Advanced Topics In Software Engineering

NetworkX

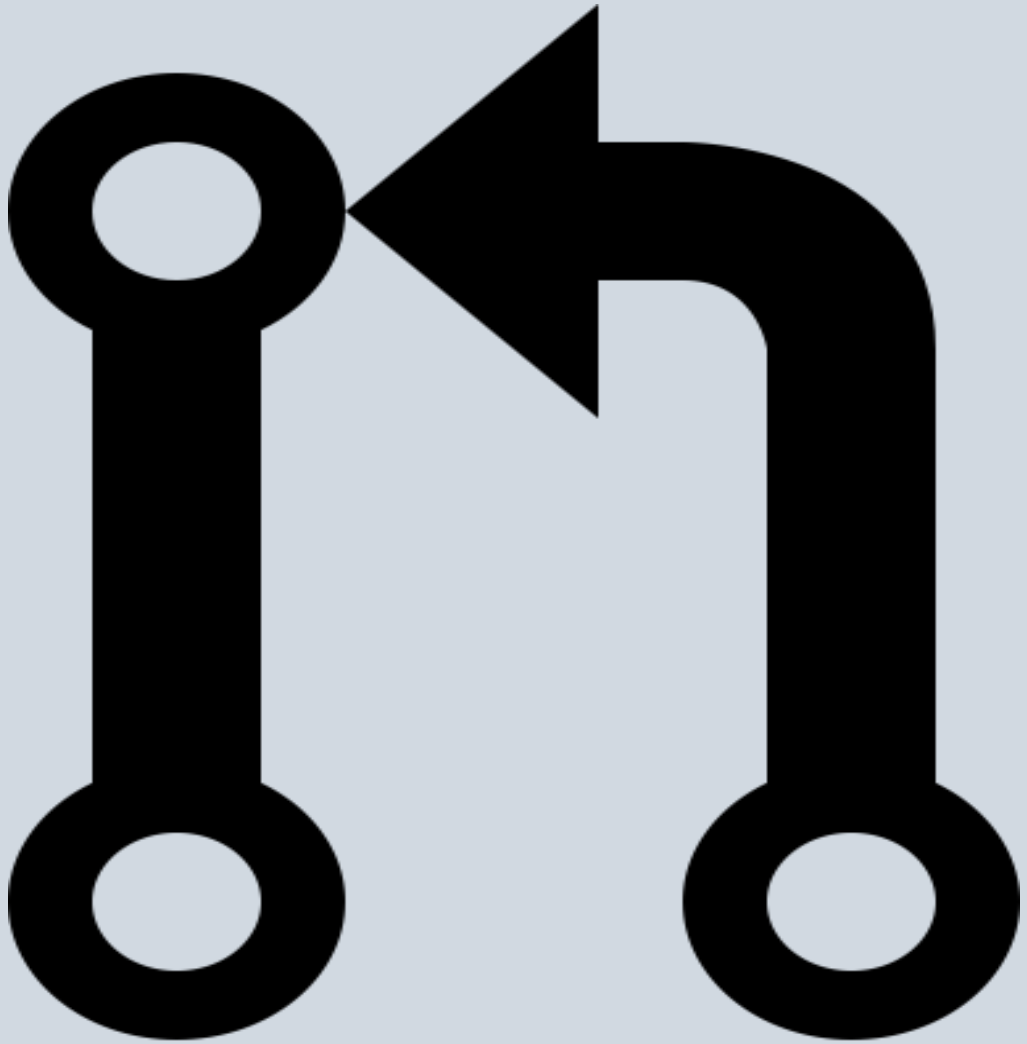
IMPLEMENTATION TEAM: OVERRIDERS

- KONSTANTINOS KARAKATSANIS
- THODORIS SOTIROPOULOS

Pull Requests

Open: 4

Merged: 3



Travis CI – Coveralls Checks(Passed)

