Question 1:

1. BH->E holds.
   We prove it by computing the transitive closure of BH:
   With H->AC, we derive BH->BHAC
   With AB->DE, we derive BH->BHACDE,
   Hence, BH->E holds.

2. T = Ø
   Find a super key X.
   Let X := {B, E, G, H, I}
   X+ = {A, B, C, D, E, G, H, I, J}
   Try to remove B, {E, G, H, I}+ = {A, C, E, G, H, I }
   Thus, B cannot be removed.
   Try to remove E, {B, G, H, I}+ = {A, B, C, D, E, G, H, I, J}
   Thus, X:= { B, G, H, I }
   Try to remove G, {B, H, I}+ = {A, B, C, D, E, G, H, I, J}
   Thus, X:= { B, H, I }
   Try to remove H, {B, I}+ = {B, I}
   Thus, H cannot be removed.
   Try to remove I, {B, H}+ = {A, B, C, D, E, G, H, J}
   Thus, I cannot be removed.

   So {B, H, I} is a candidate key and add to T.

   Find another super key X.
   Let X := {B, C, G, I, J}
   X+ = {A, B, C, D, E, G, H, I, J}
   Try to remove B, {C, G, I, J}+ = {A, C, E, G, H, I, J}
   Thus, B cannot be removed.
   Try to remove C, {B, G, I, J}+ = {B, G, I, J}
   Thus, C cannot be removed.
   Try to remove G, {B, C, I, J}+ = {A, B, C, D, E, G, H, I, J}
   Thus, X:= {B, C, I, J}
   Try to remove I, { B, C, J }+ = {B, C, G, J}
   Thus, I cannot be removed.
   Try to remove J, {B, C, I}+ = {A, B, C, D, E, G, H, J}
   Thus, X:= {B, C, I}

   So {B, C, I} is a candidate key and add to T.

   Cannot find other super keys that do not contain any candidate key in T.
   So candidate keys are {B,H,I}, {B,C,I}.

3. 1NF since it only contains atomic attribute values.
   Not in 2NF since CI->A violates 2NF, the non-prime attribute A is partially dependent on key {B,C,I}.

4. Step 1. Reduce right, F is converted to
   F1={AB->D,AB->E,BC->G,CDE->H,CDE->J,H->A,H->C,CI->E,CI->G,CI->H,DHI->E,DHI->J}
   Step 2. Reduce Left. F1 is converted to (DHI->E is reduced to HI->E)
   F2={AB->D,AB->E,BC->G,CDE->H,CDE->J,H->A,H->C,CI->E,CI->G,CI->H,HI->E,DHI->J}
   Step3. Remove redundancy, F2 is converted to (remove redundancy: CI->E, DHI->J)

   Thus, F_{min} = {AB->D,AB->E,BC->G,CDE->H,CDE->J,H->A,H->C,CI->G,CI->H,HI->E}
   Or F_{min} = {AB -> DE, BC ->G, CDE -> HJ, H -> AC, CI ->GH, HI-> E}

5. Not lossless since there's no row contains "a" entirely.

	A	B	C	D	E	G	H	I	J
R1	a	a	b	b	a	b	a	b	b
R2	b	b	a	a	b	b	a	a	a
R3	b	a	b	b	b	a	a	b	a
	A	B	C	D	E	G	H	I	J
R1	a	a	a	b	a	a	a	b	b
R2	a	b	a	a	b	b	a	a	a
R3	a	a	a	b	a	a	a	b	a

