Student name:

Student ID: 19520113

Class:

(Đáp án chỉ mang tính chất tham khảo)

HK1: 2019 – 2020

Question 1:

1.a.

- Considering the data unit A: Pair(1, 13) $\Longrightarrow$ T1 → T2.

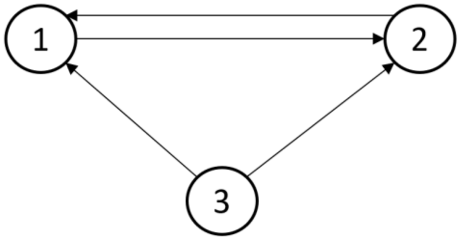
- Considering the data unit B: Pair(5, 9) $\Longrightarrow$ T3 → T2.

- Considering the data unit C:

+ Pair(3, 11) $\Longrightarrow$ T2 → T1.

+ Pair(3, 15) $\Longrightarrow$ T3 → T1.

+ Pair(11, 15) $\Longrightarrow$ T3 → T2.



The wait graph has a cycle, so schedule S has a deadlock.

1.c.

	T1 TS = 20	**T2** TS = 10	**T3** TS = 30	**A** RT = 0 WT = 0	**B** RT = 0 WT = 0	**C** RT = 0 WT = 0	**D** RT = 0 WT = 0	
1		R(A)		RT = 10 WT = 0				WT(A) <= TS(T2) → T2 read A
2	R(C)					RT = 20 WT = 0		WT(C) <= TS(T1) → T1 read C
3		W(B)			RT= 0 WT = 10			RT(B) <= TS(T2), WT(B) <= TS(T2) → T2 write B
4	R(D)						RT= 20 WT = 0	WT(D) <= TS(T2) → T1 read D
5			R(B)		RT= 30 WT = 10			WT(B) <= TS(T3) → T3 read B
6		W(C)						RT(C) > TS(T2) → Roolback(T2)
7								

Re-initialize T2 so that TS(T2) > TS(T3). Assume TS(T2) = 40.

	T1 TS = 20	**T2** TS = 40	**T3** TS = 30	**A** RT = 0 WT = 0	**B** RT = 0 WT = 0	**C** RT = 0 WT = 0	**D** RT = 0 WT = 0	
1	R(C)					RT = 20 WT = 0		WT(C) <= TS(T1) → T1 read C
2	R(D)						RT= 20 WT = 0	WT(D) <= TS(T2) → T1 read D
3			R(B)		RT= 30 WT = 0			WT(B) <= TS(T3) → T3 read B
4	W(A)			RT = 0 WT = 20				RT(A) <= TS(T1), WT(A) <= TS(T1) → T1 write A
5			W(C)			RT = 20 WT = 30		RT(C) <= TS(T3), WT(C) <= TS(T3) → T3 write C
6		R(A)		RT = 40 WT = 20				WT(A) <= TS(T2) → T2 read A
7		W(B)			RT= 30 WT = 40			RT(B) <= TS(T2), WT(B) <= TS(T2) → T2 write B
8		W(C)				RT= 20 WT = 40		RT(C) <= TS(T2), WT(C) <= TS(T2) → T2 write C

⟹ Schedule S is serializable in the order T1, T3, T2.

Question 2:

<T4, B, 8>

T4 is incompleted → Recover B = 80.

~~<COMMIT T2>~~

~~T2 is completed → do nothing~~

<T4, F, 70>

T4 is incompleted → Recover F = 70.

<T3, D, 40>

T3 is incompleted → Recover D = 40.

<T3, B, 20>

T3 is incompleted → Recover B = 20.

Reach <START CKPT>

Stop

HK1 18-19

Question 1:

- Considering the data unit A: Pair(1, 8) $\Longrightarrow$ T2 → T1.

- Considering the data unit B:

+ Pair(3, 4) $\Longrightarrow$ T4 → T1.

+ Pair(3, 6) $\Longrightarrow$ T3 → T1.

+ Pair(4, 6) $\Longrightarrow$ T3 → T4.

- Considering the data unit C:

+ Pair(2, 7) ⟹ T1 → T2.

