

Mini Project: Graph Representation and Analysis

Instructions:

You are provided with two CSV files, one containing information about nodes in the graph (nodes.csv) and the other containing connections between nodes (connections.csv). Your task is to choose the appropriate data structure to represent this non-oriented graph. Once the graph is represented, follow these steps:

1. Graph Analysis Report:

- a. Create a main function that, after receiving the graph, generates a detailed report on the graph. This report should include:

i. Node Degree Calculation:

1. Compute the degree of each node in the graph.
2. Define and calculate the graph degree.

ii. Graph Connectivity:

1. Determine whether the graph is connected or not.

iii. Cycle Detection:

1. Design an algorithm to detect if there is a cycle in the graph.
2. Provide a step-by-step explanation of your algorithm.

iv. Shortest Path Calculation:

1. Implement a function that receives two nodes A and B (identified by their indices i, j), calculates the shortest path between the two points, and provides its value and the list of nodes to traverse.

v. Graph Illustration (Bonus):

1. Generate an illustration of the graph and the path in different colors.

vi. CSV Output Function:

1. Implement a function that receives a list of pairs of nodes A and B and writes the response (path value followed by the list of nodes to traverse) in a CSV file.

vii. Time Calculation:

1. Display the time required to calculate and write all paths at the end.

Example Graph:

Consider a graph with 6 nodes (1, 2, 3, 4, 5, 6) with the following 2D coordinates:

- Node 1: (0, 0)
- Node 2: (1, 2)
- Node 3: (2, 4)
- Node 4: (3, 1)
- Node 5: (4, 3)

- Node 6: (5, 0)

Connections (edges) between nodes:

- (1, 2)
- (1, 3)
- (2, 4)
- (3, 5)
- (4, 5)
- (5, 6)

Detailed Steps and Example Responses:

1. **Node Degree Calculation:**

Example output:

Node Degrees: {1: 2, 2: 2, 3: 2, 4: 2, 5: 3, 6: 1}

2. ****Graph Degree Calculation:****

Example output:

Graph Degree: 3

3. ****Graph Connectivity:****

Example output:

Graph is connected: True

4. ****Cycle Detection:****

Example output:

Graph has cycle: True

5. ****Shortest Path Calculation:****

Example output for nodes 1 and 6:

Shortest Path from 1 to 6: [1, 3, 5, 6] Path Length: 3

6. **Graph Illustration (Bonus): - Use a library like `matplotlib` to generate a visual representation of the graph and the path. 7. ****CSV Output Function:******

Example output:

CSV file with paths: Start,End,Path Length,Path 1,6,3,[1, 3, 5, 6] 2,5,2,[2, 4, 5]

8. ****Time Calculation:****

```
import time
```

```
def calculate_and_write_paths(graph, pairs, filename):
    start_time = time.time()
```

```
write_paths_to_csv(graph, pairs, filename)
end_time = time.time()
print(f"Time taken: {end_time - start_time} seconds")
```