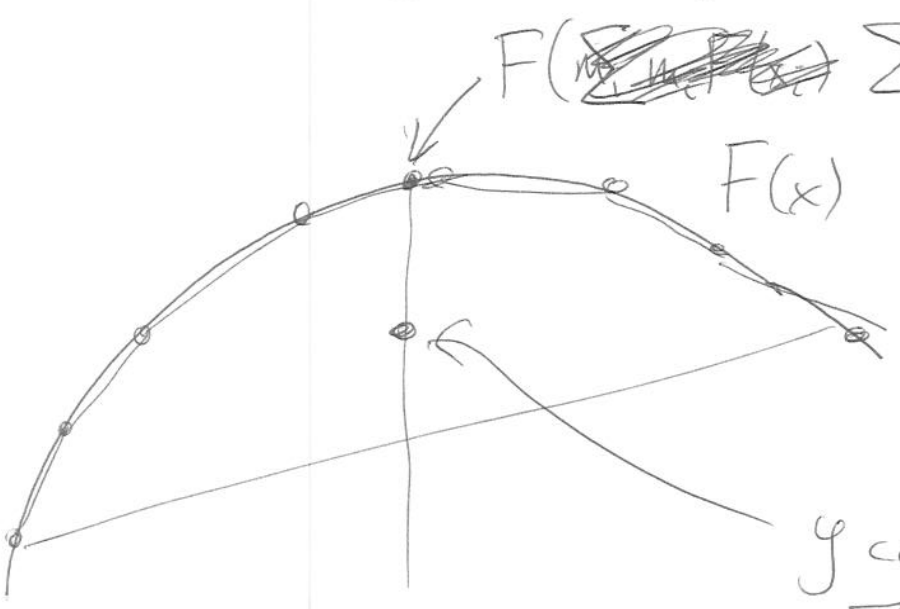
Out of possible 40 pts

Total out of 32
because it was long

Class average $\approx$ 49/100



$$\sum m_i F(x_i) \le F\left(\sum m_i x_i\right)$$

$$m_i = 1/n \qquad x_i = (i+1)^3 - i^3$$

$$\sum_{i=0}^{n-1} \frac{1}{n}\sqrt{(i+1)^3 - i^3} \le \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}\left((i+1)^3 - i^3\right)}$$

$$\frac{1}{n}\sum_{i=0}^{n-1}\sqrt{3i^2+3i+1} \le \sqrt{\frac{1}{n}\,n^3}$$

Entropy

Unicity distance

Trees

- tree from heights
- Huffman trees

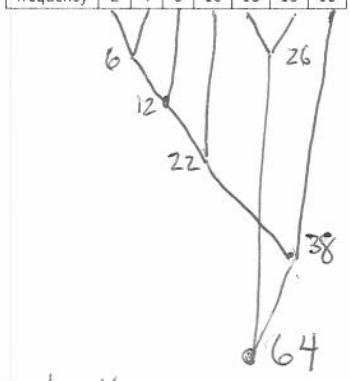
Perfect secrecy

$h_i = \lceil \log_2 1/p_i \rceil$ height at which we should place a letter at.

Huffman Code

Begin with a text file with the following frequencies

letter	A	B	C	D	E	F	G
frequency	2	4	6	10	13	13	16



Expected code length

$ECL = 5 \cdot 2 + 5 \cdot 4 + 4 \cdot 6 + 3 \cdot 10 + 2 \cdot 13 + 2 \cdot 13 + 2 \cdot 16$

$= 10 + 20 + 24 + 30 + 26 + 26 + 32$

$= 112 + 56 = 168$

Ebits per letter $= 168/64$

Huffman Code

A	B	C	D	E	F	G
2	4	6	10	13	13	16

(1) Say that you have a cryptosystem with two plaintext messages m_0 = "The British are coming" and m_1 = "The sky is falling" that each occur with probability 1/2. Also say that there are 4 keys which are equally likely k_0, k_1, k_2, k_3 which send the plaintext messages to one of the four cyphertexts

$$c_0 = \text{"cheese sandwiches."}$$
$$c_1 = \text{"milk and cookies"}$$
$$c_2 = \text{"mashed potatoes"}$$
$$c_3 = \text{"Ted Danson."}$$

Say that message m_i will be sent under key k_j to the cyphertext $c_{2i+j(mod4)}$.

(a) It is agreed in advance that today key that is being used is k_2. You receive the message "mashed potatoes." What plaintext does this represent?

(b) Does this system achieve perfect secrecy? Why or why not?

(c) Compute $H(K|C)$.

(d) Now assume that the 4 keys are not chosen with equal probability and instead k_0 and k_2 are chosen with probability 1/8 and k_1 and k_3 are chosen with probability 3/8. Does this system achieve perfect secrecy? Why or why not?

