Process concurrency :- (ek process ko different processor mein kaise execute karenge.)

Suppose we have four statement.

S1 $a = x + y$] independent st.
S2 $b = x + 1$
S3 $c = a + b$ → depend on S_1 and S_3.
S4 $d = c - 1$

Read set : represented by Si (statement no.) (only read not modify.)

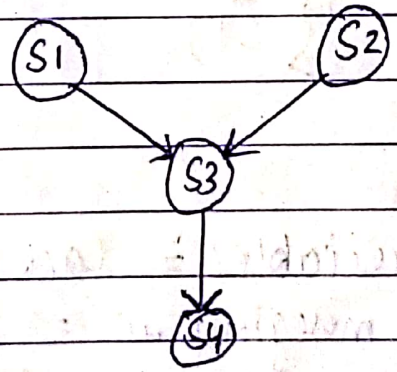
$R(S_1) = \{x, y\}$
$R(S_2) = \{x\}$
$R(S_3) = \{a, b\}$
$R(S_4) = \{c\}$

Write set

$W(S_1) = \{a\}$
$W(S_2) = \{b\}$
$W(S_3) = \{c\}$
$W(S_4) = \{d\}$

Here, (read and modify both.)

Precedance Graph
Draw node for each statement.



S_1, S_2 are independent node mean parallel.

(Parallel + all the condition)

Page no.:
Date: / /

Concurrency Condition

check krne ke liye ki parallel S_1, S_2 ya kuch bhi stm. hoga ki nhi. (intersection na ho).

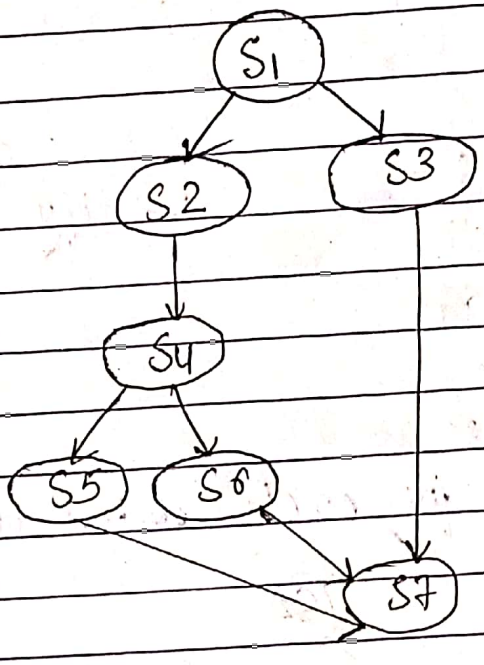
→ $W(S_i) \cap R(S_j) = \phi$,

→ $W(S_i) \cap W(S_j) = \phi$,

→ $W(S_j) \cap R(S_i) = \phi$

→ $R(S_i) \cap R(S_j) \neq \phi$ (may or may not).

*



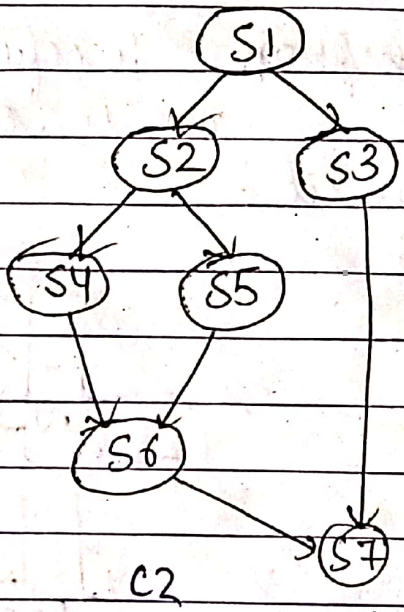
began = to start
count = no. of variables = last end node. = count
value = lines of merging it = 3
count = 3.
forks = for subdivided jobs
join = jab join hoga last node.

concurrency Program. :1

Page no.:
Date: / /

began	C1	C2
count=3		
S1	S3	S6
L1: fork S2	go to L3	go to L3
	end	end
~~L2~~: fork S4		
L2: fork S5		
L3: join count		
S7		
end.		

join (count)
count = count - 1;
if count ≠ 0 then quit



began	C1	C2
count=2	count=2	
S1	S3	S5
L1: fork S2	go to L4	go to L3
	end	end
L2: fork S4		
L3: join count 1 S6		
L4: join count 2. S7 end.		

