

Computer Science & IT

Database Management System

Transaction
&
Concurrency control

Lecture No. 07



By- Vishal Sir

Recap of Previous Lecture



- ✓ Topic Classification based on serializability
- ✓ Topic Conflicting and non-conflicting operations
- ✓ Topic Conflict serializable schedule



Topics to be Covered



✓ Topic

Precedence graph

✓ Topic

Topological order

✓ Topic

Practice questions on Conflict serializable schedule

Topic

View serializable schedule

Topic

View equivalence condition

Topic

Practice questions on View serializable schedule



Topic : Testing condition for conflict serializable schedule

- ① If precedence graph is cyclic, then corresponding schedule is not a Conflict serializable schedule
{ But we don't know whether the schedule is a serializable schedule or not. }
- ② If precedence graph is acyclic, then corresponding schedule is a Conflict serializable schedule (hence a serializable schedule), and Conflict equivalent serial schedule w.r.t. given schedule can be given by "topological order" of the precedence graph



Topic : Topological order

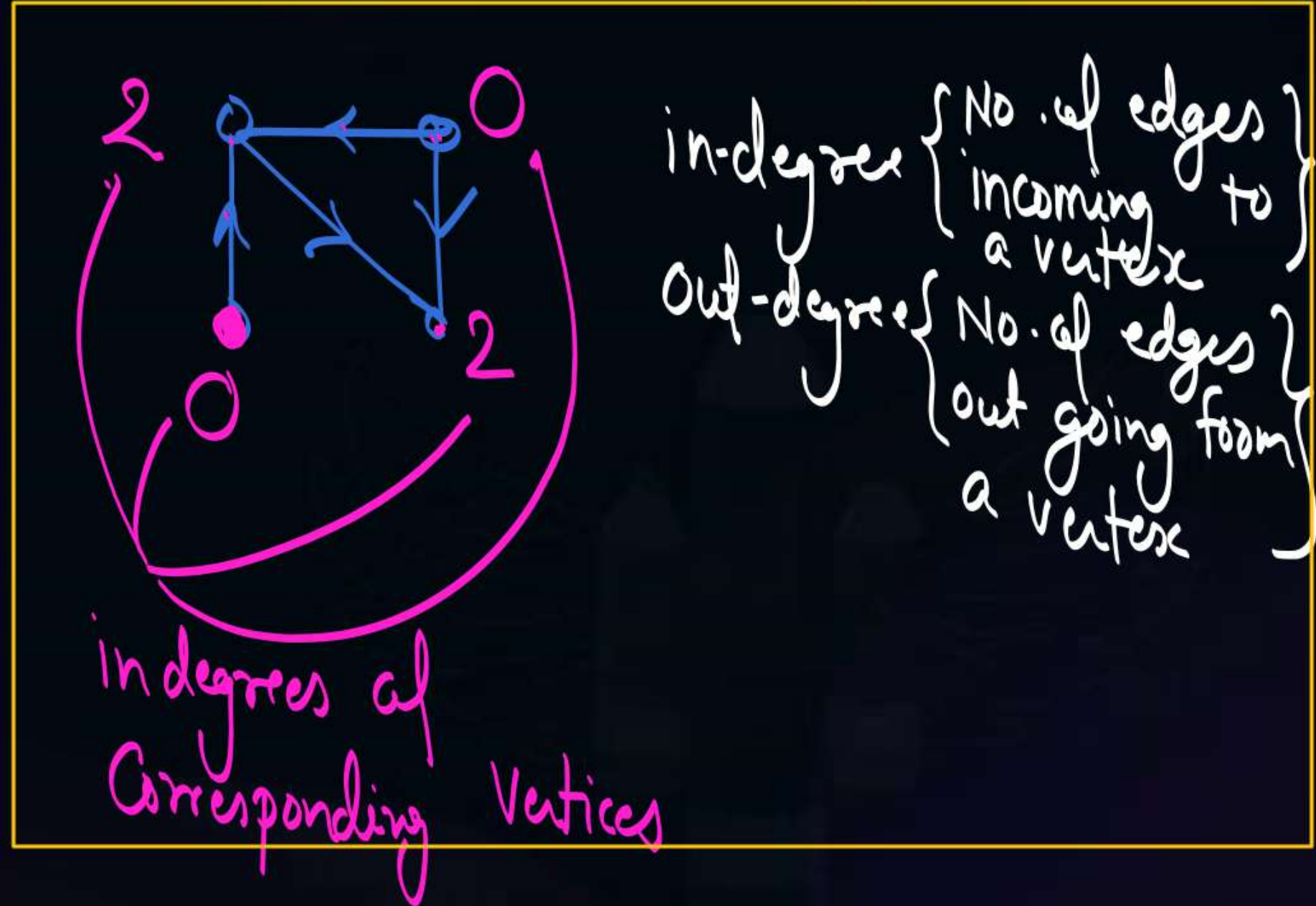


Let $G = (V, E)$ be an acyclic precedence graph. ie., a directed graph

Step-1 :- Visit a vertex $V \in G$, such that indegree of vertex $V = 0$, and delete vertex V from graph G .

