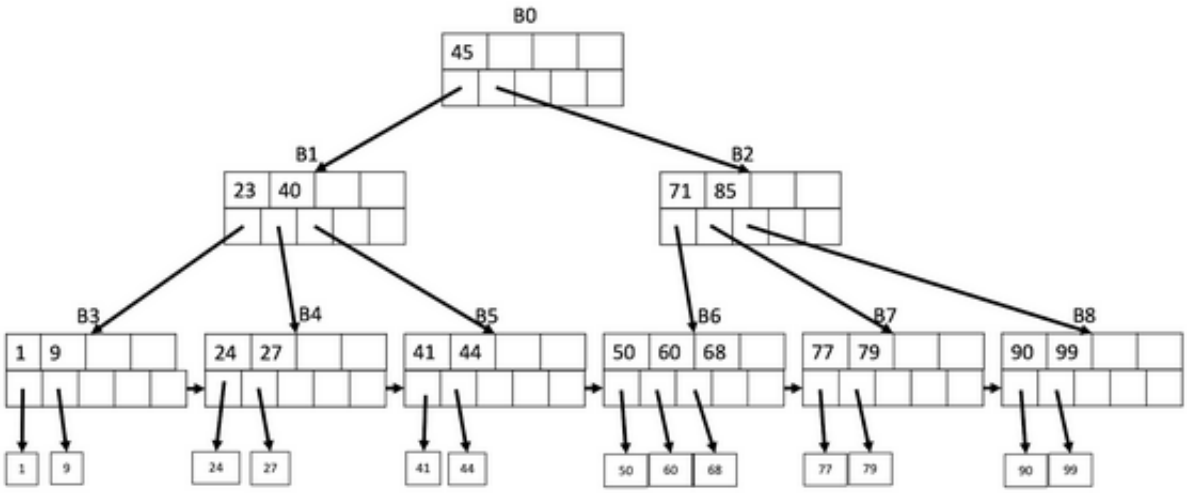
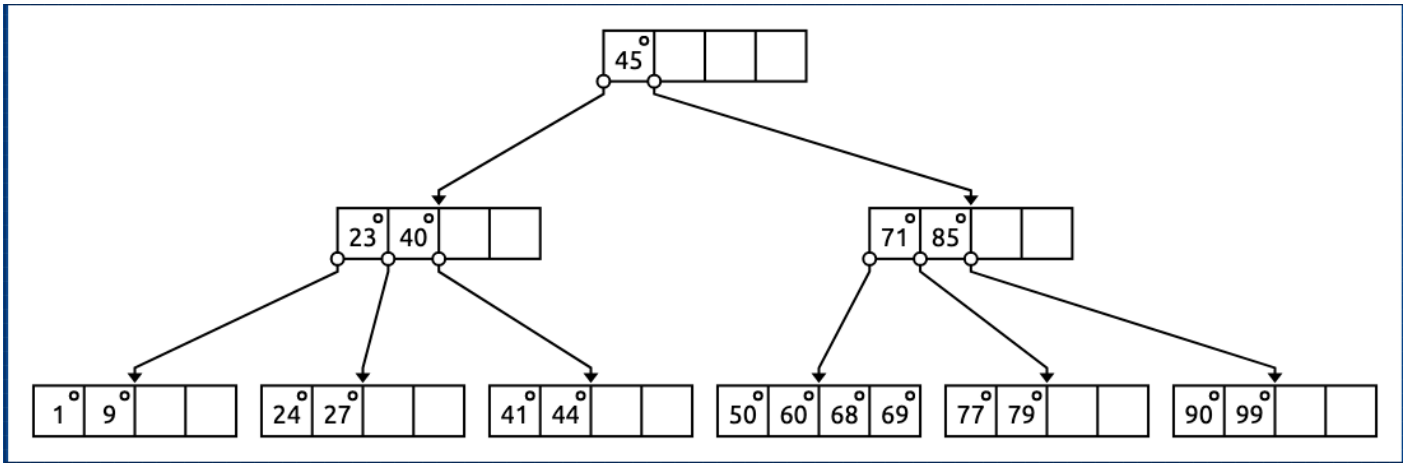
Homework 5

