

CS171- HW3
Due date: May 27th, 2009

- 1) State which communication primitives (reliable, atomic, causal multicast) will you use and why to achieve:
 - a) A distributed jigsaw-puzzle is played over the network. All players try to build together the same copy of the puzzle. When the game starts there is one piece on the board. Any piece should be connected to the pieces that currently exist on the board. Only the correct piece matches its place. The target is to finish the jigsaw-puzzle as soon as possible.
 - b) A distributed color map (Assume 4x4 blocks each block of one unique color) is held by servers. Clients can query the color of a certain slot in the map and update it (a client can only act through one server). The color maps held by different servers must stay consistent with each other. There are no process crashes or network partitions. How can you achieve this using group communication.
- 2) Explain how FIFO is achieved in case of reliable broadcast/multicast?
- 3) Suppose that the Byzantine generals only need to reach an approximate agreement, That is, the following two conditions must hold:
 - a) All loyal generals attack within 10 minutes of each other.
 - b) If the commander is loyal, all loyal generals attack within 10 minutes of the time given by the general's order.Show that approximate agreement cannot be guaranteed unless more than two thirds of the generals are loyal.
- 4) Suppose that two causal multicast groups have overlapping members. Modify the vector timestamp causal multicast algorithm to ensure causal message delivery in spite of overlapping multicast groups.
- 5) Modify Algorithm 14.1 of the textbook to design an early-stopping algorithm for consensus under fail-stop failures, that terminates within $f' + 1$ rounds, where f' , the actual number of stop-failures, is less than f . Prove the correctness of your algorithm.
(Hint: A process can be required to send a message in each round, even if the value was sent in the earlier round. Process should also track the other processes that failed, which is detectable by identifying the processes from which no message was received).