Page no.:
Date: / /

Deadlock

- Indefinite or infinite waiting state.
- Starvation - finite waiting.

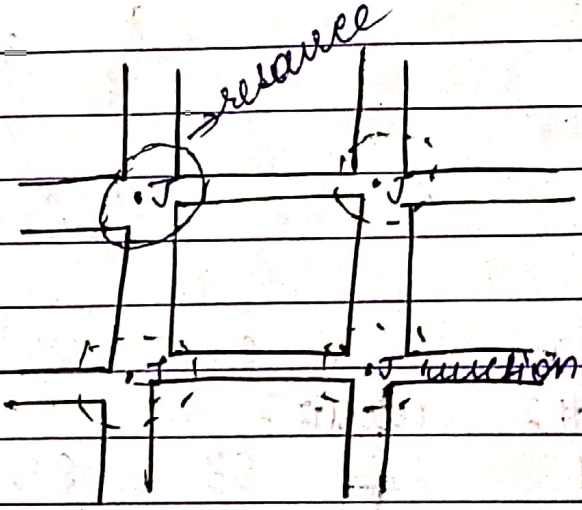
Train A ——→ ←—— Train B.

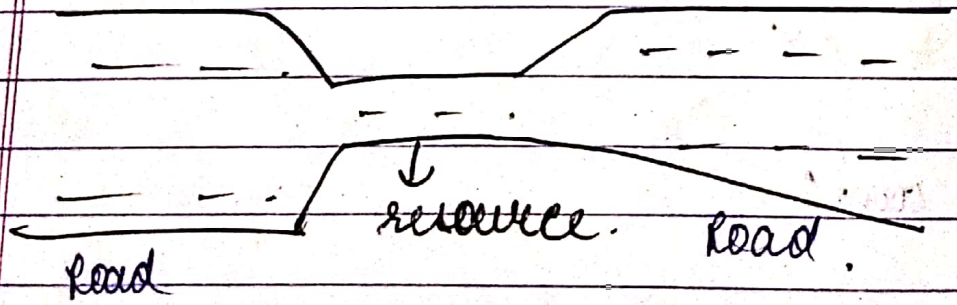
eg:-

Necessary conditions (System is deadlock or not)

1) Mutual exclusion :- non-shareable resource.
2) Hold & wait :-
3) No-preemption :- forcefully resource wapas nhi le sakte process se.
4) Circular wait :-

eg:-





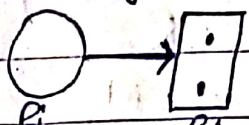
circular wait is the extension of hold & wait.

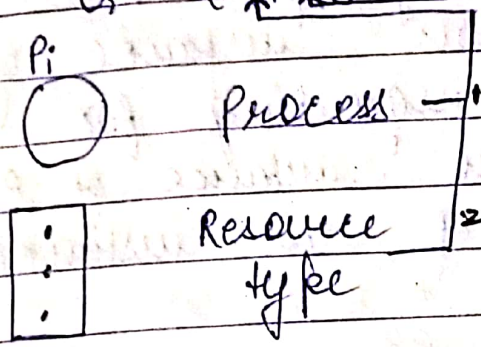
Process P_1 P_2
Tape Drives T_1 T_2

mutual exclusion T_1 P_1 hoga toh P_2 usse nhi karega.
system me deadlock hai ki nhi yeh check karne ke liye
Resource Allocation Graph :-

$G = \{V, E\}$

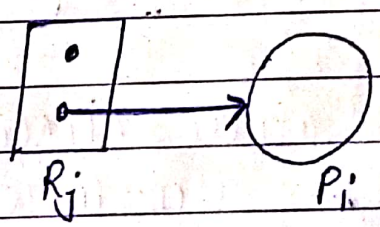
Request edge
suppose process P_i requesting resource R_j i.e (P_i, R_j).





no. of instances (exactly same resource) is represented by .

