Distributed Algorithms - Homework 2

1. The question relates to the Burns-Pachl algorithm for self-stabilization.

Prove or disprove:

The algorithm works correctly when the following conditions are removed from Rule A (one at a time):

(a) \( L_i \neq 0 \)
(b) \( T_{i-1} = 0 \)
(c) \( T_{i-1} \neq L_i - L_{i-1} \)
(d) \( T_{i-1} < T_i \)

Note: Whenever you disprove, give a counter example (an execution) which is as short as possible.

2. The question relates to Franklin’s \( 2n \log n \) algorithm for election bidirectional in rings (or Peterson’s 1st algorithm for election in unidirectional rings).

Assume a set of \( n \) identities, that all \( n! \) permutations are equally probable, and that all processors start the algorithm as candidates. What is the expected number of processors that will remain candidates after the first phase?

Note: Give a direct combinatorial proof. In particular do not use probabilistic tools (e.g. linearity of expectation).

3. The question relates to Peterson’s 1st algorithm to find a leader in a unidirectional ring.

(a) Give a proof of correctness.
(b) What is the worst case message complexity of the algorithm? Prove.

4. The question relates to Peterson’s 2nd algorithm to find a leader in a unidirectional ring.

(a) Prove that the algorithm always determines exactly one processor as a leader.
(b) Show that this leader is not always the processor with the maximal identity. Give an example with as few processors as possible where this happens.
(c) Show exactly the point in the execution which is crucial.
(d) Modify the algorithm so that it will determine the processor with largest identity as a leader.
   It is not allowed to first isolate a single processor, and then have it send a message around the ring to identify the maximal identity.

(e) Prove the correctness of your algorithm.