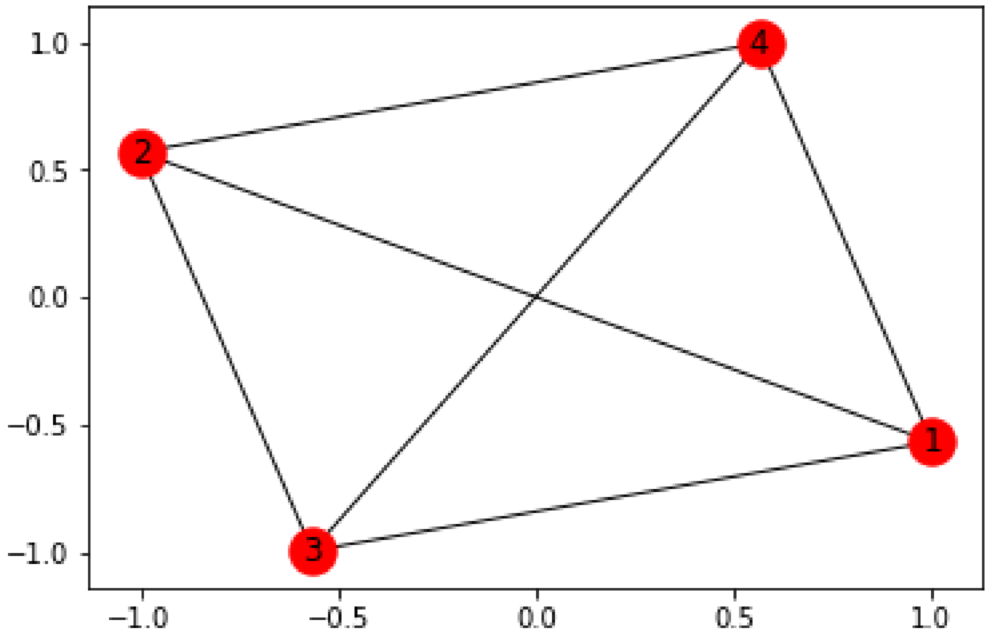
graphs

April 7, 2019

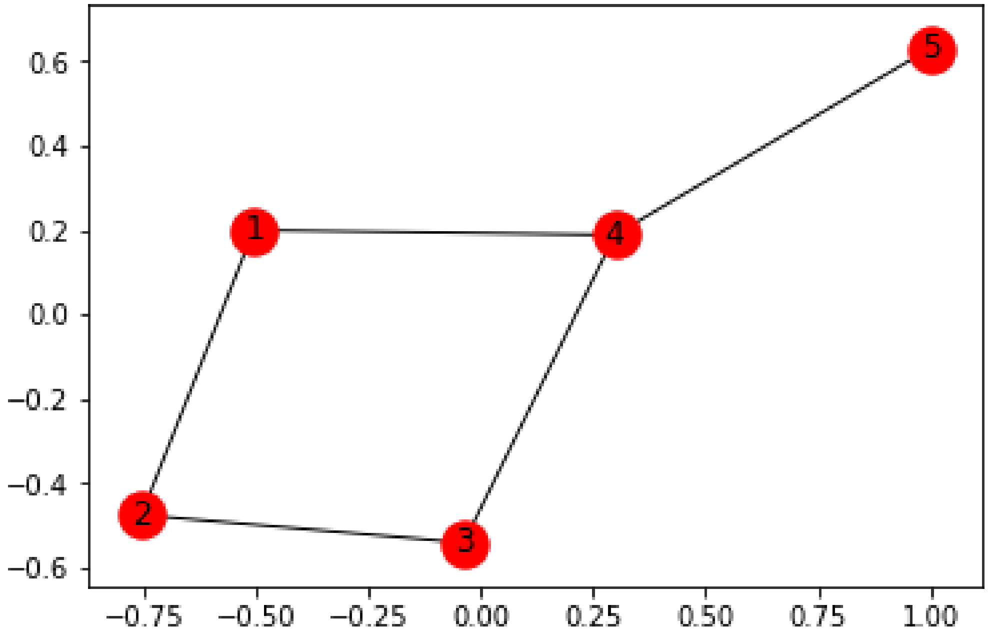
```
In [1]: %matplotlib inline
        import matplotlib.pyplot as plt
        import networkx as nx
```

```
In [2]: G = nx.Graph()
        G.add_edge(1,2)
        G.add_edge(2,3)
        G.add_edge(3,1)
        G.add_edge(3,4)
        G.add_edge(4,1)
        G.add_edge(4,2)
        nx.draw_networkx(G)
        plt.show()
```

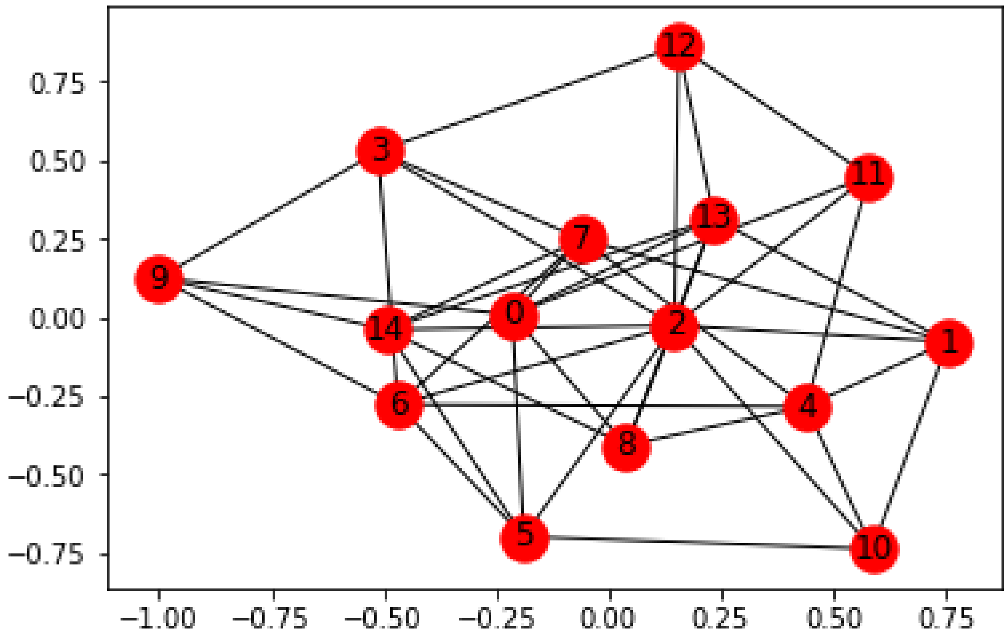
```
d:\python\lib\site-packages\networkx\drawing\nx_pylab.py:611: MatplotlibDeprecationWarning: is
  if cb.is_numlike(alpha):
```



```
In [3]: G = nx.Graph()
        G.add_edges_from([(1,2), (2,3),(3,4), (4,1), (5,4)])
        nx.draw_networkx(G)
        plt.show()
```

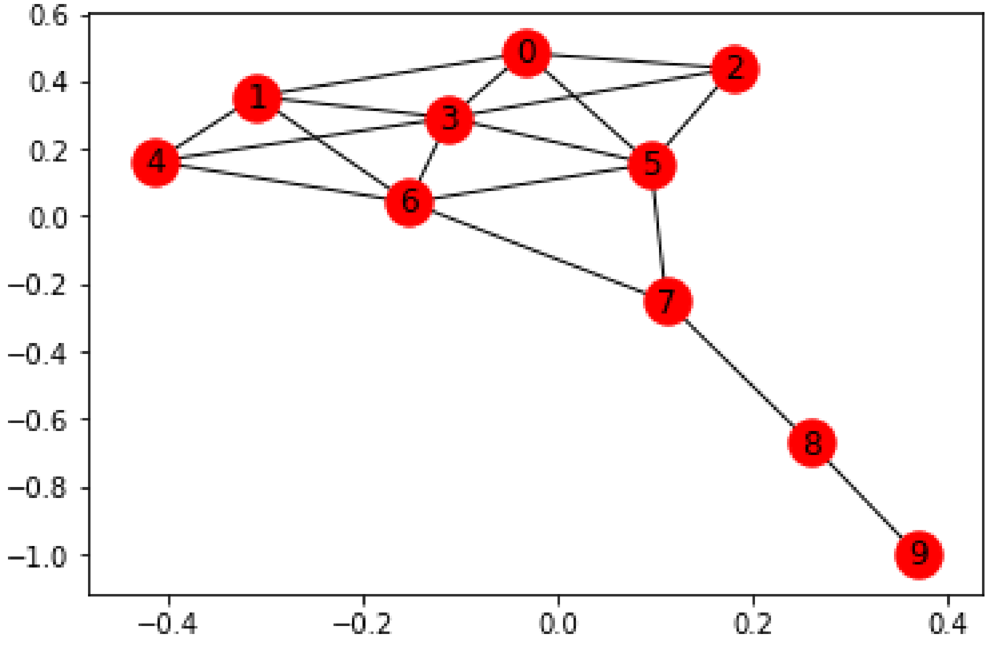


```
In [6]: k = nx.fast_gnp_random_graph(15,0.40)
        nx.draw_networkx(k)
        plt.show()
```



```
In [8]: G = nx.krackhardt_kite_graph()
        nx.draw_networkx(G)
        plt.show()
```

```
d:\python\lib\site-packages\networkx\drawing\nx_pylab.py:611: MatplotlibDeprecationWarning: is