2. assignment edge (R_j, P_i)
ek resource process ko allocate kiya



Resource Allocation Graph

Page no.:
Date: / /

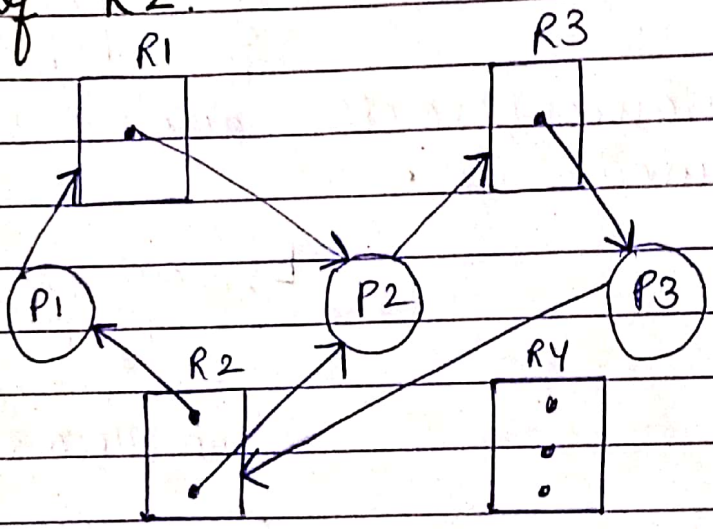
Q-1). There are three processes P_1, P_2, P_3 and 4 resources R_1, R_2, R_3, R_4

R1 has one instance — R1 — 1
R2 has two instance — R2 — 2
R3 has one instance — R3 — 1
R4 has three instance — R4 — 3

(i) Process P1 is holding an instance R2 and requesting for R1.

ii) P2 is holding an instance of R1 and R2. and requiring for R3.

iii) P3 is holding an instance of R3 and requesting of an instance of R2.



→ agar cycle ban rahi h toh deadlock ho sakta (it might be).
↳ yaha unsafe system ho sakta h)!

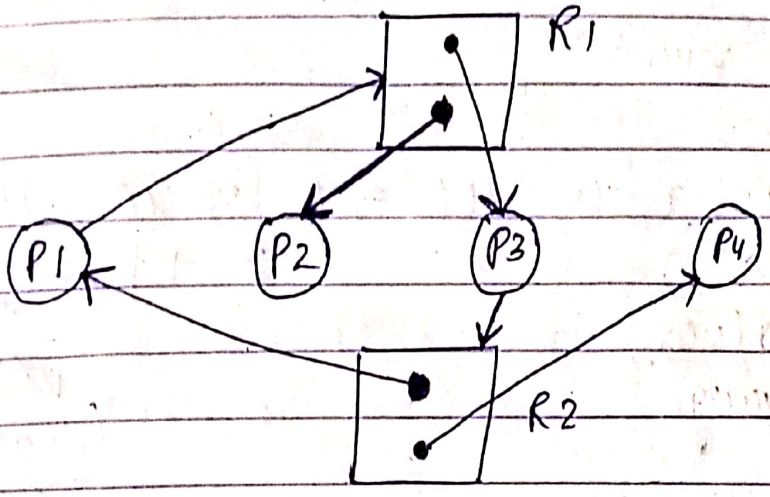
→ aur agar circle nhi banega toh system deadlock mai nhi h (system safe h tab)

Q-2) There are four processes P_1, P_2, P_3, P_4 and two resource type R_1 and R_2 each having two instances.

i) P_1 is holding an instance of R_2 and requesting for an instance of R_1

ii) P_2 is holding an instance of R_1 and P_3 is holding an instance of R_1 an requesting for R_2.

iii) P_4 is holding an instance of R_2.



→ Method of Handling Deadlock.

Q-3) A system is having three user processes each requesting two units of resource ~~are~~ R. Find the min. no. of units of R such that no deadlock will occur.

max R — Resources.

1 — P1 = 2 unit

1 — P2 = 2 unit

+1 — P3 = 2 unit

total 3

[R min = R max + 1]

R min. poocha h toh = R max nikalenge + 1

3 + 1 = 4.

↑ min R (deadlock free hogaya system)

(max. one mai system deadlock state per jaiga)

Q-4. P1 — 2 units
P2 — 3 units
P3 — 4 unit

Max instances.

1 ← P1 — 2 instances
2 ← P2 — 3 instances
+ 3 ← P3 — 4 instances.

6

R min = Rmax + 1
= 6 + 1

R min = 7

8-5) A computer system has 6 tape drives with n processes competing for them. Each process needs three tape drives. Find the max. value of n for which the system guaranteed to be deadlock free.

resource distribution.

process
1 = 3
2 = 3+3
3 = 2+2+2 → deadlock.
or 3+1+1 → not deadlock.