NetworkX 1.10

(Negatively) Weighted Graphs (Merged)

```
Examples
--------
>>> G = nx.path_graph(4)
>>> nx.is_weighted(G)
False
>>> nx.is_weighted(G, (2, 3))
False

>>> G = nx.DiGraph()
>>> G.add_edge(1, 2, weight=1)
>>> nx.is_weighted(G)
True
```

```
Examples
--------
>>> G=nx.Graph()
>>> G.add_edges_from([(1, 3), (2, 4), (2, 6)])
>>> G.add_edge(1, 2, weight=4)
>>> nx.is_negatively_weighted(G, (1, 2))
False
>>> G[2][4]['weight'] = -2
>>> nx.is_negatively_weighted(G)
True
>>> G = nx.DiGraph()
>>> G.add_weighted_edges_from([
... ('0', '3', 3), ('0', '1', -5), ('1', '0', -2)])
>>> nx.is_negatively_weighted(G)
True
```

Johnson’s Algorithm (Merged)

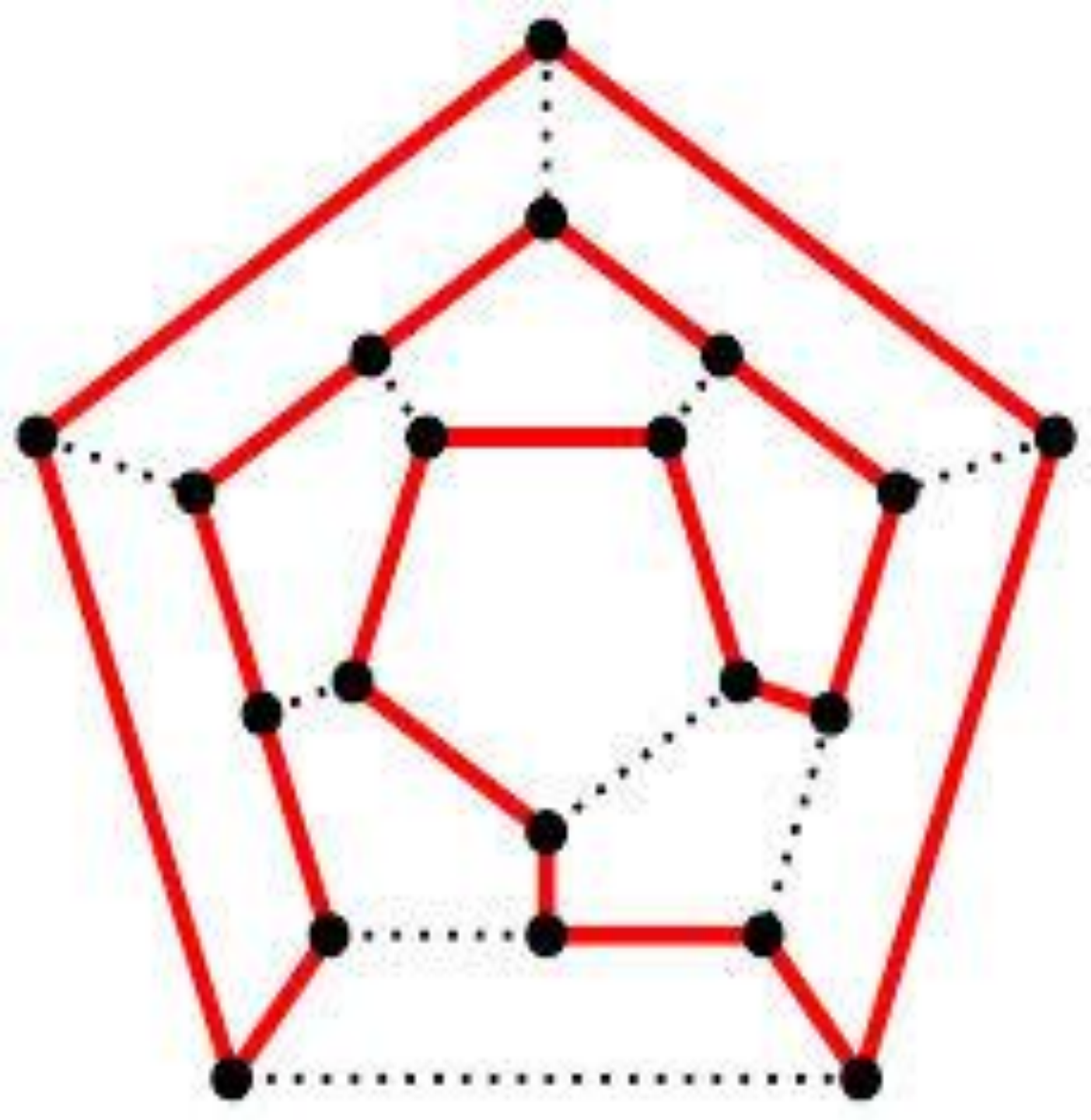
```
>>> import networkx as nx
>>> graph = nx.DiGraph()
>>> graph.add_weighted_edges_from([('0', '3', 3), ('0', '1', -5),
... ('0', '2', 2), ('1', '2', 4), ('2', '3', 1)])
>>> paths = nx.johnson(graph, weight='weight')
>>> paths['0']['2']
['0', '1', '2']
```