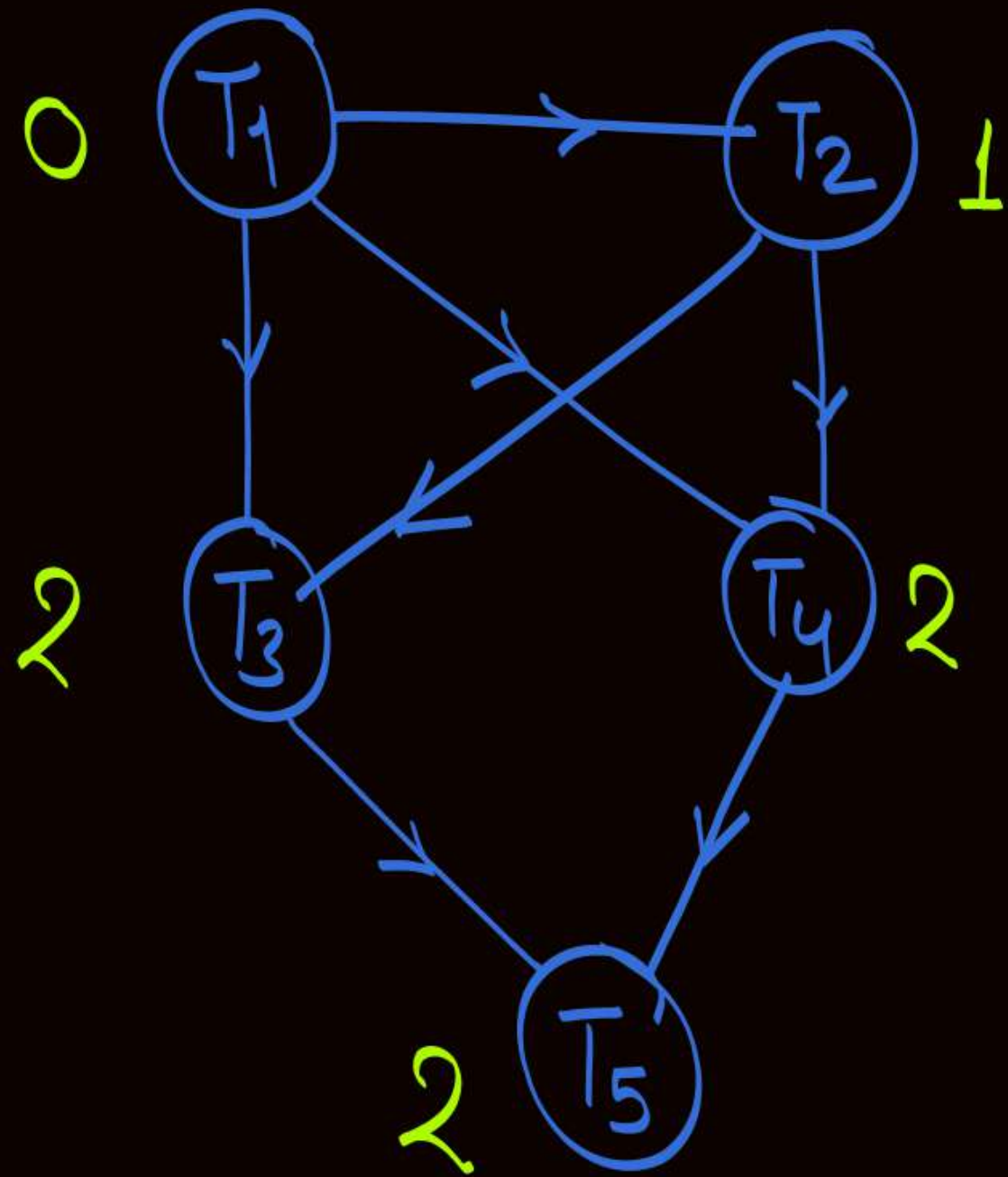
Step-2 :- Repeat Step-1 until graph becomes Empty

The order in which the vertices of graph G can be visited defines the topological order of that graph.

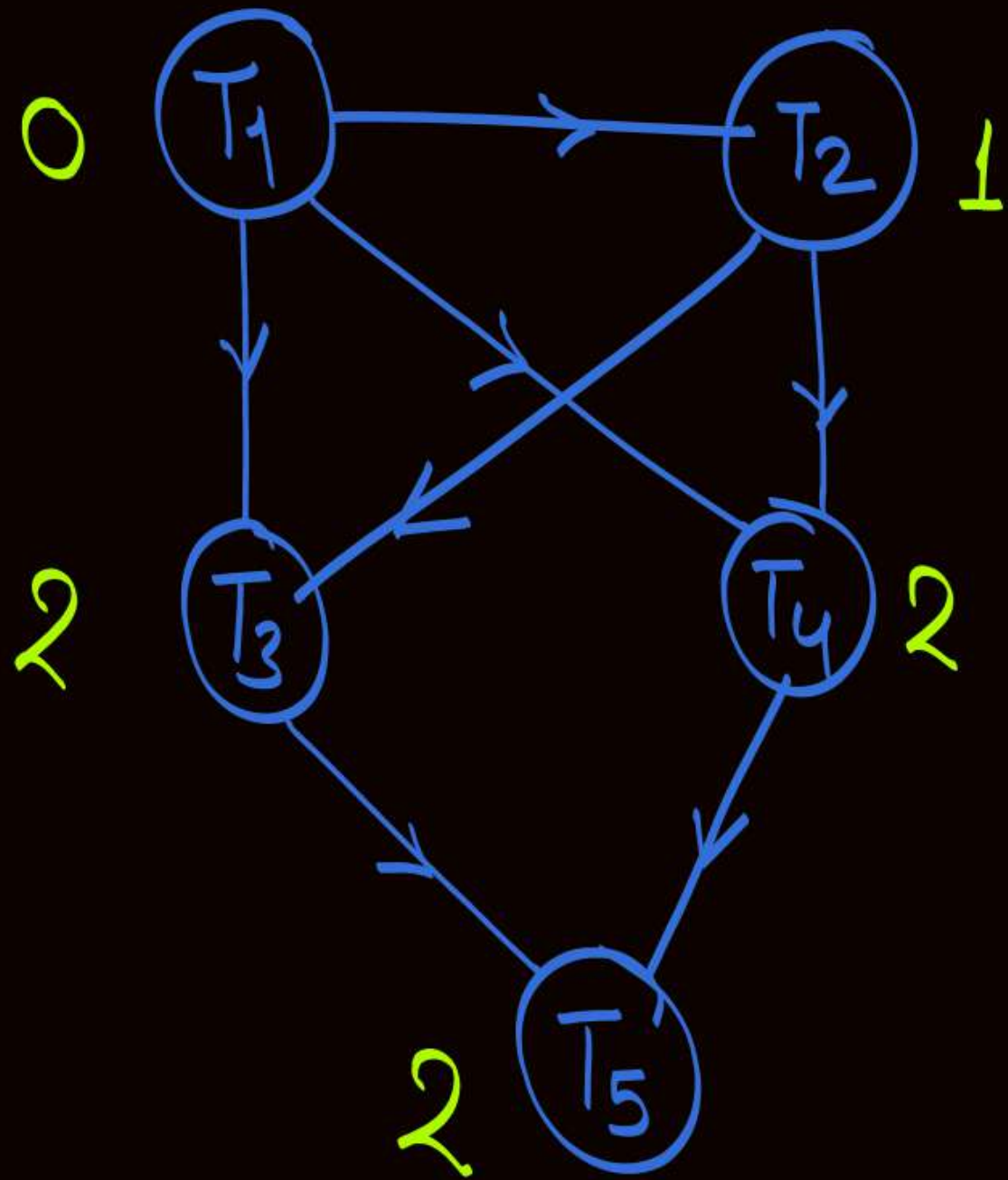


Note:- There may be more than one topological order
for a given acyclic precedence graph