4 = 3+1+1+1
5 = 1+1+1+1+2

is case mai deadlock nahi hoga.

answer:- 4 is the max process kuki deadlock free ho ja raha. Isnai process wait kar raha tha har process ke liye.

→ mac OS mai prevent deadlock method use karte h

Page no.:
Date: / /

Method of Handling Deadlock :→

1) Avoid / prevent deadlock < Prevention / Avoidance

2) Detect deadlock and then Recovery → (now a days mobile phone)

3) Ignore. (windows / linux follow).

1) System will never enter deadlock.
2) Allow system to deadlock & then recover.
3) Ignore deadlock.

→ Deadlock prevention

1) Mutual Exclusion
2) Hold & wait
3) No - Preemption
4) circular wait.

Hold & wait solution:

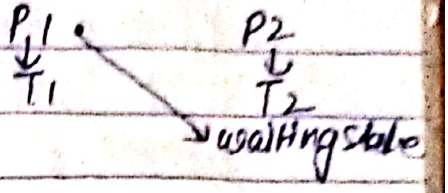
- → Grant all resources before execution.
  → drawback.
  1) initially pata hona chahiye ki kitne resource ki jarurat hogi.
  2) un-utilisation of resource.

- Grant resource based on numbering. increasing order mai hi number allot hoga.
  (ek-ek kar ke kaam complete hoga)

for ex!- File → 5
secondary memory → 10
printer → 15

ek kaam complete hone ke bad usko release kar denge then next job.

3) No Preemption.
agar koi resource available nhi to jiske pass resource available hoga use de denge aur agar dusre ke pass available nhi h to use forcefully le lenge.



4) circular wait.
(increasing order mai resource grant karnge toh circular nhi banegi)

eg:- File → 5
SM → 10
printer → 15

(ek direction mai allow karenge bas)

(wapas nhi jaenge nhi toh circular wait ho jayega.) → printer ka task complete hone ke bad wapas file mai nhi jainge. nhi toh circular wait ban jayega.

→ Deadlock avoidance
Safety algorithm

- Safe state → in a particular direction
  unsafe state → in particular way.

→ is ques. mai total tape drive diya h usse available nikalenge, agar available diya hota h, nhi nikalte.

Q-1) Consider a system with 12 magnetic tape drives and three process P_0, P_1, P_2 ; P_0 requires 10 Tape drives P_1 requires 4. P_2 requires 9 Suppose that at time to process P_0 is holding 5 Tape drives, P_1 is holding 2, P_2 is holding 2.

→ request reject when the process takes more than 12 tape drives.

Tape drive = 12.

	Need	holding	needed more is
P_0	10	− 5	= 5
P_1	4	− 2	= 2
P_2	9	− +2 / 9	= 7.

available = 12 − granted(9) = 3.

$3 + \boxed{2}_{P_1} = 5 + \boxed{5}_{P_0} = 10 + \boxed{2}_{P_2} = 12$

order is $P_1 \rightarrow P_0 \rightarrow P_2$.

→ 3 ke andar P_1 wala a rha h to uska holding add kardenge.

Page no.:
Date: / /

Q-i)

	max need		currently holding		need more
P_0	6	-	2	=	4
P_1	7	-	4	=	3
P_2	6	-	5	=	1
P_3	2	-	0	=	2
			11		

available resource 1.

Sequence. $P_2 \rightarrow P_0 \rightarrow P_1 \rightarrow P_3$.

$1 + 5 = 6$

↑ max 6. hogaya h na toh iss kaam wale mai se koi bhi sequence mai le lenge.

↳ Deadlock avoidance

Based on Resource instance. Iss algorithm are of two types

1) Single instance
2) Multiple instance (Bankers algo)

(Algo used is Resource allocation Graph Algo)

Theory Question with one example with steps

Page no.:
Date: / /

Resource allocation graph algorithm

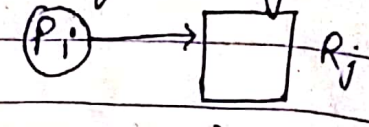
• dot nhi hoga. inmai.

$G = \{ V, E \}$

1) Process

Resource

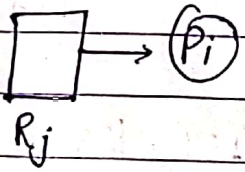
2) Request edge.



