Homework Exercise 6
Due: November 12, 2003

6. (a) In class, we studied a proof that any message-passing n-process consensus algorithm that tolerates f halting failures must use at least $f + 1$ rounds in some execution. The proof uses a chain of executions, starting with a failure-free execution where every input is 0 and ending with a failure-free execution where every input is 1. Each pair of consecutive executions in the chain are indistinguishable to some correct process. If you actually wrote down an explicit list of all the executions in this chain, how many would there be (as a function of f and n)? Use big-O notation to state your answer (i.e. don't worry too much about constant factors). Explain why your answer is correct.

(b) Let $k \geq 1$. In the *k-set consensus* problem, each process starts with an integer input value and eventually terminates and outputs an integer value. The following two conditions must be satisfied:
Validity: Every output value is the input value of some process.
k**-Agreement**: The number of different output values is at most k.
Note that 1-set agreement is just ordinary consensus.

The following synchronous algorithm is a generalization of the one given in class for consensus (notice that the two are very similar when $k = 1$).

Code for process j:

$V_j \leftarrow$ {process j's input value} % Stores set of values j has seen so far
for $i \leftarrow 1..\left\lfloor \frac{f}{k} \right\rfloor + 1$
    send a message containing V_j to all other processes
    insert into V_j all values received in the messages from other processes
end for
output minimum element of V_j.

Show that this algorithm solves k-set consensus in a synchronous message-passing system where up to f halting failures can occur ($f < n$).

Hint: Let $M_i = \{\min(V_{ji}) : V_{ji}$ is the value of V_j at end of round $i\}$. Show $M_i \subseteq M_{i-1}$. How big can M_i be if f_i processes fail during round i?