

Final Project Write-Up

This rust program is designed to perform six-degree of separation algorithm do do comprehensive analysis on graph data, particularly focusing on the concepts of graph traversal and statistical analysis of graph properties. It uses the algorithm called Breadth-First Search(BFS) and statistical computations like degree distribution, mean, variance, standard deviation, mode, and graph density to achieve such purpose.

The dataset: euroroad: http://konect.cc/networks/subelj_euroroad/

It is the international E-road network, a road network located mostly in Europe. The network is undirected; nodes represent cities and an edge between two nodes denotes that they are connected by an E-road.

Key Features:

1. Graph Parsing: It is used to efficiently parses the given CSV file to construct a graph represented as an adjacency list
2. Breadth-First Search(BFS): Implements BFS algorithm to explore the graph and find the shortest path between two nodes(The start vertex and the goal vertex)
3. Statistical Analysis: In this program, it contains the functionality of calculating basic statistical measures such as degree distribution, mean, variance, standard deviation, mode, the amount of the connected components, and the graph density of the graph.
4. Modular Design: The program has three modules, bfs.rs, statistics.rs, and parser.rs, where the bfs.rs contains the Breadth-First search algorithm to compute the shortest path between two nodes. The parser.rs contains the code that is used to read the csv file and

convert it into the graph. The statistics.rs is the module that has the statistical analysis function

5. The program also contains two test module. The test.rs is used for making sure that the statistical analysis part is working correctly, while the graph_test.rs is used for testing the graphing of the program is working well.

How to run the program:

1. Prerequisites: Ensure Rust and Cargo are properly installed in the system
2. Compilation: navigate to the project's root directory and compile the program using the command: `cargo build --release`
3. Running the program using the command: `cargo run`
4. Testing: `cargo test`

Input Format:

The program is expected to receive a CSV file as input, where each line contains two values representing an edge in the graph, for example:

node1, node2

node3, node4

node5, node6

...

Program Output:

The program will output several different information as following:

- BFS output: shows the shortest path between two specified nodes, if there is one; If not, output the text says there is no path between these two nodes.

- Statistical Output: degree distribution, mean, standard deviation, variance, mode, graph density, number of connected components

Example:

Degree Distribution: {"1076": 1, "1115": 1, "1118": 4, "359": 2, "834": 1, "375": 4, "1023": 1, "1111": 1, "63": 1, "1152": 1, "585": 1, "482": 3, "600": 5, "499": 7, "417": 3, "166": 1, "644": 2, "896": 5, "649": 1, "195": 3, "653": 1, "906": 2, "918": 4, "702": 3, "1033": 2, "331": 2, "1090": 3, "687": 4, "1052": 1, "637": 2 ...

Distance between 316 and 330 is 9

Mean degree: 2.41396933560477

Variance: 1.4129564760613882

Standard Deviation: 1.1886784578099276

Mode: [2]

Number of Connected Components: 26

Graph Density: 0.002057944872638338