Bankers Algorithm

Scenario:

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2	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

Work = Available = [3 3 2]

Finish[i] = false for i = 0, 1, ... n-1

Finish[0] = false

$Need_0$ = [7 4 3] $\leq$ [3 3 2] false (Work)

$Need_1$ = [1 2 2] $\leq$ [3 3 2] True ✓

Work = Work + Allocation

= [3 3 2] + [2 0 0]

= [5 3 2]

$Finish_1$ = True

$Need_2$ = [6 0 0] $\leq$ [5 3 2] false

$Need_3$ = [0 1 1] $\leq$ [5 3 2] True ✓

Work = Work + $Allocation_3$

= [5 3 2] + [2 1 1]

= [7 4 3]

$Need_4 = [4\ 3\ 1] \leq Work\ [7\ 4\ 3]$ True✓

$Work = Work + Allocation_4$

$= [7\ 4\ 3] + [0\ 0\ 2]$

$= [7\ 4\ 5]$

$Finish[4] = True$

$Need_2 = [6\ 0\ 0] \leq [7\ 4\ 5]$ (Work) True✓

$Work = Work + Allocation_2$

$= [7\ 4\ 5] + [3\ 0\ 2]$

$= [10\ 4\ 7]$

$Finish[2] = True$

$Need_0 = [7\ 4\ 3] \leq [10\ 4\ 7]$ (Work) True✓

$Work = Work + Allocation_0$

$= [10\ 4\ 7] + [0\ 1\ 0]$

$= [10\ 5\ 7]$

$Finish[0] = True$

$Finish[i] = True$, for $i = 0$ to 4

Hence there is a safe sequence.

The safe sequence is $<P_1, P_3, P_4, P_2, P_0>$

Hence the system is in a safe state.

Now, suppose process P_1 requests one additional instance of resource type A and two instances of resource type C.

$Request_1 = (1, 0, 2)$

Check whether this request can be granted.
To check this, use the resource request algorithm

$Request_1 \leq Need$

$[1\ 0\ 2] \leq [1\ 2\ 2]$ True

$Request_1 \leq Available$

$[1\ 0\ 2] \leq [3\ 3\ 2]$ True

Pretend that the request is granted

$Available_1 = Available - Request_1$

$= [3\ 3\ 2] - [1\ 0\ 2]$

$= [2\ 3\ 0]$

$Allocation_1 = Allocation_1 + Request_1$

$= [2\ 0\ 0] + [1\ 0\ 2]$

$= [3\ 0\ 2]$

$Need_1 = Need_1 - Request_1$

$= [1\ 2\ 2] - [1\ 0\ 2]$

$= [0\ 2\ 0]$

The state of the system becomes like this

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

Now apply the safety algorithm

Work = Available = [2 3 0]

Finish[i] = False, for i = 0, 1, 2 4

Finish[0] = False

$Need_0 = [7\ 4\ 3] \leq [2\ 3\ 0]$ false

Finish[1] = false

$Need_1 = [0\ 2\ 0] \leq [2\ 3\ 0]$ True ✓

Work = [2 3 0] + [3 0 2]

= [5 3 2]

Finish[1] = True

$Need_2 = [6\ 0\ 0] \leq [5\ 3\ 2]$ false

Finish[3] = false

$Need_3 = [0\ 1\ 1] \leq [5\ 3\ 2]$ True ✓

Work = [5 3 2] + [2 1 1] = [7 4 3]

Finish[3] = True

Finish[4] = false

$Need_4 = [4\ 3\ 1] \leq [7\ 4\ 3]$ True ✓

Work = [7 4 3] + [0 0 2]

= [7 4 5]

Finish[4] = True

Finish[0] = false

$Need_0 = [7\ 4\ 3] \leq [7\ 4\ 5]$ True ✓

Work = [7 4 5] + [0 1 0]

= [7 5 5]

Finish[0] = True

$Need_2 = [6\ 0\ 0] \leq [7\ 5\ 5]$ True ✓

Work = [7 5 5] + [3 0 2]

= [10 5 7]

Finish[2] = True

Safe sequence is $<P_1, P_3, P_4, P_0, P_2>$

Hence the system will be in a safe state.

Hence the request can be granted.