1. Since this question posed many misunderstandings in Assignment 1, we would ask you to revise and resubmit your answers. See some comments after the question.

Given a message-passing asynchronous network $N$ and a distributed algorithm $A$, recall the three definitions for time:

- $f(A)$ is the *synchronous time* of $A$. That is, the time it takes the algorithm to terminate assuming all processors start simultaneously, and working synchronously (so that all messages sent at time $t$ are received before time $t+1$).
- $g(A)$ is the *longest-chain time* of $A$. This is the longest chain of messages, assuming any non-empty subset of processors as starters.
- $h(A)$ is the *bounded-delay* time. This is the longest time until termination, assuming any non-empty subset of processors as starters, and assuming that each message is delivered within less than 1 time unit.

Consider the 5 possible cases: all three measures are equal, only two of them are equal and differ from the other (three cases), and all three measures are distinct. Give examples of networks and algorithms, for as many of the five as you can.

Note the following:

(a) In order to prove the needed, you may define any (simple) algorithm you want. There is no obligation to stick to the algorithms taught in class.

(b) In asynchronous executions, any subset of the processes may start spontaneously; moreover, these initiators may not start in the same time.

(c) All processes must terminate. Make sure your algorithm does not leave any process waiting forever.

2. The question relates to Dijkstra’s 1st algorithm for mutual exclusion for ring networks.

(a) We saw in class that the algorithm, that is using $k$-state machines, self-stabilizes if $k \geq n - 1$ with a *centralized scheduler*. Prove that the algorithm self-stabilizes if $k \geq n$ with a *distributed scheduler*. (Recall that in a single step a distributed scheduler allows any non-empty subset of privileged processors to simultaneously make a move.)
(b) Give examples, with \( K = N - 2 \), in which the system never stabilizes under a centralized scheduler. Give examples, with \( K = N - 1 \), in which the system never stabilizes under a distributed scheduler. You have to give examples for infinite values of \( N \) (if possible, for every \( N \)).

(c) Estimate the number of steps that it takes the system to self-stabilize in the worst case, for the case where \( K \geq N - 1 \) under a centralized scheduler. Give a best possible upper bound, and a best possible lower bound (that is satisfied by infinite number of cases).

(d) Estimate the number of steps that it takes the system to self-stabilize in the worst case, for the case where \( K \geq N \) under a distributed scheduler. Give a best possible upper bound, and a best possible lower bound (that is satisfied by infinite number of cases).

3. Consider Dijkstra’s 3rd algorithm for mutual exclusion for ring networks.
   
   Gives upper and lower bounds for the time needed until stabilization. Give a best possible upper bound, and a best possible lower bound (which holds for infinite values of \( N \)).