25 Mar 2025 - Operating Systems - II - Week 11

Banker's Algorithm.

↳ $O(mn^2)$

BFS → $O(n^2)$

Example of Banker's Algorithm

Total :

A	B	C
10	5	7

Avail

A	B	C

Allocation :

A	B	C
0	1	0
2	0	0
3	0	2
2	1	1
0	0	2

Max :

A	B	C
7	5	3
3	3	2
9	0	2
2	2	2
4	3	3

Total :

A	B	C
10	5	7

Avail

A	B	C
3	3	2

10	5
−2	−1
−3	−1
−2	

Need:

	A	B	C
P_0	7	4	3
P_1	1	3	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

$$\text{Need}[i] = \text{Max}[i] - \text{Allocation}[i]$$

Avail_0 → Work

A	B	C
3	3	2

$$\text{Need}_i \leq \text{avail}_i$$

P_0	×
P_1	✓ →
P_2	×
P_3	×
P_4	×

	A	B	C
$Work_1$	5	3	2
P_1 ✓ $Work_2$	7	4	3
P_3			
P_4			

$Work_1$ ↓ 1^{st} iteration

→ Allocation +

→ now all can be satisfied (work ↑) more than req.

Devising this algorithm is complicated
↓
applying is easy

P_s : Max Allocation Need

P_s ↓ single central process maintains this str.

Drawback SMP

↓ every process needs to inform P_s → P_s becomes slow

AMP ?? one core

If not Ps, who? → OS x

→ itself a collection of processes

Obtain lock over

lock → meta lock

Max
Allocation
Need

Since this is one single lock it will not lead to deadlock

→ <u>drawback</u>

lock needs to be acquired

→ implement this compare each.

Little book on Semaphores

→ Not practical.

Deadlock Detection

Allow system to enter deadlock state

Detection algorithm

Recovery scheme

Garbage collection

✓ system automatically reclaims (no process uses it)

→ Java

↳ Frequently garbage collector runs →

too frequently	less
over head	memory ↓

⊗ → "manage your memory" on your own

→ stack → local
→ global

→ heap

C++	C
new delete	malloc free

Deadlock ⇝ break deadlock

→ hold and wait

Resource - Allocation Graph

Similar Algorithm $O(mn^2)$

26 Mar 2025

→ Resource allocation graph and corresponding wait - for graph.

→ Here you don't have max no. of resources

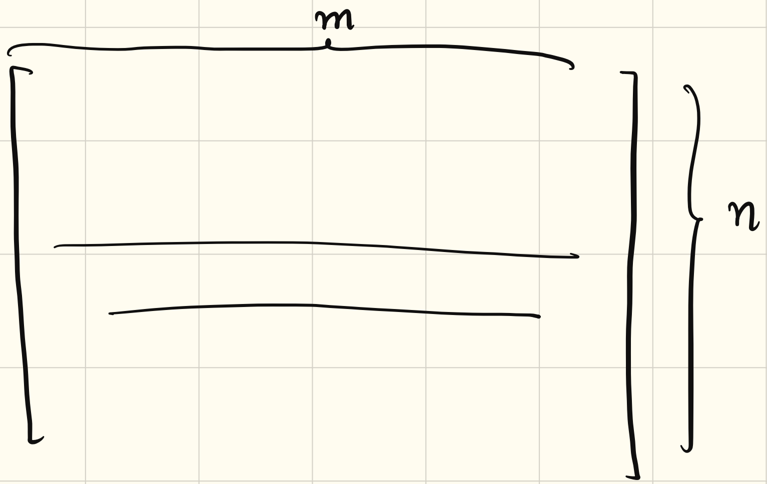
* Available
* Allocation
* S

$\text{Allocation}[i, j] = k$

T_i has been allocated k instances of resource type R_j

→ Detection Algorithm

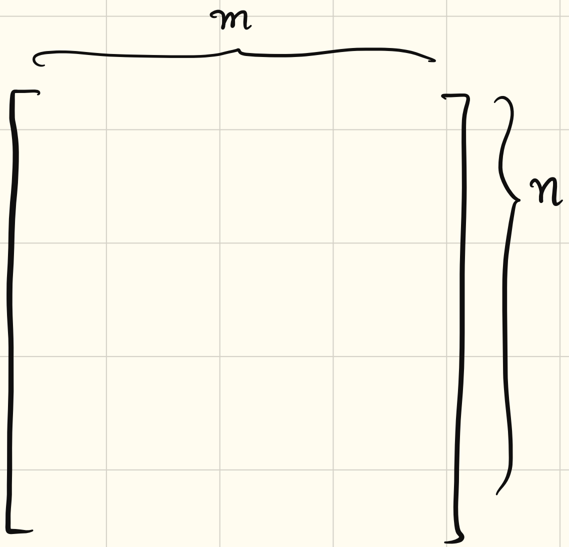
Alloc



$n \times m$

if $Alloc_i = 0$ → will never be involved in deadlock
(hold and wait)
not satisfied

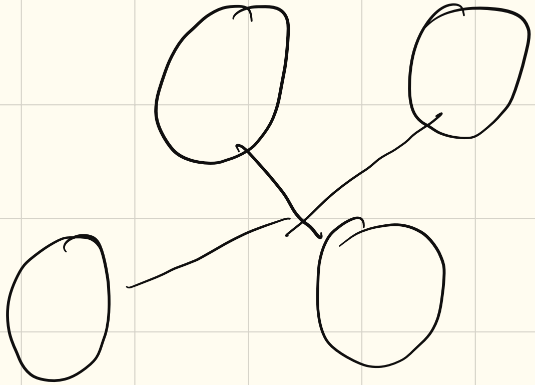
Req



$n \times m$

→ Keep checking periodically

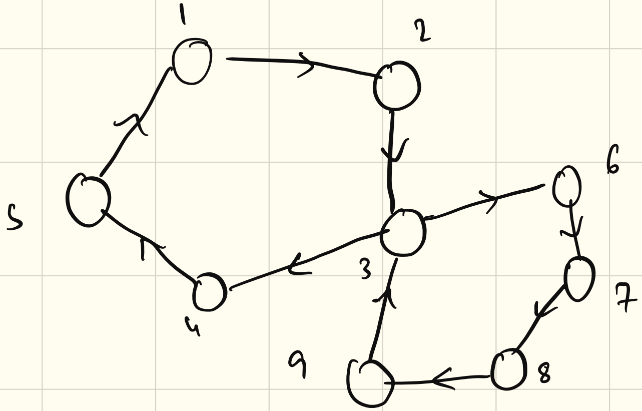
cycles in the graph



Recovery from Deadlock

→ terminate a process that is involved in most no. of cycles

NP-complete?



→ Abort and rollback

→ logs for recovery

→ RAID

Most popular method for deadlock prevention :

→ lock resources in order.

→ Transaction Memory

Resource Preemption

— see slides -