CS101: Discrete Mathematics PROJECT-2

Name: Som Nainwal
Entry Number: 2023CSB1163

April 25, 2024

Question 3

Exploring Minimum Degree of Separation in Social Networks

Minimum Degree Of Separation
The minimum degree of separation in a network is the smallest number of edges needed to connect any pair of nodes. This concept is important in transportation systems, communication networks, and social networks.

Problem Statement:-

Given a directed graph representing a social network between students (nodes), determine the average minimum degree of separation between all possible pairs of nodes in the graph.

Solution:-

To find the minimum degree of separation in a directed graph, we will use breadth-first search (BFS). By traversing the graph from each node and keeping track of the minimum distance to all other nodes, we can determine the minimum degree of separation.

1. Read the data representing the directed graph from a CSV file.
2. Create a directed graph object using NetworkX library.
3. Implement a function to find the minimum degree of separation between two nodes in the graph using breadth-first search (BFS).
4. Implement a function to compute the average minimum degree of separation between all possible pairs of nodes in the graph.

5. Iterate over all possible pairs of nodes in the graph and calculate the minimum degree of separation for each pair.

6. Compute the average minimum degree of separation by dividing the total minimum degree of separation by the number of valid pairs.

Mathematical Background

- **Directed Graph:** A graph consists of a set of nodes (vertices) and a set of edges. In a directed graph, edges have a direction associated with them, indicating a one-way relationship between nodes.
- **Breadth-First Search (BFS):** BFS is a graph traversal algorithm used to explore nodes in a graph level by level. It starts at a given node (source) and explores all its neighbors at the current level before moving to the next level. BFS can be used to find the shortest path between two nodes in an unweighted graph.
- **Minimum Degree of Separation:** The minimum degree of separation between two nodes in a graph is the smallest number of edges that must be traversed to reach from one node to another.
- **Average Calculation:** To compute the average minimum degree of separation, the total minimum degree of separation is divided by the number of valid pairs. If no valid pairs are found, the average is considered to be zero.

Time Complexity

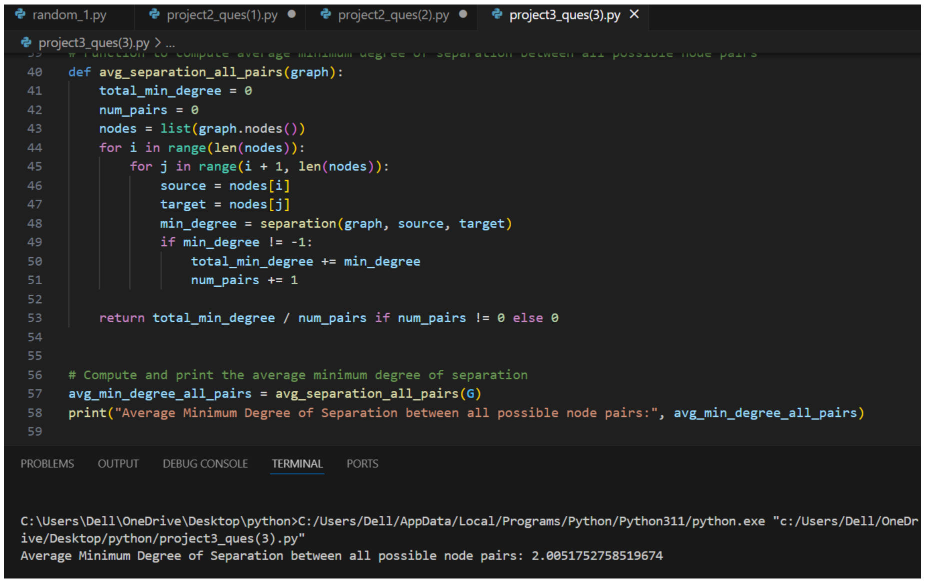
The time complexity of the algorithm depends on the graph traversal method used and the size of the graph. BFS have a time complexity of $O(V+E)$, where V is the number of vertices (nodes) and E is the number of edges in the graph.

Optimality Of The Code Using BFS:-

Level-based Exploration:
When using BFS, nodes are explored level by level, starting from the source node. This means that the first time the target node is encountered during the exploration, it guarantees that the source and target nodes have the minimum degree of separation. Since BFS explores nodes in increasing order of distance from the source node, it always finds the shortest path between the source and target nodes, ensuring optimality.

Queue Dequeuing Order:

In BFS, nodes are dequeued from the front of the queue. This ensures that nodes at the same level are explored before moving on to the next level. This order of exploration guarantees that the first time the target node is encountered during the BFS traversal, it corresponds to the minimum degree of separation between the source and target nodes.



```python
# Function to compute average minimum degree of separation between all possible node pairs
def avg_separation_all_pairs(graph):
    total_min_degree = 0
    num_pairs = 0
    nodes = list(graph.nodes())
    for i in range(len(nodes)):
        for j in range(i + 1, len(nodes)):
            source = nodes[i]
            target = nodes[j]
            min_degree = separation(graph, source, target)
            if min_degree != -1:
                total_min_degree += min_degree
                num_pairs += 1

    return total_min_degree / num_pairs if num_pairs != 0 else 0


# Compute and print the average minimum degree of separation
avg_min_degree_all_pairs = avg_separation_all_pairs(G)
print("Average Minimum Degree of Separation between all possible node pairs:", avg_min_degree_all_pairs)
```

```
C:\Users\Dell\OneDrive\Desktop\python>C:/Users/Dell/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Dell/OneDr
ive/Desktop/python/project3_ques(3).py"
Average Minimum Degree of Separation between all possible node pairs: 2.0051752758519674
```

Figure 1: Snapshot of the code and output

Hence, using BFS we found the average minimum nodes between two nodes as 2.005