Conflict Serializable Schedule - Solved Exercise

Question:

For each of the schedules below, indicate whether they are conflict serializable. If you answer yes, then give the equivalent serial order of the transactions.

(i) Schedule S1: R1(A), R1(B), W1(A), R2(B), W2(D), R3(C), R3(B), R3(D), W2(B), W1(C), W3(D)

(ii) Schedule S2: R1(A), R1(B), W1(A), R2(B), W2(A), R3(C), R3(B), R3(D), W2(B), W1(C), W3(D)

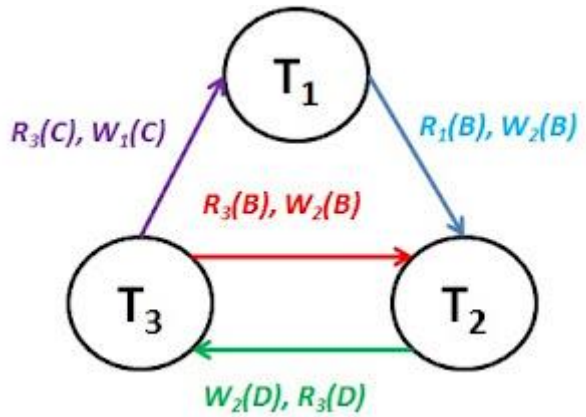
Solution:

> ***Conflict serializable schedule*: A schedule is conflict serializable if it can be transformed into an equivalent serial schedule by swapping pairs of non-conflicting instructions. Two instructions conflict if they involve the same data item and at least one of them is a WRITE.**

To find whether the given schedule is conflict serializable or not, we can draw precedence graph. Precedence graph can be constructed by carefully analyzing the conflicting instructions. For every conflicting instruction, an edge can be inserted into the precedence graph. At the end, if the graph has not formed a cycle, then we would say that the schedule S is conflict serializable, otherwise not.

(i) Schedule S1:

Schedule S1 is not conflict serializable because the precedence graph for S1 forms a cycle (refer the image below).



You could see two cycles here; $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$ and $T_2 \rightarrow T_3 \rightarrow T_2$. Second one is discussed below;

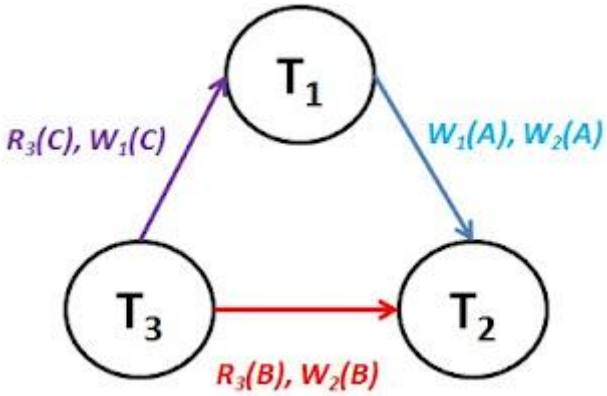
The conflicting instructions are highlighted in color below;

R1(A), R1(B), W1(A), R2(B), **W2(D)**, R3(C), **R3(B)**, **R3(D)**, **W2(B)**, W1(C), W3(D)

In the first conflict (**green color**), transaction T_2 is writing D and T_3 is reading D. In the second one, transaction T_3 is reading B and T_2 is writing B. And these two conflicts end up in creating a cycle. **Hence, schedule S1 is not conflict-serializable**.

(ii) Schedule S2:

Schedule S2 is conflict serializable because the precedence graph f or S3 doesn't have a cycle in it (refer the image below).



Serialization order for this schedule is T_3, T_1, T_2. That is, the execution of this schedule will be same as executing transactions in the order T_3, T_1 and T_2.