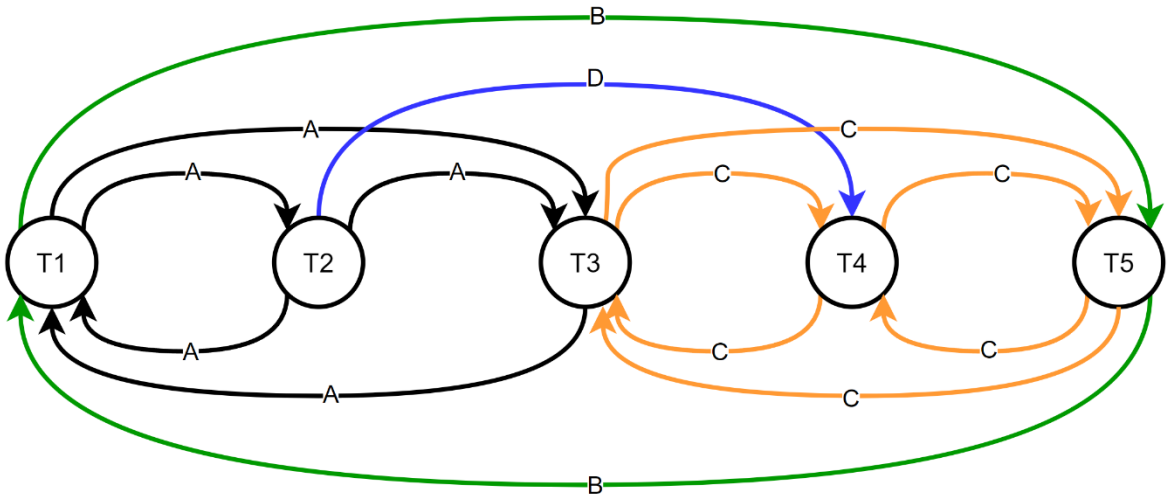
6. R violates BCNF because of AB->DE, decompose R into R1(A,B,D,E), R2(A,B,C,G,H,I,J).
   R2 violates BCNF because of BC->G, decompose R2 into R21(B,C,G), R22(A,B,C,H,I,J).
   R22 violates BCNF because of H->AC, decompose R22 into R221(H,A,C), R222(B,H,I,J).
   R222 violates BCNF because of BH->J, since BH-> BAC -> DEC->J, decompose R222 into R2221(B,H,J), R2222(B,H,I).

   R can be decomposed into R1(A,B,D,E), R21(B,C,G), R221(H,A,C), R2221(B,H,J) and R2222(B,H,I).

Question 2:

1. T1, T5: Redo
   T3, T4: Undo
   T2: Do Nothing

2. The transaction schedule is not conflict serializable because its precedence graphs is not acyclic



3. T1 waits T5 for B, T5 waits for T3 for C, T3 waits T1 for A.

T1	WL(A)	R(A)	...				WL(B)	R(B)	W(B)	W(A)	...			UL(A)	UL(B)				
T2	...																		
T3			WL(C)	R(C)	...				WL(A)	R(A)	W(C)	W(A)	...			UL(C)	UL(A)		
T4	...																		
T5					WL(B)	W(B)	...				WL(C)	R(C)	W(C)	...				UL(B)	UL(C)

Question 3:

1.

LRU													
QUERY	Initinal	1	3	3	2	3	4	5	2	1	5	2	6
Buffer 1	N	1	1	1	1	1	4	4	4	1	1	1	6
Buffer 2	N	N	3	3	3	3	3	3	2	2	2	2	2
Buffer 3	N	N	N	N	2	2	2	5	5	5	5	5	5
Page Fault:		F	F		F		F	F	F	F			F
Hit:				H		H					H	H	
Number of Page Fault:		8											

2.

MRU													
QUERY	Initinal	1	3	3	2	3	4	5	2	1	5	2	6
Buffer 1	N	1	1	1	1	1	1	1	1	1	1	1	1
Buffer 2	N	N	3	3	3	3	4	5	5	5	5	5	5
Buffer 3	N	N	N	N	2	2	2	2	2	2	2	2	6
Page Fault:		F	F		F		F	F					F
Hit:				H		H			H	H	H	H	
Number of Page Fault:		6											

3.

FIFO													
QUERY	Initinal	1	3	3	2	3	4	5	2	1	5	2	6
Buffer 1	N	1	1	1	1	1	4	4	4	4	4	2	2
Buffer 2	N	N	3	3	3	3	3	5	5	5	5	5	6
Buffer 3	N	N	N	N	2	2	2	2	2	1	1	1	1
Page Fault:		F	F		F		F	F		F		F	F
Hit:				H		H			H		H		
Number of Page Fault:		8											

4. MRU performs the best in the given query because it has the least number of page faults.