(e) Under this new system calculate $H(K|C)$.

(2) Calculate the unicity distance of the Vernam cipher with $p = 7$ and $q = 5$ (the lengths of the two keys). Use the table from the notes to estimate the entropy of English and assume that all keys are equally likely.

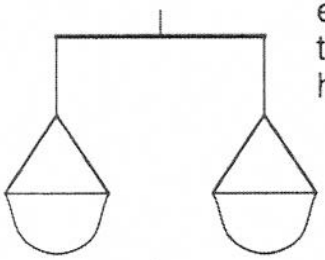
1. In the enciphering system MIX45 the message is first Vigenere encrypted with a 4-letter keyword, and then subjected to a rectangular transposition of period 5. Determine the unicity distance of MIX45. Assume all ciphers are equally likely.

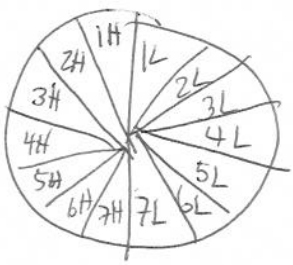
Suppose you are to write a program to simulate the output of a fortune wheel producing 1 2 3 4 5 6 with respective probabilities

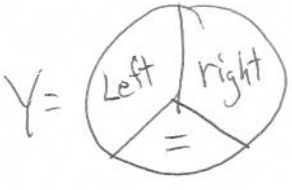
1/8 1/8 1/4 1/6 1/6 1/6

Suppose you have already written a random number generator yielding a random variable W uniformly distributed in [0,1] and that the only thing missing in your program is the procedure which converts W into one of the numbers 1 2 3 4 5 6. Draw the decision tree that carries out this conversion with the smallest expected number of comparisons.

A scale compares weights, testing if two objects weigh the same, the left is heavier, or the right is heavier. There are seven coins, each look the same but one of the seven is heavier or lighter than the others. Draw the decision tree that determines which is the heavier or lighter coin in a minimum number of weighings.







$H(Y) \leq \log_2 3 = 1.59$

$H(X) = \log_2(14)$
$= 3.8$

Expected Code Length

Theorem 2 *The best possible expected code length (bits per letter) is*

$$H = \sum_{i=1}^{n} p_i \log_2 1/p_i$$

Proof.

Letter frequenices $N_1, N_2, \ldots, N_k$ $(N = \sum_{i=1}^{k} N_i)$

Code lengths $h_1, h_2, \ldots, h_k$ (from a binary tree)

$p_i = N_i/N$ and $q_i = 1/2^{h_i}$

$$\begin{aligned}
\text{File length} &= \sum_{i=1}^{k} N_i h_i \\
&= \sum_{i=1}^{k} N_i \log_2 2^{h_i} \\
&= N \sum_{i=1}^{k} p_i \log_2 1/q_i \\
&\geq N \sum_{i=1}^{k} p_i \log_2 1/p_i = NH
\end{aligned}$$

For tree from heights

~~ECL~~

average bits per letter $= \sum_{i=1}^{n} p_i \cdot \lceil \log_2 1/p_i \rceil$

$\lceil \log_2 1/p_i \rceil = h_i$

if we use a complete tree then final count will have

average bits per letter $\leq \sum_{i=1}^{n} p_i \lceil \log_2 1/p_i \rceil$

$$\leq \sum_{i=1}^{n} p_i \left(\log_2 1/p_i + 1 \right)$$

$$\leq \sum_{i=1}^{n} p_i \log_2 1/p_i + \sum_{i=1}^{n} p_i$$

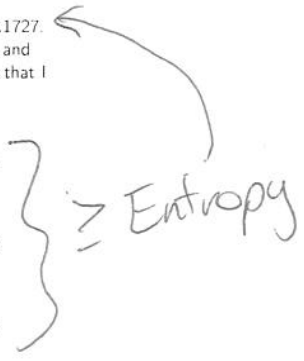
$$\leq \sum_{i=1}^{n} p_i \log_2 1/p_i + 1$$