+ Pair(2, 9) ⟹ T3 → T2

+ Pair(7, 9) ⟹ T3 → T1

- Considering the data unit D: + Pair(5, 10) ⟹ T3 → T4

HK2 18-19

Question 1:

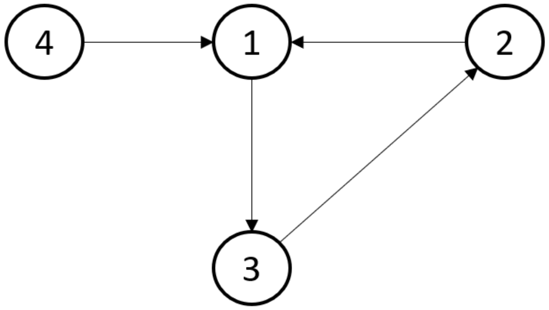
1.a.

- Considering the data unit A: + Pair(1, 8) ⟹ T4 → T1

- Considering the data unit B: + Pair(2, 6) ⟹ T3 → T2

- Considering the data unit C: + Pair(3, 7) ⟹ T2 → T1

- Considering the data unit D: + Pair(4, 9) ⟹ T1 → T3



The wait graph has a cycle, so schedule S has a deadlock.

1.c.

	T1 TS = 10	**T2** TS = 20	**T3** TS = 30	**T4** TS = 40	**A** RT = 0 WT = 0	**B** RT = 0 WT = 0	**C** RT = 0 WT = 0	**D** RT = 0 WT = 0	**E** RT = 0 WT = 0	
1	R(A)				RT = 10 WT = 0					WT(A) <= TS(T1) → T1 read A
2		R(B)				RT = 20 WT = 0				WT(B) <= TS(T2) → T2 read B

3	W(C)						RT = 0
							WT = 10
										RT(C) <= TS(T1)
										WT(C) <= TS(T1)
										→ T1 write C

4			R(D)					RT = 30 WT = 0		WT(D) <= TS(T3) → T3 read D
5				R(E)					RT = 40	WT(E) <= TS(T4)

									WT = 0	→ T4 read E

6			W(B)			RT = 20
						WT = 30
										RT(B) <= TS(T3)
										WT(B) <= TS(T3)
										→ T3 write B

7		W(C)					RT = 0
							WT = 20
										RT(C) <= TS(T2)
										WT(C) <= TS(T2)
										→ T2 write C

8				W(A)	RT = 10
					WT = 40
										RT(A) <= TS(T4)
										WT(A) <= TS(T4)
										→ T4 write A

9	W(D)									RT(D) > TS(T1) → Roolback(T1)

Re-initialize T1 so that TS(T1) > TS(T4). Assume TS(T1) = 50.

	T1 TS = **5**0	T2 TS = 20	T3 TS = 30	T4 TS = 40	A RT = 0 WT = 0	B RT = 0 WT = 0	C RT = 0 WT = 0	D RT = 0 WT = 0	E RT = 0 WT = 0	
1		R(B)				RT = 20 WT = 0				WT(B) <= TS(T2) → T2 read B
2			R(D)					RT = 30 WT = 0		WT(D) <= TS(T3) → T3 read D
3				R(E)					RT = 40 WT = 0	WT(E) <= TS(T4) → T4 read E

4			W(B)			RT = 20
						WT = 30
										RT(B) <= TS(T3)
										WT(B) <= TS(T3)
										→ T3 write B

5		W(C)					RT = 0
							WT = 20
										RT(C) <= TS(T2)
										WT(C) <= TS(T2)
										→ T2 write C

6				W(A)	RT = 0
					WT = 40
										RT(A) <= TS(T4)
										WT(A) <= TS(T4)
										→ T4 write A

7	R(A)				RT = 50 WT = 40					WT(A) <= TS(T1) → T1 read 4

8	W(C)						RT = 0
							WT = 50
										RT(C) <= TS(T1)
										WT(C) <= TS(T1)
										→ T1 write C

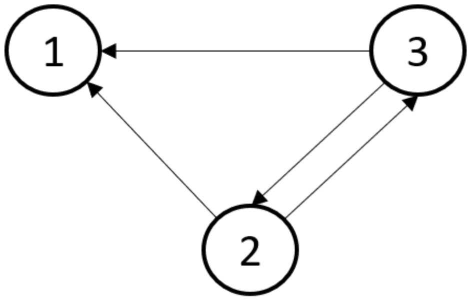
9	W(D)							RT = 30
								WT = 50
										RT(D) <= TS(T1)
										WT(D) <= TS(T1)
										→ T1 write D

⟹ Schedule S is serializable in the order T2, T3, T4, T1.

