Ma 450: Mathematics for Multimedia

Solution: to Homework Assignment 6

Prof. Wickerhauser

Due Sunday, April 30th, 2023

1. Fix an integer $q \geq 2$, let $N = 2^q \geq 4$ and consider a graph with vertices labeled $0, 1, \ldots, N-1$. Suppose that vertex i is connected by an edge to vertex j if and only if the base-two expansions for i and j differ by exacly two bitflips. Compute the total number of edges.

   Solution: The number of ways to choose two distinct bits to flip from a q-bit string is

   $$\binom{q}{2} = \frac{q(q-1)}{2},$$

   so that is the number of edges that start from each vertex (also called the *degree* of the vertex). To count total edges, multiply the degree of each vertex by the number of vertices, then divide by two since each edge will be counted twice (once for each of its endpoint vertices). This gives $\frac{q(q-1)}{2}\frac{N}{2} = q(q-1)2^{q-2}$ total edges. □

2. Construct a prefix code for the alphabet A={a,b,c,d,e,f} with codeword lengths 1,3,3,3,4,4 or prove that none exists.

   Solution: One exists. Compute $2^{-1} + 3 * 2^{-3} + 2 * 2^{-4} = 1$ and apply Lemma 6.1.

   Many solutions exist, of which the following is an example:

a	1
b	001
c	010
d	011
e	0000
f	0001

□

3. Construct a prefix code for the 24-letter Greek alphabet $A = \{\alpha, \beta, \gamma, \ldots, \omega\}$ with longest codeword 4, or prove that none exists.

   Solution: None exists, by Lemma 6.1, since even if all codewords had length 4 bits the total number of letters the code could represent would be $2^4 = 16 < 24$. □

4. Suppose we have two prefix codes, $\mathbf{c}_0(a,b) = (1,0)$ and $\mathbf{c}_1(a,b) = (0,1)$, for the alphabet $A = \{a,b\}$. Show that the following *dynamic encoding* is uniquely decipherable by finding a decoding algorithm:

Simple Dynamic Encoding Example

```
dynamicencoding0( msg[], M ):
[0] Initialize n=0
[1] For m=1 to M, do [2] to [3]
[2]    Transmit msg[m] using code n
[3]    If msg[m]==‘b’, then toggle n = 1-n
```

(This encoding is called dynamic because the codeword for a letter might change as a message is encoded, in contrast with the *static encodings* studied in this chapter. It gives an example of a uniquely decipherable and instantaneous code which is nevertheless not a prefix code.)

Solution: The receiver must keep track of the decoded message and alter the decoding every time a ‘b’ is encountered:

Simple Dynamic Decoding Example

```
dynamicdecoding0( bit[], L ):
[ 0] Initialize n=0 and k=1
[ 1] While k<=L, do [2] to [10]
[ 2]    If n==0, then do [3] to [4]
[ 3]       If bit[k]==1, then let OUT=‘a’
[ 4]       Else let OUT = ‘b’
[ 5]    Else do [6] to [7]
[ 6]       If bit[k]==0, then let OUT = ‘a’
[ 7]       Else let OUT = ‘b’
[ 8]    Increment k += 1
[ 9]    If OUT==‘b’, then toggle n = 1-n
[10]    Print OUT
```

□

5. A k-ary tree is called *extended* if every *interior*, or non-leaf, vertex has all k children. Let N_d be the number of extended k-ary trees of depth d or less.

(a) Find, with proof, a recursive formula for N_{d+1} in terms of N_d.

(b) Compute N_3.

Solution: (a) First note that $N_0 = 1$, since the only depth=0 extended k-ary tree is the one consisting of just the root. Also $N_1 = 2$, with the possibilities being the childless root or the root with all k children.

Now observe that any extended k-ary tree of depth at most $d+1$ consists either of just a childless root or else a root with k nonempty extended k-ary subtrees of depth at most d. These k subtrees may be chosen independently from the N_d possibilities (the empty subtree is not counted in N_d), so the following recursion relation holds:

$$N_{d+1} = 1 + N_d^k$$

(b) Apply part (a) to compute $N_2 = 1 + 2^k$ and thus $N_3 = 1 + (1 + 2^k)^k$. □

6. Fix a positive integer n and consider the alphabet $A = \{a_1, \dots, a_n, a_{n+1}\}$ with occurrence probabilities $p(a_i) = 2^{-i}$ for $i = 1, \dots, n$, and $p(a_{n+1}) = 2^{-n}$.

   (a) Construct a Huffman code for the alphabet and compare its bit rate with $H(p)$.

   (b) Construct a canonical Huffman code for this alphabet, with the property that no letter has a codeword consisting of just 1-bits. Compute its bit rate.

