1. Fix an integer \( q > 0 \), let \( N = 2^q > 1 \) 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 exactly one bitflip. Compute the total number of edges.

**Solution:** This graph can be represented by the vertices and edges of the unit cube in \( \mathbb{R}^q \). That cube has \( q \) edges at each vertex, for a total of \( qN/2 = q2^{q-1} \) edges.

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

**Solution:** None exists. Compute \( 2^{-1} + 2 * 2^{-2} + 3 * 2^{-3} > 1 \) and apply Lemma 6.1.

3. Construct a prefix code for the 24-letter Greek alphabet \( A = \{\alpha, \beta, \gamma, \ldots, \omega\} \) with longest codeword 5, or prove that none exists.

**Solution:** A fixed-length code with 5 bits per codeword will work as it can represent \( 2^5 = 32 \geq 24 \) letters.

4. Suppose we have two prefix codes, \( c_0(a, b) = (1, 0) \) and \( c_1(a, b) = (0, 1) \), 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
[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
[ 3] If bit[k]==1, then let OUT='a'
[ 4] Else let OUT = 'b'
[ 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. Suppose that $A$ is a finite set, $s$ is a fixed positive integer, and $p_k : A \rightarrow [0,1]$ is a probability function on $A$ for each $k = 1,\ldots,s$.

(a) Show that the function $p : A^s \rightarrow [0,1]$ defined by

$$p(x_1, x_2, \ldots, x_s) \overset{\text{def}}{=} p_1(x_1)p_2(x_2)\cdots p_s(x_s)$$

is a probability function on $A^s$.

(b) Compute the entropy $H(p)$ in terms of $H(p_1),\ldots,H(p_s)$.

Solution: For convenience, write $x = (x_1, x_2, \ldots, x_s) \in A^s$ for an arbitrary $s$-tuple of elements of $A$.

(a) Since $p(x)$ is a product of numbers in $[0,1]$, it is itself in $[0,1]$. Compute

$$\sum_{x \in A^s} p(x) = \sum_{x \in A^s} p(x_1, x_2, \ldots, x_s) = \sum_{x \in A^s} p_1(x_1)p_2(x_2)\cdots p_s(x_s)$$

$$= \sum_{x_1 \in A} \sum_{x_2 \in A} \cdots \sum_{x_s \in A} p_1(x_1)p_2(x_2)\cdots p_s(x_s)$$

$$= \left( \sum_{x_1 \in A} p_1(x_1) \right) \left( \sum_{x_2 \in A} p_2(x_2) \right)\cdots \left( \sum_{x_s \in A} p_s(x_s) \right) = 1.$$

Hence $p$ has the right normalization to be a probability function.

(b) From part (a), $p$ is a probability function. To compute its entropy, use Equation 6.4:

$$H(p) = \sum_{x \in A^s} p(x_1, x_2, \ldots, x_s) \log_2 \frac{1}{p(x_1, x_2, \ldots, x_s)}$$
\[
\sum_{x \in A^s} p_1(x_1) p_2(x_2) \cdots p_s(x_s) \left( \sum_{i=1}^{s} \log_2 \frac{1}{p_i(x_i)} \right)
\]

\[
= \sum_{i=1}^{s} \left[ \left( \sum_{x_i \in A} p_i(x_i) \log_2 \frac{1}{p_i(x_i)} \right) \prod_{j \neq i} \left( \sum_{x_j \in A} p_j(x_j) \right) \right],
\]

after interchanging the order of summation and then extracting one factor \( p(x_i) \) from the \( i \)th new summand to combine with its corresponding logarithm. But each \( x_i \) is an independent random variable with occurrence probabilities \( p_i \), so for all \( k = 1, \ldots, s \),

\[
\sum_{x_k \in A} p_k(x_k) = 1 \quad \text{and} \quad \sum_{x_k \in A} p_k(x_k) \log_2 \frac{1}{p_k(x_k)} = H(p_k).
\]

Conclude that \( H(p) = \sum_{i=1}^{s} H(p_i) \). \( \square \)

6. A \( k \)-ary tree is called extended if every interior, or non-leaf, vertex has all \( k \) children. Count, with proof, the extended \( k \)-ary trees of depth 3 or less.

**Solution:** Let \( N_d \) be the number of extended \( k \)-ary trees of depth \( d \) or less. Then \( N_0 = 1 \), 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:

\[
N_{d+1} = 1 + N_d^k
\]

Thus \( N_2 = 1 + 2^k \) and \( N_3 = 1 + (1 + 2^k)^k \). \( \square \)

7. Fix a positive integer \( n \) and consider the alphabet \( A = \{ a_1, \ldots, a_n, a_{n+1} \} \) with occurrence probabilities \( p(a_i) = 2^{-i} \) for \( i = 1, \ldots, n \), 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 \( c(a_1, a_2, a_3, \ldots, a_n, a_{n+1}) = (0, 10, 110, \ldots, 1 \cdots 10, 1 \cdots 11) \), having corresponding codeword lengths list \( n = (1, 2, 3, \ldots, n, n) \). 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 \( c(a_1, a_2, a_3, \ldots, a_n, a_{n+1}, b) = (0, 10, 110, \ldots, 1 \cdot 10, 1 \cdot 110, 1 \cdot 111) \), having description \( L = n + 1 \), \( M = (1, 1, 1, \ldots, 1) \), and \( A' = a_1, a_2, a_3, \ldots, a_n, a_{n+1} \). 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 costs vanishingly little extra for this family of occurrence probabilities as long as \( n \) is reasonably large.