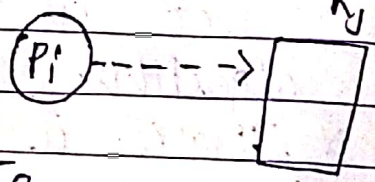
(P_i, R_j).

3) Assignment edge.

(R_j, P_i)

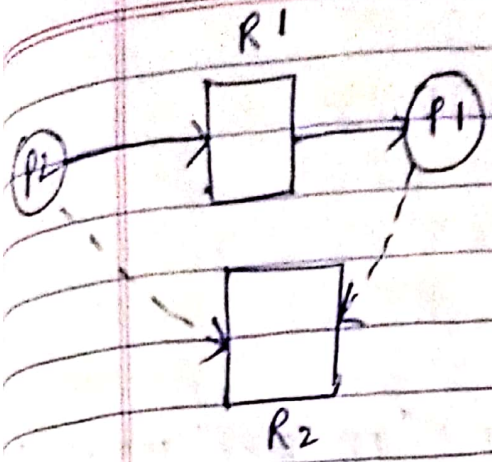


4,1) Claim Edge (P_i, R_j) (dotted line)



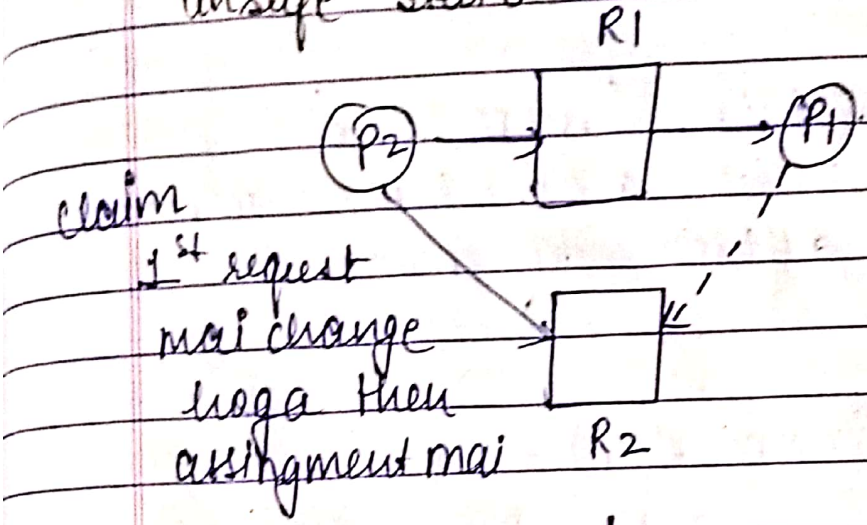
[Process P_i, resource R_j ke liye request kar sakta in future]

Two process P_1, P_2, two resource R_1, R_2 R_1 is allocated to P_1 and P_2 is requesting for R1. P1 and P2 can request for R2.

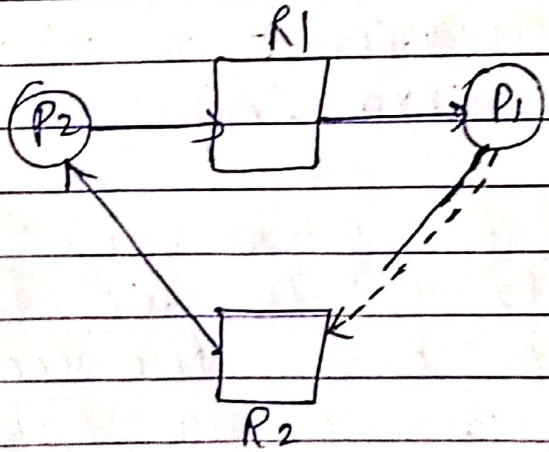


No cycle → safe state.

unsafe state.



claim
1st request
mai change
hoga then
assignment mai



cycle → unsafe state.

Numerical

Page no.:
Date: / /

2) Banker's algo

→ Used data structures are :-
→ n = process
→ m = Resource type.

i) Availability
represented by vector (m-size) column).

ii) Max
→ This is a matrix. (n x m size).
→ kisi process ko max. kitne resource chahiye defined by max.

iii) Allocation
matrix (n x m size)

iv) Need
Max - allocation = matrix.
Matrix = (n x m size).