   Solution:

   (a) One Huffman code for this combination of alphabet and occurrence probabilities is $\mathbf{c}(a_1, a_2, a_3, \dots, a_n, a_{n+1}) = (0, 10, 110, \dots, 1\cdots 10, 1\cdots 11)$, having corresponding codeword lengths list $n = (1, 2, 3, \dots, n, n)$. The entropy lower bound on the bitrate is

$$H(p) = \sum_{x\in A} p(x)\log_2 \frac{1}{p(x)} = \left(\sum_{i=1}^{n} 2^{-i} \times i\right) + 2^{-n} \times n.$$

   The Huffman code's bit rate is

$$\sum_{x\in A} p(x)n(x) = \left(\sum_{i=1}^{n} 2^{-i} \times i\right) + 2^{-n} \times n.$$

   These are evidently the same, so Huffman coding achieves the optimal bit rate for this family of occurrence probabilities.

   (b) In this case, adding an extra letter b with occurrence probability 0 deepens the tree by one level. One canonical Huffman code for this appended alphabet and occurrence probabilities is $\mathbf{c}(a_1, a_2, a_3, \dots, a_n, a_{n+1}, b) = (0, 10, 110, \dots, 1\cdots 10, 1\cdots 110, 1\cdots 111)$, having description $L = n + 1$, $M = (1, 1, 1, \dots, 1)$, and $A' = a_1, a_2, a_3, \dots, a_n, a_{n+1}$. The canonical Huffman code's bit rate is

$$R = \sum_{x\in A'} p(x)n(x) = \left(\sum_{i=1}^{n} 2^{-i} \times i\right) + 2^{-n} \times (n+1).$$

   Without the extra inactive letter, the Huffman code would have space for two active codewords at level n, and its bitrate would be

$$R_0 = \sum_{x\in A} p(x)n(x) = \left(\sum_{i=1}^{n} 2^{-i} \times i\right) + 2^{-n} \times n.$$

   The difference in efficiency is $R - R_0 = 2^{-n}$ bits per character, so canonical Huffman coding very little extra for this family of occurrence probabilities as long as n is reasonably large. □

7. (a) Find a binary code with five 13-bit or shorter codewords, wherein restoration to the nearest codeword corrects any three or fewer bit flips.

   (b) Can part (a) be solved with 12-bit codewords?

   (c) How many 13-bit codewords can you find satisfying the three-bitflip correction condition?

   Solution: Use Gilbert's ideas from Theorem 6.12 and the algorithm from Solution 12.0 as implemented on the class website.

   NOTE: the textbook implementation is wrong, in that it removes the Hamming ball of radius $2r + 1$ rather than the ball of radius $2r$ as in the proof of the theorem. A corrected version is below:

Repaired Gilbert's Algorithm for an Error Correcting Code

```
gilbertcode( bits, words, corrects ):
[0] Let N = 1<<bits, let radius = 2*corrects
[1] For n=0 to N-1, allocate tag[n] with tag[n]=LIVE
[2] Allocate c[1],...,c[words] and let c[1] = 0
[3] For j=1 to words-1, do [4] to [7]
[4]   For n=c[j]+1 to N-1 do [5]
[5]     If dist(n,c[j])<=radius, then let tag[n] = DEAD
[6]   For n=c[j]+1 to N-1 do [7]
[7]     If tag[n]==LIVE, then let c[j+1]=n and goto [3]
[8] For j=1 to words, print c[j]
```

This is implemented in the file `gilbertcode.m` on the class website.

The call `gilbertcode(13,5,3)` starts with codeword $\mathbf{c}(1) = 000\,00000\,00000$ and eliminates Hamming spheres of radius $6 = 2 * 3$ from the 13-dimensional unit hypercube, then searches for the first survivor $\mathbf{c}(2) = 000\,00011\,11111$. It then repeats the elimination and survivor location steps three more times to find the following codewords:

```
>> gilbertcode(13,5,3)
ans =
0000000000000
0000001111111
0011110000111
0011111111000
1100110011001
```

(b) The following shows that 13 codeword bits are the minimum needed for five codewords:

```
>> gilbertcode(12,5,3)
ans =
00000000000
00001111111
11110000111
11111111000
00000000000
```

The output is a list with the last codeword being a repeat of the first, indicating that the survivor search was unsuccessful after four codewords.

(c) The following, using Gilbert's method, shows that there are eight codewords with three-bitflip correction, but not nine.

```
>> gilbertcode(13,9,3)
ans =
0000000000000
0000001111111
0011110000111
0011111111000
1100110011001
```

```
1100111100110
1111000011110
1111001100001
0000000000000
```

□

8. (a) Prove that casting out seventeens will detect all one-digit errors in hexadecimal arithmetic.

   (b) Find an example one-hexadecimal-digit error undetected by casting out fifteens.

