Session 3: Leader election
Christian Fischer Pedersen, cfp@ece.au.dk

---

1 Exercise: Leader election

1. In general, why and when is leader election relevant in distributed systems?
   (a) Discuss applications that would / would not work without leader election.
2. Does leader election entail a computational and/or network burden?
   (a) If so, how can time complexity be used as a metric for the computational burden?
   (b) If so, how can message complexity be used as a metric for the network burden?
3. Discuss the mechanics of the presented leader election algorithms, e.g. by considering
   (a) Concretely, how do they function?
   (b) What is their scope of applicability (i.e. their problem and application domains)?
   (c) Discuss pros and cons of competing algorithms.
4. Bully Election
   (a) Derive the message complexity
   (b) Implement the algorithm (preferably in Python) and verify it works
   (c) Use the implementation to demonstrate the message complexity in practice
5. Hirschberg-Sinclair
   (a) Derive the message complexity
   (b) Implement the algorithm (preferably in Python) and verify it works
   (c) Use the implementation to demonstrate the message complexity in practice