Chapter 9

Source Coding

9.1 Problems

Problem 9.1 (Modeling a source with a die toss) *A die has six sides and produces a symbol corresponding to the number of dots showing on the top face. A fair die produces symbols that are equiprobable. What are the symbols and their probabilities?*

(ans: One obvious choice for symbols has the index number equal to the number of dots, as

$$X_1, X_2, X_3, X_4, X_5, X_6$$

The six equiprobable probabilities sum to one, giving

$$P[X_1] = P[X_2] = P[X_3] = P[X_4] = P[X_5] = P[X_6] = \frac{1}{6}$$

)

Problem 9.2 (Source entropy of the die-toss source) *Compute $\mathcal{H}_S$ for the source that produces symbols X_1 through X_6 with probabilities modeled by a fair die toss.*

(ans: The six probabilities are

$$P[X_1] = P[X_2] = P[X_3] = P[X_4] = P[X_5] = P[X_6] = \frac{1}{6}$$

$$\mathcal{H}_S = -\sum_{i=1}^{6} P[X_i] \log_2 P[X_i]$$

$$\frac{1}{6}\log_2\frac{1}{6} = -\frac{1}{6}\log_2 6 = -\frac{1}{6}\frac{\log_{10} 6}{\log_{10} 2} = -\frac{2.58}{6} = -0.431$$

$$\mathcal{H}_S = -\sum_{i=1}^{6}(-0.431) = -6(-0.431) = 2.585 \;\; \textit{bits/symbol}$$

For $m = 6$ equiprobable symbols

$$\mathcal{H}_S = \log_2 6 = \frac{\log_{10} 6}{\log_{10} 2} = \frac{0.778}{0.301} = 2.585 \ \textit{bits/symbol}$$

)

Problem 9.3 (Source entropy of the unfair die-toss source) *Compute $\mathcal{H}_S$ for the source that produces symbols with probabilities modeled by an unfair die toss, in which a 1 appears one quarter of the time and 2 through 6 have equal probabilities.*

(ans: $P[X_1] = 0.25$ gives the other probabilities as

$$\sum_{i=2}^{6} P[X_i] = 0.75$$

Making these probabilities equal gives

$$P[X_2] = P[X_3] = P[X_4] = P[X_5] = P[X_6] = \frac{0.75}{5} = 0.15$$

The source entropy equals

$$\mathcal{H}_S = -0.25 \log_2(0.25) - 5[0.15 \log_2(0.15)] = 0.5 + 2.053 = 2.553 \ \textit{bits/symbol}$$

Note that this entropy is smaller than that for 6 equiprobable symbols.
)

Problem 9.4 (Source entropy of the decimal digit source) *Compute $\mathcal{H}_S$ for the source that produces ten symbols that have equal probabilities.*

(ans: For $m = 10$ equiprobable symbols

$$\mathcal{H}_S = \log_2 10 = \frac{\log_{10} 10}{\log_{10} 2} = \frac{1}{0.301} = 3.322 \ \textit{bits/symbol}$$

)

Problem 9.5 (Data sequences having specified $\mathcal{H}_S$) *Compose a representative sequence containing ten or more symbols that would be typical of a source having the source entropy value.*

1. $\mathcal{H}_S = 0$ *bits/symbol*

2. $\mathcal{H}_S = 1$ *bit/symbol*

3. $\mathcal{H}_S = 2$ *bits/symbol*

(ans:

