CS HW3

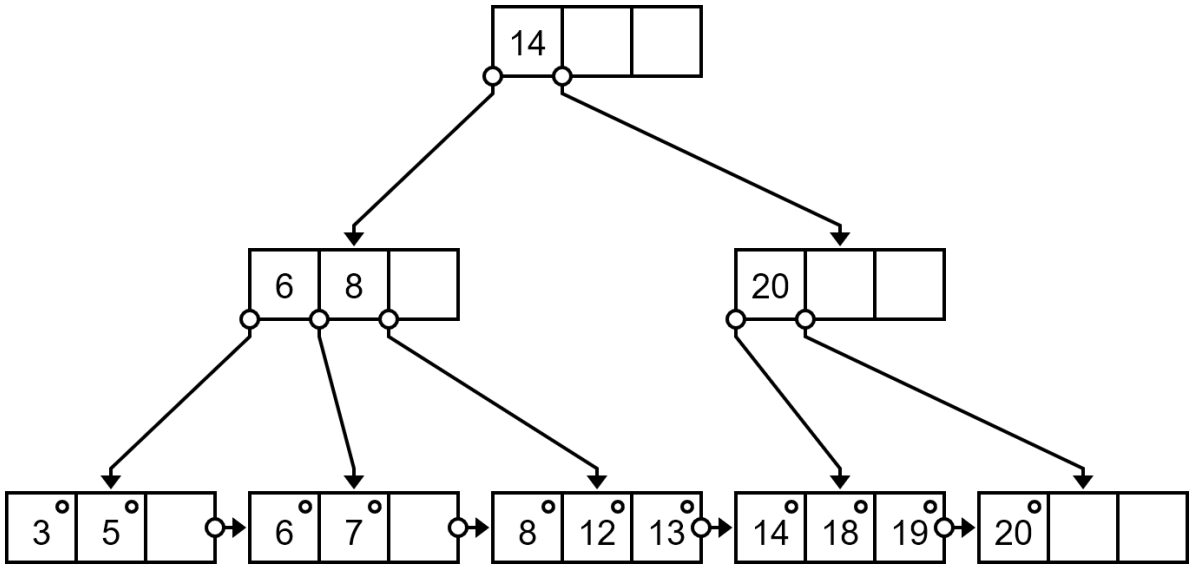
A)
Record size (r = 12000), Block size (B = 2048), Block pointer (P = 6), Record pointer (Pr = 7)

A1)
$R = 10 + 4 + 5 + 12 + 12 + 15 + 4 = 62$
$\text{bfr } = \lfloor \frac{B}{R} \rfloor = \frac{2048}{62} = 33$
$\text{Number of file blocks } = b = \lceil \frac{r}{\text{bfr}} \rceil = \lceil \frac{12000}{33} \rceil = 364$

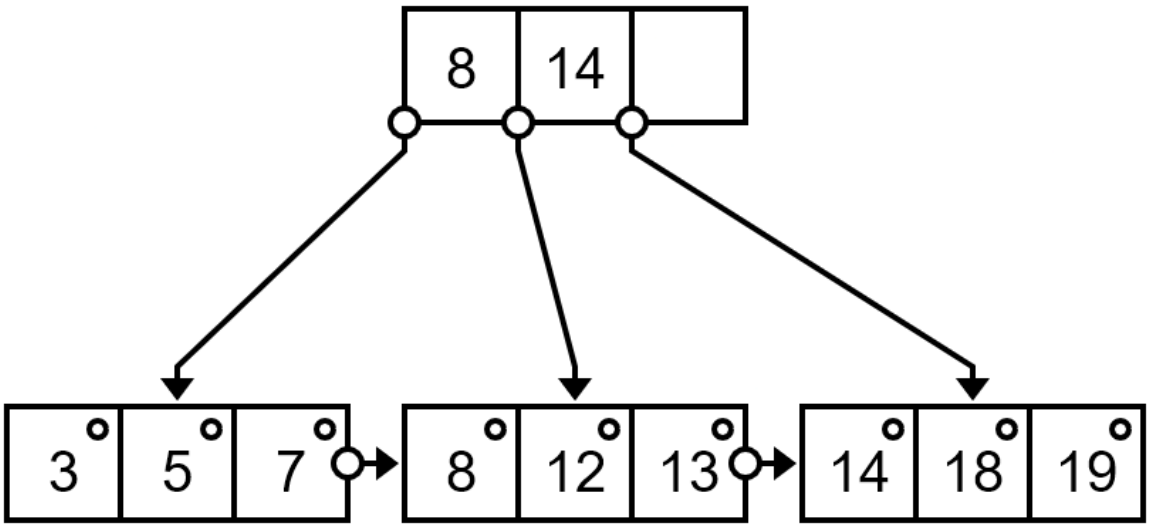
A2)
$\text{Employee ID} = 15 \text{ bytes}$
$Ri = V_{\text{EmployeeID}} + P = 15 + 6 = 21$
$bfri = \lfloor \frac{B}{Ri} \rfloor = \lfloor \frac{2048}{21} \rfloor = 97$
$\text{Total number of index entries } r_1 = \text{number of blocks} = 364 \text{ entries } = b$
$\text{Number of first level index blocks } b_1 = \lceil \left(\frac{r_1}{bfri} \right) \rceil = \lceil \left(\frac{364}{97} \right) \rceil = 4 \text{ blocks}$

