PROJECT REPORT

The project that I decided to work on is based on road network analysis of California using graph theory principles implemented in rust. The implementation tries to answer the following questions:
 1. Average distances between pairs of vertices to understand the connectivity and accessibility of different regions in California.
 2. Degree distributions to analyze the connectivity patterns of intersections and endpoints in the road network.

CODE STRUCTURE

The project is structured into several modules within the src directory, each responsible for different aspects of the program:

Modules

1. Parser (parser.rs)-
   Purpose: manages the road network data's reading and parsing from text files.
   Key Functions:

- read_file(path: &str) -> Result<Vec<(usize, usize)>, Error>: Reads the road network data from a file and returns a list of edges represented as tuples of vertices.
- find_max_index_parallel(path: &str) -> Result<usize, Error>: Determines the maximum vertex index to help in constructing the graph correctly.

2. Graph (graph.rs):
   Purpose: Provides the data structures and methods to represent and manipulate the graph.
   Key Functions:

- create_undirected_optimized(n: usize, edges: &[(usize, usize)]) -> Graph: Constructs an undirected graph from a list of edges, optimizing the structure for quick access and modifications.

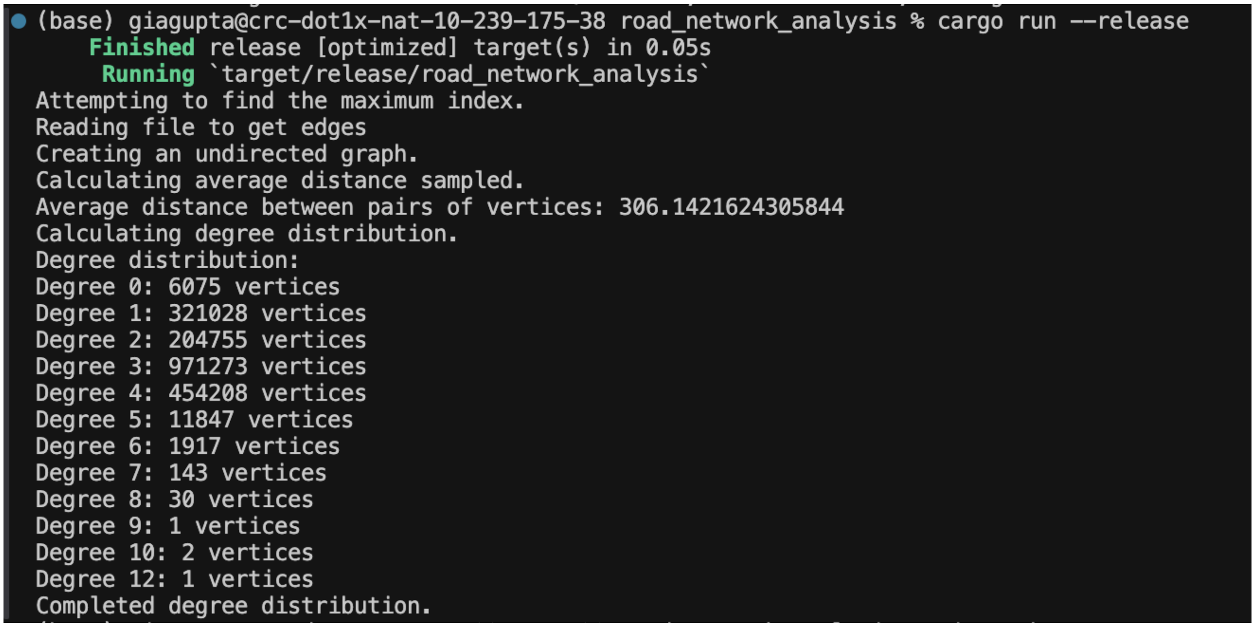
3. Analysis (analysis.rs):
   Purpose: Contains functions to compute various analytical metrics on the graph.
   Key Functions:

- average_distance_sampled(graph: &Graph, samples: usize) -> f64: Calculates an estimate of the average distance between pairs of vertices using a sampling method.
- degree_distribution(graph: &Graph) -> HashMap<usize, usize>: Computes the degree distribution of the graph, showing how many vertices have each possible degree.

→ The project also contains a test module which is necessary for the system's functionality to be accurate and reliable. It includes automated tests intended to confirm that every function—including data parsing, graph building, and analytical computations—is implemented correctly across a range of modules. Regular testing helps to identify regressions or errors introduced during development, which leads to faster updates and maintenance. It also increases confidence in the project's outputs.

OUTPUT

Average Distance Between Vertices: This function estimates the average distance between vertices and gives information about how dispersed and connected the road network is.
Degree Distribution: Provides the graph's degree distribution as a formatted list that indicates the degree and the quantity of vertices that possess it. This measure is essential to comprehending the road network's topology.



```
● (base) giagupta@crc-dot1x-nat-10-239-175-38 road_network_analysis % cargo run --release
      Finished release [optimized] target(s) in 0.05s
       Running `target/release/road_network_analysis`
  Attempting to find the maximum index.
  Reading file to get edges
  Creating an undirected graph.
  Calculating average distance sampled.
  Average distance between pairs of vertices: 306.1421624305844
  Calculating degree distribution.
  Degree distribution:
  Degree 0: 6075 vertices
  Degree 1: 321028 vertices
  Degree 2: 204755 vertices
  Degree 3: 971273 vertices
  Degree 4: 454208 vertices
  Degree 5: 11847 vertices
  Degree 6: 1917 vertices
  Degree 7: 143 vertices
  Degree 8: 30 vertices
  Degree 9: 1 vertices
  Degree 10: 2 vertices
  Degree 12: 1 vertices
  Completed degree distribution.
```

Result

The project produced an average calculated distance of 306.14 between vertices, indicating a widely dispersed network typical of large-scale road networks. Significant insights were gleaned from the degree distribution: most vertices have one or two connections (321,028 and 204,755 respectively), typical for linear road segments and simple intersections, while 6,075 vertices are isolated (degree 0), indicating unused or endpoint sections. Higher degree vertices (up to 12) draw attention to intricate intersections or significant network hubs that are essential for managing traffic flow and urban planning. This analysis aids in planning network expansions and improving network connectivity in addition to helping to comprehend the current network structure.