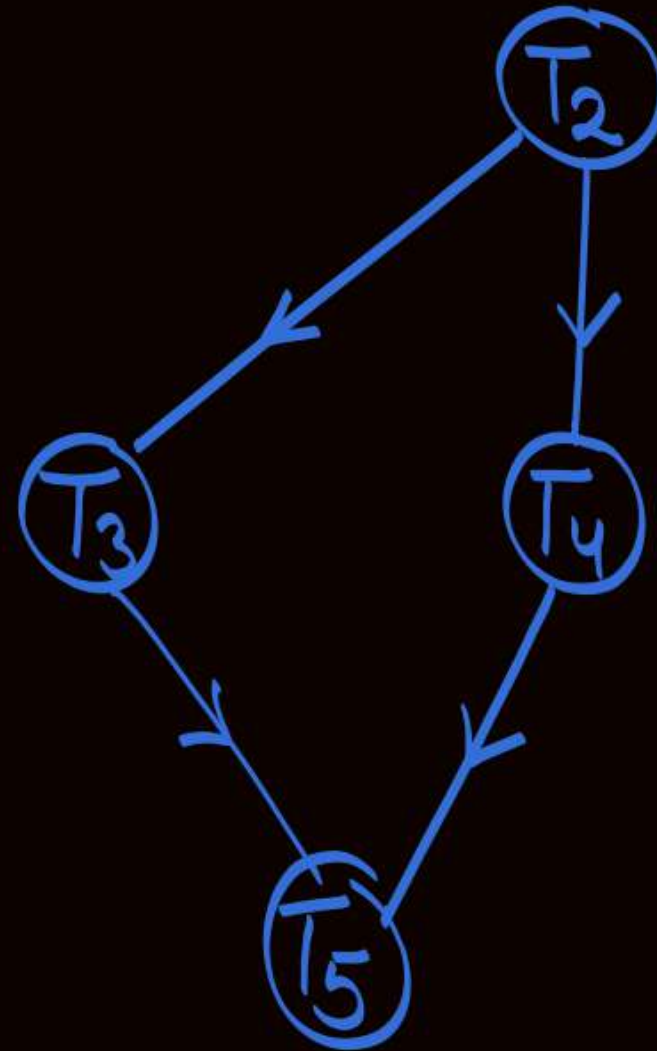
eg: Find the topological order of the following graph,



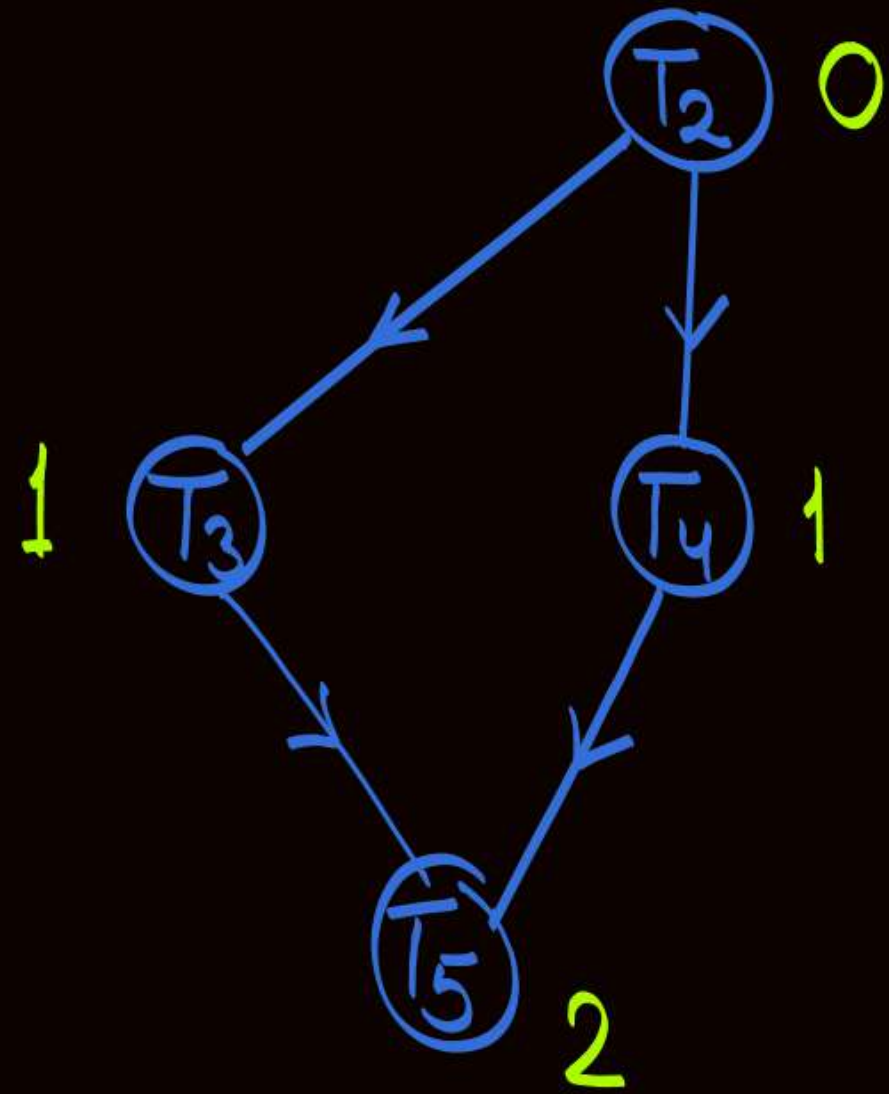
eg: Find the topological order of the following graph,



