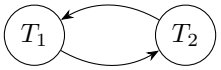
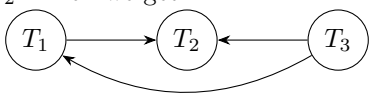
Response: No. T_1 does not finish completely before T_2 starts.

Response: We write(A) in T_1 before T_2 read(A), so $T_1 \to T_2$. But we read(B) in T_2 before T_1 write(B), so $T_2 \to T_1$. Therefore,we have



so there is a cycle, and therefore it is not conflict serializable.

Response: We write(A) in T_1 before write(A) in T_2 and read(A) in T_3 so $T_1 \to T_2$. We write(B) in T_3 before read(B) in T_1, T_2 so $T_3 \to T_1, T_2$. Then we get



Response: Yes, there is no cycle in the graph, so we can find a serial ordering of the transactions. One possible ordering using topological sort is to go from $T_3 \to T_1 \to T_2$. This is because T_3 is a source node. T_1 becomes a source node when we visit T_3. T_2 is the last node left so we visit it last.