  if cb.is_numlike(alpha):
```



```
In [10]: print(nx.has_path(G, source=1, target=9))
         print(nx.shortest_path(G, source=1, target=9))
         print(nx.shortest_path_length(G, source=1, target=9))
```

```
True
[1, 6, 7, 8, 9]
4
```

```
In [14]: all_paths = list(nx.shortest_simple_paths(G, source=1, target=9))
         for path in all_paths:
             print(path)
```

```
[1, 6, 7, 8, 9]
[1, 0, 5, 7, 8, 9]
[1, 6, 5, 7, 8, 9]
[1, 3, 5, 7, 8, 9]
[1, 4, 6, 7, 8, 9]
[1, 3, 6, 7, 8, 9]
[1, 0, 2, 5, 7, 8, 9]
[1, 0, 5, 6, 7, 8, 9]
[1, 6, 3, 5, 7, 8, 9]
[1, 3, 5, 6, 7, 8, 9]
[1, 4, 3, 5, 7, 8, 9]
[1, 4, 6, 5, 7, 8, 9]
[1, 3, 0, 5, 7, 8, 9]
[1, 3, 6, 5, 7, 8, 9]
[1, 0, 3, 5, 7, 8, 9]
[1, 4, 3, 6, 7, 8, 9]
[1, 3, 2, 5, 7, 8, 9]
[1, 0, 3, 6, 7, 8, 9]
[1, 3, 4, 6, 7, 8, 9]
[1, 0, 2, 3, 5, 7, 8, 9]
[1, 0, 2, 5, 6, 7, 8, 9]
[1, 0, 5, 3, 6, 7, 8, 9]
[1, 6, 4, 3, 5, 7, 8, 9]
[1, 6, 3, 0, 5, 7, 8, 9]
[1, 4, 3, 5, 6, 7, 8, 9]
[1, 4, 6, 3, 5, 7, 8, 9]
[1, 3, 0, 2, 5, 7, 8, 9]
[1, 3, 0, 5, 6, 7, 8, 9]
[1, 0, 3, 5, 6, 7, 8, 9]
[1, 4, 3, 0, 5, 7, 8, 9]
[1, 4, 3, 6, 5, 7, 8, 9]