Satwik Singh satwiks2

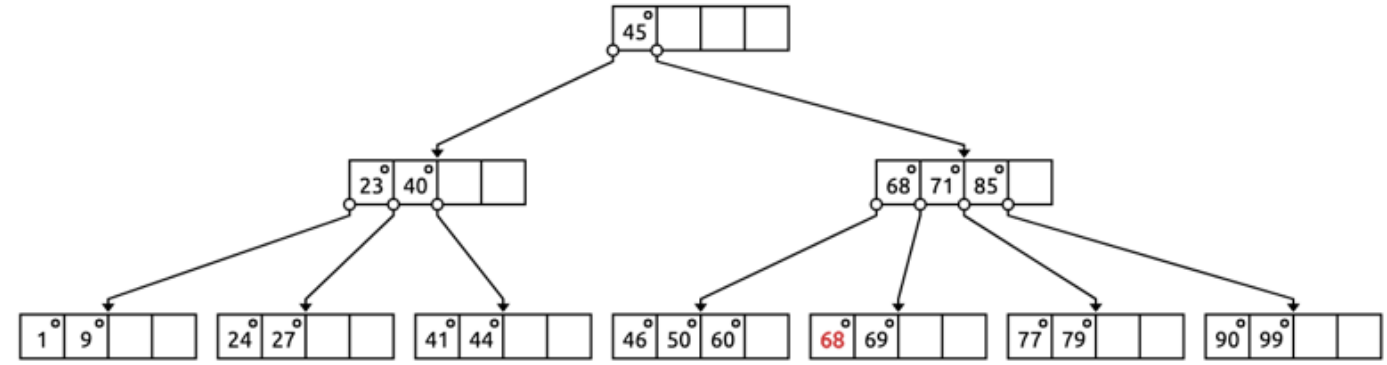
Problem 1.



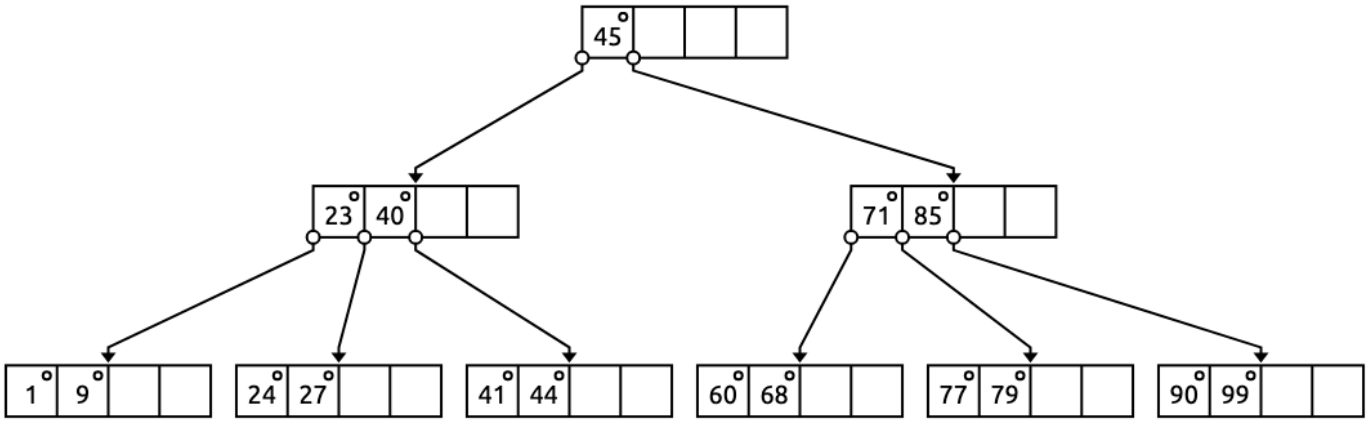
1. 24: B0, B1, B4
2. 78: B0, B2, B7
3. [26,63]: B0, B1, B4, B5, B6
4. Insert 69, 46
    1. 69