1. $\mathcal{H}_S = 0$ *bits/symbol means that* $P[X_1] = 1$ *because* $\log_2 1 = 0$ *and there is only one symbol possible. Hence, the following ten symbols are typical*

$$X_1X_1X_1X_1X_1X_1X_1X_1X_1X_1$$

2. $\mathcal{H}_S = 1$ *bit/symbol. The simplest example is two equiprobable symbols (bits). Hence, five realizations of each symbol would occur in typical sequences, as in the following two:*

$$X_1X_2X_1X_2X_1X_2X_1X_2X_1X_2$$

or

$$X_1X_1X_1X_1X_1X_2X_2X_2X_2X_2$$

3. $\mathcal{H}_S = 2$ *bits/symbol. The simplest example is four equiprobable symbols (dibits). Typically, three realizations of each symbol would occur in typical sequences having a length of 12 , as in the following two:*

$$X_1X_2X_3X_4X_1X_2X_3X_4X_1X_2X_3X_4$$

or

$$X_1X_1X_1X_2X_2X_2X_3X_3X_3X_4X_4X_4$$

)

Problem 9.6 (Effective probabilities of symbols in a text file) *A text file contains the characters:*

```
i need a vacation!!!
```

Include the space as a separate symbol. What are the values for the vocabulary size m_x*, total file size* n_T*, and the effective probabilities of the symbols in this file?*

(ans: The symbols, their counts and their effective probabilities are given in the following table.

i	X_i	n_{X_i}	cum. count	$P_e[X_i]$
1	i	2	2	0.1
2	space	3	5	0.15
3	n	2	7	0.1
4	e	2	9	0.1
5	d	1	10	0.05
6	a	3	13	0.15
7	v	1	14	0.05
8	c	1	15	0.05
9	t	1	16	0.05
10	o	1	17	0.05
11	!	3	20	0.15

Hence, $m_x = 11$*,* $n_T = 20$*, and the effective probabilities are computed in the fifth column.*
)

Problem 9.7 (Effective source entropy of a text file) *Use the values in Problem 9.6 to compute $\hat{\mathcal{H}}_S$.*

(ans:

$$\begin{aligned}\hat{\mathcal{H}}_S &= -\sum_{i=1}^{m_x} P_e[X_i] \log_2 P_e[X_i] \\ &= -3 \times 0.15 \times \frac{\log_{10}(0.15)}{\log_{10} 2} - 3 \times 0.1 \times \frac{\log_{10}(0.1)}{\log_{10} 2} - 5 \times 0.05 \times \frac{\log_{10}(0.05)}{\log_{10} 2} \\ &= 1.23 + 1.00 + 1.08 = 3.31 \textit{ bits/symbol}\end{aligned}$$

)

Problem 9.8 (Huffman code a text file) *Generate a Huffman code for the text file in Problem 9.6. Encode the file with the variable-length code and compare the total bit count to the $\hat{\mathcal{H}}_f$ value.*

(ans: The effective file entropy is computed from $n_T = 20$ and $\hat{\mathcal{H}}_S$ as

$$\hat{\mathcal{H}}_f = n_T \hat{\mathcal{H}}_S = 20(3.31) = 66.2 \textit{ bits}$$

Note that 11 symbols require a 4-bit code words to encode using a constant-length code, and the 20 symbols would need 80 bits.

The first to fourth sorts are given by the next table.

X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit
sp	0.15			sp	0.15			sp	0.15			sp	0.15	
a	0.15			a	0.15			a	0.15			a	0.15	
!	0.15			!	0.15			!	0.15			!	0.15	
i	0.1			i	0.1			i	0.1			v-c-d	0.15	
n	0.1			n	0.1			n	0.1			i	0.1	
e	0.1			e	0.1			e	0.1			n	0.1	
d	0.05			t-o	0.1			t-o	0.1			e	0.1	1
v	0.05			d	0.05			v-c	0.1	1		t-o	0.1	0
c	0.05			v	0.05	1		d	0.05	0				
t	0.05	1		c	0.05	0								
o	0.05	0												

Note: If the $P_e[X_i]$ values are awkward (because n_T is an accommodating value), use n_{X_i} for the sorting operation. This works because $n_{X_i} \propto P_e[X_i]$.

