1.
(a) Consider a scenario where a Proposer p completes phase 1 for a proposal number n1. Another proposer q then completes phase 1 for a proposal number n2>n1. Proposer p’s phase 2 accept requests for a proposal numbered n1 are ignored because the acceptors have all promised not to accept any new proposal numbered less than n2. So, proposer p then begins and completes phase 1 for a new proposal number n3 > n2, causing the second phase 2 accept requests of proposer q to be ignored. And so on.

(b) To guarantee progress, a distinguished proposer must be selected as the only one to try issuing proposals. If the distinguished proposer can communicate successfully with a majority of acceptors, and if it uses a proposal with number greater than any already used, then it will succeed in issuing a proposal that is accepted. By abandoning a proposal and trying again if it learns about some request with a higher proposal number, the distinguished proposer will eventually choose a high enough proposal number.

2.
(a) Replicas each use more than required memory to store the information on commands that have already been decided. Even though some of these may be used in future by the replicas to filter out duplicates, most of them are unnecessary

(b) Once all replicas in a configuration have learned the decision for a slot, the corresponding leader and acceptor state for that slot can be removed from memory.