DS 210 Final Project Report
Crime Data from 2020 to Present (Los Angeles Open Data)

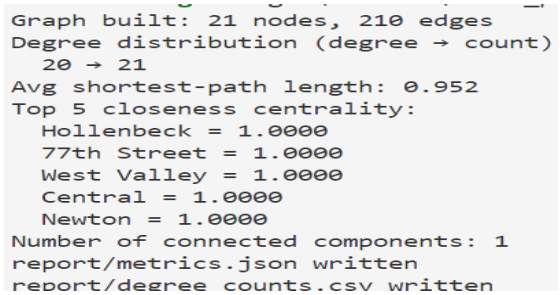
A. Project Overview
Goal: Model how criminal activity “spreads” across L.A. neighborhoods by constructing and analyzing a graph.
Dataset: “Crime Data from 2020–Present” (LA Open Data; ~ 1,000,000 records of timestamp, crime type, and AREA_NAME).

B. Data Processing
Python (scripts/0_preprocess.py): load the raw CSV with pandas, extract DAY and AREA_NAME columns, drop duplicates → data/day_area.csv.
Rust (src/main.rs): read data/day_area.csv with csv::ReaderBuilder, parse DAY into NaiveDate, bucket neighborhoods per day.

C. Code Structure
scripts/0_preprocess.py: pandas-based deduplication.
src/main.rs:
– bfs_distances: BFS to compute a distance map from one node.
– main(): builds an undirected petgraph::UnGraph<String, ()>, adds edges between neighborhoods co-occurring on the same day, computes degree distribution, average shortest-path length (via repeated BFS), closeness centrality (top 5), connected components, and writes report/metrics.json and report/degree_counts.csv.
report/python.py: uses pandas/matplotlib to plot the log-log degree distribution from degree_counts.csv.

D. Tests
After adding unit tests in src/main.rs, you can run:
cargo test
Output:
test test_bfs_chain … ok (# verifies BFS on a 3-node chain)
test test_connected_components … ok (# two isolated nodes → 2 components)

E. Results



- F. Usage
  Prerequisites: Python 3.x (pandas, matplotlib), Rust (1.70+).
  - Preprocess raw data:
    py -3 scripts/0_preprocess.py
  - Build & run analysis:
    cargo build --release
    cargo run --release
    (approx. 5 seconds runtime)
  - Generate plot:
    py -3 report/python.py