**Codes that causes the error: history is not linearizable**

IIn raft.go, Line 302, int sendAppendEntries

if rf.matchIndex[server] < args.PrevLogIndex+len(args.Entries) {

            rf.matchIndex[server] = args.PrevLogIndex + len(args.Entries)

            rf.nextIndex[server] = rf.matchIndex[server] + 1

            DPrintf("Leader %d set nextIndex of server %d to %d/%d in term %d\n", rf.me, server, rf.nextIndex[server], rf.GetLastIndex(), rf.currentTerm)

        }

Must not have: if rf.matchIndex[server] < args.PrevLogIndex+len(args.Entries)

Cause applied log missing and not linearizable.

The reason behind this situation may be: what really matters to the outside world client is the commitIndex, to which the logs are committed and applied. Match index is only used to track the transient state of the follower, and what already matched log entries should be able the hold the chance that it would be overwritten.

I was mislead by the wording in the Raft paper saying “increases monotonically”, debugging for 10+ hours for that, but fortunately my previous implementation didn’t do so and passed the 3A test, making able to find out the problem in comparison.

The other Raft\*.go codes are some of my other trials, all causing “history is not linearizable”, but still hold the chance that it can work after modification, and probably better performance?