Homework 3 Report

Beyza Türk
150220704

My code , implements several graph operations using data from Freebase. The operations include loading data from the given file, finding neighbors and printing them out, calculating degree centrality, and finding the shortest path between two entities. I have slightly changed the skeleton code's main function to be able to process command line arguments appropriately.

1. Using given tsv files to obtain data: The code begins by loading two types of data from given tsv files:

- mid2name.tsv: This file contains mappings of unique identifiers (MIDs) to entity names.
- freebase.tsv: This file contains relationships between entities in the Freebase knowledge graph.

2. Construction of graphs: The code represents the Freebase graph using an adjacency list , where each entry is a node, and relationships between entities are edges. The freebase.tsv file is used to construct the graph. For each line in the file, representing a relationship between two entries, nodes corresponding to the entries are created or retrieved from the graph_map. Bidirectional edges are established between the nodes, and relationship information is stored.

4. Operations of graph : The code provides functionality for three graph operations:

- **Finding Neighbors (part1):**
    - Given a MID, the code prints the number of neighbors and name of the neighbors with the relationship.
- **Calculating Degree Centrality (part2):**
    - Degree centrality measures the importance of a node based on the number of edges it has. findTopCentralEntities function creates a vector for every nodes centrality.
    - Bubble sort function compares the degree centrality of each entry.
    - After finding top 10, findTopCentralEntities prints the top 10 entities with the highest degree centrality.
- **Finding Shortest Path (part3):**
    - Using the breadth first search (BFS) algorithm and visited set, the code finds the shortest path between two given entries identified by their MIDs and signed them as visited. With using a queue, it prints the shortest path and size of path between the entities if exists.