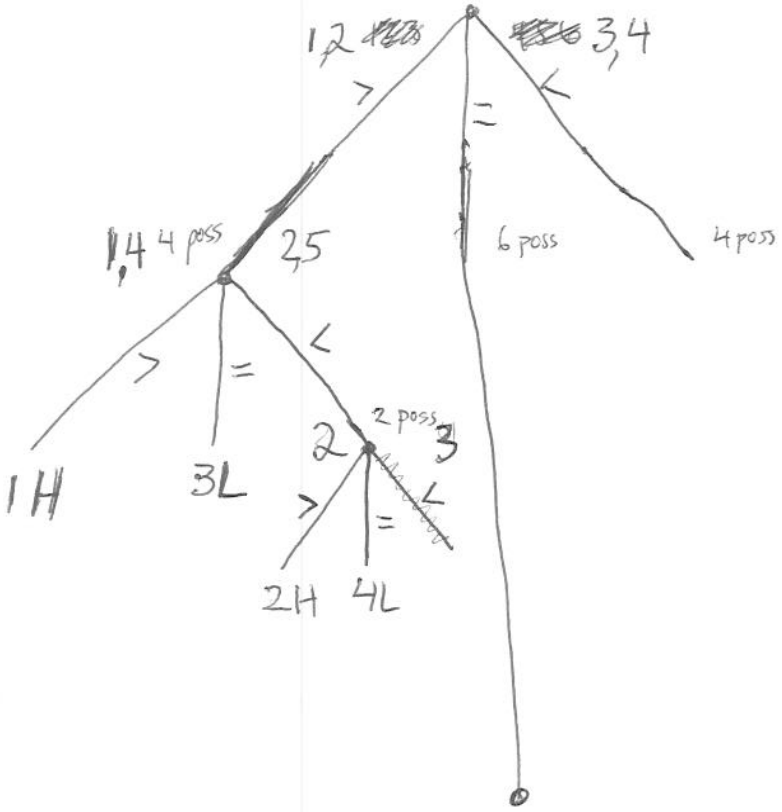
Using this text file with 96,558 characters and entropy 4.1727. Using three UNIX file compression programs `zip`, `compress` and `gzip`. I wanted to see how close to the theoretical minimum that I could get.

- `compress`:
  file length = 45,122 bytes or 360,976 bits. The average number of bits per character is approximately 3.7384.
- `gzip`:
  file length = 39,584 bytes or 316,672 bits. The average number of bits per character is approximately 3.2796.
- `zip`:
  file length = 39,706 bytes or 317,648 bits. The average number of bits per character is approximately 3.2897.
- Wait!? How is it possible? You got better than the theoretical minimum? Oops! Read the instructions, and notice that they are encoding 32 bits at a time (not 8 bits).

Using this text file with $4 \times 96,558$ characters and entropy 4.1727. Using three UNIX file compression programs `zip`, `compress` and `gzip`. I wanted to see how close to the theoretical minimum that I could get.

- `compress`:
  file length = 62,159 bytes or 497,272 bits. The average number of bits per character is approximately 5.15.
- `gzip`:
  file length = 57,404 bytes or 459,232 bits. The average number of bits per character is approximately 4.76.
- `zip`:
  file length = 57,526 bytes or 317,648 bits. The average number of bits per character is approximately 4.77.
- Thats better. These values are close (but larger than) the theoretical minimum.




1,2
3,4
>
=
<
1,4 4 poss
2,5
6 poss
4 poss
>
=
<
1H
3L
2 poss
2
3
>
=
<
2H
4L

Theorem

Perfect secrecy is achieved when

1. *All keys are equally likely*
2. *For each pair (m_i, c_j) there is a unique key, k_s, such that*

$$E_{k_s}(m_i) = c_j$$

On the other hand

$$\begin{aligned} P(M = m_i, C = c_j) &= \sum_{E_{k_s}(m_i)=c_j} P(M = m_i)P(K = k_s) \\ &= P(M = m_i)\frac{1}{S} \\ &= P(M = m_i)P(C = c_j) \end{aligned}$$

by (*) there is only one term

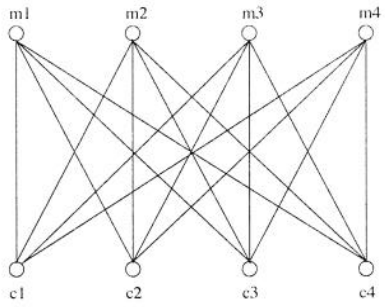
$\Rightarrow$ we have perfect secrecy

BUT there are other ways of acheiving perfect secrecy

Latin Squares

of Keys = # of Ciphers = # of Plaintexts

	m_1	m_2	m_3	m_4
k_1	1	2	3	4
k_2	2	3	4	1
k_3	3	4	1	2
k_4	4	1	2	3



A latin square is an $n \times n$ array where the integers 1 through n appear exactly once in each row and column.

Theorem

Perfect secrecy is achieved when

1. *All keys are equally likely*
2. *For each pair (m_i, c_j) there is a unique key, k_s, such that*

$$E_{k_s}(m_i) = c_j$$

(**) $P(K=k_s) = 1/S$ where S = # of keys

(*)

Theorem

Perfect secrecy is achieved when

1. *All keys are equally likely*
2. *For each pair (m_i, c_j) there is a unique key, k_s, such that*

$$E_{k_s}(m_i) = c_j$$

Proof.

