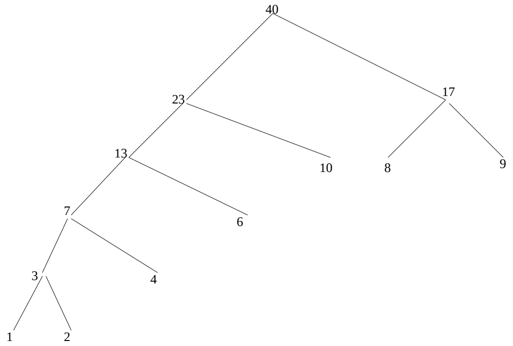
1. (Compression)

   (a) A Huffman tree for symbols with the frequencies $\{8, 4, 1, 2, 6, 9, 10\}$ is shown below.



   (b) Let f be a bag of frequencies in which x and y are the two smallest items. Applying a step of the Huffman algorithm, f will be transformed to g, where g is $f - \{x, y\} \cup \{x + y\}$. We have to show that the sum of the two smallest items in g is at least $x + y$.

   Suppose the two smallest items in g are u and v. If one of them, say u, is $x + y$, then their sum, $u + v$, i.e., $x + y + v$, is at least $x + y$. If neither u nor v is $x + y$, then they are both from $g - \{x + y\}$, i.e., $f - \{x, y\}$, and, hence, from f. Since x and y are the two smallest items in f, both u and v are at least as large as any of x and y; so, $u + v \geq x + y$.

2. (Error Correction)

   (a) A nonempty set of words, W, is *dependant* iff $\hat{W} = 0$, where $\hat{W}$ is the exclusive-or of all the words in W. Given that W' is $W \cup \{t\}$ or $W - \{t\}$, $\hat{W'} = \hat{W} \oplus t = t$. Therefore, if $t \in W'$, remove t from W' to get W and if $t \notin W'$, add t to W' to get W.

   (b) First, we compute the hamming distances between the codewords, see Table 1. Since the minimum distance is 3, we can detect 2 errors and correct 1 error.

   (c) We compute the distances of the received strings from the codewords, see Table 2. Since the minimum distance for 01010 is 1, we can correct the error and claim that 01110 was sent. For 10100, the minimum distance is 2, and we can only claim that there has been an error in the transmission.

	00000	01110	11001	10111
00000	0	3	3	4
01110	3	0	4	3
11001	3	4	0	3
10111	4	3	3	0

Table 1: Hamming distances between the codewords, Problem 2

	00000	01110	11001	10111
01010	2	1	3	4
10100	2	3	3	2

Table 2: Hamming distances between codewords and received word, Problem 2

(d) Proof is by induction on n.

$n = 0$: There is only one row and it consists of all 1s.

$n + 1$, $n \geq 0$: From the definition.

$$H_{n+1} = \begin{bmatrix} H_n & H_n \\ \\ H_n & \overline{H_n} \end{bmatrix}$$

The very top row of the matrix is composed of two copies of the first row from H_n, each of which is all 1s, from the induction hypothesis; so, it is all 1s. Each of the other rows in the top half of H_{n+1} is composed of two copies of the corresponding row from H_n. Again, from the induction hypothesis, each copy has equal number of 0s and 1s; so, their concatenation has the same property. The very top row of the lower half has all 1s in its left half and all 0s in its right half; so, it also has equal number of 0s and 1s. For the other rows in the bottom half of H_{n+1}, the same argument is applied as for the top half.

3. (Cryptography)

(a) (8 points) Each voter needs to send a number between 0 and 4. If these numbers are sent directly, there are 5 possible messages, and it is easy for an interceptor to decode even an encrypted message (he just constructs all 5 encoded strings using the public key of the trusted party). Therefore, we need a scheme where the number of possible messages is large, but each message can be decoded into one of five possible values using an algorithm. (The requirement that an algorithm has to be used eliminates the possibility of sending messages such as, “Bush is my man”.)

Let each voter send a number n in encrypted fashion, where $1 \leq n \leq M$, for a suitably large M. The candidate is $n \bmod 5$. The tally protocol is: decode the message to get n, compute $n \bmod 5$ and add 1 to the tally of this candidate.

(b) (6 points) Let the i^{th} plaintext block be p_i, and the encrypted block be b_i, $i > 0$. Let the secret key be denoted by b_0. Then,

$$b_i = p_i \oplus b_{i-1},\ i > 0$$

Then, for any $i > 0$, $b_i \oplus b_{i-1} = p_i \oplus b_{i-1} \oplus b_{i-1} = p_i$. All b_i except b_0 are available. Hence all blocks except b_1 can be decrypted.

(c) (8 points) Let the i^{th} plaintext block be p_i, and the encrypted block be b_i, $i > 0$. Let the secret key be denoted by p_0. Then,

$$b_i = p_i \oplus p_{i-1},\ i > 0$$

Each block is the exclusive-or of two p blocks. Taking exclusive-or of any two b_is yields a block which is exclusive-or of even number of p terms, because even number of terms (possibly 0) get cancelled by taking exclusive-or. Therefore, no single p term can ever be isolated.