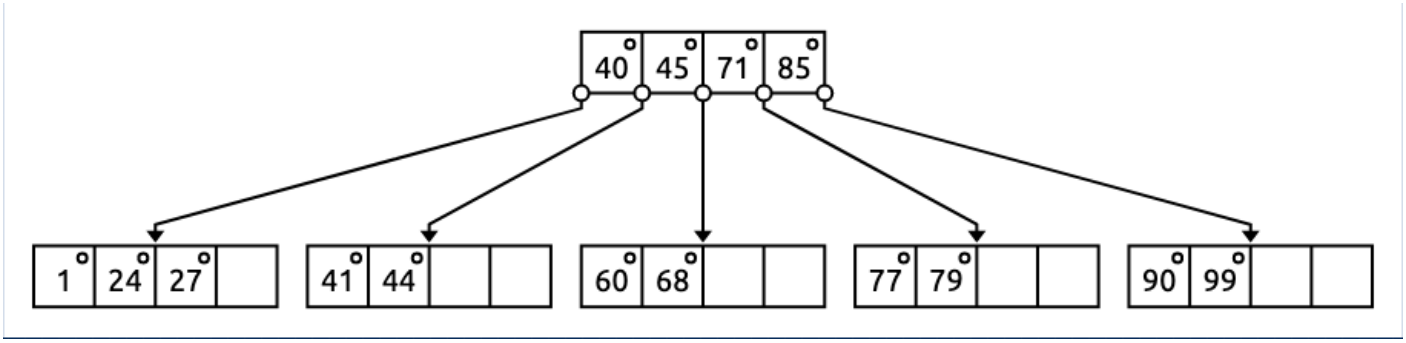
2. 46



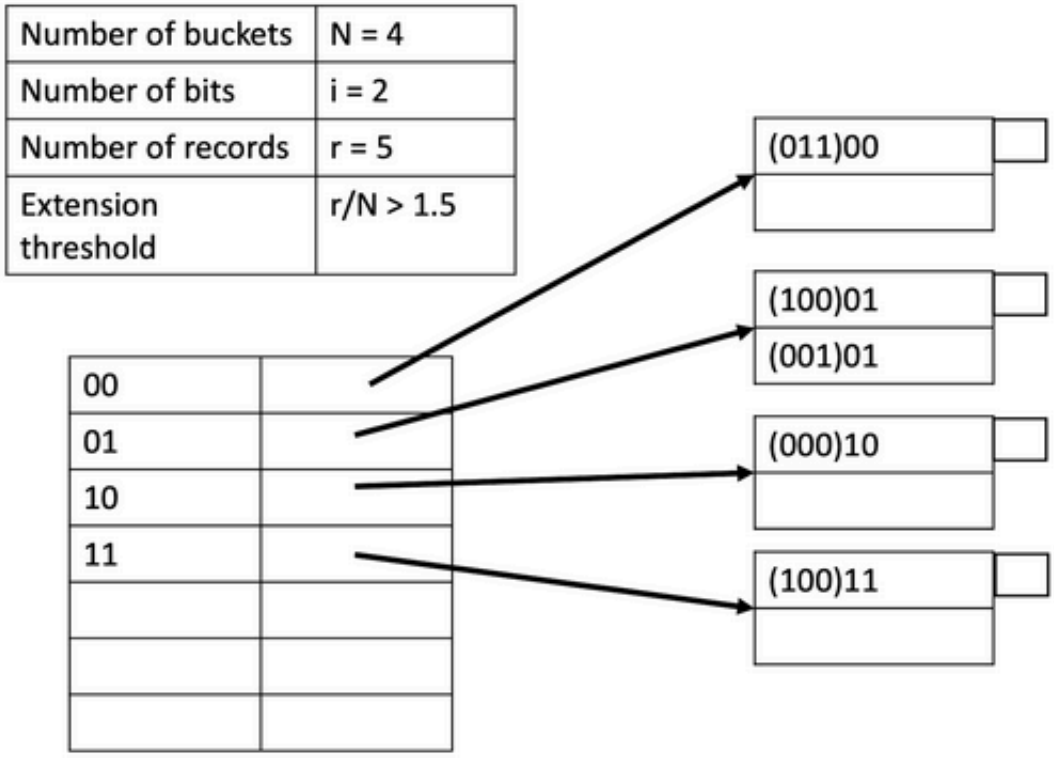