A3)
$\text{Number of first level entries } = r_1 = \text{ number of blocks b } = 364 \text{ entries}$
$\text{Number of first level index blocks } b_1 = \lceil \left(\frac{r_1}{bfri} \right) \rceil = \lceil \left(\frac{364}{97} \right) \rceil = 4 \text{ blocks}$
$\text{Number of second level index entries } r_2 = b_1 = 4 \text{ entries}$
$\text{Number of second level index blocks } b_2 = \lceil \frac{r_2}{bfri} \rceil = \lceil \frac{4}{97} \rceil = 1 \text{ block}$
$\text{Total number of blocks for the index } = 4 + 1 = 5 \text{ blocks}$

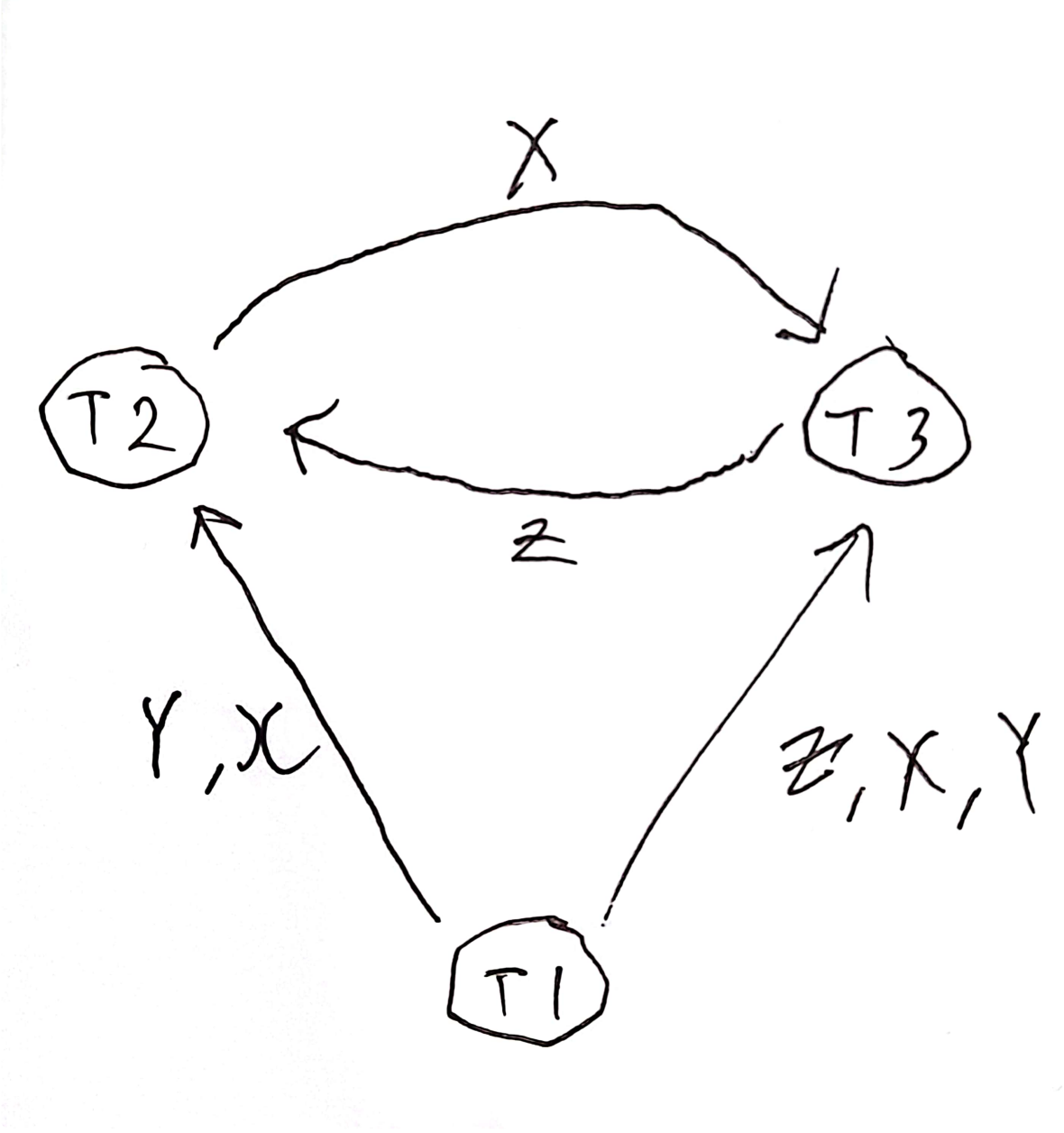
B1)



B2)