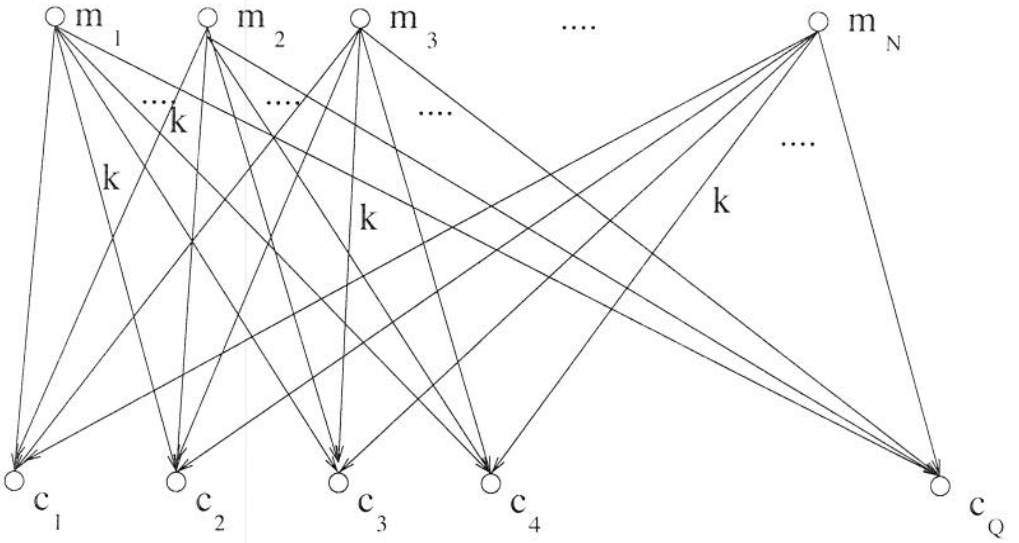
$$P(C = c_j) = \sum_{i=1}^{N} P(M = m_i) \boxed{\sum_{E_{k_s}(m_i)=c_j} P(K = k_s)}$$

this has only one term by (*)

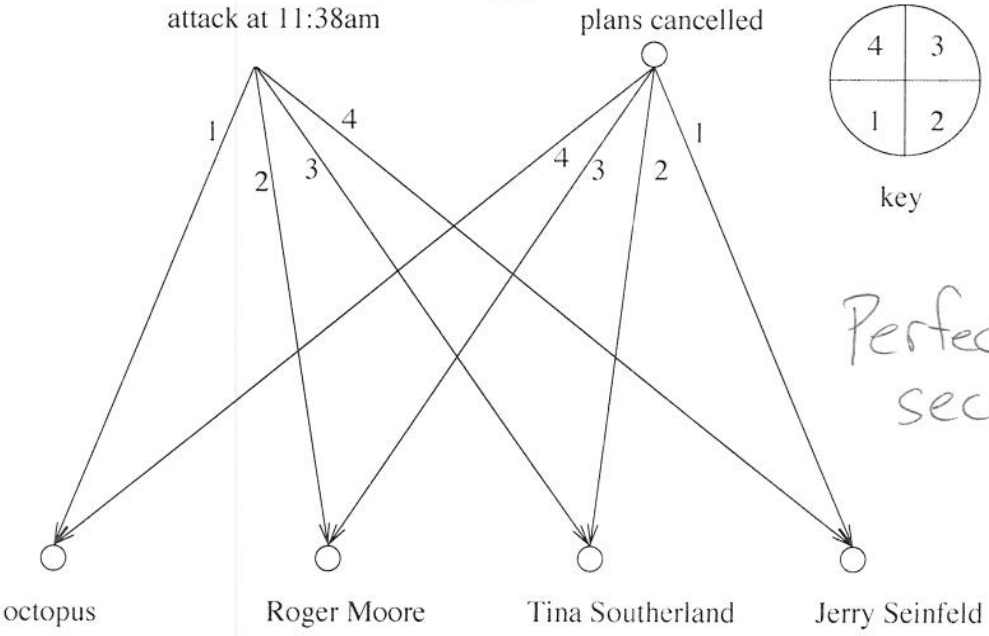
But if there is only one key k_s yielding $E_{k_s}(mi) = c_j$ then the inner sum reduces to a single term, and if all keys are equally likely then $P(K = k_s) = 1/S$

$$P(C = c_j) = \sum_{i=1}^{N} P(M = m_i)\frac{1}{S} = \frac{1}{S}$$

$$P[M = m_i, C = c_j] = P[M = m_i]P[C = c_j]$$



Since if we fix a key k we see every message is sent to a different cyphertext we must have that the number of cyphertexts is larger or equal to the number of plaintexts.



Perfect secrecy