**Graph Measures Report**

In designing these classes, I opted for a simple and intuitive object-oriented approach.

The `Node` and `Edge`classes represent the basic elements of a graph, while the `Graph` class

encapsulates these elements and provides methods for manipulating them.

This design allows for easy expansion and modification,

aligning with principles of good software design.

The design uses the NetworkX library in Python,

which provides built-in functions for calculating centrality measures.

For Betweenness Centrality, the `nx.betweenness\_centrality` function is used,

which internally uses Dijkstra's algorithm or Bellman-Ford's algorithm to compute shortest paths

(geodesics) between all pairs of nodes.

Degree centrality: {1: 0.25, 2: 0.5, 3: 0.5, 4: 0.5, 5: 0.25}

Closeness centrality: {1: 0.4, 2: 0.5714285714285714, 3: 0.6666666666666666, 4: 0.5714285714285714, 5: 0.4}

Betweenness centrality: {1: 0.0, 2: 0.5, 3: 0.6666666666666666, 4: 0.5, 5: 0.0}