Bully Algorithm (Garcia-Molina 1992)

Types of messages: ELECTION, COORDINATOR, OK

Initially:
\[
\begin{align*}
\text{leader} &= \bot \\
\text{holdingElection} &= \text{false}
\end{align*}
\]

To initiate election:
\[
\begin{align*}
\text{holdingElection} &= \text{true} \\
\text{send ELECTION to all processes with higher IDs} \\
\text{wait for } T \text{ time units to receive OK messages from these processes} \\
\text{if no responses received} \\
& \quad \text{leader} = \text{self} \\
& \quad \text{send COORDINATOR to all processes with lower IDs} \\
& \quad \text{holdingElection} = \text{false} \\
\text{else} \\
& \quad \text{wait for } T' \text{ time units to receive COORDINATOR message} \\
& \quad \text{if COORDINATOR received from } \text{q} \\
& \quad \quad \text{leader} = \text{q} \\
& \quad \text{else} \\
& \quad \quad \text{initiate election}
\end{align*}
\]

On receiving ELECTION from process q with lower ID:
\[
\begin{align*}
\text{send OK to } \text{q} \\
\text{if } \neg \text{holdingElection} \text{ initiate election}
\end{align*}
\]

On recovery from failure:
\[
\text{initiate election}
\]

On receiving COORDINATOR message from q:
\[
\begin{align*}
\text{if } \text{self.ID} > \text{q.ID} \text{ and } \neg \text{holdingElection} \quad \text{(can be avoided at network layer)} \\
& \quad \text{initiate election} \\
\text{else} \\
& \quad \text{leader} = \text{q} \\
& \quad \text{holdingElection} = \text{false} \\
& \quad \text{cancel any ongoing election}
\end{align*}
\]
Ring Algorithm  (Simplified version of Chang and Roberts, 1979)

Initially:
   leader = ⊥

To initiate election:
   send (ELECTION, MY_ID) to successor

On receive (ELECTION, PID):
   if (PID ≠ MY_ID)
      if MY_ID > PID
         send (ELECTION, MY_ID) to successor
      else
         send (ELECTION, PID) to successor
   else
      leader = MY_ID
      send (COORDINATOR, MY_ID) to successor

On receive (COORDINATOR, PID):
   if (PID ≠ MY_ID)
      leader = PID
      leader = PID
      send (COORDINATOR, PID) to successor