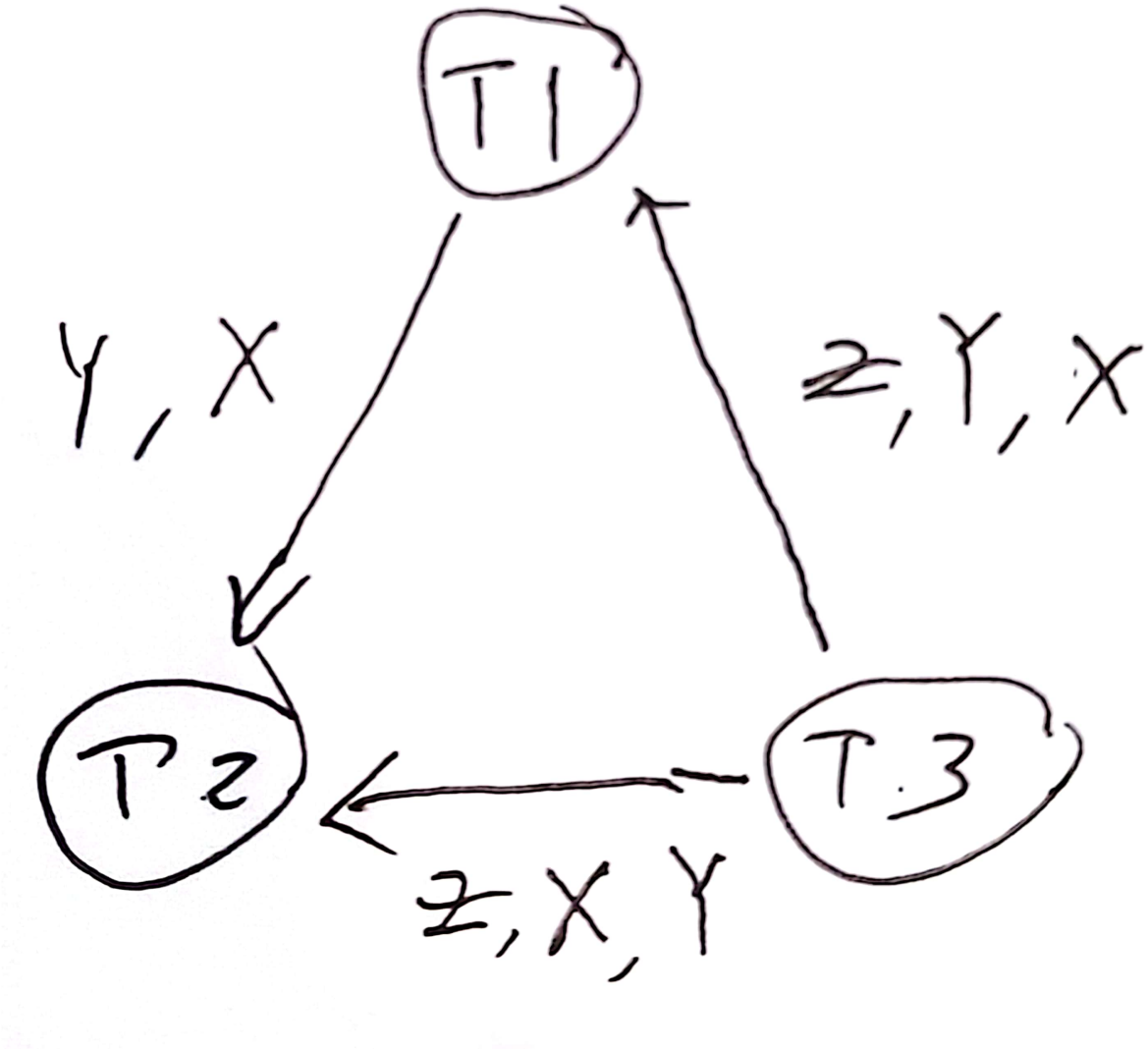


C) S1



Not serializable, because cycle exists between T2 and T3

S2



Is serializable. Equivalent serial schedule is T3→T1→T2

	T1	T2	T3
1			w3(Z)
2			r3(X)
3			r3(Y)
4			w3(X)
5			c3
6	r1(X)		
7	w1(Y)		
8	r1(Z)		
9	c1		
10		r2(Y)	
11		w2(X)	
12		r2(Z)	
13		w2(Y)	
14		c2	

C2)

S1
Strict:
No, T2 can read Y and write X from T1 before T1 commits.

Cascadeless:
No. T2 reads Y from T1 before T1 commits, and T1 commits before T2. Should T1 fail, both T1 and T2 have to be rolled back.

Recoverable/Non-recoverable:
Recoverable. T2 reads Z written by T3, and T2 commits after T3. Should T3 abort after T2 commits, value of Z not recoverable.

S2
Strict:
Yes. No transaction reads or writes from other transactions before it commits.

Cascadeless:
Yes. Because T1 reads X,Z after T3 commits, T2 reads Y,Z after both T3 and T1 commits.

Recoverable/Non-recoverable:
Recoverable. All transactions commit after all of their respective read write operations. T1 reads values from T3 after T3 commits, and T1 commits after T3. T2 reads from T1 after T1 commits, and T2 commits after T1.