[1, 3, 2, 0, 5, 7, 8, 9]
[1, 3, 2, 5, 6, 7, 8, 9]
```

```
[1, 0, 3, 2, 5, 7, 8, 9]
[1, 0, 3, 6, 5, 7, 8, 9]
[1, 3, 4, 6, 5, 7, 8, 9]
[1, 0, 2, 3, 6, 7, 8, 9]
[1, 6, 3, 2, 5, 7, 8, 9]
[1, 4, 3, 2, 5, 7, 8, 9]
[1, 0, 3, 4, 6, 7, 8, 9]
[1, 0, 2, 3, 5, 6, 7, 8, 9]
[1, 0, 2, 5, 3, 6, 7, 8, 9]
[1, 0, 5, 2, 3, 6, 7, 8, 9]
[1, 0, 5, 3, 4, 6, 7, 8, 9]
[1, 6, 4, 3, 0, 5, 7, 8, 9]
[1, 6, 3, 0, 2, 5, 7, 8, 9]
[1, 4, 6, 3, 0, 5, 7, 8, 9]
[1, 3, 0, 2, 5, 6, 7, 8, 9]
[1, 4, 3, 0, 2, 5, 7, 8, 9]
[1, 4, 3, 0, 5, 6, 7, 8, 9]
[1, 3, 2, 0, 5, 6, 7, 8, 9]
[1, 0, 3, 2, 5, 6, 7, 8, 9]
[1, 0, 2, 3, 4, 6, 7, 8, 9]
[1, 0, 2, 3, 6, 5, 7, 8, 9]
[1, 6, 3, 2, 0, 5, 7, 8, 9]
[1, 4, 3, 2, 0, 5, 7, 8, 9]
[1, 4, 3, 2, 5, 6, 7, 8, 9]
[1, 0, 3, 4, 6, 5, 7, 8, 9]
[1, 6, 4, 3, 2, 5, 7, 8, 9]
[1, 4, 6, 3, 2, 5, 7, 8, 9]
[1, 0, 2, 5, 3, 4, 6, 7, 8, 9]
[1, 0, 5, 2, 3, 4, 6, 7, 8, 9]
[1, 6, 4, 3, 0, 2, 5, 7, 8, 9]
[1, 4, 6, 3, 0, 2, 5, 7, 8, 9]
[1, 4, 3, 0, 2, 5, 6, 7, 8, 9]
[1, 0, 2, 3, 4, 6, 5, 7, 8, 9]
[1, 4, 3, 2, 0, 5, 6, 7, 8, 9]
[1, 6, 4, 3, 2, 0, 5, 7, 8, 9]
[1, 4, 6, 3, 2, 0, 5, 7, 8, 9]
```