5. Delete 50, 9
    1. 50



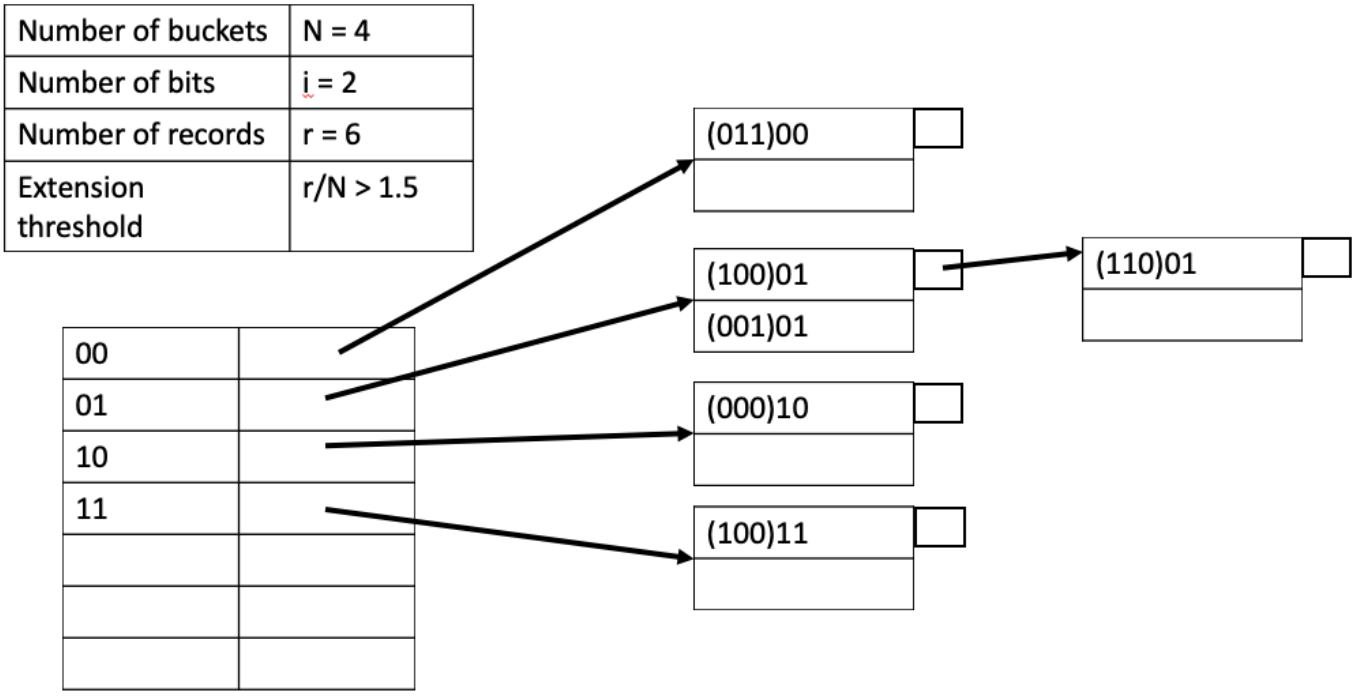
    2. 9



Problem 2.