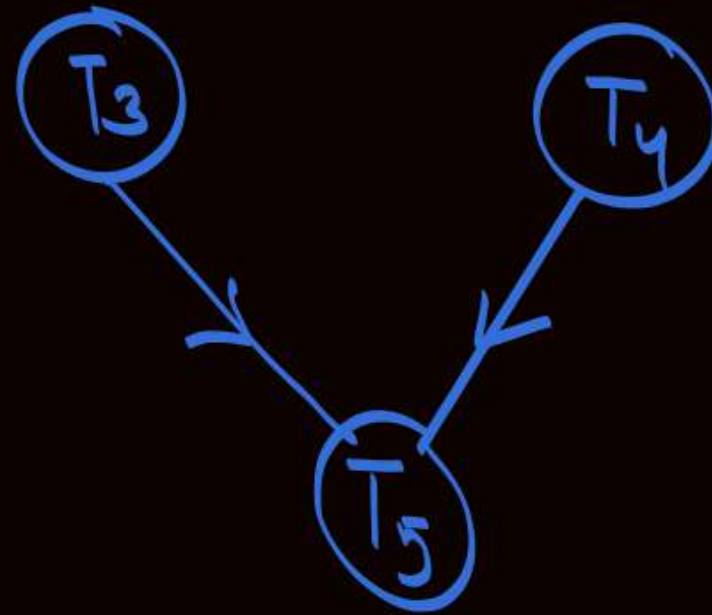
① T_1 deleted



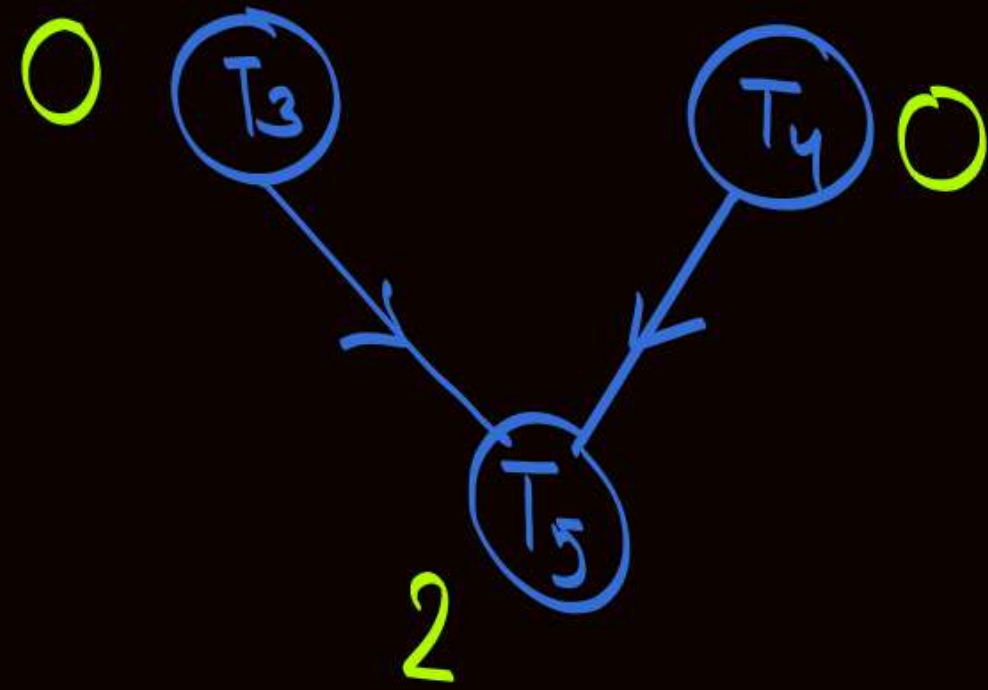
eg: Find the topological order of the following graph,



② T_2 deleted



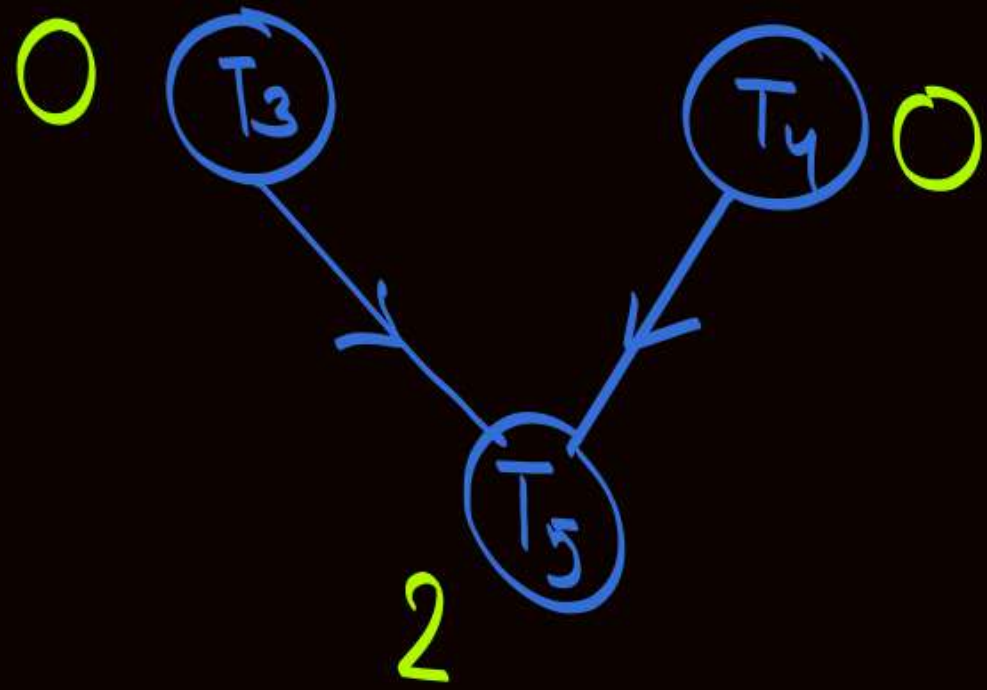
eg: Find the topological order of the following graph,



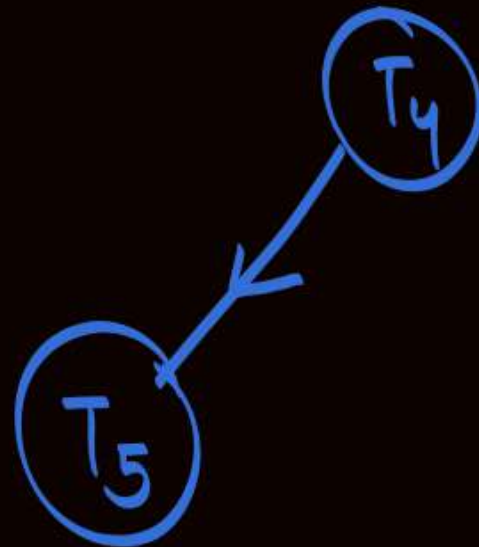
- There are two vertices of indegree = 0
 \therefore Multiple topological order will exist.

eg: Find the topological order of the following graph,

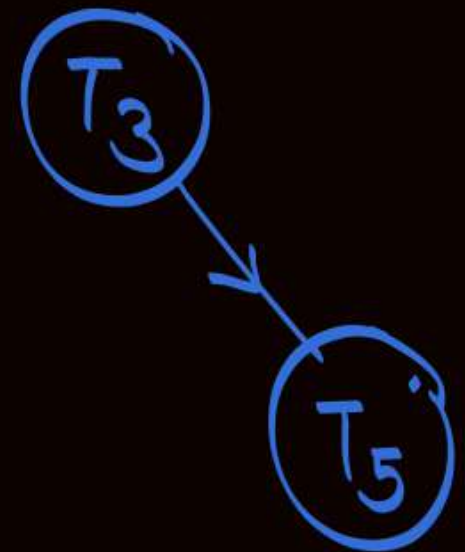
- There are two vertices of indegree = 0
 \therefore Multiple topological order will exist.