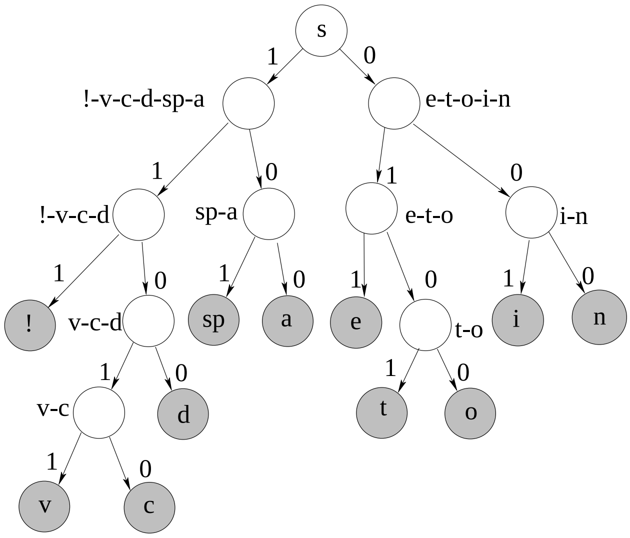
The fifth to eighth sorts are given by the next table.

X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit		X_i	$P_e[X_i]$	bit
e-t-o	0.2			e-t-o	0.2			!-v-c-d	0.3			!-v-c-d	0.3	
sp	0.15			i-n	0.2			e-t-o	0.2			sp-a	0.3	
a	0.15			sp	0.15			i-n	0.2			e-t-o	0.2	1
!	0.15			a	0.15			sp	0.15	1		i-n	0.2	0
v-c-d	0.15			!	0.15	1		a	0.15	0				
i	0.1	1		v-c-d	0.15	0								
n	0.1	0												

The ninth and tenth sorts are given by the next table. The Huffman code tree and code table are shown below.

X_i	$P_e[X_i]$	*bit*		X_i	$P_e[X_i]$	*bit*
e-t-o-i-n	*0.4*			*!-v-c-d-sp-a*	*0.6*	*1*
!-v-c-d	*0.3*	*1*		*e-t-o-i-n*	*0.4*	*0*
sp-a	*0.3*	*0*				

symbol X_i	*code assignment*
!	*111*
space	*101*
a	*100*
e	*011*
i	*001*
n	*000*
d	*1100*
t	*0101*
o	*0100*
v	*11011*
c	*11010*



Encoding the sequence gives

$$\overbrace{001}^{i}\ \overbrace{101}^{sp}\ \overbrace{000}^{n}\ \overbrace{011}^{e}\ \overbrace{011}^{e}\ \overbrace{1100}^{d}\ \overbrace{101}^{sp}\ \overbrace{100}^{a}\ \overbrace{101}^{sp}$$

$$\overbrace{11011}^{v}\ \overbrace{100}^{a}\ \overbrace{11010}^{c}\ \overbrace{100}^{a}\ \overbrace{0101}^{t}\ \overbrace{001}^{i}\ \overbrace{0100}^{o}\ \overbrace{000}^{n}\ \overbrace{111}^{!}\ \overbrace{111}^{!}\ \overbrace{111}^{!}$$

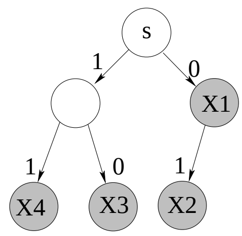
The total bit count is 67, which is approximately equal to $\hat{\mathcal{H}}_f = 66.2$ *bits.*

)

Problem 9.9 (Valid or invalid Huffman code) *Does the following code represent a valid Huffman code? (Hint: draw the tree.)*

$$X_1 : 0 \quad X_2 : 01 \quad X_3 : 10 \quad X_4 : 11$$

(ans: The Huffman code tree shows that X_1 *is not a terminal node, or leaf, as valid Huffman codes require.*



)

Problem 9.10 (Huffman code for digits) *Generate the Huffman code for the ten digits* 0 *through* 9 *that have equal probabilities.*

(ans: The first to fourth sorts are given by the next table.