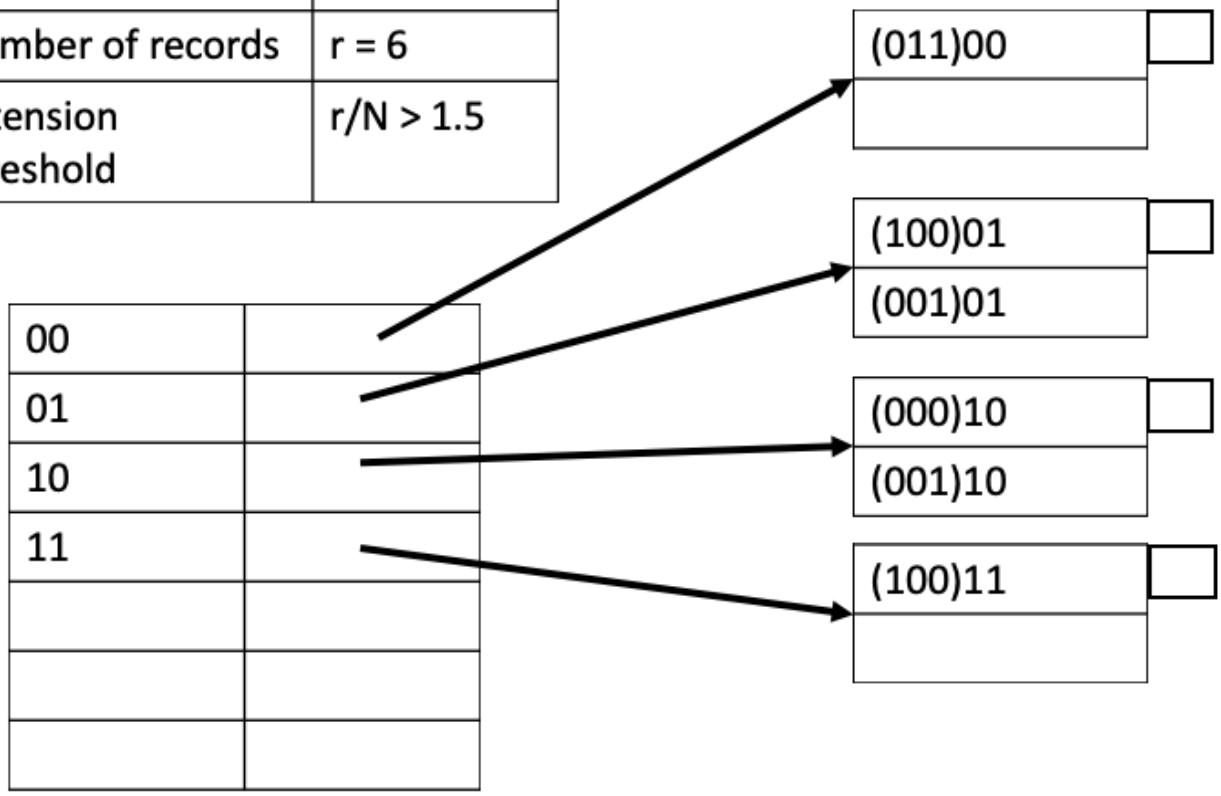


1. r/N = 1.25
2. Insert 11001



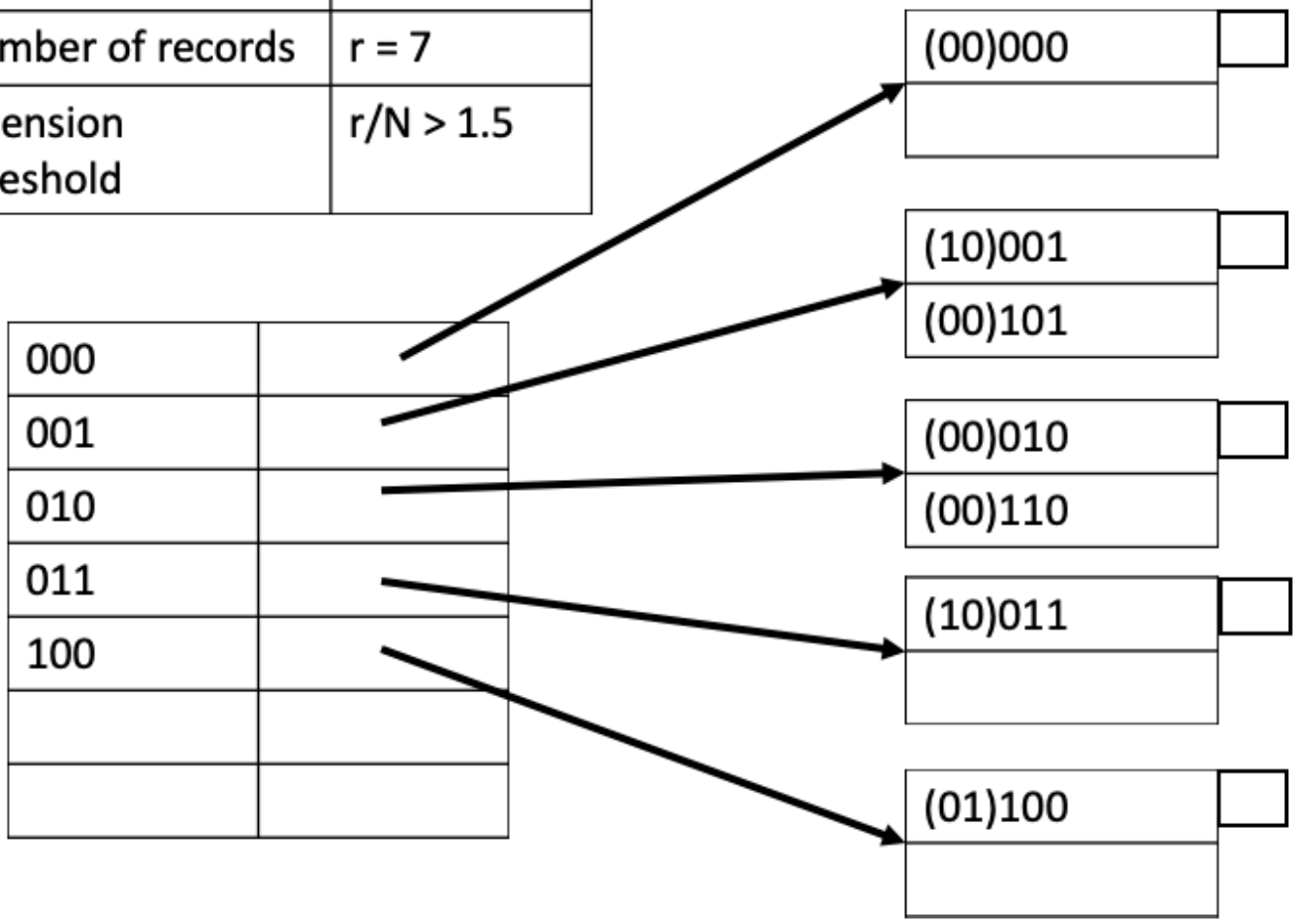
3. Insert 00110

Number of buckets	N = 4
Number of bits	i = 2
Number of records	r = 6
