```python
class Process:
  def __init__(self, chanID, procID, procIDSet, initTX):
    self.txID       = initTX       # Your own most recent transaction
    self.leader     = self.procID  # Who you believe may become leader
    self.lastTX     = self.txID    # What is the most recent transaction
    self.noleader   = False        # Are you still in the race for leader?

  def receive(self):
    while True:
      msg             = self.chan.recvFrom(self.otherProcs)
      sender, payload = msg[0], msg[1]
      if payload[0] == ELECTION: # A process started an election
        voteID, voteTX = payload[1], payload[2]

        if self.lastTX < voteTX: # You’re not up to date on most recent transaction
          self.leader = voteID   # Record the suspected leader
          self.lastTX = voteTX   # As well as the likely most recent transaction

        elif (self.lastTX == voteTX) and (self.leader < voteID): # Wrong leader
          self.leader = voteID   # Update your suspected leader

        elif (self.procID > voteID) and (self.txID >= voteTX) and (not self.noleader):
          # At this point, you may very well be the new leader (having a sufficiently
          # high process identifier as well as perhaps the most recent transaction).
          # No one has told you so far that you could not be leader. Tell the others.
          self.chan.sendTo(self.otherProcs, (LEADER, self.procID, self.txID))

      if payload[0] == LEADER:
        # Check if the sender should indeed be leader
        if ((self.lastTX < payload[2]) or
            ((self.lastTX == payload[2]) and (self.leader <= payload[1]))):
          # The sender is more up-to-date than you, or is equally up-to-date but
          # has a higher process identifier. Declare yourself follower.
          self.chan.sendTo(sender, (FOLLOWER, self.procID))
        else:
          # Sender is wrong: you have information that the sender based its decision
          # on outdated information
          self.chan.sendTo(sender, (NOLEADER))
```