東京工業大学
Tokyo Institute of Technology

Complex Network

Quiz 6

WANG BIYUAN

18M38156

Department of Computer Science

Lecturer: Tsuyoshi Murata

December 20, 2018

Code

```
import networkx as nx
import matplotlib.pyplot as plt

star = nx.star_graph(9)
plt.subplot(231)
nx.draw(star, node_size=400, node_color='red', with_labels=True,
    font_weight='bold')
print("star transitivity:", nx.transitivity(star))
print("mean local clustering coefficient of star:",
    nx.average_clustering(star))

# print("local clustering coefficient of star:", nx.clustering(star))
# print("mean local clustering coefficient of star:",
    sum(nx.clustering(star).values())/nx.number_of_nodes(star))


petersen = nx.petersen_graph()
plt.subplot(232)
nx.draw_shell(petersen, nlist=[range(5, 10), range(5)], node_size=200,
    node_color='red', with_labels=True, font_weight='bold')
print("petersen transitivity:", nx.transitivity(petersen))
print("mean local clustering coefficient of petersen:",
    nx.average_clustering(petersen))

# print("local clustering coefficient of petersen:",
    nx.clustering(petersen))
# print("mean local clustering coefficient of petersen:",
    sum(nx.clustering(petersen).values())/nx.number_of_nodes(petersen))


cycle = nx.cycle_graph(10)
plt.subplot(233)
nx.draw_spring(cycle, node_size=400, node_color='red', with_labels=True,
    font_weight='bold')
print("cycle transitivity:", nx.transitivity(cycle))
print("mean local clustering coefficient of cycle:",
    nx.average_clustering(cycle))

# print("local clustering coefficient of cycle:", nx.clustering(cycle))
# print("mean local clustering coefficient of cycle:",
    sum(nx.clustering(cycle).values())/nx.number_of_nodes(cycle))

K_10 = nx.complete_graph(10)
plt.subplot(234)
nx.draw_circular(K_10, node_size=200, node_color='red', with_labels=True,
    font_weight='bold')
print("k_10 transitivity:", nx.transitivity(K_10))
print("mean local clustering coefficient of k_10:",
    nx.average_clustering(K_10))
```

```
# print("local clustering coefficient of k_10:", nx.clustering(K_10))
# print("mean local clustering coefficient of k_10:",
    sum(nx.clustering(K_10).values())/nx.number_of_nodes(K_10))

K_5_5 = nx.complete_bipartite_graph(5, 5)
plt.subplot(235)
nx.draw_circular(K_5_5, nlist=[range(5, 10), range(5)], node_size=200,
    node_color='red', with_labels=True, font_weight='bold')
print("k_5_5 transitivity:", nx.transitivity(K_5_5))
print("mean local clustering coefficient of k_5_5:",
    nx.average_clustering(K_5_5))

# print("local clustering coefficient of k_5_5:", nx.clustering(star))
# print("mean local clustering coefficient of k_5_5:",
    sum(nx.clustering(K_5_5).values())/nx.number_of_nodes(K_5_5))

wheel = nx.wheel_graph(10)
plt.subplot(236)
nx.draw(wheel, node_size=400, node_color='red', with_labels=True,
    font_weight='bold')
print("wheel transitivity:", "{0:.2f}".format(nx.transitivity(wheel)))
print("mean local clustering coefficient of wheel:",
    "{0:.2f}".format(nx.average_clustering(wheel)))

# print("local clustering coefficient of wheel:", nx.clustering(wheel))
# print("mean local clustering coefficient of wheel:",
    "{0:.2f}".format(sum(nx.clustering(wheel).values())/nx.number_of_nodes(wheel)))
```

Results

Make a program of computing
(i) transitivity and
(ii) mean local clustering coefficient of
(a) star graph,
(b) Petersen graph,
(c) circle graph,
(d) complete graph,
(e) complete bipartite graph, and
(f) wheel graph.

Network	Transitivity	Mean Local Clustering Coefficient
Star Graph	0	0
Petersen Graph	0	0
Circle Graph	0	0
Complete Graph	1	1
Complete Bipartite Graph	0	0
Wheel Graph	0.43	0.63