

HOMWORK #1

Do 50 points of the following problems (due 1/15/03).

- 15 pts. **1** The Chemical Abstract Service uses the following scheme for registry numbers: $a_1a_2\dots a_7$ are the identification digits, and the check digit is $a_8 = 7 \cdot a_1 + 6 \cdot a_2 + 5 \cdot a_3 + 4 \cdot a_4 + 3 \cdot a_5 + 2 \cdot a_6 + 1 \cdot a_7 \pmod{10}$. What mistyped digits will not be detected? What about transpositions?

This scheme will pick up any transposition. Suppose a_i and a_{i+1} are transposed: then the contribution to the sum will be $ia_{i+1} + (i+1)a_i$ instead of $ia_i + (i+1)a_{i+1}$. If we suppose they are equal, that implies that $ia_{i+1} + (i+1)a_i = ia_i + (i+1)a_{i+1}$, or $a_i = a_{i+1}$. However, that really means that there was not a transposition error (which only occurs when consecutive numbers are not equal).

For mistyped digits, any even term will mistake a mistype of magnitude 5. For example, in the second position the code will miss a 0 switching with a 5, a 1 with a 6, etc. In the third spot, the scheme will miss any mistake of even magnitude. This is because 5 times any even number is 0 mod 10 and 5 times any odd number is 5 mod 10. All other errors will be picked up.

- 15 pts. **2** Generalize problem 1.5, p.10, to a 3-ary $(k, M, 2)$ -code (do parts (i) and (ii)).

$A_3(k, 2) = 3^{k-1}$. We first argue that $A_3(k, 2) \leq 3^{k-1}$. Suppose we have a code of length k whose minimum distance is 2. If we puncture the last component, the remaining code must have a minimum distance of 1 implying that all these codewords of length $k-1$ are distinct. Since there are 3^{k-1} distinct codewords of that length, that shows the upper bound. In order to construct a code with that many codewords, take the list of all 3-ary codewords of length $k-1$. Add a parity check bit on the end (make the sum of all k digits add to 0 mod 3). If the first $k-1$ bits are only different in one place, then those codewords must have a different parity check bit implying a distance of 2. If the first $k-1$ bits are different in 2 or more places, then we don't need to worry about the parity check bit because the codewords are already at least

2 apart. Thus, we have constructed a code with 3^{k-1} codewords and minimum distance of 2.

- 30 pts., **3** If you have 16 messages to send, and you want to be able to correct errors, one way to do that is to use a binary (12,16,3) repetition code. This code is created by numbering the 16 messages in binary, then repeating them 3 times while sending. Discuss the error correcting capabilities of this system (when can it correct 2 errors, 3 errors, ...)? Explain how you would do your error correcting. If each digit has a probability of .99 of being received properly and we assume that errors in digits are independent events, what is the probability that the word is received correctly?

We discussed the error correcting capabilities in class: this code can correct up to 4 errors (if they are in the right places). We also explained how to do the error correcting. The probability of correctly decoding the message is $(.99)^{12} + 12(.99)^{11}(.01) + 12(9)(.99)^{10}(.01)^2/2 + 12(9)(6)(.99)^9(.01)^3/6 + 12(9)(6)(3)(.99)^8(.01)^4/24 = .9988078$.

- 20 pts. **4** Let C be the binary code of length 6 made up of all of the codewords of even weight. What is M? What is d? What received words can be decoded unambiguously? What received words will give problems? What properties will this code have if it is repeated twice? How would you correct errors?

$M = 32$: to show this, this code is equivalent to the code of all codewords of length 5 with a parity check bit added at the end. $d = 2$: suppose two codewords had distance 1. Then one of the codewords would have an odd weight, but that is not possible in the code of even weight codewords. Therefore, the minimum distance is at least 2, and 000000 and 000011 demonstrate that there are codewords distance 2 apart. Only even weight codewords can be decoded unambiguously. Any odd weight codeword will be distance 1 from 6 even weight codewords. If the code is repeated twice, then it will have a minimum distance of 4 and will be able to correct errors. If a single error is made, then one 6-tuple will have even weight and the other (the one with the error) will have odd weight: the message is the even weight 6-tuple.