BUBT | BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY

Computer Science and Engineering

Lab Report

Comparison between RAG vs Banker's Algorithm

OPERATING SYSTEMS (CSE-210)

Submitted By:
Kaniz Fatema
20245103154
Intake 53 (01)

Submitted To:
Mishal Al Rahman
Lecturer
Dept. of CSE

November 2, 2025

Definitions

Resource Allocation Graph (RAG)

A directed bipartite graph with **process nodes** and **resource nodes**. An edge $P \to R$ means a process *requests* a resource; an edge $R \to P$ means a resource instance is *allocated* to a process.

Banker's Algorithm

A **deadlock-avoidance** algorithm for systems with **multiple instances per resource type**. It maintains matrices/vectors **Max**, **Alloc**, **Need** ($= \text{Max} - \text{Alloc}$), and **Available**; a request is granted only if the resulting state is **safe**.

Where and Why We Use Them

Resource Allocation Graph (RAG):

- Used to **visualize** and **diagnose** waits/deadlocks.
- Best for systems where each resource has **one instance**; a cycle then **implies deadlock**.
- With claim edges it can aid basic avoidance, but it does not scale well to many instances per resource type.

Banker's Algorithm:

- Used by a resource manager to **decide grant vs. wait at runtime** and **avoid deadlocks**.
- Suited for **multiple-instance** resources; effective when approximate **maximum demand** per process is known.
- Trades added bookkeeping overhead for system **safety**.

Short Comparison

- **Purpose: RAG** is used to *model & detect*; **Banker's** is used for *policy & avoid*.
- **Best fit: RAG** for single-instance resources & analysis; **Banker's** for multi-instance run-time control.
- **Data needed: RAG** needs current requests/allocations; **Banker's** needs **Max/Alloc/Need/Availab** (max demand known).
- **Outcome: RAG** finds cycles (evidence of trouble); **Banker's** ensures a safe sequence before granting.
- **Trade-offs: RAG** is simple/visual but weak for multi-instance prevention; **Banker's** is stronger avoidance but needs more info and adds overhead.

Conclusion

Banker's Algorithm is generally better than RAG for deadlock avoidance in systems with multiple instances of resources. It proactively checks that the state remains safe before each grant and can produce a safe sequence, whereas RAG primarily aids modeling and detection and is weaker for prevention in multi-instance settings.

[1] [2]

[1]Kaniz Fatema
[2]ID: 20245103154