Solution: (a) If x and y differ at exactly one hexadecimal digit, then $x - y = \pm d \times 16^k$ for some integers $k \geq 0$ and $d \in \{1, 2, \ldots, 9, A, B, C, D, E, F\}$. But $c_{17}(x) = c_{17}(y)$ if and only if $x - y = 0 \pmod{17}$, which requires $d = 0 \pmod{17}$ since $16^k \neq 0 \pmod{17}$ for any $k \geq 0$. But no d in the stated range satisfies that congruence.

(b) Note that $c_{15}(110 \text{ (base 16)}) = c_{15}(11F \text{ (base 16)})$ is a one-hexadecimal-digit difference undetected by casting out fifteens. In general, changing any 0 digit into F will not change c_{15}, since the difference will be divisible by $15 = F$ (base 16). Neither will transposing two digits, though transposing unequal adjacent digits will change the value of c_{17}. □

9. Find a mod-2 polynomial of degree 4 that is relatively prime to $p(t) = t^7 + t^6 + t^3 + t$. (Hint: use Euclid's algorithm for mod-2 polynomials.)

Solution: Following the hint, use Euclid's algorithm for mod-2 polynomials. Write $p(2) = 202 = 11001010$ (base 2) as an integer, and consider all mod-2 polynomials q whose corresponding integers $q(2)$ are in the range $16 = 10000$ (base 2) to $31 = 11111$ (base 2). There are 16 of these in all.

Find Relatively Prime Mod-2 Polynomials of Given Degree

```
intmod2polyrelativeprime( p, d ):
[0] Initialize num = 0
[1] Let dp = intmod2polydegree(p)
[2] If dp > 0 then do [3] to [8]
[3]   Let qmin = (1<<d)
[4]   Let qmax = 2*qmin - 1
[5]   For q=qmin to qmax do [6] to [8]
[6]     If intmod2polygcd(p,q)==1 then do [7] to [8]
[7]       Print q
[8]       Increment num += 1
[9] Return num
```

Solve this with the class website Octave implementation in `intmod2poly.txt` as follows:

```
p=sum(2.^[7 6 3 1]) % pointwise exponentiation, summed gives p=202
intmod2polyrelativeprime(p,4) % 19 25 31
dec2bin(intmod2polyrelativeprime(p,4)) % 10011  11001  11111
```

Thus there are three mod-2 polynomials of degree 4 that are relatively prime to p. They are $q_1(t) = t^4 + t + 1$, $q_2(t) = t^4 + t^3 + 1$, and $q_3(t) = t^4 + t^3 + t^2 + t + 1$, corresponding to $q_1(2) = 19 = 10011$ (base 2), $q_2(2) = 25 = 11001$ (base 2), and $q_3(2) = 31 = 11111$ (base 2). □

10. Suppose that b is a prime number. Write $b = \ldots b_2 b_1 b_0$ (base 2) and let $p(t) = b_0 + b_1 t + b_2 t^2 + \cdots$ be the associated mod-2 polynomial. Prove or find a counterexample to the claim that p must be irreducible.

Solution: It is tempting to believe that p must be irreducible, since any factorization $p(t) = q(t)r(t)$ seems to give a factorization $b = p(2) = q(2)r(2)$. However, the multiplication in these factorizations is not the same as the one used by integers.

A simple counterexample is $b = 5 = 101$ (base 2), which corresponds to the mod-2 polynomial $p(t) = t^2 + 1 = (t+1)(t+1)$. This p is evidently not irreducible.

The converse of the claim is also false. The footnote on page 214 of the text gives the counterexample of an irreducible mod-2 polynomial S corresponding to a non-prime integer $S(2)$. □

11. Find integers j, k, $0 < k < j < 32$, such that $s(t) = t^{32} + t^j + t^k + 1$ is an irreducible mod-2 polynomial, or prove that none exists. (Hint: try dividing one such $s(t)$ by $t + 1$.)

Solution: This may be done by an exhaustive search through the $\binom{31}{2} = 31 \times 30/2 = 465$ possibilities. However, there is a simple proof based on the hint. Note that

$$s(t) = t^{32} + t^j + t^k + 1 = (t^{32} + t^j) + (t^k + 1) = t^j(t^{32-j} + 1) + (t^k + 1).$$

But any mod-2 polynomial of the form $t^n + 1$ with $n > 0$ is divisible by $t + 1$:

$$t^n + 1 = (t+1)(t^{n-1} + t^{n-2} + \cdots + t + 1).$$

Hence both $t^j(t^{32-j} + 1)$ and $t^k + 1$ are divisible by $t + 1$, so $s(t)$ must be divisible by $t + 1$.

Thus, no four-term mod-2 polynomial of degree 32 and not divisible by t is irreducible. Any that are divisible by t are likewise not irreducible, so in fact no four-term mod-2 polynomials of degree 32 are irreducible.

Nothing in this proof requires the degree to be 32, only that it is at least 3 to leave room for four terms. It is easy to generalize the result as follows: If $K > 1$ is an even integer, then any K-term mod-2 polynomial of degree at least $K - 1$ must be divisible by $t + 1$. □