(a) ③ T_3 deleted

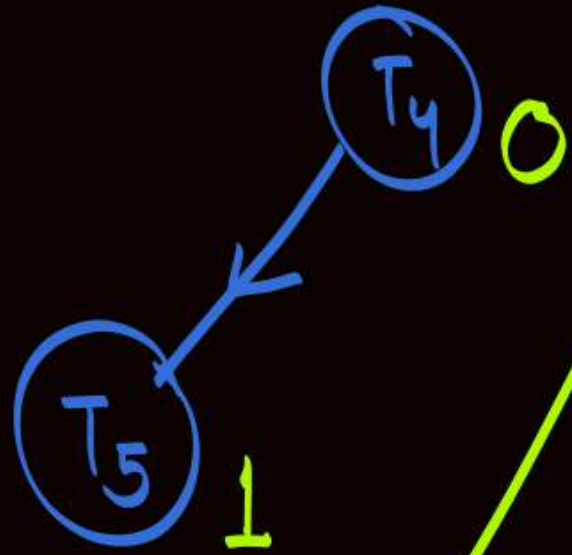


(b) ③ T_4 deleted



eg: Find the topological order of the following graph,

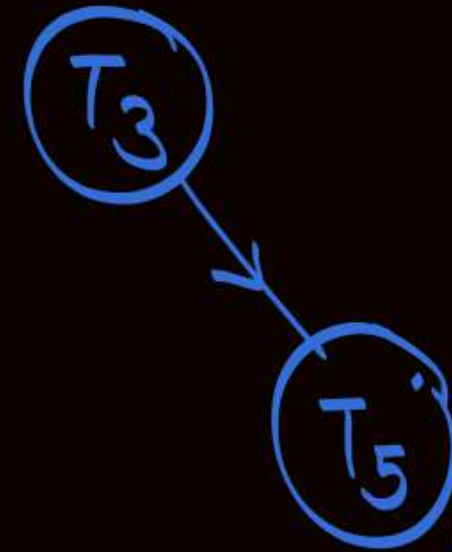
(a) ③ T_3 deleted



(a) ④ T_4 deleted



(b) ③ T_4 deleted



(b) ④ T_3 deleted



eg: Find the topological order of the following graph,

(a) ④ T_4 deleted

④
⑤

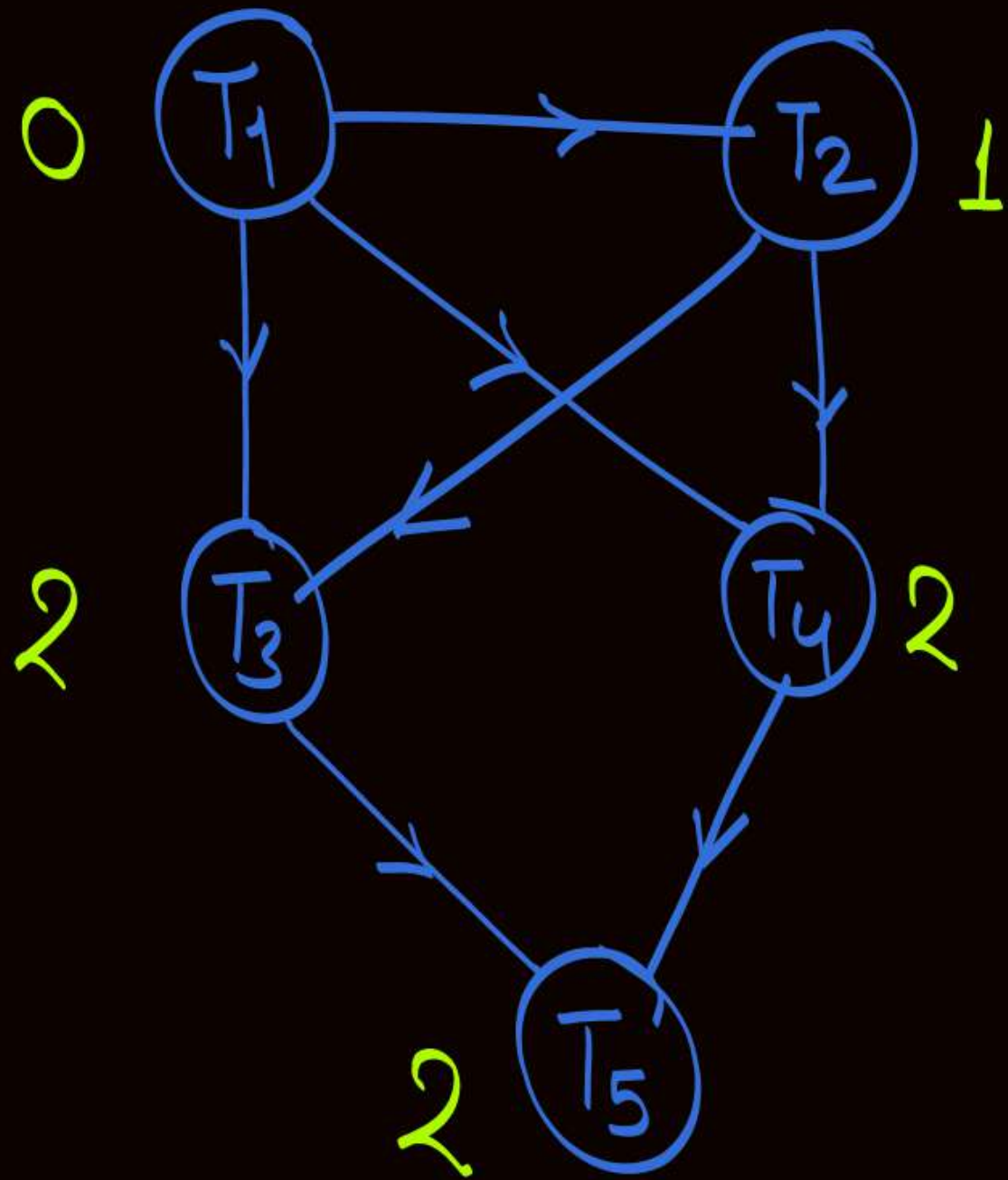
(a) ⑤ T_5 deleted
graph becomes
Empty

(b) ④ T_3 deleted

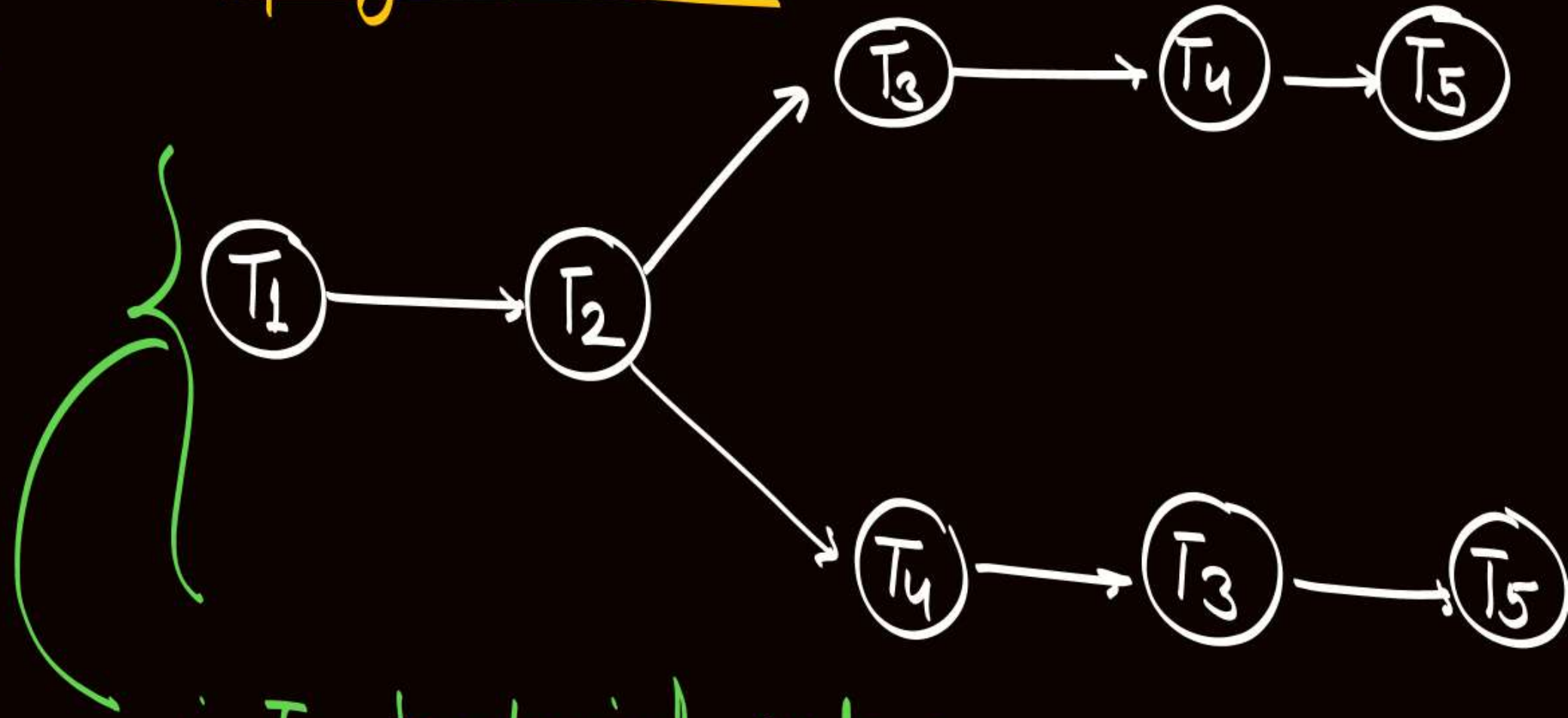
④
⑤

(b) ⑤ T_5 deleted
graph becomes
Empty

eg: Find the topological order of the following graph,



Topological orders:



∴ Two topological orders.

(i) $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5$

& (ii) $T_1 \rightarrow T_2 \rightarrow T_4 \rightarrow T_3 \rightarrow T_5$

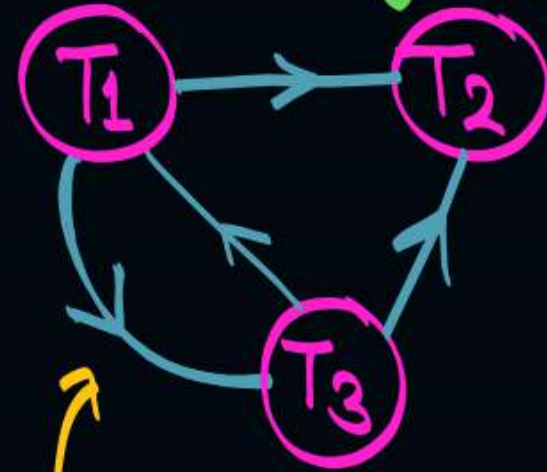
Note:- If there are more than one topological order for the given precedence graph, then it means that given schedule is conflict equivalent to more than one serial schedules, and those serial schedules are defined by different topological orders.

