1. 考虑下面表中给出了 DBMS 对某四个事务操作五个数据项的调度按时间分布的表，其中$R(X)$和 $W(X)$分别代表对数据项X的“读”和“写”，T_i表示事务i，t_n表示第n个时刻，请阅读下表，并回答以下几个问题:

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
T_1	R(A)			R(C)			R(B)		W(C)		
T_2						R(C)				W(D)	W(E)
T_3		R(E)			W(A)						
T_4			W(B)					R(D)			

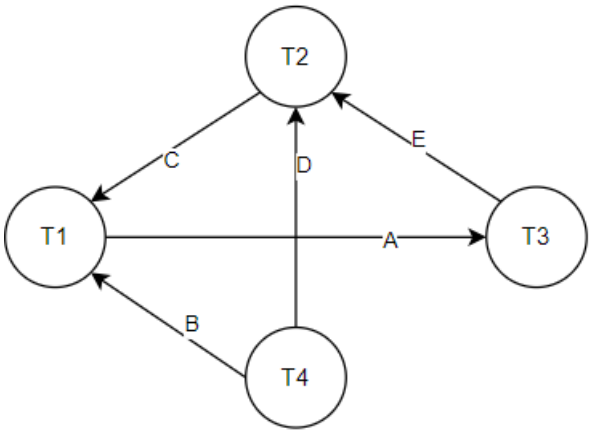
1.1 这是否是一个串行的调度?

答案:

否

1.2 请仿照课上幻灯片中的画法，画出上表中所示事务的优先图（请在箭头上标明事务操作的数据项）

答案:



1.3 根据你画出的事务优先图，该调度是否是个冲突可串行的调度?

答案:

否

2.设 $r_i(X)$与 $w_i(X)$分别表示事物 T_i 读、写数据单元 X，则一个并发调度可以抽象为读、写串。基于上述表示，画出优先图，并判断下面两个并发调度是否是可串行化的，为什么?

调度S_1：

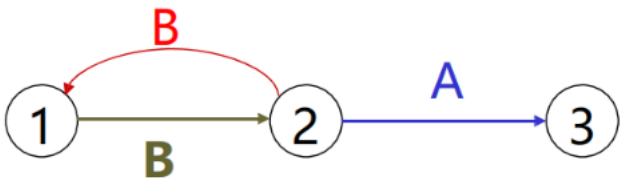
$$r_2(A);r_1(B);w_2(A);r_2(B);r_3(A);w_1(B);w_3(A);w_2(B)$$

调度S_2：

$$r_2(A);r_1(B);w_2(A);r_3(A);w_1(B);w_3(A);r_2(B);w_2(B)$$

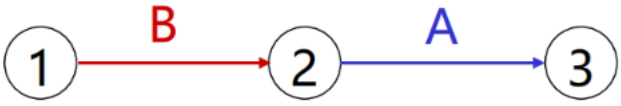
答案：

(1)根据S_1调度，可构造如下优先图：



由于上面优先图有环，所以该调度是不可串行化的。

(2) 根据S_2调度，可构造如下优先图：



由于上面优先图无环，所以该调度是可串行化的。

3. 假设我校淘乐果园水果店架设了一套数据库系统，同学在超市挑选好商品后，带商品到结算处结算付款，结算处有多名收银员使用多台机器进行结算。收银员负责扫同学购买水果的种类和数量，由系统后台结算程序计算出同学购买商品的总金额，修改商品表的水果库存量，并将销售信息写入销售表。请根据上述描述，回答以下问题。

假设有两位同学同时购买同一种类的水果，结算事务修改该水果的库存量（记为数据项）所产生的部分调度如下表所示：

$T1$	$T2$
$a \leftarrow Read(X)$	
	$a \leftarrow Read(X)$
$a = a - 1$	
$Write(X, a)$	
	$a = a - 2$
	$Write(X, a)$

3.1 如果数据项X购买前的初值为114514，则上述调度执行完成后，X的值是多少?这属于哪一类不一致性?

