Rabin's algorithm for Byzantine Generals Problem.
Notes by Sanjeev Arora, Fall 1995 (updated Fall 2005)

Byzantine Generals Problem: There are $n = 3t + 1$ processors and at least $2t + 1$ of them are working correctly. (The remaining t could be arbitrarily misbehaved or faulty.) Initially, processor i holds a bit $b_i \in \{0, 1\}$. The goal is for them to exchange messages and agree upon a bit b_{final}. If all b_i for the good processors were the same, then we require b_{final} to be that same bit. Otherwise there is no hard requirement on what b_{final} should be so long as all good processors agree what it is.

Sometimes this problem is also called *Byzantine Agreement.* It can be solved with a deterministic algorithm in poly(n) steps. It is known that no distributed algorithm exists if the number of faulty processors exceeds $n/3$.

Now we describe a simple randomized algorithm due to Rabin which assumes that there is a global *random coin* that is tossed at each step, and which is visible to all processors.

The processors maintain at all times a bit, vote, which is initially b_i for processor i. At the start of each round each processor sends its vote value to every other processor. Each processor examines all $3t + 1$ values of vote it receives (including its own). It identifies maj = the majority bit among these values of vote, and $tally$ = the number of times it saw this bit among the vote values. Notice, the values of maj and tally can vary widely among the processors, since a faulty/malicious processors could try to confuse things by sending different values of vote to different processors.

Each processor now applies the following algorithm at each step.

`Do the following depending upon the value of tally:`

1. `tally` $\geq 2t + 1$: `Set vote` $=$ `maj.`

2. `tally` $\leq 2t$: `Look at global-coin-toss. If it is ‘‘Heads," then set vote` $= 1$, `else set vote` $= 0$.

Proof of Correctness: Note that if all the good processors have the same initial value, then they all set their votes to this value in the first round. In all other cases, we show that with probability at least 1/2, all the processors assign the same value to vote. (Note that as soon as this happens, then from then on, tally $\geq 2t + 1$ for all processors, and so all processors will continue executing line (1) in the algorithm.)

There are two cases. (i) Some processor sees tally $\geq 2t + 1$, and maj $= b$ for some $b \in \{0, 1\}$. Since only t processors are faulty, we conclude that at least $t + 1$ good processors must have sent b as their value of vote. Thus no other processor will see both tally $\geq 2t + 1$ and maj $= 1 - b$ in the same round. Hence regardless of whether the other processors execute step (1) or (2) in the above algorithm, the probability is at least 1/2 that they will all set vote to b.

(ii) No good processor sees tally $\geq 2t + 1$. Then all of them execute step (2), and with probability 1 set vote to the same value.

It is a homework exercise to modify the algorithm so that processors can *detect* when the algorithm has succeeded (namely, all good processors have the same value of vote).