#Q. Check whether the schedule is conflict serializable schedule or not? ✓

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 |
|----------------------|----------|----------------------|
| | | $R_3(x)$ $R_3(z)$ |
| $R_1(x)$ $W_1(x)$ | | |
| | | $W_3(y)$ $W_3(x)$ |
| | $R_2(x)$ | |
| $R_1(y)$ $W_1(y)$ | | |

Precedence graph

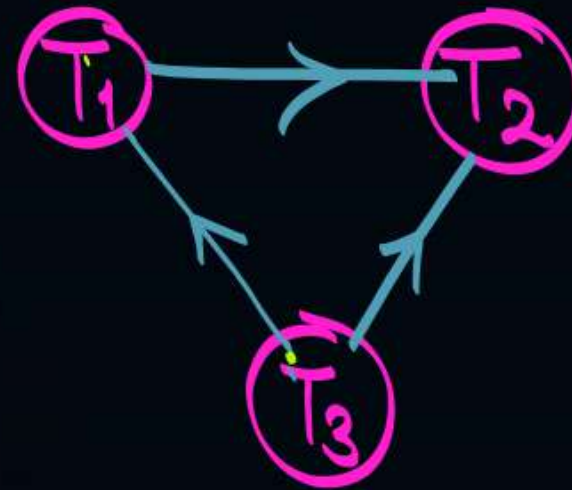


Cyclic precedence graph
∴ Not a C.S.S.

#Q. Check whether the schedule is conflict serializable schedule or not?

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 |
|----------------------|--|----------------------|
| | | $R_3(y)$ $R_3(z)$ |
| $R_1(x)$ $W_1(x)$ | | |
| $R_1(y)$ $W_1(y)$ | $R_2(z)$ | $W_3(y)$ $W_3(z)$ |
| | $R_2(y)$ $W_2(y)$ $R_2(x)$ $W_2(x)$ | |



Acyclic precedence graph

∴ Schedule is a Conflict serializable schedule

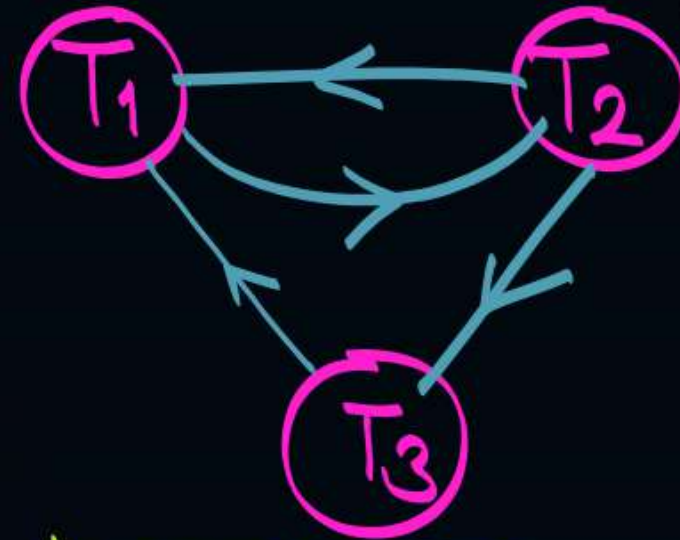
Topological order is
 $T_3 \rightarrow T_1 \rightarrow T_2$

∴ Conflict equivalent serial schedule is
 $T_3 \rightarrow T_1 \rightarrow T_2$

#Q. Check whether the schedule is conflict serializable schedule or not?

