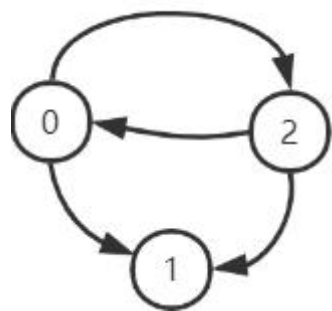


1、考虑下面的三个事务和它们的一个调度 S（时间从上往下依次增大）。判断 S 是否是冲突可串行化的调度？要求画出优先图并给出判断依据。

T0	T1	T2
r0(A)		
w0(A)		
		r2(A)
		w2(A)
	r1(A)	
r0(B)		
		r2(B)
w0(B)		
		w2(B)
	r1(B)	

解：
不是。



因为优先图中存在环，存在 $r0(B)r2(B)w0(B)w2(B)$ ， $r2(B)$ 与 $w0(B)$ 冲突， $w0(B)w2(B)$ 冲突，无论怎么调整，都不能转化为一个串行调度，所以这不是一个冲突可串行化调度。

2、设 T1、T2、T3 是如下三个事务：

T1: $A:=A+4$

T2: $A:=A*3$

T3: $A:=A^2$

初始 $A=2$

(1) 设三个事务都遵守两段锁协议，按 T2-T3-T1 的顺序执行，请给出一个不产生死锁的可串行化调度，并给出最终 A 的结果

时间	T1	T2	T3
1		LOCK-X(A)	
2		READ(A)	
3		$A:=A*3$	
4		WRITE(A)	
5		UNLOCK(A)	
6			LOCK-X(A)
7			READ(A)
8			$A:=A^2$
9			WRITE(A)

10			UNLOCK(A)
11	LOCK-X(A)		
12	READ(A)		
13	A:=A+4		
14	WRITE(A)		
15	UNLOCK(A)		

A=40

(2) 若这三个事务都遵循两段锁协议，请给出一个产生死锁的调度。

时间	T1	T2	T3
1	LOCK-S(A)		
2	READ(A)		
3		LOCK-S(A)	
4		READ(A)	
5	LOCK-X(A)		
6	等待		
7		LOCK-X(A)	
8		等待	
9			LOCK-S(A)
10			READ(A)
11			LOCK-X(A)
12			等待

3、考虑两个事务 T1, T2。其中，T1 显示账户 A 与 B 的内容：

T1: Read(B);
Read(A);
Display(A+B).

T2 表示从账户 B 转 50 美元到账户 A, 然后显示两个账户的内容：

T2: Read(B);
B := B-50;
Write(B);
Read(A);
A := A+50;
Write(A);
Display(A+B).

请给出一个满足时间戳协议的一个可能的调度。（注：Display(A+B)仅表示显示账户 A 和账户 B 的内容）

解：令 TS (T1) , TS (T2) 分别是事务 T1 和 T2 开始的时间戳，并且 TS (T2) < TS (T1)

T1	T2	W-ts(A)	R-ts(A)	W-ts(B)	R-ts(B)
	Read(B);				TS(T2)
	B := B-50;				
	Write(B);			TS(T2)	
Read(B);					TS(T1)
	Read(A);		TS(T2)		
	A := A+50;				
Read(A);			TS(T1)		
	Write(A);	TS(T2)			
Display(A+B).					
	Display(A+B).				

4、一个带检查点的日志内容如下，结束处发生了故障，请简述恢复算法的过程，并给出 Undo-List 和 Redo-List，以及数据库系统恢复后的 A,B,C 的值。

Start of the logs

...

<T4 start>

<T4, B, 1000, 1700>

<T5 start>

<checkpoint {T4, T5}>

<T5, C, 300, 100>

<T5 commit>

<T6 start>

<T6, A, 700, 600>

<T4, B, 1000>

<T4 abort>

← System crash, start recovery

解：找到最后的 checkpoint，令 Undo-List=L，本题中为{T4,T5}，从此处向下扫描，若是 start 的事务则加入 Undo-List，若是 commit 或 abort 的事务则从 Undo-List 移除，并添加到 Redo-List。将 Undo-List 中事务涉及的 ABC 赋旧值，将 Redo-List 中的事务重新执行，即对 ABC 赋新值。

Undo-List: T6

Redo-List: T4, T5

A=700, B=1000, C=100

5、设一个数据库系统启动后中，执行 4 个事务 T0、T1、T2 和 T3。四个事务的内容如下：

T0: A := A + 20 (读入数据库元素 A 的值，加上 20 后，再写回 A 的值)

T1: B := B - 10 (读入数据库元素 B 的值，减去 10 后，再写回 B 的值)

T2: C := C * 2 (读入数据库元素 C 的值，乘以 2 后，再写回 C 的值)

T3: D := D + 15 (读入数据库元素 D 的值，加上 15 后，再写回 D 的值)

除了这四个事务外，系统中无其他事务执行。设四个事务开始前，数据库元素 A、

B、C、D 的值分别为 A = 50, B = 30, C = 35, D = 15。在执行这四个事务的过程中，系统发生了故障。系统重启后，经故障恢复，数据库元素 A、B、C、D 的值被恢复为 A = 50, B = 20, C = 70, D = 15。故障恢复时，数据库系统日志文件中包含如下 12 条日志记录，这里只给出部分日志记录。已知该数据库管理系统使用基于 undo-redo 日志的故障恢复技术，这段日志中仅有 1 个不停机检查点(又称模糊检查点)。

1	
2	
3	
4	<start checkpoint (T0, T2)>
5	<end checkpoint>
6	
7	
8	
9	
10	
11	
12	

请根据上述信息，回答下列问题：

a.将日志文件补充完整，直接在上面的日志文件中填写。一个事务 T 启动时向日志文件中写入日志记录<T, start>;提交时向日志文件中写入日志记录<T, commit>;中止时向日志文件中写入日志记录<T, abort>;对数据库元素 X 进行修改时向日志文件中写入日志记录<T, X,X 的旧值, X 的新值>。

1	<T0, start>
2	<T0, A, 50, 70>
3	<T2, start>
4	<start checkpoint (T0, T2)>
5	<end checkpoint>
6	<T1, start>
7	<T1, B, 30, 20>
8	<T1, commit>
9	<T2, C, 35, 70>
10	<T3, start>
11	<T3, D, 15, 30>
12	<T2, commit>

b.在故障恢复过程中，哪些事务需要 redo，哪些事务需要 undo。说明理由。

解：由日志文件知，T1、T2 事务已提交，所以需要 redo

T0、T3 事务未提交，所以需要 undo

c.在故障恢复过程中，还会向日志文件添加什么日志记录？说明理由。

解：检查点记录，系统周期性的执行检查点，内存中所有的日志记录输出到日志

文件，将内存中所有修改了的数据块输出到数据文件，将一个日志记录输出到日志文件，该日志记录称为检查点记录，检查点记录的内容：**建立检查点时刻所正在执行的事务清单，这些事务最近一个日志记录地址**。