Part 3

A)

T1	T2	T3
L(A)		
R(A)		
W(A)		
		L(A)
		R(A)
		W(A)
	L(A)	
	R(A)	
L(B)		
R(B)		
		L(B)
		R(B)
W(B)		
	L(B)	
	R(B)	
	U(B)	
	COMMIT	
U(A)		
U(B)		
COMMIT		
		U(A)
		U(B)
		COMMIT

← Now transaction will execute in order: T1 → T3 → T2

B) We need strict 2PL because it ensures both conflict-serializability and recoverability. Strict 2PL never releases a lock after using it, it holds all locks until committing and releases all locks at once. Implementing strict 2PL prevents any kind of situation from arising where the possibility of a deadlock exists in the transaction.

In addition, strict 2PL avoids cascading aborts.