答案：

数据项X的值是114512，属于更新丢失引起的不一致性。

3.2 引入独占锁指令$lock()$和解锁指令$Unlock()$，对"问题 1"中存在并发调度进行重写，要求满足两段锁协议，且事务 T1、T2 首条指令的相对请求时间与"问题 1"中的相同。

$T1$	$T2$
$lock(X)$	
$a \leftarrow Read(X)$	
	$lock(X)$
$a = a - 1$	
$Write(X, a)$	
$Unlock(X)$	
	$a \leftarrow Read(X)$
	$a = a - 2$
	$Write(X, a)$
	$Unlock(X)$

4、已知某数据库采用即时更新方法（undo-redo 方法）记录 WAL 日志。设故障发生时 WAL 日志文件内容如下：

```
<T1, begin>
<T1, A, 114, 114514>
<T2, begin>
<T1, B, “hit”, “hitcs”>
<T1, commit>
<T3, begin>
<T3, B, “hitcs”, “hitcsdb”>
<T2, A, 114514, 1919810>
```

当系统重启后，DBMS 基于该 WAL 日志文件进行故障恢复。回答下列问题：

4.1　该数据库系统的的日志恢复策略为 Undo/Redo 型，那么其对应的缓冲区处理策略是什么（Steal？No Steal + Force？No Force）？该策略的每一项的具体内容都有什么？

答案：

策略：Steal + No force

具体内容：

Steal：允许在事务 commit 之前把内存中的数据写入磁盘。

No force：内存中的数据可以一直保留，在 commit 之后过一段时间再写入磁盘。

4.2 .当 DBMS 进行故障恢复时，需要对那些事务进行 undo？对那些事务进行 redo？给出具体理由

答案：

要对事务 T2 和 T3 进行 undo， 要对事务 T1 进行 redo。（2 分）

原因：在 undo-redo 方法中，已提交的事务可能有部分写操作尚未写盘，因此 T1 要 redo；未提交的事务 T2 和 T3 可能有部分写操作已经落盘，因此 T2 和 T3 要 undo。

4.3　当故障恢复完成时，对象 A 和 B 的值分别是什么？描述故障恢复的具体过程。

答案：

当恢复完成时，A = 114514，B = “hitcs”。（1 分）

恢复过程有 2 个阶段。

Redo 阶段：对日志文件从前向后进行扫描，对已提交事务 T1 进行 redo，A 的值被写为 114514，B 的值被写 为“hitcs”。（1 分）

Undo 阶段：对日志文件从后向前进行扫描，对未提交事务 T2 和 T3 进行 undo，A 的值被写为 114514，B 的值被写为“hitcs”。（1 分）

5. 当前有一个支持故障恢复技术的 DBMS，其采用了 STEAL 和 NO-FORCE 缓冲区策略，假设每当 DBMS 执行检查点时会将数据缓冲区里面所有的脏数据块写入相应的数据文件，保证数据库的一致性。

以下是数据库发生故障后的一份 WAL 日志 ，请阅读该日志并回答以下几个问题，

LSN	WAL Record
1	<T1 BEGIN>
2	<T1, X, 1, 2>
3	<T2 BEGIN>
4	<T2, Y, 1, 2>
5	<T1 COMMIT>
6	<T2, Y, 2, 3>
7	<T3 BEGIN>
8	<T3, Z, 1, 2>
9	<T2, X, 2, 3>
10	<CHECKPOINT>
11	<T2, Y, 3, 4>
12	<T3, Z, 2, 3>
13	<T3 COMMIT>
14	<T2, Z, 3, 4>

其中< T_i, X, A, B >分别代表着事务T_i，数据项X，更新前数据项的值A，更新后数据项的值B

5.1 请问 DBMS 需要进行恢复数据的时候，数据库的磁盘文件上数据项 X Y Z 的值分别是多少？

答案：

X=3，Y 和 Z 的数值无法确定

提示：检查点时刻 DBMS 会将所有内容写入到磁盘，但随后数据项 Y、Z 被检查点之后的事务修改。由于该 DBMS 使用了 STEAL 和 NO-FORCE 策略，DBMS 则可以在事务提交后批量地将数据落盘，任何脏页页可能在提交前写入磁盘，因此我们不知道崩溃时磁盘上数据项 Y 和 Z 的内容。

