Question 2
Consider the Wuu-Bernstein algorithm for the Replicated Log Problem.
For each statement below, indicate whether the statement is TRUE or FALSE. If the statement is TRUE, provide a justification. If the statement is FALSE, provide a counter-example.

1. For any pair of processes \( p_i \) and \( p_j \), it always holds that \( T_i(j, i) \leq T_j(j, i) \).
2. For any pair of processes \( p_i \) and \( p_j \), \( i \neq j \), it always holds that \( T_i(i, i) \leq T_j(j, j) \).

Question 3
Dr. Science proposes to improve Lamport’s Mutual Exclusion algorithm by allowing a process to access the resource once it has received permission from a majority of processes. Assume there are \( N \) processes, where \( N \) is even. Dr. Science’s algorithm variation proceeds as follows

- \( p_i \) requests resource:
  - send \( \text{req}(i, C_i) \) to all processes, excluding itself.
  - put \( \text{req}(i, C_i) \) in own queue

- When \( p_j \) receives \( \text{req}(i, C_i) \):
  - put \( (i, C_i) \) in its queue
  - send GRANT to \( p_i \)

- \( p_i \) can access resource when:
  - it has received GRANT from \( \frac{N}{2} \) processes and its request is at the head of its own queue
    (note that a process implicitly receives permission from itself when it adds its request to its own queue)

- When \( p_i \) finishes with resource:
  - send RELEASE to all processes from which it has received a GRANT message
  - remove \( (i, C_i) \) from queue

- When \( p_j \) receives RELEASE from \( p_i \)
  - remove \( (i, C_i) \) from queue

The system model is the same as defined for Lamport’s algorithm: reliable, FIFO, asynchronous communication, and no process crashes.

1. Does Dr. Science’s proposed algorithm variation guarantee safety? Answer YES or NO and give a justification.
2. Does Dr. Science’s proposed algorithm variation guarantee liveness? Answer YES or NO and give a justification.

Question 4
Consider an execution of Raymond’s algorithm in the tree network shown below. Site \( F \) has the token and is accessing the resource. Before it is finished, sites \( A \) and \( D \) both issue requests for the resource. Suppose that the ordering of the requests and messages results in the worst-case number of messages for getting the token to \( A \) and to \( D \) (not necessarily in that order). Assume that no other processes request the resource.

1. Explain the steps of Raymond’s algorithm that are executed after site \( F \) is finished with the resource to grant the two requests.
2. What is the total number of messages that are sent to grant the requests of \( A \) and \( D \)?
**Question 5** Recall that Maekawa’s original mutual exclusion algorithm is prone to deadlock. Sanders proposed a revision of the algorithm that is deadlock free. In this revision, a process must relinquish its locks only if it receives both an INQUIRE message and a FAIL message. Give an example execution where a process receives an INQUIRE message (but no FAIL message), and there is no deadlock.