```
In [18]: paths = nx.all_pairs_shortest_path(G)
         for path in paths:
             print(path)
```

```
(0, {0: [0], 1: [0, 1], 2: [0, 2], 3: [0, 3], 5: [0, 5], 4: [0, 1, 4], 6: [0, 1, 6], 7: [0, 5,
(1, {1: [1], 0: [1, 0], 3: [1, 3], 4: [1, 4], 6: [1, 6], 2: [1, 0, 2], 5: [1, 0, 5], 7: [1, 6,
(2, {2: [2], 0: [2, 0], 3: [2, 3], 5: [2, 5], 1: [2, 0, 1], 4: [2, 3, 4], 6: [2, 3, 6], 7: [2,
(3, {3: [3], 0: [3, 0], 1: [3, 1], 2: [3, 2], 4: [3, 4], 5: [3, 5], 6: [3, 6], 7: [3, 5, 7], 8
(4, {4: [4], 1: [4, 1], 3: [4, 3], 6: [4, 6], 0: [4, 1, 0], 2: [4, 3, 2], 5: [4, 3, 5], 7: [4,
(5, {5: [5], 0: [5, 0], 2: [5, 2], 3: [5, 3], 6: [5, 6], 7: [5, 7], 1: [5, 0, 1], 4: [5, 3, 4]
(6, {6: [6], 1: [6, 1], 3: [6, 3], 4: [6, 4], 5: [6, 5], 7: [6, 7], 0: [6, 1, 0], 2: [6, 3, 2]
(7, {7: [7], 5: [7, 5], 6: [7, 6], 8: [7, 8], 0: [7, 5, 0], 2: [7, 5, 2], 3: [7, 5, 3], 1: [7,
(8, {8: [8], 7: [8, 7], 9: [8, 9], 5: [8, 7, 5], 6: [8, 7, 6], 0: [8, 7, 5, 0], 2: [8, 7, 5, 2
(9, {9: [9], 8: [9, 8], 7: [9, 8, 7], 5: [9, 8, 7, 5], 6: [9, 8, 7, 6], 0: [9, 8, 7, 5, 0], 2:
```

```
In [ ]:
```