Generalized Shortest Path Problem (Merged)

```
def _dijkstra(G, source, get_weight, pred=None,
paths=None, cutoff=None, target=None):
```

- Duplicate code removed
- Distance ≠ $\sum weight(e)$

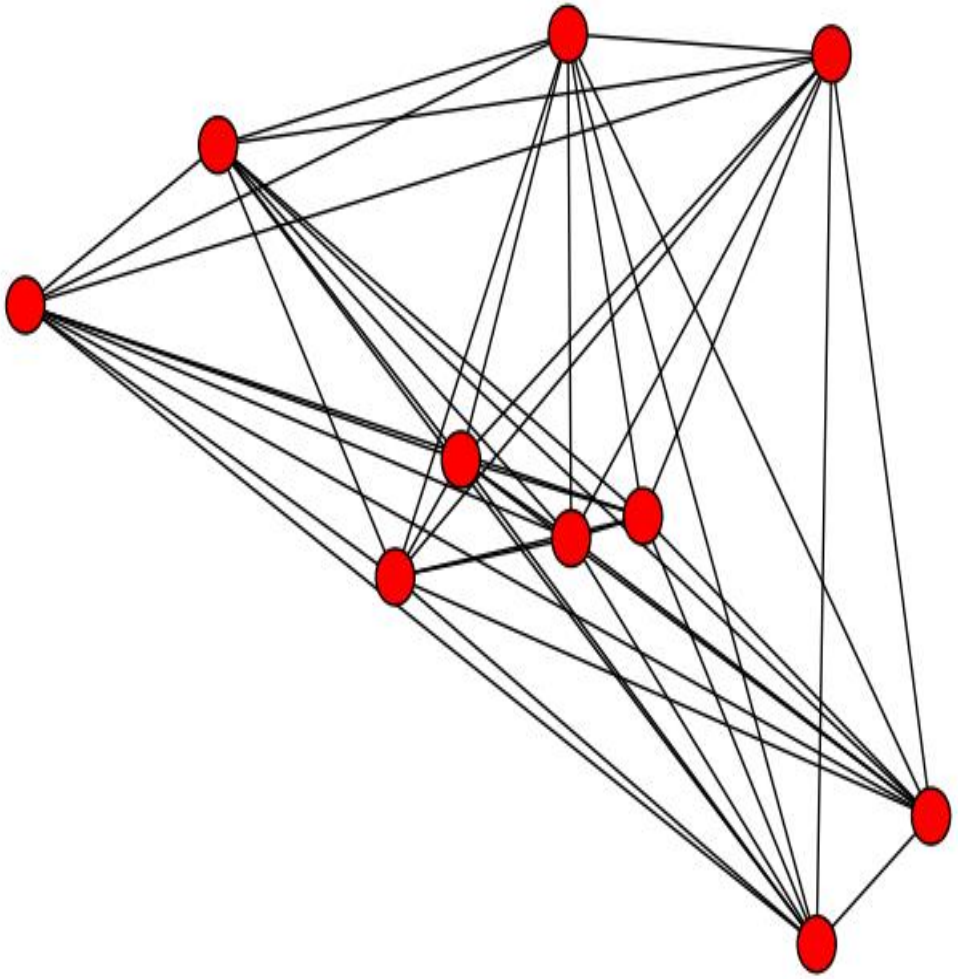
NetworkX 2.0



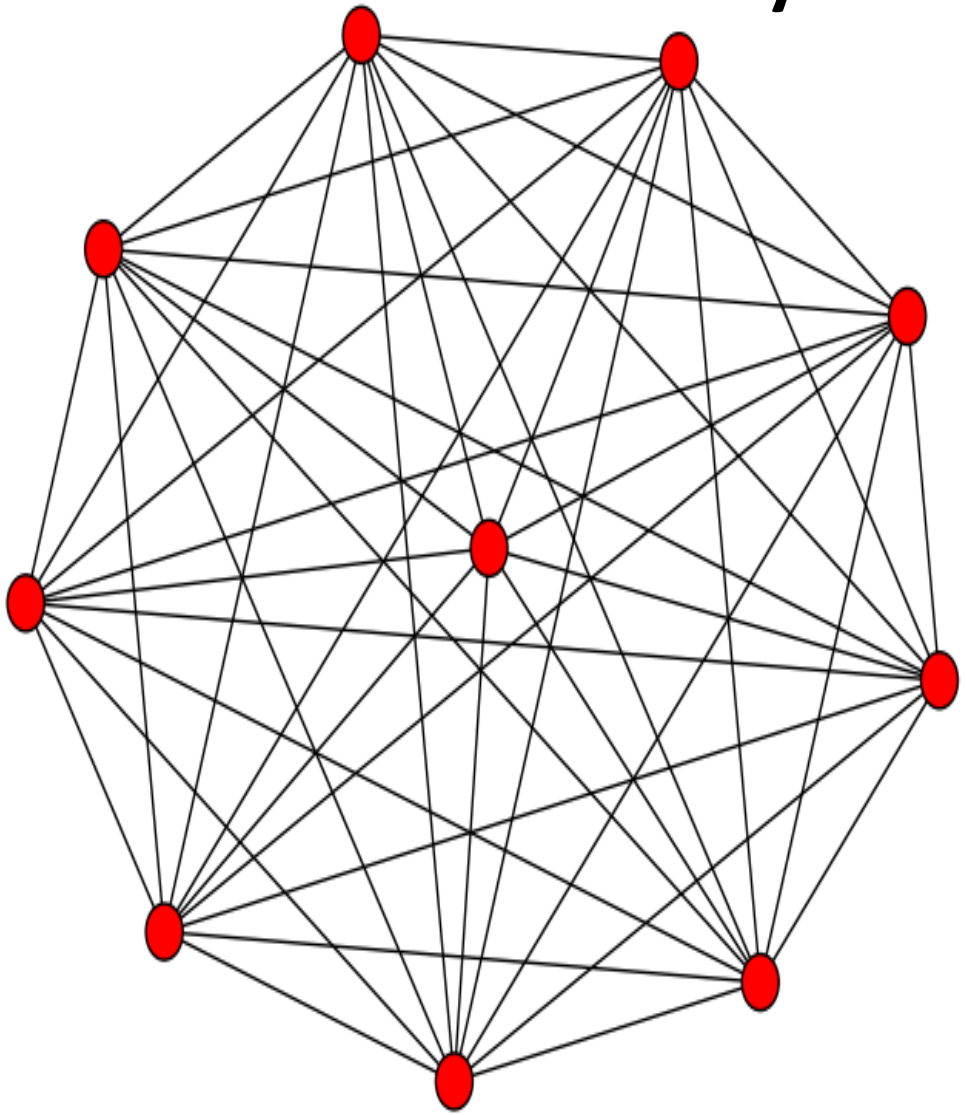
Travelling Salesman Problem (2 PR Open)

