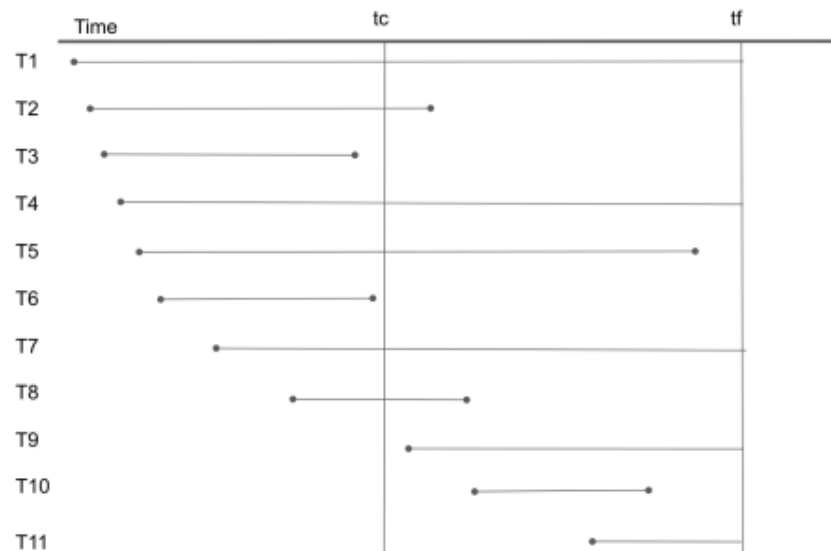


Task 5

5.a) Using the below diagram I have stated the three stages and what transactions are involved for each transaction. According to that I have done the using the recovery procedure for write through.



T1 - it must be rolled back, it was still running at the point of failure (redo)

T2 - it reached completion (it will need to be redone/rollforward, but that is not what this question is asking about)(redo)

T3 - It hasn't reached the time of checkpoint and T3 can be ignored at the checkpoint the system will have ensured the data was written to the disk (undo)

T4- it has completed the transaction starting from checkpoint time to time of failure.

T5- It has completed the checkpoint but it is on the way to the time of failure.(redo)

T6 - It hasn't reached the time of checkpoint and T6 can be ignored at the checkpoint the system will have ensured the data was written to the disk(undo)

T7 - T7 must be rolled back, it was still running at the point of failure

T8 - T8 has completed the checkpoint and it is still running towards the point of failure

T9 - T9 must be rolled back, it was still running at the point of failure

T10 - T10 does not need to be rolled back, it reached completion (it will need to be redone/rollforward)

T11 - T11 must be rolled back, it was still running at the point of failure

Using the recovery procedure for write through

Completed Transactions recorded in the log (green complete)

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
----	----	----	----	----	----	----	----	----	-----	-----

