

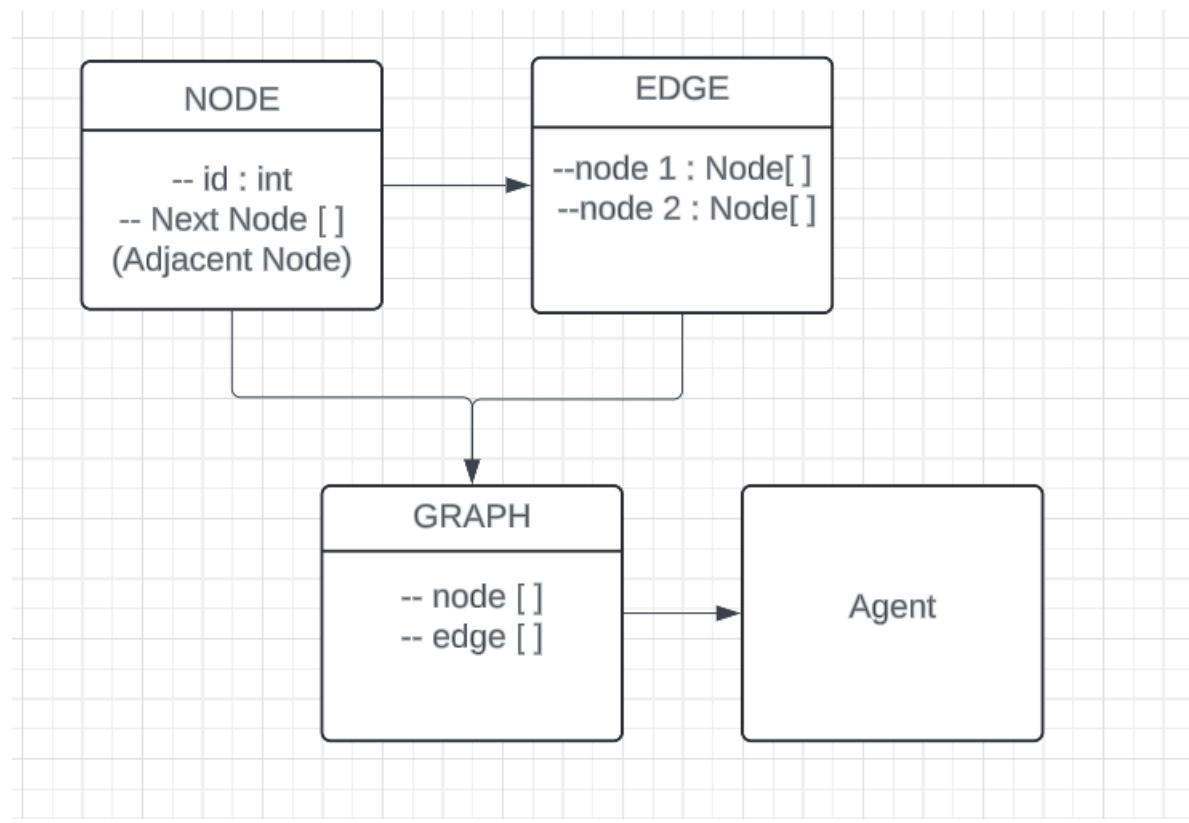
Artificial Intelligence  
(5COM2003-0105-2023)

Project:  
**Report on Graph Measures**

Name : HARDIK MATHUR

Student ID:22000375

## Implementation Of Graph



In this design, **Node** represents a node in the graph with each node having an identification **id** and a list of adjacent nodes. **Edge** connects two nodes together and is part of the graph. **Graph** contains list of nodes and edges and provides methods to add nodes and edges to the graph. I chose this design as it separates classes of node and edges making it easier to manage and manipulate graphs. In this design to make graph you have to create instances of Node and Edge and also use methods like `add_node` and `add_edge`.

