

关于几张试卷中多次出现的 SUX 锁并发问题的答案，以下面这题为例：

6. 假设有两项业务对应的事务 T1、T2 与存款关系有关：

转账业务——T1(A,B,s)，从账户 A 向账户 B 转 s 元；

计息业务——T2，对当前所有账户计算利息（即原金额为 X 元，计息后为 $X*1.1$ ）。

上述两个事务的一个并发调度如下表所示：

T1(A,B,s)	T2
Read(A)	
A:=A-s	Read(A)
Write(A)	A:=A*1.1
...	Write(A)
	...
Read(B)	Read(B)
B:=B+s	B:=B*1.1
Write(B)	Write(B)

(1) 若两个事务的操作遵循 (S,X) 锁，初始时 A=100, B=40, S=10，最终 A, B 结果如何？该并发调度正确与否，为什么？

	锁申请		
已拥有锁		S	X
	S	Y	N
	X	N	N

(2) 若两个事务的操作遵循 (S,U,X) 锁（相容矩阵如下），初始时 A=100, B=40, S=10，最终 A, B 结果如何？该并发调度正确与否？为什么？

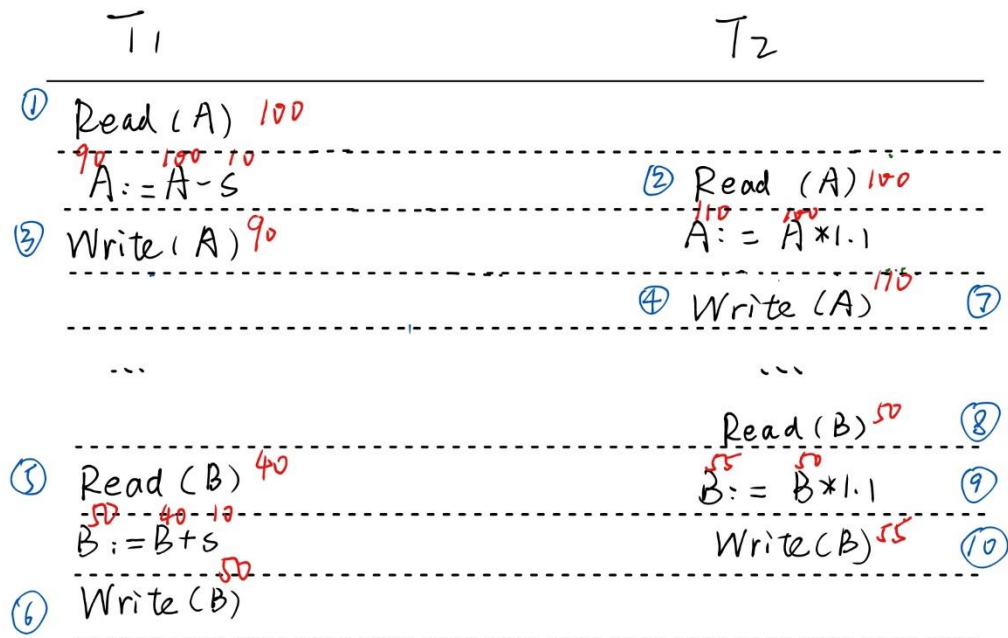
	锁请求			
已拥有锁		S	U	X
	S	Y	Y	N
	U	Y	Y	Y
	X	N	N	N

解答：

(1) 略

(2) 首先明确:

- 1) 本题引入了 U 锁, 那么 **Read-内存里改数据-write** 整个过程就被定义成了 update 操作
- 2) !!!!!!! 注意!!!!!!! 本题中 SUX 相容矩阵和书上不一样, 按照题目里给的来, 一定注意仔细看!



- ① T_1 申请 A 的 U 锁, 批准
- ② T_2 申请 A 的 U 锁, 此时 A 上有 T_1 的 U 锁, 根据相容矩阵, 批准, 读的是 A 的原数据
- ③ T_1 申请 A 升级到 X 锁, 此时 A 上有 T_2 的 U 锁, 根据相容矩阵, 批准 (这里易错!! 根据相容矩阵来!!) 写入 A, 为 90, 根据 2PL 原则, 此时 T_1 事务还没有结束, 不释放锁, 因此仍然持有 A 的 X 锁
- ④ T_2 申请升级到 X 锁, 此时 A 上有 T_1 的 X 锁, 根据相容矩阵, 不批准, T_2 卡在这里
- ⑤ T_1 申请 B 的 U 锁, 批准
- ⑥ T_1 申请 B 升级到 X 锁, 允许, 写入 B 为 50, 根据 2PL 原则, T_1 事务结束, 释放所有锁
- ⑦ T_2 可以获得 A 的 X 锁了, 写入 A 为 110
- ⑧ T_2 申请 B 的 U 锁, 批准, 这时读入的是 T_1 修改过的数据 50
- ⑨ 读入内存修改
- ⑩ 写入 B, 55