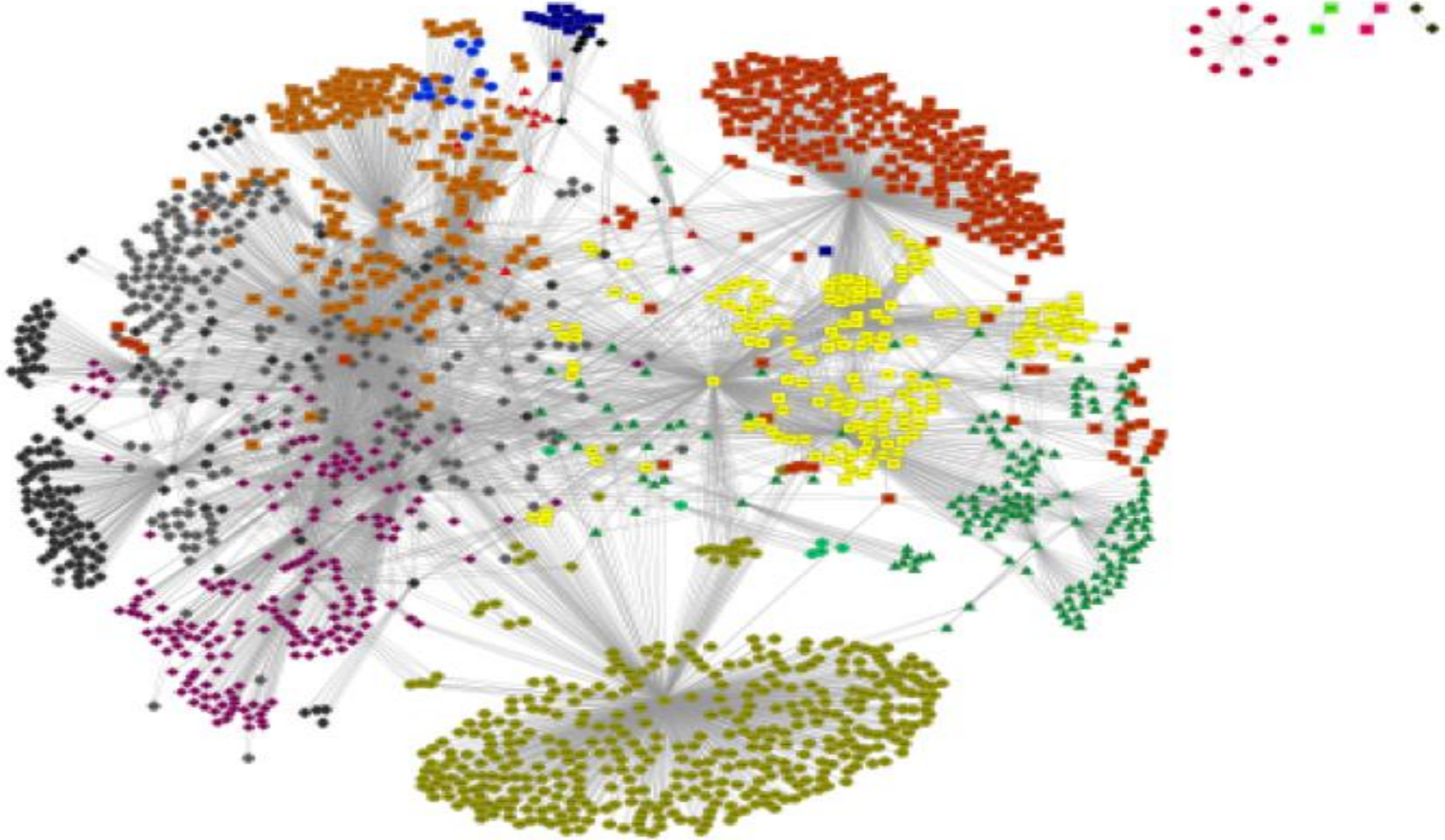
```
Examples
--------
>>> import networkx as nx
>>> G = nx.DiGraph()
>>> G.add_weighted_edges_from({('A', 'B', 3),
...                            ('A', 'C', 17), ('A', 'D', 14), ('B', 'A', 3),
...                           ('B', 'C', 12), ('B', 'D', 16), ('C', 'A', 13),
...                            ('C', 'B', 12), ('C', 'D', 4), ('D', 'A', 14),
...                            ('D', 'B', 15), ('D', 'C', 2)})
>>> cycle, weight = nx.simulated_annealing_tsp(G, 'D')
>>> cycle
['D', 'C', 'B', 'A', 'D']
>>> weight
31
>>> cycle, weight = nx.simulated_annealing_tsp(G, 'D', cycle=['D', 'B', 'A', 'C', 'D'])
>>> cycle
['D', 'C', 'B', 'A', 'D']
>>> weight
31
```

Random Layout



Kamada-Kawai Layout





Community Detection (1 PR open)

```
Examples
--------
>>> G = nx.Graph()
>>> G.add_edges_from([(1, 3), (1, 2), (2, 3),
... (3, 7), (7, 6), (6, 4), (6, 5), (5, 4), (7, 8),
... (8, 9), (9, 11), (9, 10), (10, 11), (8, 12),
... (12, 13), (12, 14), (13, 14)])
>>> comp = girvan_newman(G)
>>> comp
[([1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12, 13, 14]),
([1, 2, 3], [4, 5, 6], [7], [8], [9, 10, 11], [12, 13, 14]),
([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14])]
```

