Deadlock Avoidance (Banker's Algorithm)

$$initial\ instances\ of\ resources - total\ allocated = Available$$

$$Need_{i,j} = Max_{i,j} - Allocation_{i,j}$$

$$forall\ p_i : if\ (Need_{p_i} \leq Available)$$

$$Available = Available + Allocation_{p_i}$$

Are all the P_i used to update "Available" in the last step $\xrightarrow{Yes}$ safe state

$\xrightarrow{No}$ No safe state

if there is a request from process pi, following conditions and updates must be considered before using the algorithm:

check these conditions

$$Request_{p_i} \leq Need_{p_i}$$ $\xrightarrow{else}$ error

$$Request_{p_i} \leq Avaiable$$ $\xrightarrow{else}$ P_i must wait

apply updates

$$Allocation_{p_i} = Allocation_{p_i} + Request_{pi}$$

$$Need_{p_i} = Need_{p_i} - Request_{pi}$$

$$Available = Available - Request_{pi}$$

And then use Banker's Algorithm!

Banker's Algorithm example

Q1) 5 processes; 3 resource types

A (10 instances)

B (5 instances)

C (7 instances)

– snapshot at current time:

	Allocation	Max
	A B C	A B C
T_0	0 1 0	7 5 3
T_1	2 0 0	3 2 2
T_2	3 0 2	9 0 2
T_3	2 1 1	2 2 2
T_4	0 0 2	4 3 3

2+3+2 ← 7 2 5

Is this system in safe state? Yes

A1) Solution:

A B C

Total allocated = (7, 2, 5)

Available

Available = (10, 5, 7) – (7, 2, 5) = $\boxed{(3, 3, 2)}$

Need = Max – Allocation =

	Need
	A B C
T_0	7 4 3
T_1	1 2 2
T_2	6 0 0
T_3	0 1 1
T_4	4 3 1

Example → (1, 2, 2) ≤ (3, 3, 2)

$Need_{T_1} \leq Available$

if ($Need_{T_i} \leq Available$)

$Available = Available + Allocation_{T_i}$

Available

$(3,3,2) \xrightarrow{T_1} (5,3,2) \xrightarrow{T_3} (7,4,3) \xrightarrow{T_4} (7,4,5) \xrightarrow{T_2} (10,4,7) \xrightarrow{T_0} (10,5,7)$

(5, 3, 2) + (2, 1, 1) Allocation T_3

* (0, 1, 1) ≤ (5, 3, 2)

$Need_{T_3} \leq Available$

⟨T_1, T_3, T_4, T_2, T_0⟩ is a safe sequence.

Q2) In Q1, T3 requests two more instances of resource B. Does the system go to safe state if we grant the request of T3? NO

A2) The Need matrix shows that T3 can request just 1 more instance of resource B. So the algorithm raises an error condition.

Need T_3

$$\underbrace{(0,2,0)}_{\text{T3 request}} \nleq (0,1,1)$$

Q3) In Q1, T4 requests (3,3,0) (A B C). should we grant this request? NO

A3)

$(3,3,0) \leq (4,3,1)$ ✓

$(3,3,0) \leq (3,3,2)$ ✓

updates:

	Allocation	Max	Available
	A B C	A B C	A B C
T_0	0 1 0	7 5 3	$\underset{0}{\cancel{3}}\,\underset{0}{\cancel{3}}\,2$
T_1	2 0 0	3 2 2	
T_2	3 0 2	9 0 2	
T_3	2 1 1	2 2 2	
T_4	$3\,\cancel{0}\,\underset{3}{\cancel{0}}\,2$	4 3 3	

	Need
	A B C
T_0	7 4 3
T_1	1 2 2
T_2	6 0 0
T_3	0 1 1
T_4	$1\,\cancel{4}\,\underset{0}{\cancel{3}}\,1$

Bankers Algorithm:

Available

$(0,0,2) \xrightarrow{?}$ None of the rows of Need is less than Available, so the state will be not-safe and we don't grant the request.

Q4 In Q1, T1 requests one instance of A, two instances of B and two instances of C. Does the system go to safe state if we grant this request? Yes

A4)

T1 request $(1,2,2) \leq (1,2,2)$ ✓

$(1,2,2) \leq (3,3,2)$ ✓

step1) updates

	Allocation	Max	Available
	A B C	A B C	A B C
T_0	0 1 0	7 5 3	~~3 3 2~~ 2 1 0
T_1	~~2 0 0~~ 3 2 2	3 2 2	
T_2	3 0 2	9 0 2	
T_3	2 1 1	2 2 2	
T_4	0 0 2	4 3 3	

	Need
	A B C
T_0	7 4 3
T_1	~~1 2 2~~ 0 0 0
T_2	6 0 0
T_3	0 1 1
T_4	4 3 1

step 2) check with Banker's Algorithm:

Available

$(2,1,0) \xrightarrow{T_1} (5,3,2) \xrightarrow{T_3} (7,4,3) \xrightarrow{T_4} (7,4,5) \xrightarrow{T_0} (7,5,5) \xrightarrow{T_2} (10,5,7)$

So, the sequence <T1, T3, T4,T0, T2> is a safe sequence, and if we grant T1's request, we are sure the system will stay in safe state.

Deadlock Detection

We can detect deadlocks by a similar algorithm to Banker's, with just minor changes:

1 – Need matrix $\xrightarrow{\text{replaced by}}$ Request matrix

2- Exactly the same steps as Banker's algorithm
but replace the Need matrix with the Request matrix

3- In case of any request from a Thread in the question, there are no conditions to check. Likewise, there is no update on Allocation matrix. Just update the Request matrix:

$$Request_{p_i} = Request_{p_i} + NEW\,Request_{pi}$$

4 – we don't have Max and hence Need cannot be computed.

Example) instances of resources are $\overset{A}{(7},\overset{B}{2},\overset{C}{6})$ and the current state of system is:

Allocation

	A	B	C
T_0	0	1	0
T_1	2	0	0
T_2	3	0	3
T_3	2	1	1
T_4	0	0	2

Request

	A	B	C
T_0	0	0	0
T_1	2	0	2
T_2	0	0	0
T_3	1	0	0
T_4	0	0	2

Available

A	B	C
0	0	0

is the system deadlocked? No

Available
$(0,0,0) \xrightarrow{T_0} (0,1,0) \xrightarrow{T_2} (3,1,3) \xrightarrow{T_3} (5,2,4) \xrightarrow{T_1} (7,1,3) \xrightarrow{T_4} (7,1,5)$

$\underbrace{(0,0,0)}_{Request\,T_2} \leq \underbrace{(0,1,0)}_{available}$ $\quad (0,1,0) + \underbrace{(3,0,3)}_{Allocation\,T_2}$