# Lab 1 Big Data Management
# Problem C

Alejandro Delgado, Viktoria Gagua

April 29, 2025

# Task C (Graph Algorithms)

For this section, we chose the PageRank algorithm, which is particularly well-suited for academic citation networks. PageRank measures the importance of nodes based on the number and quality of links to them, making it perfect for identifying influential papers or authors in a research database.

This analysis highlights papers cited by other influential works, identifying not just popular but truly impactful research. PageRank values the quality of citations, revealing foundational studies, important research bridges, and overlooked "hidden gems." It offers a more nuanced measure of impact than raw citation counts and can map influential research areas.

Practically, it helps researchers prioritize key papers for literature reviews, improves academic recommendation systems, evaluates journals by their impact, and informs funding decisions by highlighting high-value research directions. Overall, PageRank uncovers deeper patterns in citation networks beyond traditional methods.

## Queries for C

Listing 1: PageRank analysis

```
// Problem C    GRAPH DATA SCIENCE: WEIGHTED PAGERANK
// Now we switch to the Neo4j
// Graph Data Science library. We (a) sprinkle random weights on
// the existing :CITES relationships, (b) project a lightweight
//  inmemory  graph, and (c) run a weighted PageRank to surface the
// most influential papers.
// ------------------------------------------------------

// C.1 Seed weights on :CITES relationships
//     Why: The original graph has unweighted edges. By assigning
//     each citation a random weight in the range [0.5,1.0] we can
//     demo how GDS algorithms respect relationship WeightProperty.
MATCH ()  [r:CITES]  >()
SET   r.weight = 0.5 + rand() * 0.5 // uniform random  [0.5,1.0]
RETURN COUNT(r) AS relationships_updated; // sanitycheck rowcount

// C.2 Project an  inmemory  graph for GDS
//     We include only Paper nodes and :CITES edges, preserving the
//     natural direction (citing  cited) and exposing the weight
//     property so PageRank can read it.
CALL gds.graph.project(
  'paperCitationGraph', // graph name
  'Paper',              // node projection
```

```
  {
    CITES: {
      orientation: 'NATURAL',
      properties: ['weight']
    }
  }
)
YIELD graphName, nodeCount, relationshipCount;

// C.3 Run weighted PageRank
//     Parameters:
//          maxIterations = 20    quick yet stable scores
//          dampingFactor = 0.85   standard value
//          relationshipWeightProperty = 'weight'  use our new weights
//     We stream results instead of writing back so we can decide
//     later whether to persist them as node properties.
CALL gds.pageRank.stream('paperCitationGraph', {
  maxIterations: 20,
  dampingFactor: 0.85,
  relationshipWeightProperty: 'weight'
})
YIELD nodeId, score
WITH gds.util.asNode(nodeId) AS paper, score
RETURN paper.title AS title,
       score      AS pageRank
ORDER BY pageRank DESC
LIMIT 10;
// top 10 mostinfluential papers by weighted PR
//                                                              FINALE
```