1. PARALLEL QUERY PROCESSING

1. d

2. $40 \times 4 \times 1KB = 160KB$

3. a

2. DISTRIBUTED TRANSACTION

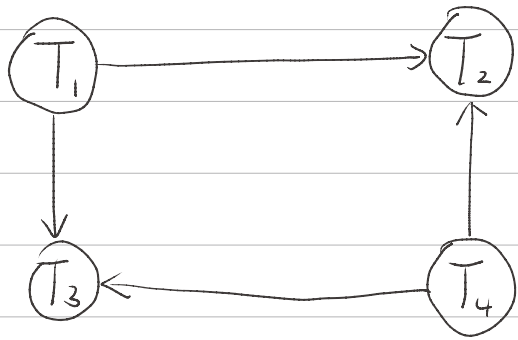
1. c

2. ad

3. b

3. CONCURRENCY CONTROL

1. a)



b) Not serial. There're interleaves between the transactions.

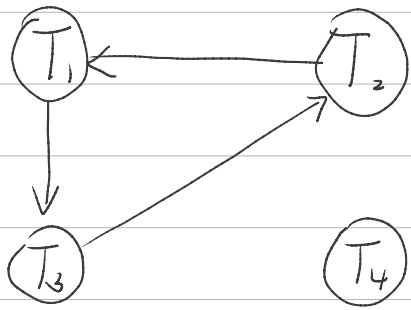
Conflict serializable, since dependency graph is acyclic.

Not possible. At t_6. T_3 needs a S(A). but T_1 holds a X(A). At t_8. T_2 needs a S(D), but T_4 holds a X(D).

2.a)

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
LM	g	g	g	b	b	g	b	b

b)



Yes, there's a deadlock, since the waits-for graph is not acyclic

c)

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
LM	g	g	g	a	a	g	g	-

d)

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
LM	g	g	g	b	b	g	g	-

4. RECOVERY

1. No. Because P5 is in the DPT, but the corresponding recLSN < 60.

   Yes. Because P2 is not in the DPT.

2. Transaction Table

Transaction	lastLSN	Status
T1	90	Running
T3	30	Running
T4	180	Aborting
T5	160	Running

Dirty Page Table

PageID	recLSN
P5	50
P1	40
P3	90
P2	160

3. 40. the smallest recLSN in DPT

40. Redo
50. Redo
60. Redo
70. recLSN > LSN
80. Only redo update and CLR
90. Redo
100. Only redo update and CLR
110. Redo
120. Only redo update and CLR
130. Redo
140. Only redo update and CLR
150. Only redo update and CLR
160 Redo
180 Redo

40, 50, 60, 90, 110, 130, 160, 180 should redo