Homework 04

Problem 4.1:

* **Mutual Exclusion**, ~~...~~ is a concurrency control property, ~~...~~ it prevents multiple processes from entering their critical sections simultaneously, thus avoiding race conditions. it ensures that only one process can access shared resources at a time, maintaining consistency and correctness.

⟹ **During Execution**, All processes are aware that they must not enter their critical section, if some other process have enter his.

• The state of wether a process is in its Critical section is globally Known, or mechanisms (Locks or semaphores) are used to enforce this exclusion.

⟹ **After Execution**, Upon exiting critical section by a process others may be allowed to enter their critical sections.

* **Leader Election**, designate a single proc (leader) among a group of distributed process to perform a ~~...~~ specific role or task that requires central coordination.

⟹ **During Execution**: processes are unaware of who would be elected leader. processes communicate their ~~...~~ identities or priorities until a consensus on a single leader is reached

* if we were guaranted that all processes would survive the execution and that none of them risks failing we would not need Leader Election ~~...~~ Simply pick any ~~...~~ arbitrary process to be leader. However, that is not the case in practique real life

⟹ **After Execution**: Once a leader is elected only him knows it, thus information (Leader Ids)

~~Msg~~ is flooded to all other processes. The leader retains its role until it fails, resigns, execution terminates or a new election is triggered, depending on the systems design.

* Mutual exclusion aims to control access to a critical section, while Leader Election aims to select one process to be coordinator.

Problem 4.2:

* In Election Algorithms for trees it's decentralized, as every leaf node represents an initiator, they start by sending ⟨wake up⟩ msg to all their neighbours and so does the latter until ⟨wake up⟩ reaches the root ~~which would~~. After D times unit (from ~~root~~ leaf to root) from the start of algorithm. Upon succesful election, Leader's identity needs to propagate to every node at each level. This dissemination of information requires time proportional to the diameter D, as it's performing a traversal of all present nodes.

* To conclude, ~~the~~ we are left with a total of $2D \simeq O(D)$ time complexity for election Algorithm in a tree topology (Upwards and downwards traversal)

Problem 4.3:

Chang-Roberts Algorithm:

* At most N different tokens are exchanged, each by at most N hops, which yields an $O(N^2)$ bound on the message complexity.

• In which scenario $\Omega(N^2)$ occurs?

⇒ Let's consider a ring network with an initial configuration where node identities are arranged in an increasing clockwise around the ring and each process is initiator, each token would have to traverse the entire network, forwarded by $N-i$ hops before being discarded by process with smaller identity, which brings the number of message passes to a total of

$$\sum_{i=0}^{N-1}(N-i) = \frac{1}{2}N(N+1) \simeq O(N^2)$$