CS 174, Spring 1998
Solution for Midterm #2

Problem #1

(A)

Here we are looking for exactly two of something, so use the general inclusion exclusion formula:

Sum_{i=0}^{i=floor(k/2)} (2+i)C(i) * (k-2-i)C(2+i) * 5^(k-2-2i)

(B)

In this problem count the case AAA seperately... note the rest of the string must not contain AA

Then add on the case with AA and AA delimited by non A characters

Problem #2

The checker should pick a random vector v, then for the supposed inverse, B, compute A(Bv) and compare this with v. If they are different, then B is not A inverse, otherwise there is less than 50% chance it is not the inverse. This must be repeated at least n times with linerly independent v's to be sure that B is the inverse. Each repetition requires two matrix, vector multiplies, or O(n^2) operations.

Problem #3

When appending T after S, consider the changes to the first level of nodes above the actual string. Only the last above S, and the first above T will change. This is a constant number of changes which will be propogated up the fingerprint tree. So we have a constant number of changes at each level, and the number of levels is bounded by log_2 n, and therefore O(log n) new nodes, and O(log n) time to append.

Problem #4

(A) No. Once one of the good processors has tally >= G then on the next round all the good processors will have tally >= G, and have all set their votes permanently.

(B) A processor can halt one round after it has set its vote permanently.