327 assignment 2: leader election

```
send_joining()---
if(guard) {
        enqueue_message(join);
        mydata->initiator = true;
}

loop()---
send_election()

send_election()---
if(initiator && !IsQueueFull() && state == COOPERATIVE)
        enqueue_message(ELECTION); //define ELECTION in header
        mydata->initiator = false;

enqueue_message(msg)-----
if (msg == ELECTION)
        data[MIN_ID] = mid_id; //define mid_id in header
else

message_rx()----- //callback method          .
        switch(data[msg])
                case ELECTION:
                        if (data[ID] == myData->left)
                                receive_election();
                        break;
                case ELECTED:
                        break;
                default:
                        break;

receive_election()-----
initiator = true;
```

have one initiator for protocol
all nodes choose node with minimum ID: solve problem
leader color: white, if not, use red color
execute protocol when node joins (this node is the initator)

node sends electing(v) to successor
m = smallest ID of node
if node gets electing(other node)
        if node with other node < m
                node forwards electing(other node) clockwise and set other node to m
                node does not become leader
        else if other node > m and node is not participating
                v sends electing(m) to successor
        else if node = other node
                node sends elected(node) clockwise

if node gets elected(other node) with other node != node
node forward elected(other node) clockwise and leader = other node