COEN 317: Distributed Systems
Spring 2020
Homework Assignment 3
Due: Tuesday, 6/2/2020 @11:59pm

1) In a certain system, each process typically uses a critical section many times before another process requires it. Explain why Ricart and Agrawala's mutual exclusion algorithm is inefficient for this case, and describe how to improve its performance.

2) In the Bully algorithm, a recovering process starts an election and will become the new coordinator if it has a higher identifier than the current incumbent. Is this a necessary feature of the algorithm?

3) A distributed system may have multiple, independent critical regions. Imagine that process 0 wants to enter critical region A and process 1 wants to enter critical region B. Can Ricart and Agrawala's algorithm lead to deadlocks? Explain your answer.

4) Consider a deadlock detection scheme for a single server. Describe precisely when edges are added to and removed from the wait-for-graph. Illustrate your answer with respect to the following transactions T, U and V at the server:

T	U	V
	Write(i, 66)	
Write(i, 55)		
		Write(i, 77)
	Commit	

When U releases its write lock on a_i, both T and V are waiting to obtain write locks on it. Does your scheme work correctly if T (first come) is granted the lock before V? If your answer is 'No', then modify your description.

5) The transactions T and U at the server are defined as follows:

T: x:= Read (i); Write(j, 44);

U: Write(i, 55);Write(j, 66);

Initial values of a_i and a_j are 10 and 20. Which of the following interleavings are serially equivalent and which could occur with two-phase locking?

(a)

T	U
x:= Read (i);	
	Write(i, 55);
Write(j, 44);	
	Write(j, 66);

(b)

T	U
x:= Read (i);	
Write(j, 44);	
	Write(i, 55);
	Write(j, 66);

(c)

T	U
	Write(i, 55);
	Write(j, 66);
x:= Read (i);	
Write(j, 44);	

(d)

T	U
	Write(i, 55);
x:= Read (i);	
	Write(j, 66);
Write(j, 44);	