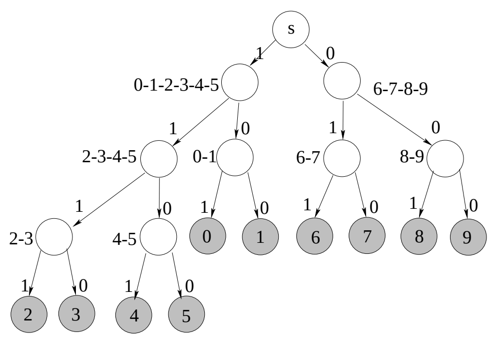
X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*
0	0.1			8-9	0.2			6-7	0.2			4-5	0.2	
1	0.1			0	0.1			8-9	0.2			6-7	0.2	
2	0.1			1	0.1			0	0.1			8-9	0.2	
3	0.1			2	0.1			1	0.1			0	0.1	
4	0.1			3	0.1			2	0.1			1	0.1	
5	0.1			4	0.1			3	0.1			2	0.1	1
6	0.1			5	0.1			4	0.1	1		3	0.1	0
7	0.1			6	0.1	1		5	0.1	0				
8	0.1	1		7	0.1	0								
9	0.1	0												

The fifth to eighth sorts are given by the next table.

X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*
2-3	0.2			0-1	0.2			6-7-8-9	0.4			6-7-8-9	0.4	
4-5	0.2			2-3	0.2			0-1	0.2			2-3-4-5	0.4	1
6-7	0.2			4-5	0.2			2-3	0.2	1		0-1	0.2	0
8-9	0.2			6-7	0.2	1		4-5	0.2	0				
0	0.1	1		8-9	0.2	0								
1	0.1	0												

The ninth and final sort is given by the next table. The Huffman code tree for the digits is shown below.

X_i	$P[X_i]$	*bit*
0-1-2-3-4-5	0.6	1
6-7-8-9	0.4	0



)

Problem 9.11 (Data compression) *Consider the following data file.*

$$AAACAAABAAACAAADAAAE$$

Implement a Huffman code to compress this file. What is the average number of bits per symbol?

(ans: The symbols, their counts and their effective probabilities are given in the following table.

i	X_i	n_{X_i}	*cum. count*	$P_e[X_i]$
1	*A*	*15*	*15*	*0.75*
2	*B*	*1*	*16*	*0.05*
3	*C*	*2*	*18*	*0.1*
4	*D*	*1*	*19*	*0.05*
5	*E*	*1*	*20*	*0.05*

Hence, $m_x = 5$*,* $n_T = 20$*, and the effective probabilities are computed in the fifth column. The first to fourth sorts are given by the next table.*

X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*
A	*0.75*			*A*	*0.75*			*A*	*0.75*			*A*	*0.75*	*1*
C	*0.1*			*C*	*0.1*			*D-E-B*	*0.15*	*1*		*D-E-B-C*	*0.25*	*0*
B	*0.05*			*D-E*	*0.1*	*1*		*C*	*0.1*	*0*				
D	*0.05*	*1*		*B*	*0.05*	*0*								
E	*0.01*	*0*												

The Huffman code is given in the table below, derived from the sorts.

symbol X_i	*code assignment*
A	*1*
B	*010*
C	*00*
D	*0111*
E	*0110*

$$\overbrace{111}^{AAA}\ \overbrace{00}^{C}\ \overbrace{111}^{AAA}\ \overbrace{010}^{B}\ \overbrace{111}^{AAA}\ \overbrace{00}^{C}\ \overbrace{111}^{AAA}\ \overbrace{0111}^{D}\ \overbrace{111}^{AAA}\ \overbrace{0110}^{E}$$

There are 30 bits for 20 symbols, or 1.5 bits/symbol.
)

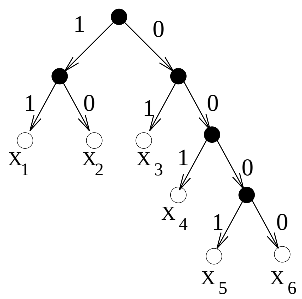


Figure 9.1: Huffman code tree for Problem 9.12

Problem 9.12 (Huffman tree encoding) *Find the binary sequence generated by the symbol sequence*

$$X_1\ X_3\ X_2\ X_5\ X_4\ X_6$$

using the Huffman code tree shown in Figure 9.1.