Extension threshold	r/N > 1.5



Insert 00000

Number of buckets	N = 5
Number of bits	i = 3
Number of records	r = 7
Extension threshold	r/N > 1.5

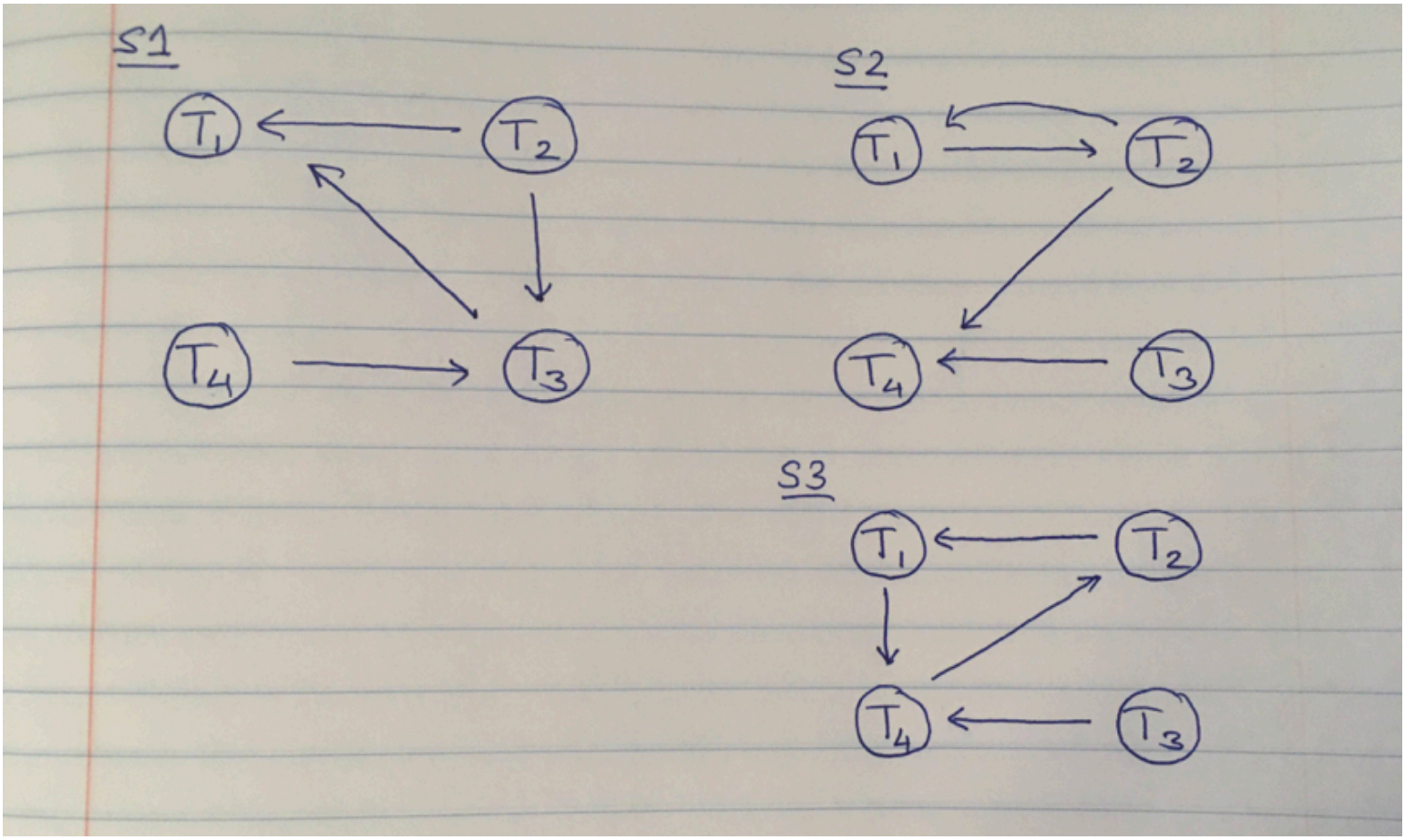


Problem 3.