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| | $R_2(A)$ | |
| $R_1(B)$ | $W_2(A)$ | |
| | | $R_3(A)$ |
| $W_1(B)$ | | |
| $W_1(A)$ | | |
| | $R_2(B)$ | |
| | $W_2(B)$ | |

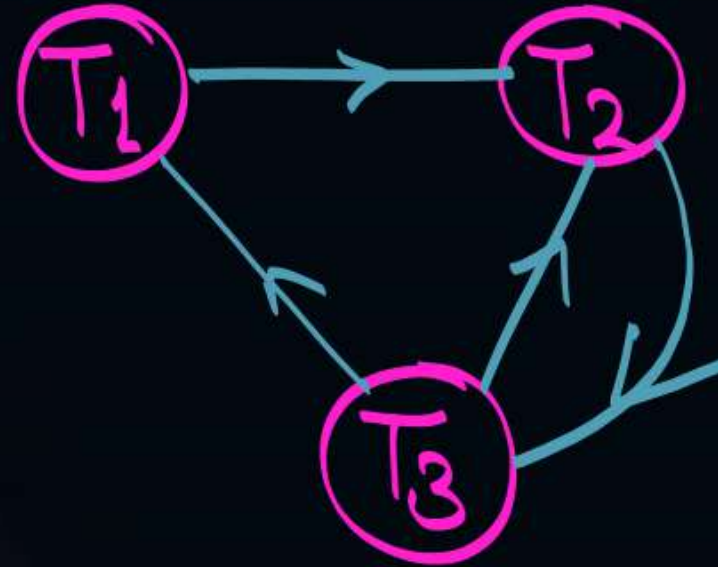


Cyclic precedence graph
is Not a C.S.S.

#Q. Check whether the schedule is conflict serializable schedule or not?

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| | | $W_3(B)$ |
| $R_1(A)$ | | |
| $W_1(B)$ | | |
| | $R_2(B)$ | |
| | $W_2(C)$ | |
| | | $R_3(C)$ |

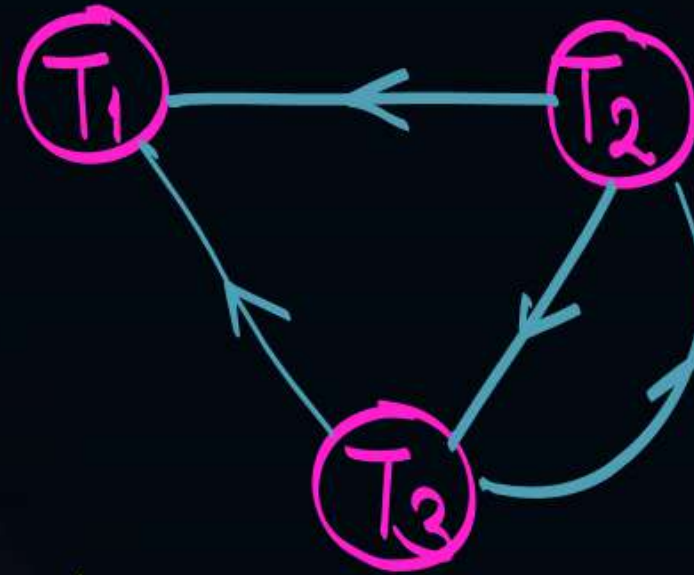


Cyclic precedence graph
 \therefore Not a C.S.S.

#Q. Check whether the schedule is conflict serializable schedule or not?

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| | $R_2(z)$ | |
| | $R_2(y)$ | |
| | $W_2(y)$ | |
| | | $R_3(y)$ |
| | | $R_3(z)$ |
| $R_1(x)$ | | |
| $W_1(x)$ | | |
| | | $W_3(y)$ |
| | | $W_3(z)$ |
| | $R_2(z)$ | |
| $R_1(y)$ | | |
| $W_1(y)$ | | |

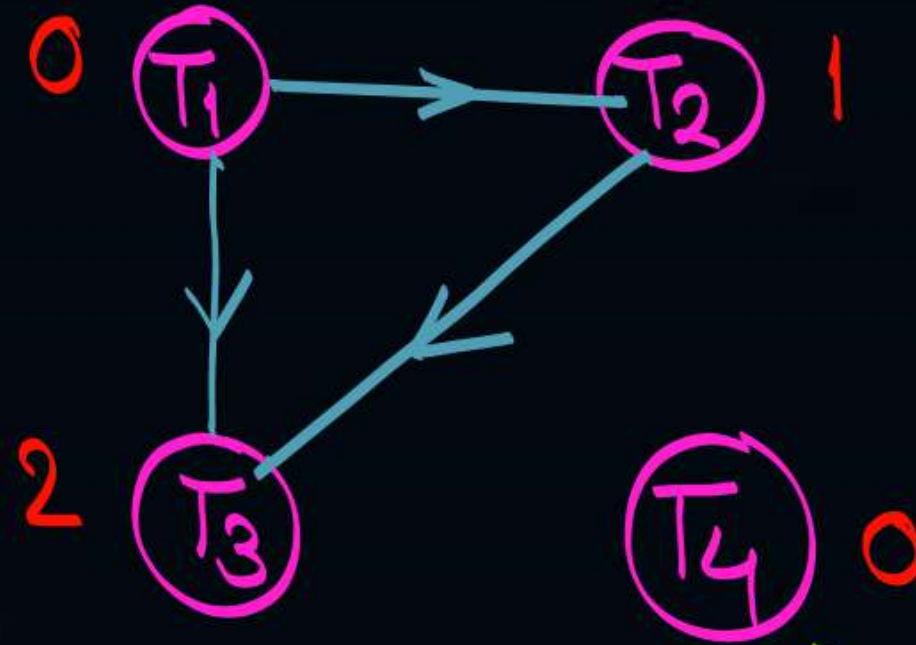


Cyclic precedence graph
 ∴ Not a C.S.S.

#Q. Check whether the schedule is conflict serializable schedule or not?

If conflict serializable schedule then identify all conflict equivalent serial schedules.

| T_1 | T_2 | T_3 | T_4 |
|----------|----------|----------|----------|
| $R_1(A)$ | $R_2(A)$ | $R_3(A)$ | $R_4(A)$ |
| $W_1(B)$ | $W_2(B)$ | $W_3(B)$ | |



Acyclic precedence graph
 ∴ Schedule is a
 Conflict serializable
 Schedule

Topological orders

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$
 $T_1 \rightarrow T_2 \rightarrow T_4 \rightarrow T_3$
 $T_1 \rightarrow T_4 \rightarrow T_2 \rightarrow T_3$
 $T_4 \rightarrow T_1 \rightarrow T_2 \rightarrow T_3$

Four topological order

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$
 $T_1 \rightarrow T_2 \rightarrow T_4 \rightarrow T_3$
 $T_1 \rightarrow T_4 \rightarrow T_2 \rightarrow T_3$
 $T_4 \rightarrow T_1 \rightarrow T_2 \rightarrow T_3$

