Exercise Session 10
More View Synchrony

Problem 1

Recall the View Synchrony algorithm based on Group Membership (slide 20 on View Synchrony). The algorithm is not live if we use Group Membership as a black box, relying on the specification in the slides (slide 8).

1. Explain why the algorithm is not live.
2. How could one change the definition of Group Membership, such that the algorithm for View Synchrony becomes live without any modifications.

Soluition

1. Imagine the following scenario. Process p_1 is correct, but never installs view v. Note that this can happen according to the specification of Group Membership, as long as other p_1 installs a later view v′. However, this means that p_1 will never trigger TRBBroadcast for view v, making all processes wait infinitely for p_1's message. Since p_1 is correct, the other processes cannot deliver the special empty message in this case.
2. The root of the problem lies in the fact that the current definition of Group Membership allows for "holes" in the view sequence, i.e. a process might never install certain views. For example, installing view 1, 2, and 4, but never installing view 3, does not violate the definition of Group Membership as currently defined.

   One possible way of "fixing" this problem is to make each process install a "full sequence" of views, by adding a property to Group Membership: If a process installs view (j, M) for $j > 1$, then it already has installed view $(j-1, M')$.

   Note, however, that it is not necessary to prevent "holes" in the view sequence to make the algorithm work. It is sufficient if all processes install exactly the same sequence of views (with the same "holes"). To achieve this, one might change the agreement property to be stronger: If a process installs view (j, M) then all correct processes install view (j, M).