Q-1) Consider a system with 5 processes P_0, P_1, P_2, P_3, P_4 and 3 resource types A, B, C. Resource A has 10 instances, B has 5 instances C has 7 instances. Suppose that at time T_0 the following snapshot of the system has been taken

allocate – total. Safe State ?

Total A B C diye hai toh available find karo.
10 5 7

Max-Allocation

Process	MAX			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	7	4	3	3	3	2
P1	3	2	2	2	0	0	1	2	2	5	3	2
P2	9	0	2	3	0	2	6	0	0	7	4	3
P3	2	2	2	2	1	1	0	1	1	7	5	3
P4	4	3	3	0	0	2	4	3	1	10	5	5
				7	2	5				10	5	7

To find Available = Total – (Allocation) sum

10-7 5-2 7-5
3 2 2

Sequence $P_1 \rightarrow P_3 \rightarrow P_0 \rightarrow P_2 \rightarrow P_4$ Safe sequence

Phele available calculate karne usmai – karnge total resource ka jo instance diya hai Allocation mai.

available + allocation add karte jaa (woh process mai ka jo liya ja raha hai).

Page no.:
Date: / /

Sem

Process	MAX. A B C D	Allocation A B C D	Need A B C D	Available A B C D
P0	0 0 1 2	0 0 1 2	0 0 0 0	~~1 5 2 0~~
P1	1 0 0 0	1 7 5 0	0 7 5 0	~~1 5 3 2~~
P2	1 3 5 4	2 3 5 6	1 0 0 2	~~2 8 8 6~~
P3	0 6 3 2	0 6 5 2	0 0 2 0	~~3 8 8 6~~
P4	0 0 1 4	0 6 5 6	0 6 4 2	~~3 14 11 8~~
				3 14 12 12
				Total.

(i) Compute the total no. of resources of each type.

ii) What is the content of the matrix need.

iii) Is the system in a safe state? if yes then find the safe sequence.

iv) If a request from process P_i arrives for (0,4,2,0) can the request be granted immediately.

Sol^n:- (i). Allocation ke same + available.

(Allocation + available) = ~~(4,27,13,10)~~ (3,14,12,12).

ii)

iii) $P_0 \rightarrow P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4$ safe sequence.

iv) Yes.

$P_0 \rightarrow P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4$

(available mai phele P_0 ka allocation add kareinge).

Total

4

~~4 17 23 16~~

3 14 12 12

Total (3, 14, 12, 12)

(for iv)

Process	MAX A B C D	Allocation A B C D	Need A B C D	Available A B C D
P_0	0 0 1 2	0 0 1 2	0 0 0 0	2 1 0 0
P_1	1 7 5 0	0 4 2 0	1 3 3 0	~~2 1 1 2~~
P_2	2 3 5 6	1 3 5 4	1 0 0 2	~~3 4 6 6~~
P_3	0 6 5 2	0 6 3 2	0 0 2 0	~~3 8 8 8~~
P_4	0 6 5 6	0 0 1 4	0 6 4 2	~~3 14 11 8~~
				3 14 12 12

(dekhna ki MAX se process ko)

↳ $P_0 \rightarrow P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4$

[Safe sequence.]

Ismai available = Total - Allocation Total.

(3, 14, 12, 12) - (1, 13, 12, 12)

(2, 1, 0, 0) → available.

Then jo bhi process max. mai se ismai range mai aega usko denge.

need dekho available se kaam hogi fabhi uske corresponding process allocation add hoga.

Page No.:
Date: 1.1

Process	MAX			Allocated			Need			available		
	R1	R2	R3	R1	R2	R3						
P1	4	3	3	1	1	2	3	2	1	~~3~~	~~1~~	~~0~~
P2	3	2	2	2	1	2	1	1	0	~~5~~	~~2~~	2
P3	9	0	2	4	0	1	5	0	1	~~6~~	~~3~~	4
P4	7	5	3	0	2	0	7	3	3	~~10~~	~~5~~	7
P5	11	2	3	1	1	2	10	1	1	~~11~~	~~6~~	~~7~~
										11	6	7

i) Determine the total amount of resources of each type.

ii) Compute the need matrix.

iii) Determine if the state is safe or not using bankers algo.

iv) Would the following request be granted in the current state

a). $P1 < 3, 3, 1>$

b) $P2 < 2, 1, 0>$.

i) Total = (11, 6, 7)

$P_1 \rightarrow P_2 \rightarrow P_4 \rightarrow P_3$ $P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$

iv) The request $P_1 <3, 3, 1>$ cannot be granted as the current state because P_1 is requesting for 3 instance of resource R_2 which is not available.