(ans:

$$\overbrace{11}^{X_1}\ \overbrace{01}^{X_3}\ \overbrace{10}^{X_2}\ \overbrace{0001}^{X_5}\ \overbrace{001}^{X_4}\ \overbrace{0000}^{X_6}$$

)

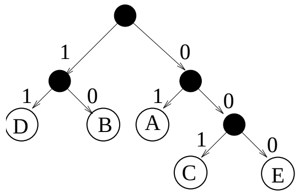


Figure 9.2: Huffman code tree for Problem 9.13

Problem 9.13 (Huffman tree decoding) *What symbols are represented by the binary sequence*

01110010001000010

using the Huffman code tree shown in Figure 9.2.

(ans:

$$\overbrace{01}^{A}\ \overbrace{11}^{D}\ \overbrace{001}^{C}\ \overbrace{000}^{E}\ \overbrace{10}^{B}\ \overbrace{000}^{E}\ \overbrace{10}^{B}$$

)

Problem 9.14 (Huffman code tree for symbols having unequal probabilities) *A coin flip produces a 0 if a tail appears and a 1 for a head. A source produces symbols that have the same probabilities as the sums that are observed when three fair coins are flipped. For example, the outcome HTH produces 101, which sums to 2. Determine the Huffman code tree for this source.*

(ans: The table below shows the equiprobable results using the binary count sequence order.

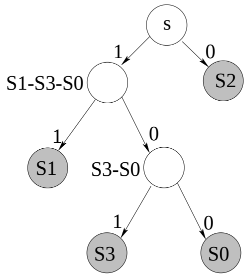
outcome	*binary values*	*sum*	*symbol*
TTT	*000*	*0*	*S0*
TTH	*001*	*1*	*S1*
THT	*010*	*1*	*S1*
THH	*011*	*2*	*S2*
HTT	*100*	*1*	*S1*
HTH	*101*	*2*	*S2*
HHT	*110*	*2*	*S2*
HHH	*111*	*3*	*S3*

The three required sorts are given by the next table.

X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*		X_i	$P[X_i]$	*bit*
S2	*3/8*			*S2*	*3/8*			*S1-S3-S0*	*5/8*	*1*
S1	*3/8*			*S1*	*3/8*	*1*		*S2*	*3/8*	*0*
S3	*1/8*	*1*		*S3-S0*	*2/8*	*0*				
S0	*1/8*	*0*								

The Huffman code table and tree are given below.

symbol X_i	*code assignment*
S0	*100*
S1	*11*
S2	*0*
S3	*101*



)

Problem 9.15 (Source encoding) *A source generates six symbols with* $P[X_1] = 0.05$, $P[X_2] = 0.47$, $P[X_3] = 0.07$, $P[X_4] = 0.20$, $P[X_5] = 0.11$, *and* $P[X_6] = 0.10$.

1. *Compute the source entropy* $\mathcal{H}_S$.

2. *Generate the Huffman code tree.*

3. *Encode the ten-symbol sequence*

$$X_3 \quad X_1 \quad X_3 \quad X_5 \quad X_6 \quad X_1 \quad X_6 \quad X_3 \quad X_1 \quad X_3$$

4. *Compute the average number of bits per symbol used for encoding the sequence.*

5. *Compare the values of the source entropy and the average bits/symbol computed.*

6. *Why does the* $\mathcal{H}_S$ *value differ significantly from the average bits/symbol?*

(ans:

1.

