CS 188-3 HW1
Spring 2021

1. Leader Election: We have two processors, p_0 and p_1. We are trying to elect a leader.

   (a) Consider the model in which every round there is one fault. This is unlike what we talked about in class where there is **at most** one failure. Here, we know that there will be **exactly** one fault per round. What is an algorithm to elect a leader?

   Hint: if p_0 does not receive a message, p_0 can be sure that p_1 did.

   (b) Consider the model in which every round there could be at most one fault, but we guaranteed that some time **there will** be a fault. I.e. the infinite run of no failure at all is excluded. Can we do election in this model? If so, show an algorithm.

   Hint: more than one round needed

2. Snapshot: We have N processors. In each round, each processor will send to all other processors all the IDs that they have in their participant set. Initially the participant set of a processor contains only its own ID. When a processor receives a processor ID that they have not yet encountered, they add that ID to their participant set. Between each pair of processors, at most one message between them may not arrive to the other side, for all $O(n^2)$ communication channels.

   Processors go through N rounds. If at the end of round k some processor p_i has heard only of a set of size k, it *returns* this set (and can continue or stop - does not matter, but remember that "return" is irrevocable.

   (a) Show inductively that at the beginning of round i, all processors have a set of at least size i

   (b) Show that at round i, if two processors p_0, p_1 return, they both return the same set.

   (c) Show that the set returned at round i is a subset of all the heard of sets at round $i + 1$

3. Immediate Snapshot: In immediate snapshot, I want a snapshot but with additional property: if processor p_i has processor p_j in its snapshot, then S_j is a subset of S_i.

   (a) Show an algorithm that achieves the required properties.
   Hint: Use the algorithm from Problem 2

   (b) Prove that k processors that execute the snapshot algorithm of Problem 2 that at least one processor must have the whole set as its snapshot.

   (c) Inductively prove that at most k processors survive to round we call k.