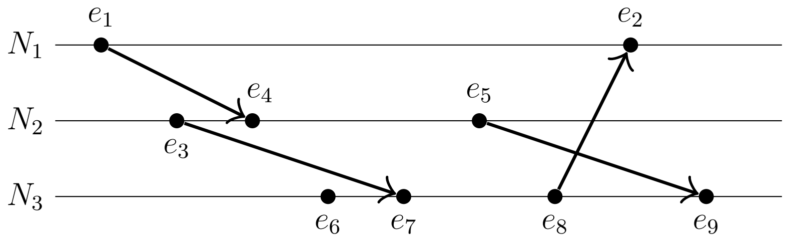
5 Concurrent and Distributed Systems (tlh20)

Distributed systems often use *logical timestamps* to capture the possible ordering between operations on different nodes.

(*a*) Three nodes $N_1 \dots N_3$ communicate using point-to-point messages:



(*i*) Give *Lamport timestamps* for each of the events $e_1 \dots e_9$, assuming clocks start from 0. [2 marks]

(*ii*) What is a necessary condition on Lamport timestamps for an event x to have *happened before* an event y? Is this a sufficient condition? Briefly justify your answer based on the events in the example. [2 marks]

(*iii*) Now give possible *vector timestamps* for each of the events $e_1 \dots e_9$. [2 marks]

(*iv*) What is a *necessary and sufficient* condition on vector timestamps for an event x to have happened before an event y? [2 marks]

(*b*) An alternate form of vector timestamps can be used to build *causal broadcast*. That is, if the broadcast of one message happened before the broadcast of another message, then all nodes must deliver those two messages in that order.

(*i*) Write pseudo-code to implement causal broadcast, assuming access to an underlying reliable broadcast protocol. You should describe the local state held at each node along with the algorithm used to broadcast a message, and to decide when a message can be delivered locally. [8 marks]

(*ii*) A colleague suggests that the space required by vector timestamps makes this algorithm inefficient. They suggest basing the algorithm on Lamport timestamps instead. Do you agree with this suggestion? [4 marks]