2-1-1

(a) Read Staff and Branch from disk : 1000 + 50 = 1050;
Execute Staff X Branch as table1;
Save table1 back to disk : 1000*50;
Read table1 from disk : 1000*50;
Do the remaining Select operation.

(b) Read Staff and Branch from disk : 1000 + 50 = 1050;
Execute Staff$\bowtie_{\text{Staff.branchNo = Branch.branchNo}}$ Branch as table1;
Save table1 back to disk : 1000;
Read table1 from disk : 1000;
Do the remaining Select operation.

(c) Read Staff and Branch from disk : 1000 + 50 = 1050;
Select Staff and Branch with given expression separately as table1 and table2;
Save table1 and table2 back to disk : 50 + 5;
Read table1 and table2 from disk : 50 + 5;
Do the remaining Join operation;

2-1-2

(a) For r2_block in r2: (transfer : b_r2 , seek 1)
Read r2_block from disk
For r2_tuple in r2_block: (n_r2):
For r1_block in r1: (transfer :b_r1, seek 1)
Read r1_block from disk
check each tuple;

transfer = b_r2 + n_r2*b_r1 = 60003000;
seek = 1 + 1 = 2;

(b) For r2_block in r2: (transfer : b_r2 , seek b_r2)
Read r2_block from disk
For r2_tuple in r2_block: (n_r2):
For r1_block in r1: (transfer :b_r1, seek b_r1)
Read r1_block from disk
check each tuple;

transfer = b_r2 + n_r2*b_r1 = 60003000;
seek = b_r2 + n_r2 = 33000 ;

(c) For 100 r1_block in r1: (transfer : b_r1 , seek b_r1/100)
Read r1_block from disk
For r2_block in r2: (transfer :b_r2, seek b_r2)
Read r2_block from disk
check each block;

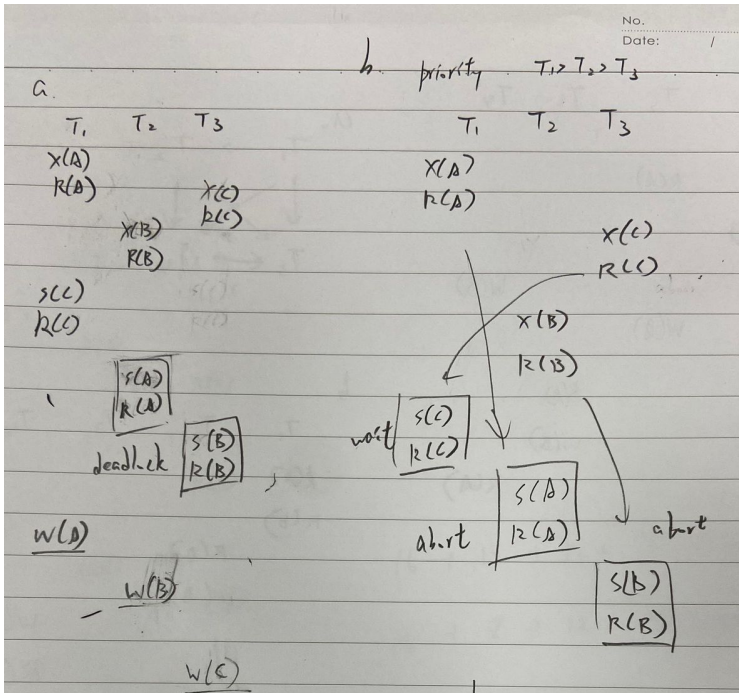
transfer = b_r1 +(b_r1/100)*b_r2 = 62000;
seek = b_r1/100 + b_r1/100 = 40 ;

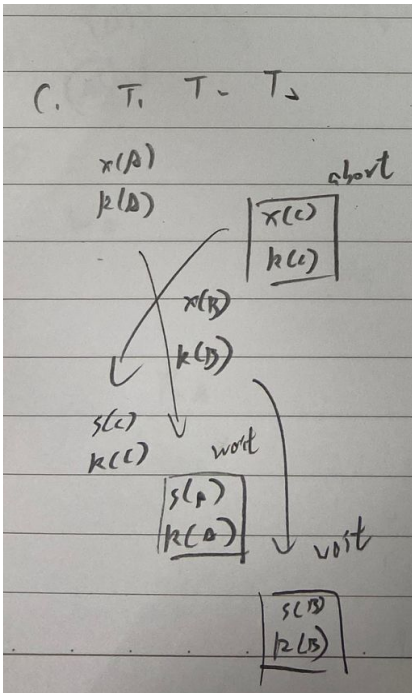
2-2-1

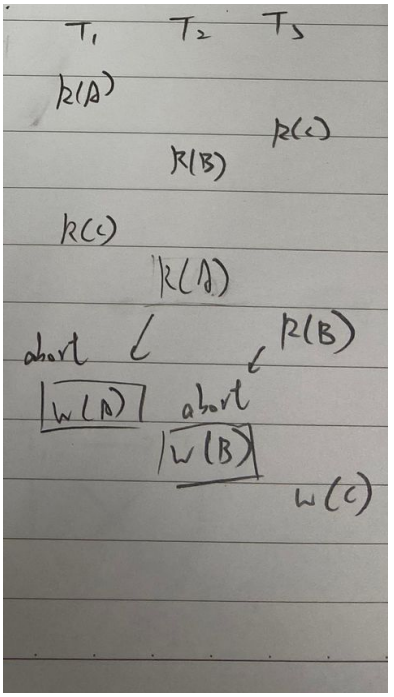
T1	T2	T3	T4
	S(B)		S(C)
	R(B)		R(C)
	S(A)		S(A)
S(B)	R(A)		R(A)
R(B)	U(A)		U(C)
X(C)	U(B)		U(A)
W(C)		X(A)	
U(B)		W(A)	
U(C)		X(B)	
		W(B)	
		U(B)	
		U(A)	

take 12s to complete

2-2-2







補充說明:

- 2-1-1:
  有解釋到每個數字怎麼來的就會給分

- 2-1-2:
  (a) seek 的部份是因為b_r1第一個pointer是固定的，視為constant value，所以每次要找的時候不用重新seek，只有第一次的時候需要，所以是1(b_r2)+1(b_r1)。

  (b) nested-loop join在沒有人可以完全fit in的時候cost會是一樣的。

- 2-2-1:
  12s滿分，15s內 -1～5，大於15 -10。

- 2-2-2:
  (a) T1 S(c),R(c)的部份有標沒標dead lock都行。