8. What is the probability of an undetected error in 8 data bits in a \( 2 \times 2 \times 2 \) array with crossed parity checks if the data bits each have an independent probability \( p \) of being flipped, but the 12 top, front, and left face parity bits are known to be correct?

Solution: If any data bit is flipped, it will cause wrong parity in its row unless the other data bit in that row is also flipped. But then the two columns will have bad parity unless the remaining two data bits are also flipped. Hence an undetected error is only possible if all eight data bits are flipped, which by independence occurs with probability \( p^8 \).

9. Find a binary code with two 10-bit or shorter codewords, wherein restoration to the nearest codeword corrects any three or fewer bit flips.

Solution: We use Gilbert’s ideas from Theorem 6.12, and the implementation from Solution 12.0.

To generate the code, call \texttt{gilbertcode}(10, 2, 3). This starts with codeword \( c(0) = 00000 00000 \) and eliminates Hamming spheres of radius \( 7 = 2 \times 3 + 1 \) from the ten-dimensional unit hypercube. It continues until it has found one additional survivor \( c(1) = 0011111111 \). Note that only eight codeword bits are actually needed, as the ninth and tenth bits are constant over the code.
10. Prove that casting out seventeens will detect all one-digit errors in hexadecimal arithmetic. Find an example one-hexadecimal-digit error undetected by casting out fifteens.

**Solution:** 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.

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 \( 15 = F \pmod{16} \). Neither will transposing two digits, though transposing unequal adjacent digits will change the value of \( c_{17} \).

11. Will the combination of checksums \( c_9 \) and \( c_{11} \) distinguish all nonequal 2-decimal-digit positive integers?

**Solution:** Yes. We will have \( c_9(x) = c_9(y) \) and \( c_{11}(x) = c_{11}(y) \) if and only if \( x - y \) is divisible by both 9 and 11, which will happen if and only if 99 divides \( x - y \). This cannot happen if \( 1 \leq x, y \leq 99 \).

12. Find a mod-2 polynomial of degree 3 that is relatively prime to \( p(t) = t^7 + t^5 + t^3 + t \). (Hint: use Euclid’s algorithm for mod-2 polynomials.)

**Solution:** Following the hint, use Euclid’s algorithm for mod-2 polynomials. Write \( p(2) = 170 = 10101010 \text{ (base 2)} \) as an integer, and consider all mod-2 polynomials \( p \) whose corresponding integers \( p(2) \) are in the range \( 8 = 1000 \text{ (base 2)} \) to \( 15 = 1111 \text{ (base 2)} \). There are eight of these in all.

Find Relatively Prime Mod-2 Polynomials of Given Degree

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

Given \( p = p(2) = 170 \), this function prints the two mod-2 polynomials of degree 3 that are relatively prime to \( p(t) = t^7 + t^5 + t^3 + t \). They are \( q_1(t) = t^3 + t + 1 \) and \( q_2(t) = t^3 + t^2 + 1 \), corresponding to \( q_1(2) = 11 = 1011 \text{ (base 2)} \) and \( q_2(2) = 13 = 1101 \text{ (base 2)} \).
13. 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 $S(2)$.

14. Suppose that $s > 0$ and $a > 1$ are integers with gcd$(a, s) = 1$. Prove that there is some integer $N > 0$ such that $a^N - 1$ is divisible by $s$, but $s$ does not divide $a^k - 1$ for any positive integer $k < N$. (This is Theorem 6.19 for integers rather than mod-2 polynomials.)

**Solution:** Rewrite the proof of Theorem 6.19.

Consider the set $\{a^k \% s : k = 0, 1, \ldots\}$, which consists of representative nonnegative integers less than $s$. There are only finitely many of those, but since $a > 1$ there are infinitely many integers in the set $\{a^k : k = 0, 1, 2, \ldots\}$. Thus there must be integers $p > q \geq 0$ such that $a^p = a^q \pmod{s}$. Since gcd$(a, s) > 1$, Lemma 1.3 implies that $s$ cannot divide $a^k$ for any $k$. Thus $a^q \neq 0 \pmod{s}$. But then $a^{p-q} = 1 \pmod{s}$, so $s$ divides $a^{p-q} - 1$. Note that $p - q$ is a positive integer.

Since the set of positive integers $\{n > 0 : a^n = 1 \pmod{s}\}$ is nonempty, it has a smallest element; denote it by $N$. By construction, $s$ does not divide $a^k + 1$ for any integer $0 < k < N$.

15. 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 $31 \times 30 = 930$ 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 $t + 1$. □