0.0327	0.3161	0.1388	0.0159
35	0.0164	0.0	0.245	0.0294	0.0049
36	0.1148	0.2482	0.4178	0.1328	0.0206
37	0.1803	0.1386	0.3987	0.3006	0.0299
38	0.1311	0.0454	0.3297	0.1966	0.0239
39	0.0328	0.0705	0.3352	0.0209	0.0078
40	0.1311	0.1431	0.404	0.2079	0.022
41	0.082	0.0233	0.2976	0.0153	0.0161
42	0.0984	0.0292	0.3297	0.081	0.0176
43	0.1148	0.0628	0.337	0.1903	0.0217
44	0.0656	0.012	0.3112	0.0778	0.0128
45	0.1803	0.0407	0.3466	0.285	0.0295
46	0.0328	0.003	0.2563	0.0297	0.0088
47	0.0984	0.0232	0.3389	0.0804	0.0173
48	0.0164	0.0	0.2328	0.0024	0.0053
49	0.0328	0.0009	0.247	0.0234	0.0089
50	0.1148	0.0334	0.3506	0.2177	0.0192
51	0.1639	0.0847	0.3297	0.2107	0.0313
52	0.0656	0.0192	0.3427	0.1296	0.0121
53	0.0328	0.0012	0.2596	0.0337	0.0082
54	0.1148	0.0991	0.3245	0.023	0.0217
55	0.0328	0.0009	0.2711	0.0521	0.0075
56	0.0328	0.0001	0.2133	0.0026	0.0083
57	0.1475	0.0842	0.302	0.0174	0.0301
58	0.0164	0.0	0.249	0.0273	0.005
59	0.082	0.0203	0.3427	0.1118	0.0148
60	0.0164	0.0	0.1784	0.0005	0.0062
61	0.0492	0.0142	0.302	0.052	0.011

PolBooks

```
polbooks = nx.read_edgelist("polbooks/_altsoph/polbooks.edges",
nodetype=int)
graph_info(polbooks, "PolBooks")
plot_centrality_table(polbooks)
```

```
The PolBooks's Graph has:
105 nodes
&
441 edges
And it's not Directional
The PolBooks Graph has a Radius of 4
The PolBooks Graph has a Diameter of 7
```

	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	PageRank
0	0.0577	0.0074	0.3455	0.0168	0.0078
1	0.0385	0.0	0.3032	0.0123	0.0055
2	0.0385	0.0005	0.3095	0.0119	0.0057
3	0.2212	0.0761	0.3714	0.0771	0.0242
4	0.0769	0.0629	0.391	0.0466	0.0099
5	0.0673	0.0129	0.3562	0.0212	0.0089
6	0.1058	0.0563	0.3796	0.0315	0.013
7	0.0769	0.0687	0.4078	0.0568	0.0094
8	0.2404	0.0749	0.3782	0.0909	0.025
9	0.1538	0.0984	0.4062	0.0666	0.0168
10	0.1442	0.0337	0.341	0.0498	0.0163
11	0.1731	0.0293	0.3611	0.0704	0.0187
12	0.2404	0.0953	0.381	0.0868	0.0256
13	0.125	0.018	0.341	0.0516	0.0139
14	0.0865	0.0588	0.3954	0.0484	0.0098
15	0.0481	0.0008	0.2921	0.0202	0.0065
16	0.0288	0.0	0.2788	0.0123	0.0044
17	0.0481	0.0001	0.2971	0.0294	0.0059
18	0.0288	0.0	0.3161	0.0164	0.0042
19	0.0481	0.0365	0.3624	0.0206	0.0066
20	0.0962	0.0262	0.381	0.0465	0.0109
21	0.0481	0.0001	0.2963	0.0276	0.0059
22	0.0673	0.0072	0.3467	0.0315	0.0081
23	0.0865	0.0019	0.298	0.0406	0.0097
24	0.0865	0.005	0.3574	0.0484	0.0095
25	0.0481	0.0008	0.321	0.0213	0.0061
26	0.0865	0.006	0.3549	0.0447	0.0095
27	0.0865	0.0029	0.3323	0.0456	0.0095
28	0.0288	0.0038	0.3421	0.048	0.0042
29	0.0385	0.0097	0.349	0.0168	0.0053
30	0.1923	0.1395	0.4143	0.2586	0.0202
31	0.1058	0.0318	0.3824	0.152	0.0114
32	0.0481	0.0002	0.2889	0.0257	0.0059
33	0.0865	0.0013	0.2946	0.0364	0.0098
34	0.0481	0.0002	0.2402	0.0104	0.0063
35	0.0962	0.0084	0.305	0.0298	0.0112
36	0.0481	0.0031	0.2857	0.0194	0.0061
37	0.0673	0.0036	0.2921	0.0256	0.008
38	0.0673	0.0026	0.2889	0.0223	0.0081
39	0.0769	0.0061	0.3041	0.0275	0.009
40	0.1731	0.0462	0.3574	0.0653	0.0183
41	0.0769	0.0033	0.326	0.0393	0.0087
42	0.0577	0.0007	0.2989	0.0267	0.0069
43	0.0481	0.0017	0.2889	0.0176	0.0063
44	0.0481	0.0002	0.3006	0.0272	0.0059
45	0.0577	0.0002	0.32	0.0338	0.0067
46	0.0385	0.0427	0.3355	0.022	0.0053
47	0.1731	0.0277	0.3514	0.0655	0.0183
48	0.0385	0.0002	0.3388	0.0167	0.0053
49	0.0769	0.1036	0.4078	0.0696	0.0091
50	0.0288	0.0033	0.3467	0.0168	0.0042
51	0.0577	0.0204	0.3562	0.0196	0.0079
52	0.0481	0.0133	0.3574	0.0192	0.0065
53	0.0481	0.0547	0.3714	0.0338	0.0059
54	0.0577	0.0002	0.298	0.0288	0.0068
55	0.0385	0.0012	0.3104	0.0149	0.0054
56	0.0385	0.0035	0.3077	0.0105	0.0056
57	0.0481	0.0111	0.3399	0.0163	0.0066
58	0.125	0.0753	0.4127	0.0629	0.0153
59	0.0481	0.0009	0.2537	0.0278	0.007
60	0.0577	0.0029	0.2744	0.0633	0.0076
61	0.0385	0.0022	0.2781	0.0263	0.0058
62	0.0577	0.0022	0.2759	0.0609	0.0076
63	0.0385	0.0	0.2518	0.0257	0.0057

