Banker's Algorithm

- Copy the alloc[i,j] table to alloc'[i,j]
- Given C, maxc and alloc', compute avail vector
- Find p_i: maxc[i,j] - alloc'[i,j] $\leq$ avail[j] for $0 \leq j < m$ and $0 \leq i < n$.
  - If no such p_i exists, the state is unsafe
  - If alloc'[i,j] is 0 for all i and j, the state is safe
- Set alloc'[i,j] to 0; deallocate all resources held by p_i; go to Step 2

Example

Maximum Claim

Process	R_0	R_1	R_2	R_3
p_0	3	2	1	4
p_1	0	2	5	2
p_2	5	1	0	5
p_3	1	5	3	0
p_4	3	0	3	3

Allocated Resources

Process	R_0	R_1	R_2	R_3
p_0	2	0	1	1
p_1	0	1	2	1
p_2	4	0	0	3
p_3	0	2	1	0
p_4	1	0	3	0
Sum	7	3	7	5

C = <8, 5, 9, 7>

- Compute total allocated
- Determine available units

avail = <8-7, 5-3, 9-7, 7-5>
= <1, 2, 2, 2>

- Can anyone's maxc be met?

maxc[2,0]-alloc'[2,0] = 5-4 = 1≤1 = avail[0]
maxc[2,1]-alloc'[2,1] = 1-0 = 1≤2 = avail[1]
maxc[2,2]-alloc'[2,2] = 0-0 = 0≤2 = avail[2]
maxc[2,3]-alloc'[2,3] = 5-3 = 2≤2 = avail[3]

- P_2 can exercise max claim

avail[0] = avail[0]+alloc'[2,0] = 1+4 = 5
avail[1] = avail[1]+alloc'[2,1] = 2+0 = 2
avail[2] = avail[2]+alloc'[2,2] = 2+0 = 2
avail[3] = avail[3]+alloc'[2,3] = 2+3 = 5

Example

Maximum Claim

Process	R_0	R_1	R_2	R_3
p_0	3	2	1	4
p_1	0	2	5	2
p_2	5	1	0	5
p_3	1	5	3	0
p_4	3	0	3	3

Allocated Resources

Process	R_0	R_1	R_2	R_3
p_0	2	0	1	1
p_1	0	1	2	1
p_2	0	0	0	0
p_3	0	2	1	0
p_4	1	0	3	0
Sum	3	3	7	2

C = <8, 5, 9, 7>

- Compute total allocated
- Determine available units

avail = <8-7, 5-3, 9-7, 7-5>
= <5, 2, 2, 5>

- Can anyone's maxc be met?

maxc[4,0]-alloc'[4,0] = 5-1 = 4≤5 = avail[0]
maxc[4,1]-alloc'[4,1] = 0-0 = 0≤2 = avail[1]
maxc[4,2]-alloc'[4,2] = 3-3 = 0≤2 = avail[2]
maxc[4,3]-alloc'[4,3] = 3-0 = 3≤5 = avail[3]

- P_4 can exercise max claim

avail[0] = avail[0]+alloc'[4,0] = 5+1 = 6
avail[1] = avail[1]+alloc'[4,1] = 2+0 = 2
avail[2] = avail[2]+alloc'[4,2] = 2+3 = 5
avail[3] = avail[3]+alloc'[4,3] = 5+0 = 5

Example

Maximum Claim

Process	R_0	R_1	R_2	R_3
p_0	3	2	1	4
p_1	0	2	5	2
p_2	5	1	0	5
p_3	1	5	3	0
p_4	3	0	3	3

Allocated Resources

Process	R_0	R_1	R_2	R_3
p_0	2	0	1	1
p_1	0	1	2	1
p_2	0	0	0	0
p_3	0	2	1	0
p_4	0	0	0	0
Sum	2	1	4	2

$C = \langle 8, 5, 9, 7 \rangle$

- Compute total allocated
- Determine available units

$$\text{avail} = \langle 8-7, 5-3, 9-7, 7-5 \rangle$$
$$= \langle 6, 2, 5, 5 \rangle$$

- Can anyone's maxc be met?

(Yes, any of them can)