HK2 2014-2015 (Đề 2)

Question 1: checked

1.a. Precedence graph for schedule S:



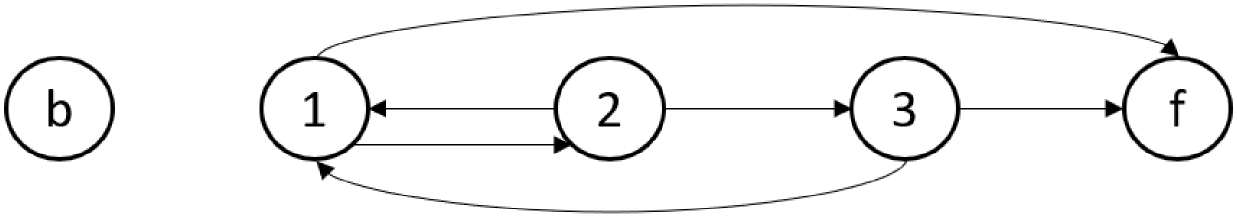
The precedence graph for schedule S contains a cycle.

→ Schedule S is not conflict-serializable.

1.b.

Tb	**T1**	**T2**	**T3**	**Tf**
W(A)				
W(B)				
		W(A)		
	R(A)			
			W(B)	
		W(B)		
			W(B)	
	W(A)			
			R(B)	
	R(B)			
				R(A)
				R(B)

Poly graph for schedule S:



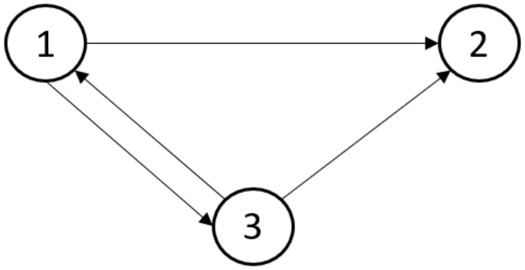
The poly graph for schedule S contains a cycle.

→ Schedule S is not view-serializable.

HK2 2014-2015 (Đề 1)

Question 1:

a.



b.

CK2 2014-2015

Question 1:

- Considering the data unit A:

+ Pair(4, 9) ⟹ T1 → T2

+ Pair(4, 12) ⟹ T1 → T4

+ Pair(7, 9) ⟹ T3 → T2

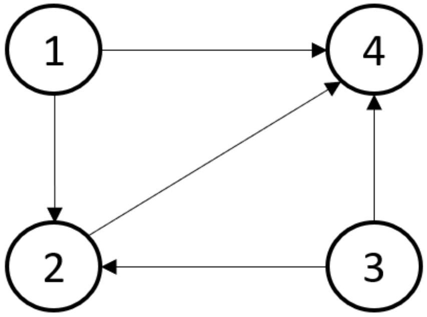
+ Pair(7, 12) ⟹ T3 → T4

+ Pair(11, 12) ⟹ T2 → T4

- Considering the data unit B:

+ Pair(5, 6) ⟹ T1 → T2

+ Pair(5, 8) ⟹ T1 → T4



The graph for S1 has no cycles, so schedule S is serializable.

Sequential schedules equivalent to S1 is T1 → T2 →T3 → T4 (COI LẠI)

Question 2:

2.a.

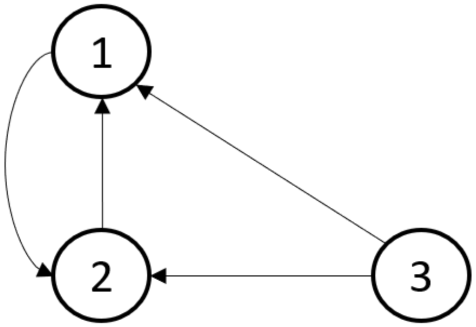
- Considering the data unit A: Pair(1, 7) ⟹ T2 → T1.