Football

```
football = nx.read_edgelist("football/_altsoph/football.edges",
nodetype=int)
graph_info(football, "Football")
plot_centrality_table(football)
```

```
The Football's Graph has:
115 nodes
&
613 edges
And it's not Directional
The Football Graph has a Radius of 3
The Football Graph has a Diameter of 4
```

	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	PageRank
0	0.1053	0.0325	0.4238	0.1065	0.0096
1	0.1053	0.0176	0.413	0.0964	0.0096
2	0.1053	0.0131	0.4071	0.1163	0.0095
3	0.1053	0.0231	0.4207	0.1062	0.0096
4	0.0965	0.0107	0.4028	0.1012	0.0089
5	0.1053	0.0107	0.3838	0.1023	0.0097
6	0.1053	0.0197	0.4254	0.1129	0.0095
7	0.1053	0.0146	0.4014	0.1207	0.0095
8	0.0965	0.005	0.3608	0.1078	0.0088
9	0.0965	0.0118	0.3972	0.1065	0.0088
10	0.0877	0.0069	0.3813	0.0903	0.0082
11	0.0877	0.0114	0.4028	0.0799	0.0083
12	0.0877	0.0101	0.3904	0.0711	0.0084
13	0.0965	0.0173	0.4207	0.106	0.0088
14	0.0877	0.0071	0.3851	0.0766	0.0083
15	0.1053	0.0209	0.4238	0.1163	0.0095
16	0.0965	0.0222	0.4238	0.1038	0.0088
17	0.0965	0.0166	0.4116	0.0818	0.009
18	0.0965	0.0085	0.38	0.0751	0.0091
19	0.0965	0.0098	0.3904	0.0835	0.009
20	0.0965	0.023	0.4028	0.0812	0.009
21	0.0965	0.0221	0.3972	0.11	0.0088
22	0.0965	0.0067	0.3677	0.1103	0.0088
23	0.0965	0.0076	0.3851	0.1045	0.0089
24	0.0877	0.0207	0.4238	0.0803	0.0083
25	0.0965	0.0178	0.413	0.0934	0.0089
26	0.0877	0.017	0.3986	0.0709	0.0084
27	0.0965	0.0107	0.3826	0.0795	0.009
28	0.0789	0.0182	0.4086	0.0701	0.0076
29	0.0965	0.009	0.4	0.0858	0.009
30	0.0965	0.0151	0.4101	0.0876	0.009
31	0.0965	0.0084	0.3787	0.0826	0.009
32	0.0965	0.0149	0.4101	0.1092	0.0088
33	0.0877	0.0074	0.3878	0.0834	0.0082
34	0.0965	0.008	0.3762	0.0768	0.0091
35	0.0965	0.0155	0.3972	0.0885	0.0089
36	0.0702	0.0132	0.4014	0.0561	0.007
37	0.0965	0.0091	0.3945	0.086	0.009
38	0.0965	0.0252	0.4043	0.0805	0.0091
39	0.0965	0.0107	0.4071	0.1044	0.0088
40	0.0965	0.014	0.3986	0.103	0.0089
41	0.0877	0.0064	0.3958	0.0991	0.0081
42	0.0614	0.0078	0.3701	0.0483	0.0063
43	0.0965	0.0136	0.3958	0.0746	0.0091
44	0.0965	0.0144	0.4	0.086	0.009
45	0.0965	0.0114	0.3972	0.0905	0.0089
46	0.0965	0.0104	0.3931	0.1196	0.0087
47	0.0965	0.0151	0.4057	0.1101	0.0088
48	0.0965	0.0178	0.4101	0.0925	0.0089
49	0.0965	0.0098	0.4043	0.1172	0.0088
50	0.0789	0.0144	0.3986	0.0743	0.0076
51	0.0965	0.0149	0.3945	0.1042	0.0089
52	0.0877	0.0099	0.3972	0.0929	0.0082
53	0.1053	0.0166	0.4116	0.1226	0.0095
54	0.0877	0.0039	0.3738	0.076	0.0083
55	0.0965	0.0117	0.3931	0.0888	0.0089
56	0.0877	0.0095	0.3813	0.0756	0.0083
57	0.0877	0.0104	0.3972	0.0756	0.0083
58	0.0877	0.0288	0.4368	0.0791	0.0084
59	0.0702	0.0109	0.3972	0.0546	0.007
60	0.0965	0.0152	0.4116	0.1019	0.0089
61	0.0965	0.0208	0.4043	0.0826	0.009
62	0.0965	0.0142	0.3918	0.0825	0.009
63	0.0789	0.0183	0.4191	0.062	0.0077
64	0.0965	0.0204	0.4176	0.1065	0.0088