Advanced Topics In Software Engineering

NetworkX

## IMPLEMENTATION TEAM: OVERRIDERS

- KONSTANTINOS KARAKATSANIS
- THODORIS SOTIROPOULOS

Some characteristics…

- Under BSD license
- 100% Python (93157 lines of code)
- Cross – Platform
- Easy to learn
- Well documented (more than 500 pages of documentation – many tutorials available)
- Well tested (more than 1800 unit tests, > 90% code coverage)

More for NetworkX...

- Graph Creation
- Graph Analysis
- Graph Visualization
- Export Data
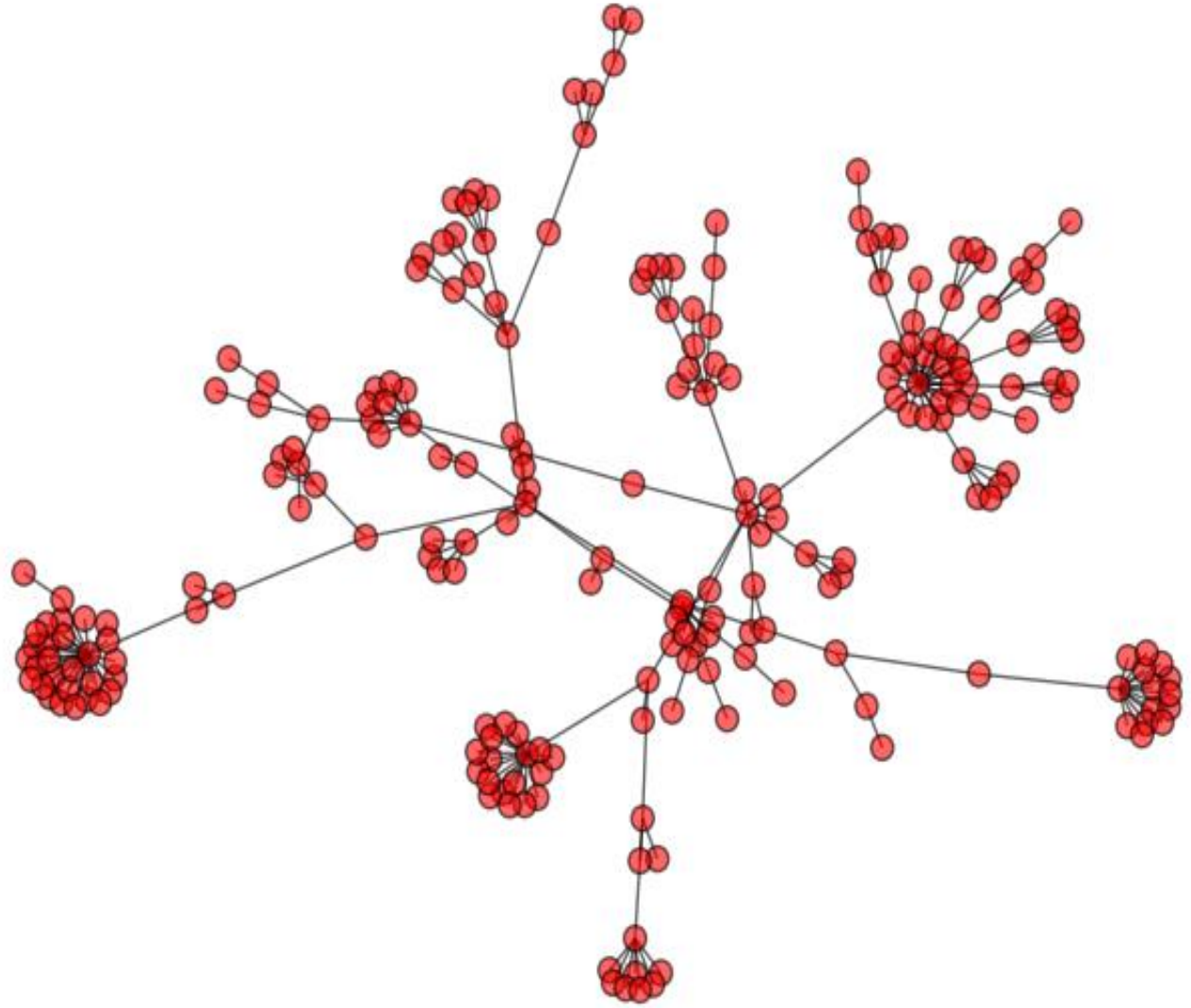
NetworkX Example (I) – Graph Creation

```python
import networkx as nx

graph = nx.Graph()
graph.add_node('Thodoris')
graph.add_node('Kostas')
graph.add_edge('Thodoris', 'Kostas')
```

