1) If events corresponding to vector timestamps $V_{t_1}$, $V_{t_2}$, ...., $V_{t_n}$ are mutually concurrent, then prove that,

$$(V_{t_1}[1], V_{t_2}[2], ...., V_{t_n}[n]) = \text{max}(V_{t_1}, V_{t_2}, ...., V_{t_n}).$$

2) If events $e_i$ and $e_j$ respectively occurred at processes $p_i$ and $p_j$ and are assigned vector timestamps $VT_{e_i}$ and $VT_{e_j}$ respectively, then show that

$$e_i \rightarrow e_j \iff VT_{e_i}[i] < VT_{e_j}[i]$$

3) Consider the following simple method to collect a global snapshot (it may not always collect a consistent global snapshot): an initiator process takes its snapshot and broadcasts a request to take snapshot. When some other process receives this request, it takes a snapshot. Channels are not FIFO.

Prove that such a collected distributed snapshot will be consistent iff the following holds (assume there are $n$ processes in the system and $V_{t_i}$ denotes the vector timestamp of the snapshot taken process $p_i$):

$$(V_{t_1}[1], V_{t_2}[2], ...., V_{t_n}[n]) = \text{max}(V_{t_1}, V_{t_2}, ...., V_{t_n}).$$

Don't worry about the channel states.

4) Consider a distributed system where every node has its physical clock and all physical clocks are perfectly synchronized. Give an algorithm to record global state assuming the communication network is reliable. (Note that your algorithm should be simpler than the Chandy-Lamport algorithm).

5) What modifications should be done to the Chandy-Lamport snapshot algorithm so that it records a strongly consistent snapshot (i.e., all channel states are recorded empty).