5.2 在对该 WAL 日志进行故障恢复的时候，事务T_1, T_2, T_3都应如何处理？请详述该过程与原因

答案：

T_1:无需处理 ，因为该事务在检查点之前就提交了，在检查点所有数据都是一致的

T_2:撤销（undo） 该事务的所有修改，因为该事务从未提交，所有的改动都需要还原

T_3:重做（redo） 该事务的所有修改，因为该事务在检查点后发生了提交，这意味着 DBMS 需要重做他的所有修改

5.3 假设恢复作业完成后，DBMS 会将所有的数据写入磁盘， DBMS 从 WAL 恢复数据库状态后数据库磁盘中数据项 X、Y、Z 的值分布是多少？

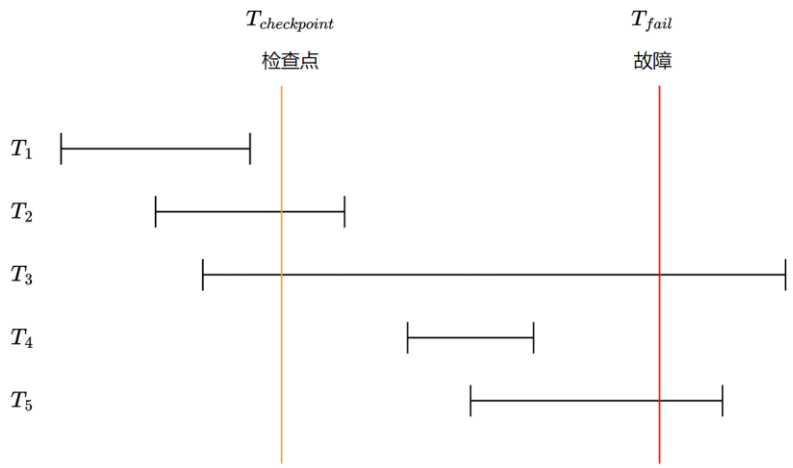
答案：

$X = 2$ （由事务T_1提交）

$Y = 1$ （由于事务T_2从未提交，需要回滚到之前的数值）

$Z = 3$ （因为事务T_3已提交，需要将其回滚到事务T_3修改后的值 ）

6. 使用检查点的数据库恢复系统将根据事务的状态和检查点的关系采取相对应的恢复策略，现在有事务T_1-T_5其执行过程如下图所示（线段左端和右端分别表示事务开始与提交），其执行过程中数据库系统发生如图所示的故障，请回答下列问题



6.1 请问在故障恢复时事务T_1-T_5那些需要撤销，那些需要重做，那些不需要操作？

不需要操作：T_1

重做：T_2 ，T_4

撤销：T_3 ，T_5

6.2 事务T_6-T_8 的日志文件如下图所示,<T_i，begin>表示事务T_i开始执行，<T_i，commit>表示事务T_i提交，<T_i，D，V_1，V_2>表示事务T_i 将数据项 D 的值由V_1修改为V_2，<crash>表示数据库发生故障

<T_6, begin>
<T_6, X, 100, 1>
<T_7, begin>
<T_7, X, 1,3>
<T_8, begin>
<T_7, Y, 50, 6>
<T_8, Y, 6, 8>
<T_8, Z, 10, 9>
<checkpoint>
<T_6,commit>
<T_8,Z,9,10>
<crash>

数据库系统发生故障时，请给出恢复子系统时需要 undo 的事务列表和需要 redo 的事务列表

答案：

因为检查点之后只有T_6提交了，所以事务T_6需要做 redo。

未提交的事务T_7、T_8需要做 undo。

6.3 请简述事务T_6-T_8 在系统故障后，基于检查点的故障恢复过程

答案：

(1)发生故障后，反向扫描日志文件，扫描至检查点，确认当前活动事务T_6，T_7，T_8；

(2)检查点之后事务T_6提交，因此要重做（REDO），具体是从事务T_6的开始标志起，正向扫描日志文件，重新执行T_6对数据库的所有操作，直到事务结束标志；

(3)检查点之后T_7、T_8未提交，因此要撤销（UNDO），具体是反向扫描日志文件，将T_7、T_8对数据库的所有更新操作执行逆操作，直到事务开始标志。