$$\begin{aligned}
\mathcal{H}_S &= -\sum_{i=1}^{6} P[X_i] \log_2 P[X_i] \\
&= -0.05 \overbrace{\log_2 0.05}^{=-4.32} - 0.47 \overbrace{\log_2 0.47}^{=-1.09} - 0.07 \overbrace{\log_2 0.07}^{=-3.84} - 0.02 \overbrace{\log_2 0.02}^{=-5.64} \\
&\quad -0.11 \overbrace{\log_2 0.11}^{=-3.18} - 0.1 \overbrace{\log_2 0.1}^{=-3.32} \\
&= 1.79 \textit{ bits/symbol}
\end{aligned}$$

2. *The first to fourth sorts are given by the next table. The probabilities have be multiplied by 100 to form* $P^*[X_i]$ *to simplify the manipulations.*

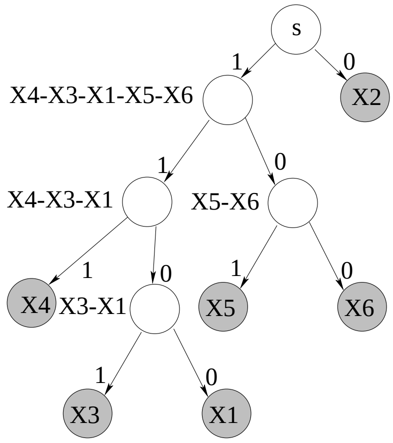
X_i	$P^*[X_i]$	*bit*		X_i	$P^*[X_i]$	*bit*		X_i	$P^*[X_i]$	*bit*		X_i	$P^*[X_i]$	*bit*
X2	47			X2	47			X2	47			X2	47	
X4	20			X4	20			X5-X6	21			X4-X3-X1	32	1
X5	11			X3-X1	12			X4	20	1		X5-X6	21	0
X6	10			X5	11	1		X3-X1	12	0				
X3	7	1		X6	10	0								
X1	5	0												

The fifth sort is given by the next table.

X_i	$P^*[X_i]$	*bit*
X4-X3-X1-X5-X6	53	1
X2	47	0

The Huffman code table and tree are given below.

symbol X_i	*code assignment*
X1	1100
X2	0
X3	1101
X4	111
X5	101
X6	100



3. *Encode the ten-symbol sequence*

$$\overbrace{1101}^{X3}\ \overbrace{1100}^{X1}\ \overbrace{1100}^{X3}\ \overbrace{101}^{X5}\ \overbrace{100}^{X6}\ \overbrace{1100}^{X1}\ \overbrace{100}^{X6}\ \overbrace{1101}^{X3}\ \overbrace{1100}^{X1}\ \overbrace{1101}^{X3}$$

4. *Compute the average number of bits per symbol used for encoding the sequence.*

$$\frac{\textit{37 bits}}{\textit{10 symbols}} = 3.7 \textit{ bits/symbol}$$

5. *Compare the values of the source entropy and the average bits/symbol computed.*

$$\mathcal{H}_S = 1.79 \textit{ bits/symbol} \quad \textit{compared to } 3.7 \textit{ bits/symbol}$$

6. *Why does the $\mathcal{H}_S$ value differ significantly from the average bits/symbol?*

The sequence contains many more $X1$ and $X3$ symbols than the theoretical probabilities indicate, making it an atypical sequence. For example,

$$P_e[X1] = \frac{3}{10} = 0.3 \quad \textit{while } P[X1] = 0.05$$

and

$$P_e[X3] = \frac{4}{10} = 0.4 \quad \textit{while } P[X3] = 0.07$$

)

Problem 9.16 (Huffman code for a symbol file) *Consider the data file containing the ten symbols*

$$X_3\ X_3\ X_5\ X_1\ X_2\ X_5\ X_1\ X_1\ X_2\ X_4$$

1. *Specify a fixed-length code to encode the symbols and compute the total number of bits that encode the file.*
2. *Compute the effective source entropy of the file $\hat{\mathcal{H}}_S$.*
3. *Generate a Huffman code to encode the file.*
4. *Encode the file using this code and compute the total number of bits that encode the file.*

