# EE 360P Practice Questions

1. Assume that you have implemented Mattern's vector clock algorithm. However some application needs Lamport's time stamps. Write a function *convert ()* that takes as input Mattern's time stamp and outputs Lamport's time stamps.

2. Let s.v be the vector clock for the local state s. Given n vectors, one from each state, one can form a matrix to represent the cross product of the local states which can be viewed as the global state. A global state is defined to be consistent if every pair of the local states are concurrent. Give a suitable condition on the matrix for this global state to be consistent. (i.e. given a matrix, how would you check if it represents a consistent global state?)

3. A vector clock designer has come up with the idea that instead of process $P_i$ sending the entire vector to $P_j$ every time, it should only send those components of the vector that are different from the last time it sent a message to $P_j$. What are the advantages and disadvantages of this scheme ?

4. To totally order all the events in a distributed system (preserving happened-before relation), one can use Lamport's logical clock with process id. Two events $e$ and $f$ are then ordered as

$$e < f \equiv (e.c < f.c) \vee ((e.c = f.c) \wedge (e.p < f.p))$$

   This ordering favors the small numbered processes when two events have the same value for Lamport's clock. Modify the scheme so that this unfairness is removed.

5. The mutual exclusion algorithm by Lamport requires that any request message be acknowledged. Under what conditions does a process not need to send an *acknowledgement* message for a *request* message?

6. Show how you will modify the centralized algorithm for mutual exclusion so that requests are granted in the order (happened-before order) they are made.