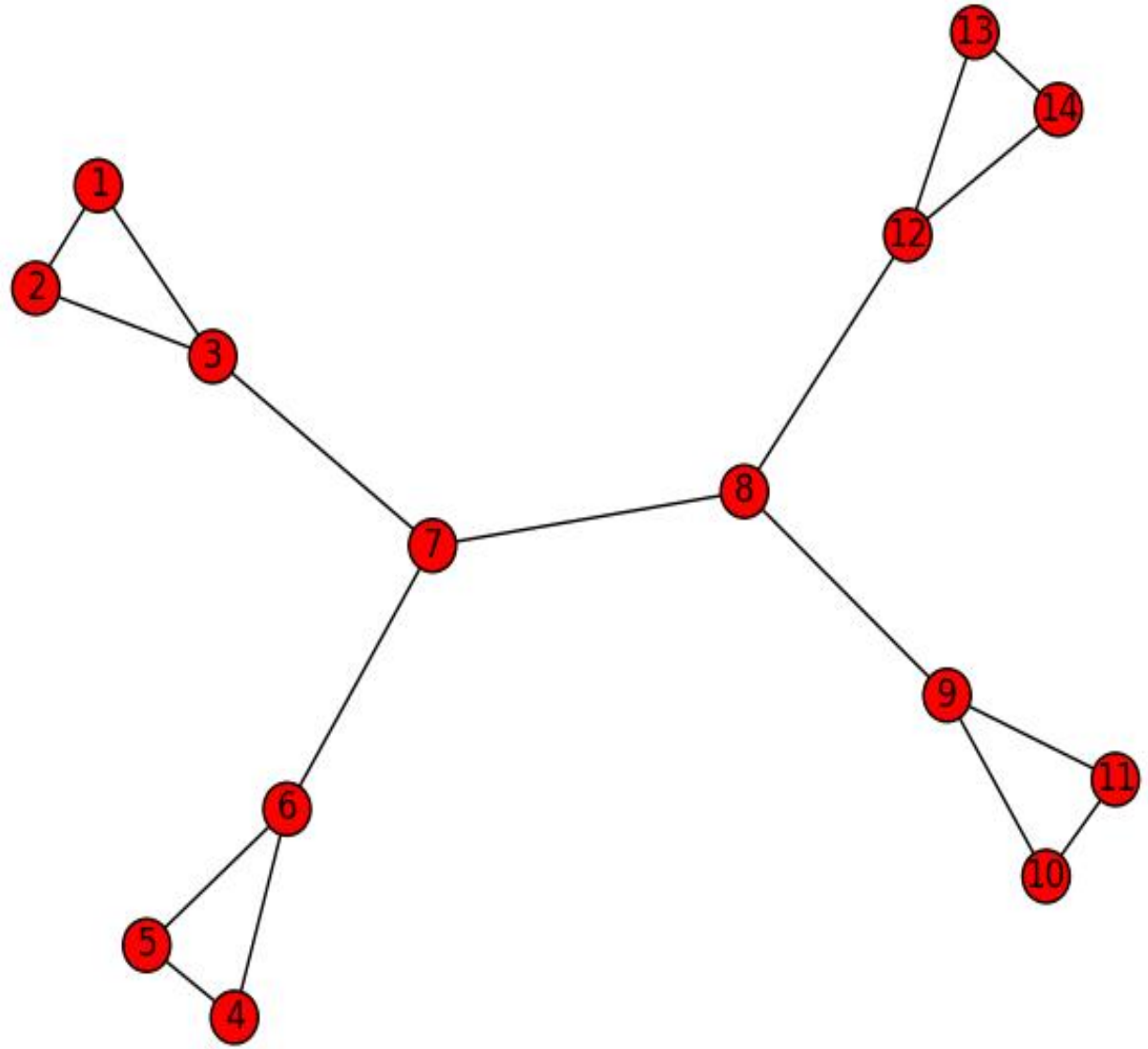
Community Detection (2 PR open)

Output:

[([1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12,13, 14]),

([1, 2, 3], [4, 5, 6], [7], [8], [9, 10, 11], [12, 13, 14]),

([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14])]



What did we gain?