Given schedule is
 Conflict equivalent to
 '4' serial schedules

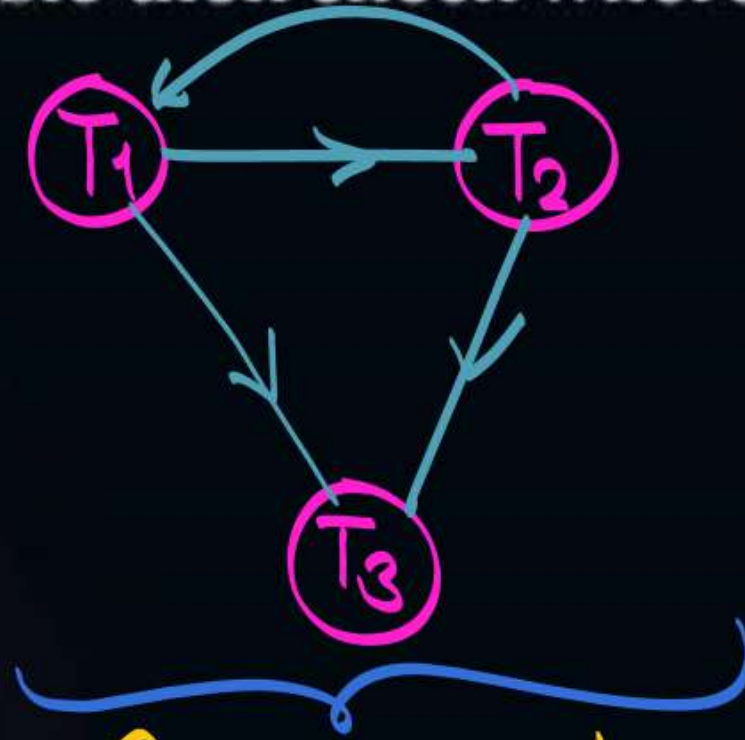
#Q. Check whether the schedule is conflict serializable schedule or not?

If not conflict serializable then check where the schedule is serializable schedule or not?

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| $R_1(A)$ | | |
| $W_1(A)$ | $W_2(A)$ | |
| | | $W_3(A)$ |

Read from initial DB

A is finally updated by T_3



Cyclic precedence graph
 ∴ Not a C.S.S.

Schedule is not a C.S.S., but still a serializable schedule

But, behaviour of given schedule is equivalent to serial schedule $T_1 \rightarrow T_2 \rightarrow T_3$

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| $R_1(A)$ | | |
| $W_1(A)$ | $W_2(A)$ | |
| | | $W_3(A)$ |

Read from initial DB

A is finally updated by T_3

∴ given schedule is a serializable schedule

eg.

(S)

| T_1 | T_2 |
|----------|----------|
| $w_1(A)$ | |
| $w_1(A)$ | $w_2(A)$ |
| $w_1(A)$ | $w_2(A)$ |
| $w_1(A)$ | |

final writer
is T_1

Precedence graph



Not a C.S.S.

Behaviours of schedule S is equivalent to $T_2 \rightarrow T_1$

Given schedule is not a C.S.S., but a serializable schedule.

\Downarrow

| T_1 | T_2 |
|----------|----------|
| | $w_2(A)$ |
| | $w_2(A)$ |
| $w_1(A)$ | |
| $w_1(A)$ | |
| $w_1(A)$ | |

final writer
is T_1

#Q. Check whether the schedule is conflict serializable schedule or not?

If not conflict serializable then check where the schedule is serializable schedule or not?

S

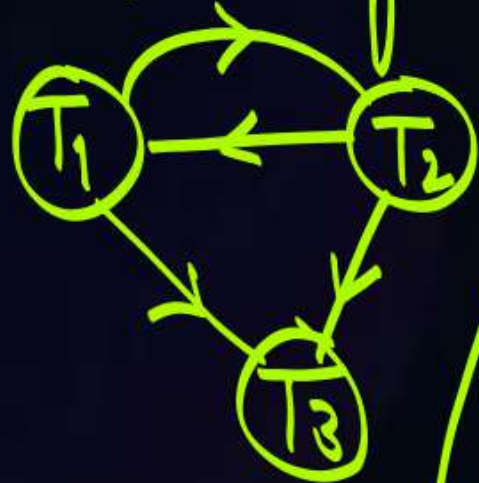
| T ₁ | T ₂ | T ₃ |
|--------------------|--------------------|--------------------|
| | W ₂ (A) | |
| W ₁ (A) | W ₂ (A) | |
| | | W ₃ (A) |

Schedule S is equivalent to two Serial Schedules

$T_1 \rightarrow T_2 \rightarrow T_3$ } $T_2 \rightarrow T_1 \rightarrow T_3$

\therefore Schedule is a Serializable Schedule

Precedence graph



Cyclic
 \therefore Not C.S.S.



Topic : NOTE



If "Schedule is not a Conflict serializable schedule but a serializable schedule"

then, "at least one blind write opⁿ exist in the schedule"

but Converse of the statement need not be true

Note:- If δ schedule is not a Conflict serializable schedule
and there is no blind write opⁿ in the
 δ schedule, then δ schedule can not be a
serializable schedule.

Note:- If δ schedule is not a Conflict serializable schedule
and there is a blind write opⁿ in the
 δ schedule, then δ schedule may or may not
be serializable



Topic : NOTE



* Conflict serializability Condition is a "if-then" Condⁿ.

i.e., if "schedule is a Conflict serializable schedule"
then "schedule is a serializable schedule."

if "if" Condⁿ is true, then "then" Condⁿ
will also be true,

if "if" Condⁿ itself is false, then there is no
restriction on then Condⁿ (i.e. schedule may or may not
be serializable)



Topic : NOTE



There are many serializable schedules which are not conflict serializable schedule

- i.e., Conflict serializability Condⁿ does not cover all serializable schedule, ∴ We define view serializability.

Note:-

View Serializability Condⁿ is "if and only if" Condⁿ

i.e. Schedule is a View serializable schedule
if and only if schedule is a serializable schedule



Topic : View serializable schedule

If given schedule is view equivalent to at least one of the serial schedule, then it is called a **view serializable schedule**.



Topic : View equivalent condition

- Consider two schedules S1 and S2
Schedules S1 and S2 are called view equivalent if the following three conditions hold true for them –
 - ❑ **Condition-01** : For each data item X, if transaction T_i reads X from the initial database in schedule S1, then in schedule S2 also, T_i must perform the initial read of X from the initial database.

Thumb Rule

- ❑ “Initial readers must be same for all the data items”.



Topic : View equivalent condition

Condition-02 :

If transaction T_i reads the value of data item X updated by the transaction T_j in schedule S_1 , then in schedule S_2 also, transaction T_i must read the value of data item X updated by the transaction T_j .

Thumb Rule

- "Write-read sequence must be same."



Topic : View equivalent condition

Condition-03 :

For each data item X , if X is finally updated by transaction T_i in schedule S_1 , then in schedule S_2 also, X must be finally updated by transaction T_i .

Thumb Rule

- ❑ “Final writers must be same for all the data items”.

Note:

Above three conditions together define the conditions for two schedules to be Equivalent



2 mins Summary

Topic

Precedence graph

Topic

Topological order

Topic

Practice questions on Conflict serializable schedule

Topic

View serializable schedule

Topic

View equivalence condition

Topic

Practice questions on View serializable schedule

THANK - YOU