NetworkX Example (II) – Graph Analysis

```python
import networkx as nx

number_of_nodes = 10
probability = 0.1
graph = nx.erdos_renyi_graph(number_of_nodes, probability)
closeness = nx.closeness_centrality(graph)
print(closeness[4])


Output: 0.401639344262
```
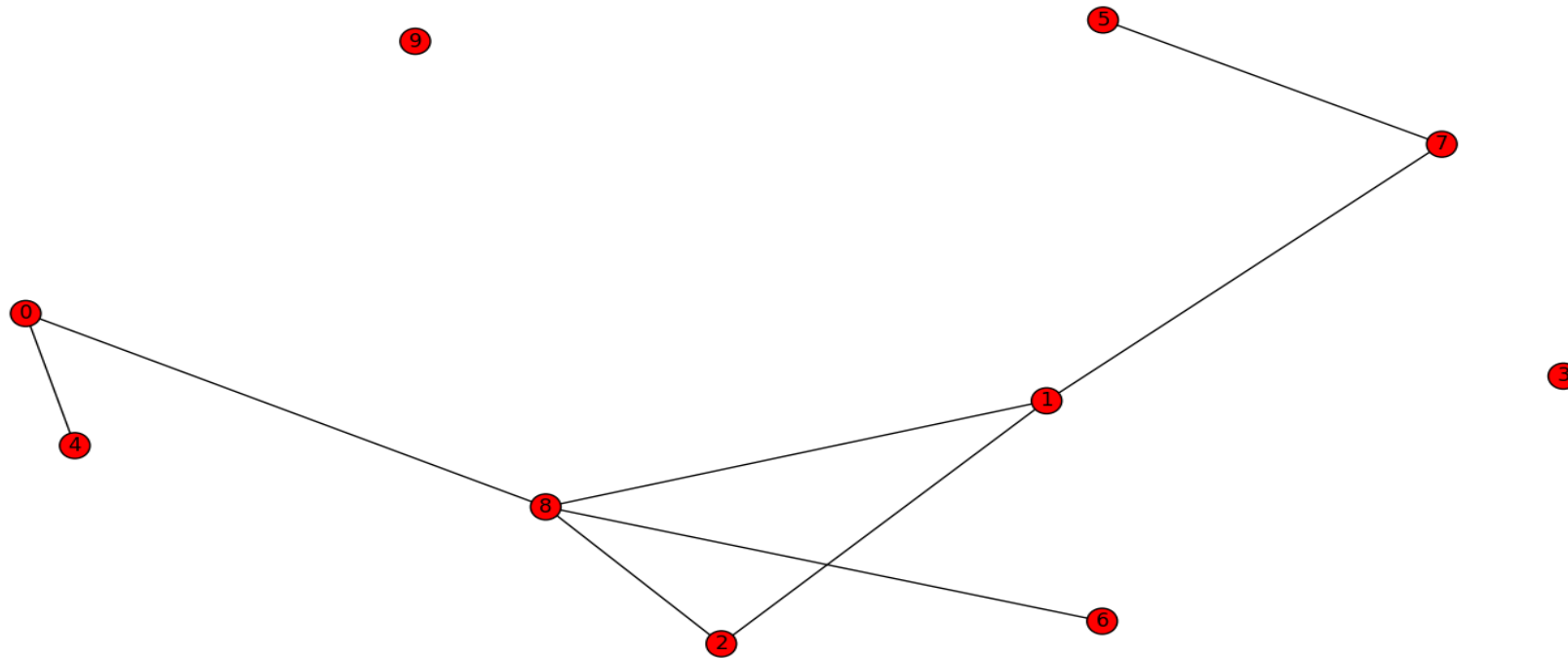
NetworkX Example (III) - Graph Drawing

```python
import networkx as nx
import pylab as plt

number_of_nodes = 10
probability = 0.1
graph = nx.erdos_renyi_graph(number_of_nodes, probability)
nx.draw_spring(graph)
plt.show()
```

NetworkX Example (III) - Output

Contribution

New Additions
- Algorithms for Community Detection (Girvan Newman, Louvain)
- Algorithms for Critical/Strongest path detection
- Checking existence of weights (negative weights)
- Directed-Force drawing

Improve current functionality
- Calculation of Average Shortest path length in a graph
- Calculation of graph's diameter

Growing Graphs
- Evolution of a graph over time (e.g. Albert Barabasi Graph)

Why NetworkX?