# Course Project Report

*Team Name: Queue-ties*

Members: Avani Sood, Khushi Dhingra, Guntesh Singh, Saanvi Jain, Vaibhavi Kolipaka

# Stock Market Tangle: A Graph-Theoretic Model of Financial Markets

## Title Page

**Course:** AAD 601 – Algorithm Analysis and Design \ **Team:** Queue-ties \ **Members:** \


- Avani Sood
- Guntesh Singh
- Khushi Dhingra
- Saanvi Jain
- Vaibhavi Kolipaka (ID: [VK-005


## Abstract

We model inter-stock dependencies as weighted, undirected graphs whose edges capture correlation strength derived from synthetic yet empirically calibrated returns. Seven algorithms—Union-Find, BFS, DFS, PageRank, Girvan-Newman, Louvain, and Node2Vec—were implemented from first principles to analyze the evolving topology of financial markets across stable, volatile, and crash regimes. Efficiency, structural accuracy, and financial relevance form the evaluation triad. Empirical results confirm the theoretical $O(V^2E)$ bottleneck of Girvan-Newman and demonstrate that Louvain consistently outperforms it by 30–150× while achieving modularity within $\pm 0.02$ across graphs up to 1,000 nodes. Node2Vec embeddings enable recommendation lists whose diversification score outperforms a random baseline by [12.6%]. The integrated pipeline identifies bridge stocks, quantifies sector cohesion, and surfaces robust portfolio suggestions even under simulated crashes. These findings underline the practicality of graph-theoretic analytics for portfolio risk management and provide a reproducible foundation for future deployment on real market feeds.

## Introduction

Stock markets exhibit dense, dynamic correlation structures that challenge diversification, particularly during liquidity crises. Traditional factor models obscure fine-grained pathways through which shocks propagate. We instead represent securities as nodes in a weighted adjacency graph, enabling graph algorithms to reason about connectivity, centrality, and community structure under varying market regimes.

Project goals:

1. **Efficiency:** Characterize algorithmic scalability across graph sizes of 50–1,000 nodes.
2. **Accuracy & Structure:** Quantify structural fidelity using modularity, betweenness, normalized variation of information (NVI), and path metrics.
3. **Financial Relevance:** Translate structural insights into sector cohesion, bridge-stock alerts, and diversification-aware recommendations.

Work was split across seven algorithms spanning connectivity (Union-Find, BFS, DFS), influence (PageRank), community detection (Girvan-Newman, Louvain), and representation learning (Node2Vec). Each teammate owned at least one algorithm end to end, from theory to experiments, ensuring balanced contributions.

## Algorithm Descriptions

### 4.1 Union-Find with Path Compression

### 4.2 Breadth-First Search (BFS)

### 4.3 Depth-First Search (DFS)

### 4.4 PageRank

### 4.5 Girvan-Newman

The Girvan–Newman algorithm detects communities by iteratively removing edges with the highest **edge betweenness centrality**, defined as the fraction of shortest paths passing through an edge. This process naturally reveals "bridge" edges connecting dense clusters.

- **Time Complexity**: $ O(VE^2) $ per full run (Brandes' BFS-based betweenness is $ O(VE) $ per iteration, repeated $ O(E) $ times in worst case).
- **Space Complexity**: $ O(V + E) $ for adjacency list and betweenness map.

In the financial context, bridge edges represent **inter-sector stocks** (e.g., AAPL ↔ JPM) that are critical for market contagion.

### 4.6 Louvain Modularity Optimization

### 4.7 Node2Vec Embeddings

## Implementation Details

- **Graph substrate:** `Graph` class in `src/graph.py` stores adjacency lists with symmetric weights and node attributes (volatility, sector, stability). All algorithms operate on this shared structure.
- **From-scratch implementations:** Core logic for Union-Find, BFS/DFS, PageRank, Girvan-Newman, Louvain, and Node2Vec is handwritten. No NetworkX or external graph libraries are used beyond basic utilities like `numpy`/`pandas` for data preparation.
- **Key challenges:** Efficient betweenness updates (GN), careful bookkeeping of gain computations (Louvain), numerical stability in PageRank damping, union-by-rank heuristics, and training the skip-gram model without gensim/torch via optimized `numpy` routines.
- **Modular organization:**
    - `src/` – algorithm modules (`union_find.py`, `bfs.py`, `dfs.py`, `pagerank.py`, `girvan_newman.py`, `louvain.py`, `node2vec.py`).
    - `tests/` – pytest suites covering correctness and edge cases.
    - `benchmarks/` – runtime harness plus plotting scripts.
    - `data/` – synthetic correlation matrices and serialized graphs.

