

Deadlock Avoidance (Banker's Algorithm)

	<u>Allocation</u>			<u>Max</u>			<u>Need</u>			<u>Available</u>		
	A	B	C	A	B	C	A	B	C	A	B	C
P ₀	0	1	0	7	5	3	7	4	3	3	3	2
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			

$n=5$, $m=3$ (A, B, C) ; $\text{Need} = (\text{Max} - \text{Allocation})$
 $\downarrow \quad \downarrow \quad \downarrow$
Total $\rightarrow 10 \ 5 \ 7$

Is the system safe? \rightarrow

work = available = (3 3 2)

finish = (false, false, false, false, false)

1. Finish[1] = false, Need₁ = (1 2 2) \leq (3 3 2)

work = work + allocation₁ = (3 3 2) + (2 0 0)
 = (5 3 2)

Finish[1] = true

2. Finish[3] = false, Need₃ = (0 1 1) \leq (5 3 2)

Finish[3] = true;

work = work + allocation₃ = (5 3 2) + (2 1 1)
 = (7 4 3)

$$3. \text{ Finish}[4] = \text{false}, \text{ Need}_4 = (4 \ 3 \ 1) \leq (7 \ 4 \ 3)$$

$$\text{Finish}[4] = \text{true};$$

$$\begin{aligned} \text{work} &= \text{work} + \text{allocation}_4 = (7 \ 4 \ 3) + (0 \ 0 \ 2) \\ &= (7 \ 4 \ 5) \end{aligned}$$

$$4. \text{ Finish}[2] = \text{false}; \text{ Need}_2 = (6 \ 0 \ 0) \leq (7 \ 4 \ 5)$$

$$\text{Finish}[2] = \text{true};$$

$$\begin{aligned} \text{work} &= \text{work} + \text{allocation}_2 \\ &= (7 \ 4 \ 5) + (3 \ 0 \ 2) \\ &= (10 \ 4 \ 7) \end{aligned}$$

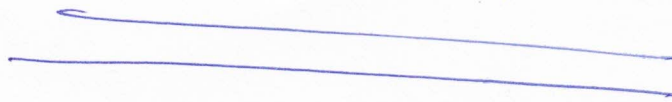
$$5. \text{ Finish}[0] = \text{false}; \text{ Need}_0 = (7 \ 4 \ 3) \leq (10 \ 4 \ 7)$$

$$\text{Finish}[0] = \text{true};$$

$$\begin{aligned} \text{work} &= \text{work} + \text{allocation}_0 \\ &= (10 \ 4 \ 7) + (0 \ 1 \ 0) \\ &= (10 \ 5 \ 7) \end{aligned}$$

\therefore Safe with sequence

$$\langle P_1, P_3, P_4, P_2, P_0 \rangle$$



Banker's Algorithm

	<u>Alloc</u>			<u>Max</u>			<u>Need</u>			<u>Available</u>		
	A	B	C	A	B	C	A	B	C	A	B	C
P ₀	0	1	0	7	5	3	7	4	3	3	3	2
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			

Let P₄ has request $\rightarrow (3 \ 3 \ 0)$

$$\text{Request}_4 = (3 \ 3 \ 0)$$

$$\text{Need}_4 = (4 \ 3 \ 1)$$

① $\text{Request}_4 \leq \text{Need}_4 \quad \checkmark$

② $\text{Request}_4 \leq \text{available}$
 $(3 \ 3 \ 0) \leq (3 \ 3 \ 2) \quad \checkmark$

③ $\text{Available} = \text{Available} - \text{Request}_4$
 $= (3 \ 3 \ 2) - (3 \ 3 \ 0)$
 $= (0 \ 0 \ 2)$

④ $\text{Allocation}_4 = \text{Allocation}_4 + \text{Request}_4$
 $= (0 \ 0 \ 2) + (3 \ 3 \ 0)$
 $= (3 \ 3 \ 2)$

⑤ $\text{Need}_4 = \text{Need}_4 - \text{Request}_4$
 $= (4 \ 3 \ 1) - (3 \ 3 \ 0)$
 $= (1 \ 0 \ 1)$

Now, the current state of the system



	A	B	C	A	B	C	A	B	C	A	B	C
P ₀	0	1	0	7	5	3	7	4	3	0	0	2
P ₁	2	0	0	3	2	2	1	2	2	↓ <u>Available</u>		
P ₂	3	0	2	9	0	2	6	0	0			
P ₃	2	1	1	2	2	2	0	1	1			
P ₄	3	3	2	4	3	3	1	0	1			
	↓ Allocation			↓ Max			↓ Need					

Is the above resource-allocation state
safe?

① work = available = (0 0 2)

② Find 'i' such that Finish[i] = false
and Need[i] ≤ work

could not find such 'i'.

→ System is unsafe.

→ P₄ has to wait.

→ Restore old state.