

Isabella Stravoiu

U33615769

## Bitcoin Transaction Network Analysis

### Project Overview:

My project looked at the structure and connectivity of the Bitcoin transaction network. It explores how wallets are connected within the given transactions, locates the most influential wallets, estimates six degrees of separation in network, and displays degree distribution in bar chart.

### Dataset:

Source: Elliptic Bitcoin Transaction Dataset

File: txs\_edgelist.csv

Size: 234,000 transactions and 200,000 wallets

Link:

[https://drive.google.com/file/d/1Q2yG\\_CIDvfdGP-fKVPSw979EYgQukjz5/view?usp=drive\\_link](https://drive.google.com/file/d/1Q2yG_CIDvfdGP-fKVPSw979EYgQukjz5/view?usp=drive_link)

Data Processing: I was able to load the CSV file into Rust, each row represented transaction between two wallet IDs which were marked as txID1 and txID2. There was no clearing necessary only making sure that it was excluded from FitHUB by using .gitignore.

### Code Structure:

Modules:

- Main.rs - Loads data, builds graph, run and generates analysis
- Graph.rs - Constructs Graph struct, edge insertion, degree distribution, and top-k hubs
- bfs.rs - Works through BFS algorithm to find shortest path
- tests.rs - Runs tests for graph and BFS

Key Functions:

- Graph Struct: Store transaction network within an adjacency list
- Add\_edge: Adds undirected edges in between wallets
- Degree\_Distribution: How many wallets have each degree counter
- Top\_k\_Nodes: Find top k wallets that have highest connectivity
- Shortest\_path: Find shortest path using BFS

### Main Workflow:

1. Load transactions
2. Build graph
3. Compute degree distribution
4. Compute top nodes
5. Estimate six degrees
6. Plot degree distribution
7. Run tests

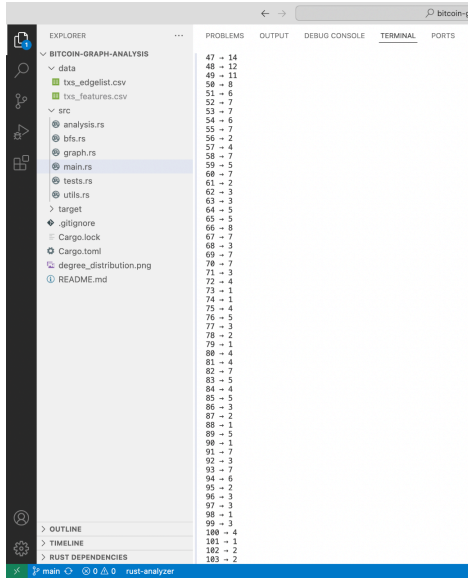
### Tests:

```
running 4 tests
test tests::tests::test_add_edge_and_counts ... ok
test tests::tests::test_neighbors ... ok
test tests::tests::test_shortest_path ... ok
test tests::tests::test_top_k_nodes ... ok
```

- test\_add\_edge\_and\_counts : Tests that node and edge counts are accurate after adding transactions
- Test\_neighbors : Tests if neighbor of node are correctly taken
- Tests\_shortest\_path : Make sure BFS is using the correct transaction in between the wallets

### Results:

Output-



**Interpretation:** Through these results it is shown the typical hub behavior between the highly connected wallets and many participants. The results also show that the average shortest path is highly connective and suggest an extremely dense bitcoin network system.

### Usage Instructions:

- Ensure txs\_edgelist.csv inside the data folder
- First build code by : cargo build --release
- Then run code : cargo run
- For tests use : cargo test
- The expected run time is 1 minute

### AI assistance:

ChatGPT ▾

Share

My data file is too big to commit to github, how can I use .gitignore?

Great question, Isabella — this is **super common** when working with large datasets, and you're smart to handle it the right way!

Here's a clear step-by-step guide:

✔ Step 1 → Create a `.gitignore` file

In the root of your project folder (same level as `Cargo.toml`), create a file named:

```
.gitignore
```

✔ Step 2 → Add the data folder or files to `.gitignore`

Inside `.gitignore`, add:

```
data/
```

✔ Step 2 → Add the data folder or files to `.gitignore`

Inside `.gitignore`, add:

```
data/txs_features.csv
```

OR if you only want to ignore the big file:

```
data/txs_features.csv
```

This tells Git:

```
ignore this file or folder — don't track it, don't try to commit it.
```

✔ Step 3 → Remove already tracked file from Git

If you already added or committed the big file, you need to untrack it:

```
git rm --cached data/txs_features.csv
```

✔ Step 4 → Commit the `.gitignore` change

```
git add .gitignore
git commit -m "Add .gitignore to exclude large data files"
git push
```

✔ Step 5 → Make sure your code works without the data on GitHub

✔ Keep a **local copy** of your data on your machine

✔ In your README, include a note like:

```
Note: The data files are excluded from the repository due to size limits. Please place th
```

Summary

Ask anything