(ans:

1. *Specify a fixed-length code to encode the symbols and compute the total number of bits that encode the file.*

   The number of symbols $m_x = 5$ and requires 3-bit code words. The ten symbols require 30 bits to encode. One possible fixed-length code table is given below.

symbol X_i	code assignment
X1	001
X2	010
X3	111
X4	100
X5	101

2. *Compute the effective source entropy of the file $\hat{\mathcal{H}}_S$. The symbols, their counts and their effective probabilities are given in the following table.*

i	X_i	n_{X_i}	cum. count	$P_e[X_i]$
1	X1	3	3	0.3
2	X2	2	5	0.2
3	X3	2	7	0.2
4	X4	1	8	0.1
5	X5	2	10	0.2

$$\begin{aligned} \hat{\mathcal{H}}_S &= -\sum_{i=1}^{5} P_e[X_i] \log_2 P_e[X_i] \\ &= -0.3 \overbrace{\log_2 0.3}^{-1.74} - 3 \times 0.2 \overbrace{\log_2 0.2}^{-2.32} - 0.1 \overbrace{\log_2 0.1}^{-3.32} \\ &= 2.25 \textit{ bits/symbol} \end{aligned}$$

3. *Generate a Huffman code to encode the file.*

4. *The first to fourth sorts are given by the next table. The probabilities have be multiplied by 10 to form $P^*[X_i]$ to simplify the manipulations.*

X_i	$P^*[X_i]$	bit		X_i	$P^*[X_i]$	bit		X_i	$P^*[X_i]$	bit		X_i	$P^*[X_i]$	bit
X1	3			X1	3			X2-X3	4			X1-X5-X4	6	1
X2	2			X5-X4	3			X1	3	1		X2-X3	4	0
X3	2			X2	2	1		X5-X4	3	0				
X5	2	1		X3	2	0								
X4	1	0												

The Huffman code table is given below.

symbol X_i	code assignment
X1	11
X2	01
X3	00
X4	101
X5	100

5. *Encode the file using this code and compute the total number of bits that encode the file.*

$$\overbrace{00}^{X3}\ \overbrace{00}^{X3}\ \overbrace{100}^{X5}\ \overbrace{11}^{X1}\ \overbrace{01}^{X2}\ \overbrace{100}^{X5}\ \overbrace{11}^{X1}\ \overbrace{11}^{X1}\ \overbrace{01}^{X2}\ \overbrace{101}^{X4}$$

$$\frac{\textit{23 bits}}{\textit{10 symbols}} = 2.3 \textit{ bits/symbol}$$

Compare to

$$n_T \hat{\mathcal{H}}_S = 10 \textit{ symbols } \times 2.25 \textit{ bits/symbol} = 22.5 \textit{ bits}$$

)

Problem 9.17 (Encryption) *Let*

$$RBS_i = 10011001$$

Use RBS_i to encrypt the data

$$\mathcal{D}_i = 11001011$$

to form the encrypted binary sequence $\mathcal{E}_i$.

(ans:

$$\begin{aligned} RBS_i &= \begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{matrix} \\ \mathcal{D}_i &= \begin{matrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{matrix} \\ \mathcal{E}_i &= \begin{matrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{matrix} \end{aligned}$$

)

Problem 9.18 (Pseudo-random number algorithm) *Letting $X_0 = 111$, generate X_1 and X_2 using the PRNG formula*

$$X_i = mod[\ (\alpha \times X_{i-1} + \beta),\ h]$$

with $\alpha = 766$, $\beta = 369$, $h = 1,000$.

(ans:

$$X_1 = mod[\ (766 \times 111 + 369),\ 1000] = mod[(85,395),\ 1000] = 395$$

$$X_2 = mod[\ (766 \times 395 + 369),\ 1000] = mod[(302,939),\ 1000] = 939$$

)

Problem 9.19 (Random numbers to random bits) *Assume the pseudo-random number algorithm in Problem 9.18 generated the eight random numbers*

$$979, 122, 475, 986, 730, 286, 66, 899$$

Describe how you would generate eight random bits from this sequence and generate the random bit sequence.