## World Metrics

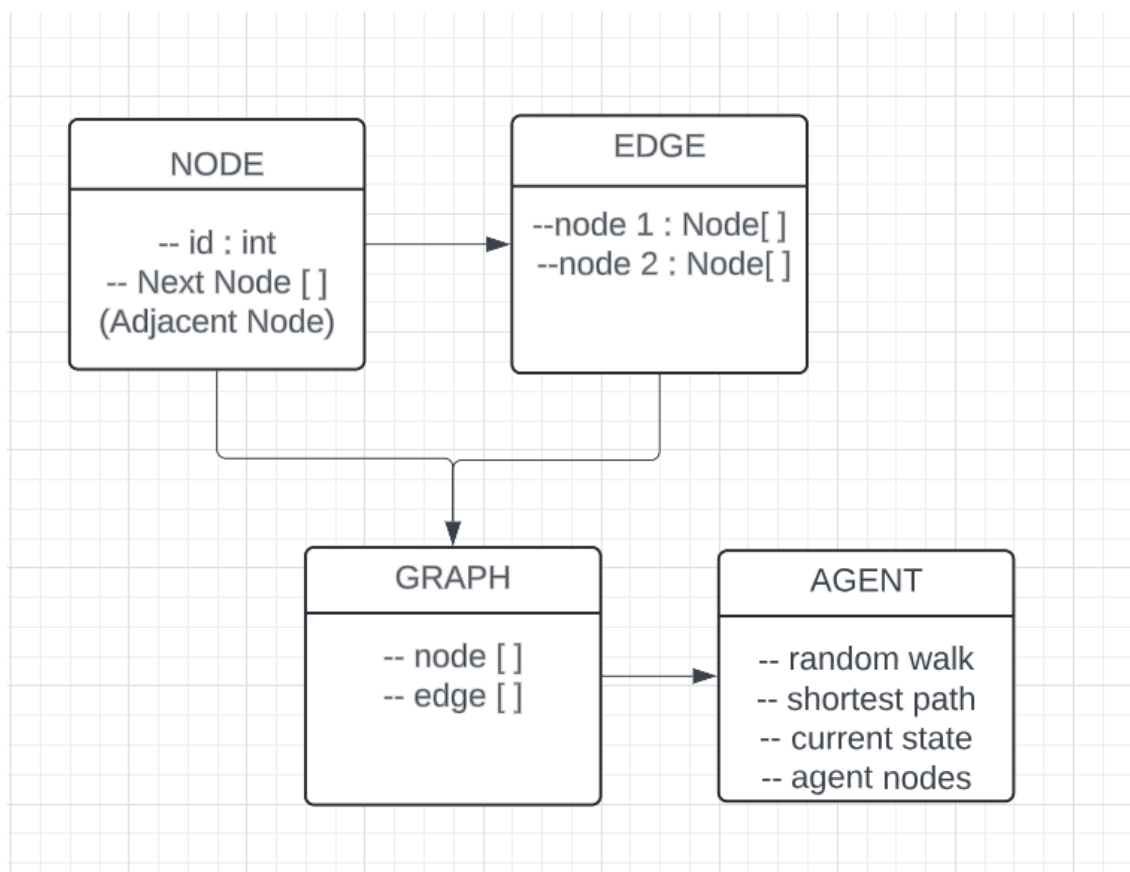
**Class Graph Metrics :** Class is designed to calculate three centrality metrics : degree centrality, closeness centrality and betweenness centrality.

**Degree Centrality :** Calculates the number of outgoing edges from the vertex.

**Closeness Centrality :** Uses Breadth – First Search (BFS) to find shortest distance from the vertex to all the other vertices, calculate closeness centrality on the reciprocal of total distance.

**Betweenness Centrality :** Implements BFS to find shortest paths for all vertex pairs and counts geodesics passing through vertex to calculate betweenness centrality.

## Agent Design



# Evaluation

```
Random Walk Results (First 10): [3, 2, 6, 3, 5, 3, 6, 7, 7, 2]
Shortest Path Results (First 10): [3, 2, 4, 3, 3, 3, 2, 3, 2, 2]
Average number of visited nodes in Random Walk: 4.439
Average number of visited nodes in Shortest Path: 2.825
Degree Centrality: {1: 3, 2: 2, 3: 2, 4: 3, 5: 2, 6: 2, 7: 2}
Closeness Centrality: {1: 0.1, 2: 0.090909090909091, 3: 0.090909090909091, 4: 0.1, 5: 0.090909090909091, 6: 0.0833333333333333, 7: 0.090909090909091}
Betweenness Centrality: {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0}
```

Evaluation on the basis of the above output provided .

Movement Mode	Average Node Visted (10 runs only)
Random Walk	4.439
Shortest Path	2.825

The main difference between average of random walk and shortest path is agent moves more randomly and covers more nodes in the graph leading to more exploration of the graph that is why the average of random walk is higher than the average of shortest path average.

Metric	Simulation Results	Graph Metrics
<b>Average Degree Centrality</b>	{1: 3, 2: 2, 3: 2, 4: 3, 5: 2, 6: 2, 7: 2}	{1: 3, 2: 2, 3: 2, 4: 3, 5: 2, 6: 2, 7: 2}
<b>Average Closeness Centrality</b>	{1: 0.1, 2: 0.090909090909091, 3: 0.090909090909091, 4: 0.1, 5: 0.090909090909091, 6: 0.0833333333333333, 7: 0.090909090909091}	{1: 0.1, 2: 0.090909090909091, 3: 0.090909090909091, 4: 0.1, 5: 0.090909090909091, 6: 0.0833333333333333, 7: 0.090909090909091}
<b>Average Betweenness Centrality</b>	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0}	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0}

Average Degree Centrality: Both align perfectly, indicating that their consistency in the graph within the number of nodes for each vertex.

Average Closeness Centrality: The values are same , representing average reciprocal of shortest path lengths to other nodes in the graph.

Average Betweenness Centrality: The values are 0, indicating that there is no vertex serves as a link between nodes .