- Considering the data unit B: Pair(3, 5) ⟹ T3 → T1.

- Considering the data unit C:

+ Pair(2, 6) ⟹ T1 → T2

+ Pair(2, 8) ⟹ T3 → T2.



The wait graph has a cycle, so schedule S has a deadlock.

2.a.c

	T1 TS = 10	**T2** TS = 20	**T3** TS = 30	**A** RT = 0 WT = 0	**B** RT = 0 WT = 0	**C** RT = 0 WT = 0	**D** RT = 0 WT = 0	
1	R(A)			RT = 10 WT = 0				WT(A) <= TS(T1) → T1 read A
2		R(C)				RT = 20 WT = 0		WT(C) <= TS(T2) → T2 read C
3	W(B)				RT = 0 WT = 10			RC(B) <= TS(T1) WT(B) <= TS(T1) → T1 write B
4		R(D)					RT = 20 WT = 0	WT(D) <= TS(T2) → T2 read D
5			R(B)		RT = 30 WT = 10			WT(B) <= TS(T3) → T3 read B
6	W(C)							RC(C) > TS(T1) → Rollback T1
7								

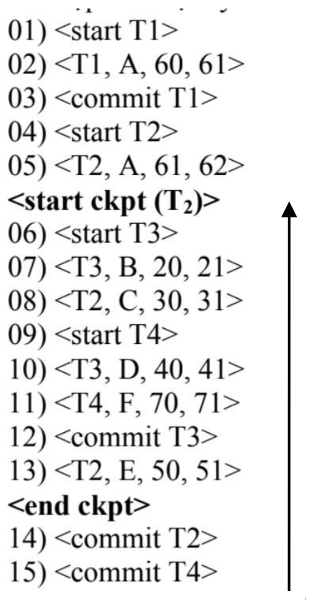
Re-initialize T1 so that TS(T1) > TS(T3). Assume TS(T1) = 40.

	T1 TS = 40	**T2** TS = 20	**T3** TS = 30	**A** RT = 0 WT = 0	**B** RT = 0 WT = 0	**C** RT = 0 WT = 0	**D** RT = 0 WT = 0	
1		R(C)				RT = 20 WT = 0		WT(C) <= TS(T2) → T2 read C
2		R(D)					RT = 20 WT = 0	WT(D) <= TS(T2) → T2 read D
3			R(B)		RT = 30 WT = 10			WT(B) <= TS(T3) → T3 read B

4		W(A)		RT = 0 WT = 20				RT(A) <= TS(T2) WT(A) <= TS(T2) → T2 write A
5			W(C)			RT = 20 WT = 30		RT(C) <= TS(T3) WT(C) <= TS(T3) → T3 write C
6	R(A)			RT = 40 WT = 20				WT(A) <= TS(T1) → T1 read A
7	W(B)				RT = 30 WT = 40			RT(B) <= TS(T1) WT(B) <= TS(T1) → T1 write B
8	W(C)					RT = 20 WT = 40		RT(C) <= TS(T1) WT(C) <= TS(T1) → T1 write C

⟹ Schedule S is serializable in the order T2, T3, T1.

Question 3:



```
01) <start T1>
02) <T1, A, 60, 61>
03) <commit T1>
04) <start T2>
05) <T2, A, 61, 62>
<start ckpt (T2)>
06) <start T3>
07) <T3, B, 20, 21>
08) <T2, C, 30, 31>
09) <start T4>
10) <T3, D, 40, 41>
11) <T4, F, 70, 71>
12) <commit T3>
13) <T2, E, 50, 51>
<end ckpt>
14) <commit T2>
15) <commit T4>
```

