Banker's Algorithm

- Used in Banking to ensure that bank never allocated its available cash in such a way that it could no longer satisfy the needs of all Customers.

Data Structures Used:-

- Available - indicates available no. of each type
- Max - max demand of each process
- Allocation - no. allocated to each process
- Need - remaining resources needed by each process

$Need_i = Max_i - Allocation_i$

Safety Algorithm:-

1) Let us initialize
   work = Available
   finish[i] = false i = 0, 1, 2 --- n

2) Find an i such that
   i) finish[i] = false
   ii) need[i] ≤ work
   If no such i exists go to 4

3. work = work + Allocation
   finish = true
   go to 2

4. If finish[i] = true for all i ②
then system is in safe state.

Example problem:
5 processes $P_0 - P_4$. Three resource types of A=10 B=5 C=7

Process	Allocation A B C	Max A B C	(new) Available	Need
P_0	0 1 0	7 5 3	~~3 3 2~~ 5 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

Available = Total - Allocation

Apply Safety Algorithm

Initially work = Available

work = 3 3 2

finish [i] = false ∀ i

P0 finish [0] = false

need [0] ≤ work.

7 4 3 ≤ 3 3 2 ✗

P1 finish [0] = false

need [1] ≤ work

1 2 2 ≤ 3 3 2 ✓

work = work + Allocation

= 3 3 2 + 2 0 0

= 5 3 2

finish [0] = True

P2

need [2] ≤ work

6 0 0 ≤ 5 3 2 ✗

finish [0] = false

P3 finish[3] = false

need3 ≤ work ✓

011 ≤ 532

finish[3] = true

work = 532 + 211 = 743

P4 finish[4] = false

need[4] ≤ work

431 ≤ 743 ✓

work = 743 + 002

= 745

finish[4] = true

P0

need0 ≤ work ✓

743 ≤ 745

work = work + Alloc

= 745 + 010

= 755

P2 need2 ≤ work

600 ≤ 755 ✓

work = work + Allocation

= 755 + 302 = 1057

Resource Request Algorithm :-

- If a process made an additional request
- Check it is granted by this algorithm

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 0 2	3 2 2	2 3 0	~~1 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

P_1 makes an additional request (1, 0, 2)

Check Request $\leq$ need & Req $\leq$ Avail

P_1 $(1\ 0\ 2) \leq (1\ 2\ 2)$ True — Yes

$(1\ 0\ 2) \leq (3\ 3\ 2)$ True — Yes

So allocate the requested resource.

$\therefore$ Avail = Avail − Req

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

$= 2\ 3\ 0$ — update it in table

Allocation = Alloc + Req

(P1) = 200 + 102

= 302 //

Need = Need - Req

(P1) = 122 - 102

= 020 //

We: