CS280D HW1

Due: Wednesady 4/10 in class, each problem separate sheet of paper

1. Consider the model of round by round adversary on two processors. Recall that the Adv can delete one message of the two sent in a round, but may not. Show that if the Adv were forced to delete a message in a round then consensus is solvable, i.e. give an algorithm.
   Now the harder part: Show that if the Adv is forced just to eventually delete a message then consensus is still solvable. Your algorithm unlike previously where it "know" that one message will be deleted in the first round, now all the algorithm "knows" is that at some future unknown round a message will be deleted.

2. We defined the wait-free adversary in the round by round model as deleting at most one message between a pair. For more than 2 processor model, suppose we generalize: The Adv can delete even the 2 messages between a pair as long as the directed graph induced by the messages left is such that for all p_i and p_j, either p_i can reach p_j or vice versa, or both. Obviously this is a Generalization. Thus the Former can trivially solve the Latter (as a task). Show that the Latter solves the Former.
   (To elaborate: the Former as a task is input to a processor is its id, each p_i returns a set S_i of processors such that for all p_j either $p_j \in S_i$ or $p_i \in S_j$, or both.
   The Latter is for each p_i and p_j there is a sequence $p_i, p_{i_1}, \ldots, p_{i_k}, p_j$ such that in the sequence if p_l is succeeded by p_m then $p_l \in S_m$, or the reverse, or both.)

3. (Bonus question - very tough, for those who like combinatorics) In the round by round adversary that in each round the Adv deletes at most a single message between a pair. Consider two successive rounds. Prove that there exist a "king" processor - a processor that transitively heard about all other processors. I.e. for p_i it either got a message from p_i in the first round. If not, then it either received a message from p_i in the second round, or it received a message in the second round from p_j who in the first round received a message from p_i.
   A much easier problem is to show that if the Adv behavior in the second round is the same as in the first round, then the above is true.