scan

Found <end ckpt>

Do nothing with T1

<T2, E, 50, 51>

T2 is completed → Recover E = 51

<T4, F, 70, 71>

T4 is completed → Recover F = 71

<T3, D, 40, 41>

T3 is completed → Recover D = 41

<T2, C, 30, 31>

T2 is completed → Recover C = 31

<T3, B, 20, 21>

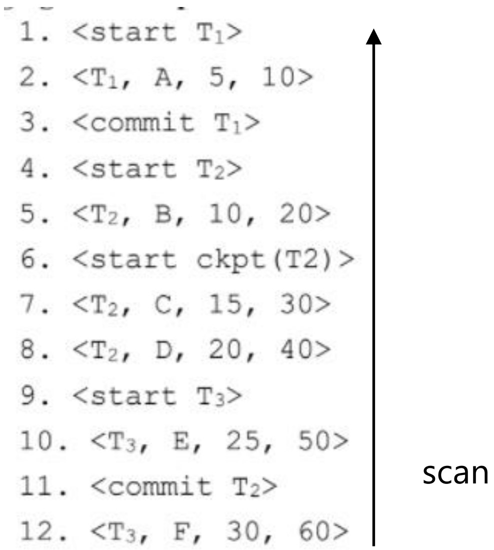
T3 is completed → Recover B = 21

Reach <start ckpt>

Stop.

CK HK2 2020-2021 (Đề 1)

Question 2: (COI LẠI)



```
1. <start T1>
2. <T1, A, 5, 10>
3. <commit T1>
4. <start T2>
5. <T2, B, 10, 20>
6. <start ckpt(T2)>
7. <T2, C, 15, 30>
8. <T2, D, 20, 40>
9. <start T3>
10. <T3, E, 25, 50>
11. <commit T2>
12. <T3, F, 30, 60>
```

The <END CKPT> is writable when T2 committed.

<T3, F, 30, 60>

T3 is incompleted → recover F = 30

<T3, E, 25, 50>

Recover E = 25

<T2, D, 20, 40>

T2 is completed → Recover D= 40

<T2, C, 15, 30>

Recover C = 30.

<T2, B, 10, 20>

Recover B = 20.

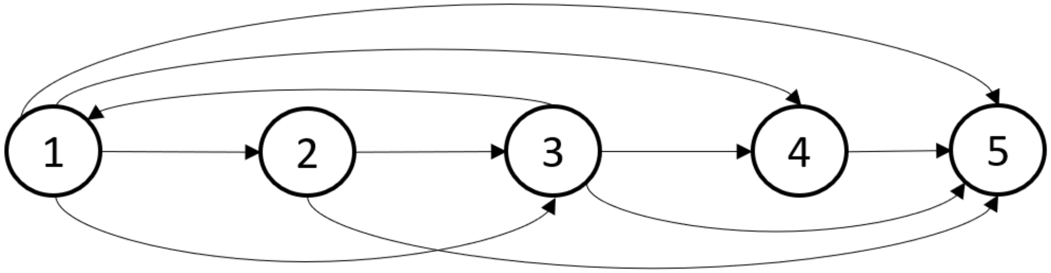
<T1, A, 5, 10>

T1 is completed → Recover A= 10

GK2 2018 – 2019

Question 1: checked

1.a. Precedence graph for schedule S:



The precedence graph for schedule S contains cycles.