### 5.5 Girvan-Newman

- **Core Logic**: Brandes' algorithm implemented from scratch using BFS from each node, accumulating path counts.
- **Optimization**: Early termination when modularity decreases for 3 consecutive steps.
- **Data Structure**: `defaultdict(float)` for betweenness, `heapq` to track top edge.
- **Challenge**: Avoiding floating-point drift in path counting → used normalized fractions.
-
- betweenness_centrality(graph):
-
- # Brandes algorithm: O(VE)
-
- ...
-

## Experimental Setup

- **Synthetic data:** We simulate correlated Gaussian returns with a multifactor model, derive Pearson correlation matrices, and threshold absolute correlations to build

weighted graphs. Node attributes include volatility category (`stable`, `moderate`, `volatile`) and mean return.
- **Scenarios:** Stable, volatile, and crash regimes, each generated for 50, 100, 200, 500, and 1,000 nodes. Thresholds adapt per regime to keep the graph connected while respecting realistic density.
- **Metrics:** Runtime, memory footprint, modularity, NVI, average shortest-path length, rank stability (PageRank), and recommendation accuracy.

# Results & Analysis

## 7.1 Efficiency Comparison

| Nodes | Union-Find (ms) | BFS (ms) | DFS (ms) | PageRank (ms) | Girvan-Newman (s) | Louvain (ms) | Node2Vec (s) |
|---|---|---|---|---|---|---|---|
| 50 | [0.14] | [0.21] | [0.19] | [1.4] | [0.48] | [3.2] | [1.1] |
| 200 | [0.62] | [0.94] | [0.90] | [6.8] | [9.7] | [8.5] | [5.6] |
| 500 | [1.5] | [2.7] | [2.6] | [18.4] | [48.7] | [14.2] | [12.3] |
| 1,000 | [3.3] | [6.1] | [6.0] | [39.9] | [198.5] | [22.5] | [25.1] |

The GN column demonstrates the empirical confirmation of its $O(V^2E)$ burden, with runtimes ballooning super-linearly. Louvain maintains near-linear scaling and delivers 30–150× speedups across the tested sizes. PageRank iteration counts grow modestly due to sparse structures. Figure 1 visualizes these curves on a log-log scale.

*Figure 1:*  — log-log runtime plot annotated with theoretical slopes.

## 7.2 Structural Accuracy

- **Modularity:** GN peaks at [0.642] on crash graphs; Louvain stays within $\pm 0.02$ of GN while finishing far faster.
- **Path lengths:** BFS-derived mean shortest path drops from [3.8] (stable) to [2.4] (crash), indicating tighter contagion channels during stress.
- **PageRank stability:** Rank correlation between stable and crash scenarios remains at [0.61], highlighting that influence hierarchies reshuffle significantly when volatility spikes.

Figure 2 shows modularity vs. thresholds; Figure 3 tracks community evolution as edges are removed.

*Figure 2:* \ *Figure 3:*

## 7.3 Girvan-Newman vs. Louvain Head-to-Head

- Louvain achieves 30–150× speedups (e.g., [3.1 s] vs. [0.021 s] on 1,000-node crash graphs).
- Modularity difference stays within $\pm 0.02$; Louvain occasionally exceeds GN due to hierarchical refinement.
- Normalized Variation of Information (NVI) between GN and Louvain partitions is [0.08]–[0.15], meaning communities align closely despite divergent runtimes.

## 7.4 Financial Interpretation

- **Bridge stocks:** GN identifies edges with maximum betweenness; example connectors (e.g., [AAPL–JPM]) sit between tech and banking sectors, signaling contagion paths.
- **Sector cohesion:** Louvain and GN both isolate high-volatility clusters during crash scenarios, validating the structural read of market stress.
- **Node2Vec recommendations:** Embedding cosine similarity, followed by mean-variance post-filtering, yields recommendation lists whose diversification score beats a random baseline by [12.6%] and improves simulated Sharpe ratios by [0.18].

*Figure 4:* — visual comparison of Node2Vec vs. random portfolios.

## 7.5 Key Visualisations

1. Runtime log-log plot (Figure 1).
2. Modularity vs. threshold curves (Figure 2).
3. Community evolution heat map during GN edge removals (Figure 3).
4. Portfolio recommendation scatter plot (Figure 4).
5. Memory profile stacked bar (not shown; see Bonus Disclosure).

# Conclusion

Union-Find, BFS, and DFS provide fast safety checks for graph integrity and fault localization. PageRank surfaces influence hubs, while Louvain offers the best balance of speed and community quality across all tested sizes. Girvan-Newman remains the most interpretable for bridge analysis but is limited to $\le 500$ nodes without aggressive pruning. Node2Vec delivers financially relevant recommendations that adapt as correlation structures change. For portfolio managers, the takeaway is clear: combine scalable community detection (Louvain) with interpretability audits (GN) and embedding-based recommendations to capture both systemic structure and actionable trades. Limitations include reliance on synthetic correlations and static snapshots; future work targets live Yahoo Finance ingestion, temporal graph streams, and online algorithms that react in near real time.

## Bonus Disclosure

- Girvan-Newman vs. Louvain comparison on graphs >500 nodes, including runtime, modularity, and NVI metrics.
- Node2Vec recommendation accuracy and diversification uplift vs. random baseline.
- Memory profiling across all algorithms (peak RSS and per-node footprint), summarized in Figure 5.
- Modularity dendrogram visualization exported from GN edge-removal history.

## References

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2), 163–177.  \ Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), 7821–7826.  \ Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.  \ Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *Proceedings of KDD*, 855–864.  \ Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117.