Question 2 (10 points)
Recall that when we studied logical clocks, we assumed a process could execute three types of events: local events, send events, and receive events. We showed that Lamport clocks satisfy the Weak Clock Condition in this system model, but they do not satisfy the Strong Clock Condition.

Consider a modified system model (with \( N \) processes) in which processes can only execute two types of events: send events and receive events. There are no local events. Further, for \( i = 1, \ldots, N - 1 \), a process \( p_i \) can only send messages to process \( p_{i+1} \). Process \( p_N \), cannot send any messages.

Do Lamport clocks satisfy the Strong Clock Condition in this system model? Justify your answer.

Question 3 (10 points)
Consider the following modified vector timestamp algorithm:

The system has \( N \) processes. Every process \( p_i \) has:
- A physical clock \( PC_i \).
- A vector clock \( V_i \), which is an \( N \)-vector of real numbers.

For both of these, assume that there is as much precision as needed to create and store any number. The physical clock is correct, with \( \rho \ll 1 \). \( PC_i(0) = 0 \), and no clock synchronization algorithm is executed after \( t = 0 \).

The algorithm (as executed by each \( p_i \)) is as follows:
Initially: \( V_i = (0, 0, \ldots, 0) \)
On event \( e \):
\[ V_i(i) = PC_i(t), \]  
where \( t \) is the absolute “wall clock time” when \( e \) occurs
If \( e \) is a send of a message \( m \)
\[ m.V = V_i \]
If \( e \) is a receive of a message \( m \)
\[ V_i(j) = \max(V_i(j), m.V(j)) \] for \( j = 1 \ldots N \)
Timestamp \( e \) with \( V_i \)

Does this modified algorithm guarantee the strong clock condition? Justify your answer.

Assume
\[ \prec \]

is defined for vector timestamps the same way as it was in class.)

Question 4 (10 points)
Consider a system with three sites: a time server \( S \) and two processes \( P_A \) and \( P_B \). These sites are connected as shown in the diagram below. The arrows indicate the directions in which messages may be sent.

Communication between \( S \) and \( P_A \) is asynchronous, and communication between \( P_A \) and \( P_B \) is synchronous. \( S \) and \( P_B \) cannot communicate directly. Assume that computation is instantaneous.

Site \( S \) is a source of UTC time. It is \( P_A \)'s job to periodically initiate a physical clock synchronization algorithm for both itself and \( P_B \). (Assume that clock drift is negligible with respect to the time it takes for the algorithm to execute.)

1. Design an algorithm to synchronize \( P_A \)'s and \( P_B \)'s clocks with \( S \) based on the two algorithms we studied in class. Describe what messages are sent and how each process sets its clock.
2. Give the accuracy of \( P_B \)'s clock immediately after it completes your algorithm. Explain how you derived your answer.