Banker's

n Processes

m Resources.

$\boxed{1}$ Safety

available: vector of length m of the number of resources available.

Max: n × m matrix of the maximum demand.

$$\begin{array}{cc} \text{Hammer} & \text{Rench} \\ \multicolumn{2}{c}{\begin{bmatrix} 5 & 3 \end{bmatrix}} \end{array}$$

Allocation: n×m matrix of the current allocation

Need : n×m matrix of the number of resources needed (in the worst case).

$$\underset{X}{\begin{bmatrix}1 & 2 & 5\end{bmatrix}} \quad \underset{Y}{\begin{bmatrix}3 & 4 & 6\end{bmatrix}}$$

$$X \leq Y \text{ iff } X_i \leq Y_i \ \underset{\text{for all}}{\forall i}$$

1) let Work = available.
Finish = False for every process Finish[n] = F;

2) Find i such that:
1) Finish[i] == False
2) $Need_i \leq Work$.

3) If no i exists goto 4

3) $Work = Work + \text{allocation}_i$ ~~available~~
Finish[i] = True;
goto 2

4) If all Finish[i] == TRUE
$\Rightarrow$ System is Safe

---

2) Request

$Request_i$ process i making a request

$$\begin{array}{cc} H & R \\ \left[\begin{matrix} 1 & 0 \end{matrix}\right. & \left.\right] \end{array}$$

1) If $Request_i > Need_i \Rightarrow$ error.
2) If $Request_i \leq available$ goto 3 else wait

3) Pretend to grant.

$$\left.\begin{aligned} Allocation_i &= Allocation_i + request_i \\ Need_i &= Need_i - Request_i \\ Available &= available - Request_i \end{aligned}\right\}$$

4) Run Safety $\Rightarrow$ If Safe grant else deny.

Ex

A	B	C
10	5	7

	Allocation A B C	Max A B C	Need A B C
P_0	0 1 0	7 5 3	7 4 3
P_1	2 0 0	3 2 2	1 2 2
P_2	3 0 2	9 0 2	6 0 0
P_3	2 1 1	2 2 2	0 1 1
P_4	0 0 2	4 3 3	4 3 1

available

A	B	C
3	3	2

$\boxed{1}$ work = [3 3 2] Finish[i] = F ; $\forall_i$

$\boxed{2}$ P_1 work = [3 3 2] + [2 0 0] = [5 3 2]

$\boxed{2}$ P_3 work = [5 3 2] + [2 1 1] = [7 4 3]

Finish	
0	F
1	T
2	F
3	~~F~~ T
4	F

$P_1 \to P_3 \to$ any Sequence [Safe]

P_0 P_2 P_4

P_2 P_0 P_4

P_2 P_4 P_0

⋮

P_1 requests $(1, 0, 2)$

$\Rightarrow$ Pretend

	Allocation A B C	Max A B C	Need A B C
P_0	0 1 0	7 5 3	7 4 3
P_1	3 0 2	3 2 2	0 2 0
P_2	3 0 2	9 0 2	6 0 0
P_3	2 1 1	2 2 2	0 1 1
P_4	0 0 2	4 3 3	4 3 1

available
A B C
2 3 0

$\boxed{1}$ work = 2 3 0

$\boxed{2}$ P_1 can finish

$\Rightarrow$ work = $[2\ 3\ 0] + [3\ 0\ 2] = 5\ 3\ 2$

⋮

Safe $\Rightarrow$ grant the request

P_4 requests $[3\ 3\ 0]$

P_4 requests [3 3 0]

→ wait

P_0 requests [0 2 0]

Pretend

	Allocation A B C	Max A B C	Need A B C
P_0	0 3 0	7 5 3	7 2 3
P_1	3 0 2	3 2 2	0 2 0
P_2	3 0 2	9 0 2	6 0 0
P_3	2 1 1	2 2 2	0 1 1
P_4	0 0 2	4 3 3	4 3 1

available
A B C
2 1 0

Safety.

$\boxed{1}$ work [2 1 0]

Can't find any i

⇒ Unsafe

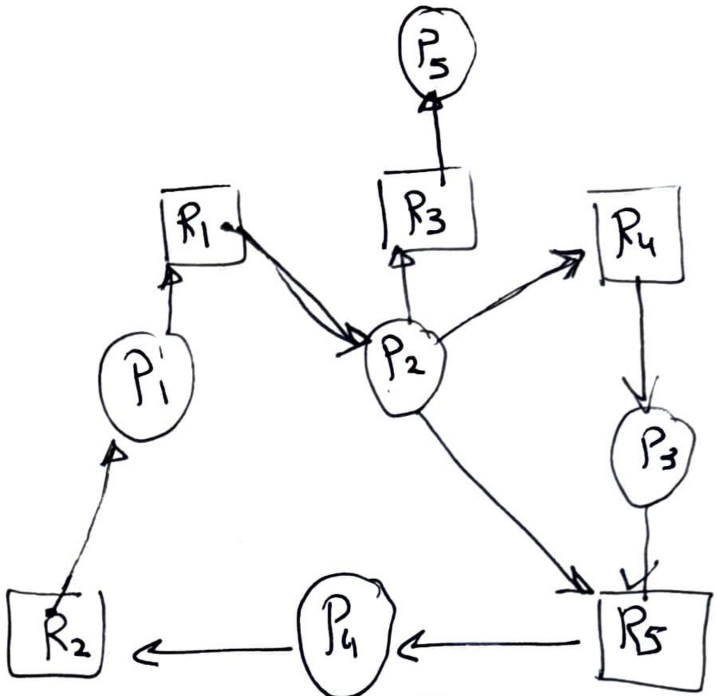
⇒ deny the request

3 Detection and Recovery

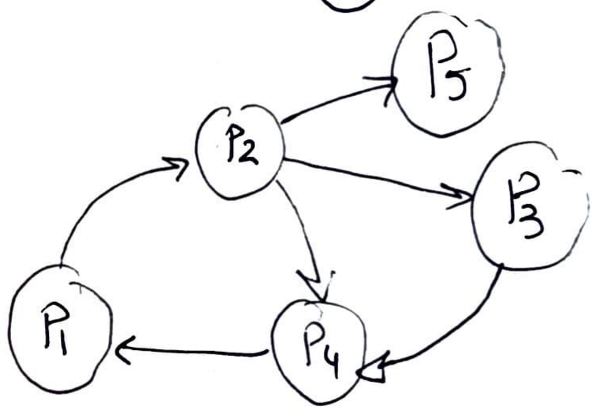
I. Single instant of each resource.

Wait-for graph

Remove resource and collapse the edges.



Deadlock exists if a cycle exists



$P_1 \quad P_2 \quad P_3 \quad P_4$

$P_1 \quad P_2 \quad P_4$