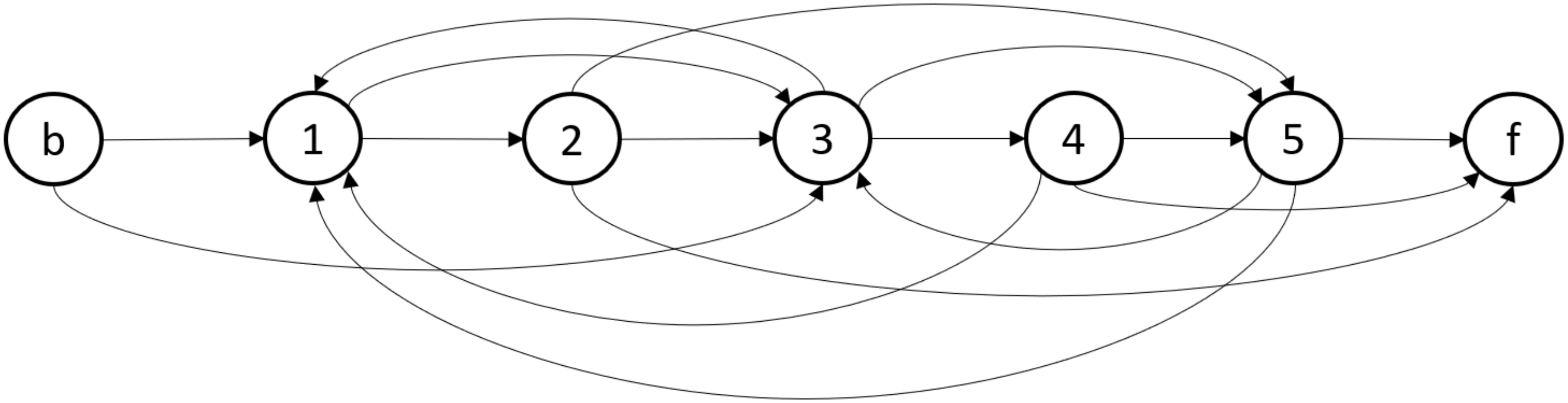
→ Schedule S is not conflict-serializable.

1.b.

Tb	T1	T2	T3	T4	T5	Tf
W(A)						
W(B)						
W(C)						
W(E)						
	R(A)					
			R(B)			
	W(B)					
		R(B)				
			W(B)			
				R(B)		
		W(C)				
					R(C)	
				W(E)		
					R(E)	
					W(B)	

						R(A)
						R(B)
						R(C)
						R(E)

Poly graph for schedule S:



The poly graph for schedule S contains cycles.

→ Schedule S is not view-serializable.

Question 2:

a.

```
CREATE OR REPLACE TRIGGER TRG_CAUA
BEFORE INSERT ON ENROLLMENT
FOR EACH ROW
DECLARE
    V_NUM_CLASS NUMBER;
BEGIN
    SELECT COUNT(CLASSID) INTO V_NUM_CLASS
    FROM ENROLLMENT
    WHERE STUDENTID = :NEW.STUDENTID;

    IF(V_NUM_CLASS >=3) THEN
        RAISE_APPLICATION_ERROR(-20122, 'Error: Moi SV khong dang ky qua 3 lop');
    END IF;
END;
```

b.

```
CREATE OR REPLACE PROCEDURE CAUB(V_COURSENO COURSE.COURSENO%TYPE)
AS
    V_COURSE_NAME COURSE.DESCRIPTION%TYPE;
    CURSOR C_CLASS
    IS
        SELECT *
        FROM CLASS
        WHERE COURSENO = V_COURSENO;
    V_NUM_STUDENT NUMBER;
BEGIN
    SELECT DESCRIPTION INTO V_COURSE_NAME
    FROM COURSE
    WHERE COURSENO = V_COURSENO;

    DBMS_OUTPUT.PUT_LINE('**Mon hoc: ' || V_COURSE_NAME || '(MaMH: ' || V_COURSENO ||
')');

    FOR R_CLASS IN C_CLASS
    LOOP
        SELECT COUNT(*) INTO V_NUM_STUDENT
        FROM ENROLLMENT
        WHERE CLASSID = R_CLASS.CLASSID;
        DBMS_OUTPUT.PUT_LINE('---Lop: ' || R_CLASS.CLASSID || ' co so luong sinh vien dang ky la: ' || V_NUM_STUDENT);
    END LOOP;


    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20010, 'Error: Ma mon hoc khong hop le');
END;
```

c.

```
CREATE OR REPLACE FUNCTION Total_cost_for_student(V_STUID STUDENT.STUDENTID%TYPE) RETURN NUMBER
AS
    V_TOTAL_COST NUMBER;
BEGIN
    SELECT SUM(COST) INTO V_TOTAL_COST
    FROM COURSE CO, CLASS C, ENROLLMENT E
    WHERE CO.COURSENO = C.COURSENO
    AND E.CLASSID = C.CLASSID
    AND E.STUDENTID = V_STUID;
    RETURN V_TOTAL_COST;
    EXCEPTION

        WHEN NO_DATA_FOUND THEN

            RETURN NULL;

END;
```