1. ◦ SLOCK1(B); R1(B); XLOCK1(B); W1(B); SLOCK2(C); R2(C); SLOCK2(A); R2(A); XLOCK2(C); W2(C); XLOCK2(A); W2(A); REL2(A,C); XLOCK1(A); W1(A); XLOCK1(C); W1(C); REL1(A,B,C);
   ◦ SLOCK1(A); R1(A); REL1(A); SLOCK2(C); R2(C); SLOCK2(B); R2(B); XLOCK2(A); W2(A); XLOCK1(B); DENIED1;
2. ◦ SLOCK1(B); R1(B); XLOCK1(B); W1(B); SLOCK2(C); R2(C); SLOCK2(A); R2(A); XLOCK2(C); W2(C); XLOCK2(A); W2(A); REL2(A,C); XLOCK1(A); W1(A); XLOCK1(C); W1(C); REL1(A,C);
   ◦ SLOCK1(A); R1(A); SLOCK2(C); R2(C); SLOCK2(B); R2(B); XLOCK2(A); DENIED2;

Problem 4.

1.



2. S2 and S3 have cycles (T1<-->T2, T1->T4->T2->T1 respectively) so they are not conflict serializable. S1 after sorting topologically we get, T4->T2->T3->T1: R4(B); W4(B); R4(A); R2(A); R2(C); W3(A); R3(C); R3(A); W1(C); R1(C);

Problem 5.

1. ◦ S1

T1	T2	T3
SLOCK(A)		
R(A)		
		XLOCK(C)
		W(C)
		SLOCK(C)
	SLOCK(B)	
	R(B)	
	SLOCK(A)	
	R(A)	
	SLOCK(C)	
	R(C)	
		R(C)
		REL(C)
	XLOCK(C)	
	W(C)	
	REL(A,B,C)	
SLOCK(B)		
R(B)		
REL(A,B)		

S1(A); R1(A); X3(C); W3(C); S3(C); S2(B); R2(B); S2(A); R2(A); S2(C); R2(C); R3(C); REL(C); X2(C); W2(C); REL(A,B,C); S1(B); R1(B); REL(A,B);

2PL is feasible.

- S2

T1	T2	T3
XLOCK(A)		
W(A)		
SLOCK(A)		
R(A)		
	XLOCK(C)	
	W(C)	
		SLOCK(B)
		R(B)
		XLOCK(B)
		W(B)
XLOCK(C)		
WAIT		

X1(A); W1(A); S1(A); R1(A); X2(C); W2(C); S3(B); R3(B); X3(B); W3(B); X1(C); WAIT1(C);

2PL is not feasble since T1 has to wait for T2 to release C.

2. ◦ S1 is not possible in Strict 2PL, thus reordering is required which leads to the following schedule. This is still equivalent to the original schedule.

T1	T2	T3
SLOCK(A)		
R(A)		
		XLOCK(C)
		W(C)
	SLOCK(B)	
	R(B)	
	SLOCK(A)	
	R(A)	
	<SLOCK(C) WAIT till T3 commits>	
		R(C)
		T3 Commits
		REL(C)
	SLOCK(C)	
	R(C)	
	XLOCK(C)	
	W(C)	
	T2 Commits	
	REL(A,B,C)	
SLOCK(B)		
R(B)		
T1 Commits		
REL(A,B)		

◦ S2 is not feasible under Strict 2PL, thus reordering is required. Deadlock happens.

T1	T2	T3
XLOCK(A)		
W(A)		
R(A)		
	XLOCK(C)	
	W(C)	
		SLOCK(B)
		R(B)
		XLOCK(B)
		W(B)
<XLOCK(C) Waiting on T2 to commit>		
		<SLOCK(A) Waiting on T1 to commit>
	<SLOCK(B) waiting on T3 to commit>	