Page no.:
Date: / /

Total.

11, 6, 7.

iv) a).

	MAX.	Allocation	Need.	Available.
P1	4 3 3	3 3 1	1 0 2	1 1 1
P2	3 2 2	2 1 2	1 1 0	
P3	9 0 2	4 0 1	5 0 1	
P4	7 5 3	0 2 0	7 3 3	
P5	11 2 3	1 1 2	10 1 1	

Total - Allocation $\geq$ Available

iv). R2 is more than available so no request granted.

iv) b).

	MAX.	Allocation	Need	Available
	A B C	A B C	A B C	A B C
P1	4 3 3	3 1 2	3 2 1	~~3 1 2~~
P2	3 2 2	2 1 0	1 1 2	~~5 2 2~~
P3	9 0 2	4 0 1	5 0 1	~~6 3 4~~
P4	7 5 3	0 2 0	7 3 3	~~10 3 5~~
P5	11 2 3	1 1 2	10 1 1	~~10 5 5~~
				11 6 7

Sequence $P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$

a) $P_2 \langle 1,0,0 \rangle$

b) $P_1 \langle 1,0,0 \rangle$

V. Imp

Process	Allocated			Max			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	2	2	3	3	6	8	1	4	5	~~7~~	~~7~~	~~10~~
P2	2	0	3	4	3	3	2	3	0	~~9~~	~~9~~	~~13~~
P3	1	2	4	3	4	4	2	2	0	~~11~~	~~9~~	~~16~~
										12	11	20

Sequence $P_1 \rightarrow P_2 \rightarrow P_3$.

Total = allocation + available = (12, 11, 20)

Hence, sequence is in safe state.

ii) a). $P_2 \langle 1,0,0 \rangle$.

Process	Allocation			MAX			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	2	2	3	3	6	8	1	4	5	~~8~~	~~7~~	~~13~~
P2	1	0	0	4	3	3	3	3	3	~~10~~	~~9~~	~~16~~
P3	1	2	4	3	4	4	2	2	0	~~11~~	~~9~~	~~16~~
										12	11	20

Available = Total - Allocation

(12, 11, 20 - 4, 4, 7) = (8, 7, 13)

Sequence = $P_1 \rightarrow P_2 \rightarrow P_3$ safe state.

P1 < 1,0,0 >

b)

Process	Allocation	MAX	Need	Available
P	A B C	A B C	A B C	A B C
P1	1 0 0	3 6 8	2 6 8	8 9 13
P2	2 0 3	4 2 3	2 2 0	9 9 13
P3	1 2 4	3 4 4	2 2 0	11 9 16
				12 11 20

$P_1 \rightarrow P_2 \rightarrow P_3$ safe state.

Q3) A system contains 6 units of resource and n processes that use the resource. What is the maximum value of n for which the system will be deadlock free if the maximum requirement of each process is 3.

Ans) Resource = 6 units.

Process	No. of Resources
if n=1	3
n=2	3 3
n=3	3 2 1
max. ← n=4	3 1 1 1 ✓ Ans
n=5	2 1 1 1 1

when deadlock occurs

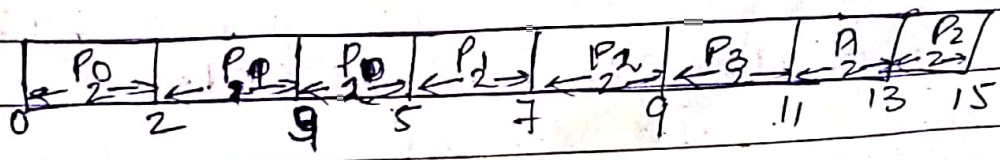
Q.4 Is it possible to have a deadlock involving only a single process? Explain

Ans)

resives = 12 = instances → apply banker's algo
if instances = 1 . then use single resources

Process	AT	BT	Priority	WT	TAT	CT
P0	0.000	3	3	2	5	5
P1	1.001	6	2	6	12	13
P2	4.001	4	1 (High)	7	11	15
P3	6.001	2	2	3	5	11

a) RR (TQ = 2)



AWT = (2+6+7+3)/4 = 4.5

TAT = 33/4 = 8.25