Step 1: Compile Redo and Undo list

REDO LIST					UNDO LIST					
T2	T4	T5	T8	T10	T1	T3	T6	T7	T9	T11

Step 2: UNDO incomplete or rolled back transactions starting from newest

T1	T3	T6	T7	T9	T11
----	----	----	----	----	-----

Step 3 : REDO committed transactions starting from oldest

T2	T4	T5	T8	T10
----	----	----	----	-----

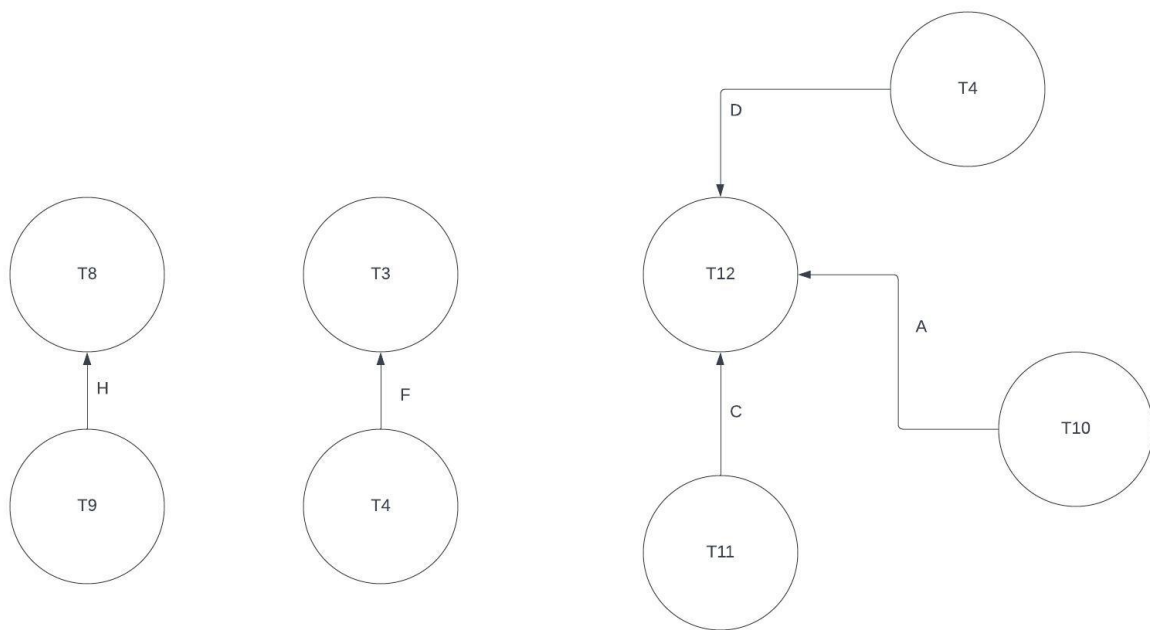
TIME	TRANSACTION	ACTION	A	B	C	D	E	F	G	H
0	T1	Read A	S(T ₁)							
1	T2	Read B		S(T ₂)						
2	T1	Read C			S(T ₁)					
3	T4	Read D				S(T ₄)				
4	T5	Read A	S(T ₅)							
5	T2	Read E					S(T ₂)			
6	T2	Update E					X(T ₂)			
7	T3	Read F						S(T ₃)		
8	T2	Read F						S(T ₂)		
9	T5	Update A	T ₅ wait on T ₁							
10	T1	Commit	X(T ₅)							
11	T6	Read A	T ₆ wait on T ₅							
12	T5	Rollback	S(T ₆)							
13	T6	Read C			S(T ₆)					
14	T6	Update C			X(T ₆)					
15	T7	Read G							S(T ₇)	
16	T8	Read H								S(T ₈)
17	T9	Read G							S(T ₉)	
18	T9	Update G							T ₉ wait on T ₇	
19	T8	Read E					S(T ₈)			
20	T7	Commit							X(T ₉)	
21	T9	Read H								S(T ₉)
22	T3	Read G							S(T ₃)	
23	T10	Read A	S(T ₁₀)							
24	T9	Update H								T ₉ wait on T ₈
25	T6	Commit								
26	T11	Read C			S(T ₁₁)					
27	T12	Read D				S(T ₁₂)				
28	T12	Read C			S(T ₁₂)					
29	T2	Update F						T ₂ wait on T ₃		
30	T11	Update C			T ₁₁ wait on T ₁₂					
31	T12	Read A	S(T ₁₂)							
32	T10	Update A	T ₁₀ wait on T ₁₂							

33	T12	Update D				T ₁₂ wait on T ₄				
34	T4	Read G							S(T ₄)	

1) This is the waits in the form for each item's transactions:

Item H: T₉ wait on T₈.
Item A: T₁₀ wait on T₁₂.
Item C: T₁₁ wait on T₁₂.
Item D: T₁₂ wait on T₄.
Item F: T₂ wait on T₃.

2) Wait for graph



3) From above given diagram in part 2 it says clearly that there is no deadlock exist in this diagram and there is transaction involved in this wait for graph happened.