(ans: Divide the h interval [0, 1000) into two equal intervals as [0, 500) and [500, 1000), assign 0 to the first interval and 1 to the second as

$$\overbrace{[0, 500)}^{0} \quad \overbrace{[500, 1000)}^{1}$$

Applying this rule to the observed random numbers gives

$$\overbrace{979}^{1}, \overbrace{122}^{0}, \overbrace{475}^{0}, \overbrace{986}^{1}, \overbrace{730}^{1}, \overbrace{286}^{0}, \overbrace{66}^{0}, \overbrace{899}^{1}$$

)

Problem 9.20 (Encryption key) *Compute the value of the encryption key using the values: $n = 10$, $a = 13$, $x = 5$, $y = 6$.*

(ans:

$$\mathcal{X} = [a^x]_{mod(n)} = [13^5]_{mod(10)} = [371,293]_{mod(10)} = 3$$

$$\mathcal{Y} = [a^y]_{mod(n)} = [13^6]_{mod(10)} = [4,826,809]_{mod(10)} = 9$$

$$K_T = [\mathcal{Y}^x]_{mod(n)} = [9^5]_{mod(10)} = [59,049]_{mod(10)} = 9$$

$$K_R = [\mathcal{X}^y]_{mod(n)} = [3^6]_{mod(10)} = [729]_{mod(10)} = 9$$

Hence, the key equals 9.
)

9.2 Excel Projects

Project 9.1 (Generating equi-probable source symbols) *Using Example 13.46 as a guide, consider a source producing $m = 9$ symbols $X1, X2, ..., X8$ that have equal probabilities. Produce $n_T = 30$ random symbols.*

(ans:

	A	B	C
1	m=		n_T=
2	9		30
3			
4	i	Rand num	Symbol
5	1	4	X4
6	2	4	X4
7	3	5	X5
8	4	9	X9
9	5	7	X7
10	6	3	X3
11	7	2	X2
12	8	5	X5
13	9	1	X1
14	10	3	X3
15	11	2	X2
16	12	3	X3
17	13	3	X3
18	14	6	X6
19	15	8	X8
20	16	1	X1
21	17	4	X4
22	18	7	X7
23	19	4	X4
24	20	2	X2
25	21	9	X9
26	22	9	X9
27	23	2	X2
28	24	6	X6
29	25	1	X1
30	26	1	X1
31	27	5	X5
32	28	5	X5
33	29	7	X7
34	30	2	X2

	A	B	C
1	m=		n_T=
2	9		30
3			
4	i	Rand num	Symbol
5	1	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B5,0))
6	=1+A5	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B6,0))
7	=1+A6	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B7,0))
8	=1+A7	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B8,0))
9	=1+A8	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B9,0))
10	=1+A9	=FLOOR($A$2*RAND()+1,1)	=CONCATENATE("X", TEXT(B10,0))

)

Project 9.2 (Computing effective entropy) *Using Example 13.47 as a guide, generate $n_T = 30$ realizations of the $m = 9$ symbols in Project 9.1, compute effective probabilities $P_e[X_i]$, effective entropy $\mathcal{H}_e$ and file entropy $\mathcal{H}_f$.*

(ans:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Symbol	X1?	X2?	X3?	X4?	X5?	X6?	X7?	X8?	X9?		X_i	$P_e[X_i]$	P_e Log_2 P_e	H_e (b/sym)	H_f (b)
2	X5	0	0	0	0	1	0	0	0	0		X1	0.400	-0.529	3.695	110.865
3	X7	0	0	0	0	0	0	1	0	0		X2	0.200	-0.464		
4	X3	0	0	1	0	0	0	0	0	0		X3	0.133	-0.388		
5	X9	0	0	0	0	0	0	0	0	1		X4	0.000	0.000		
6	X1	1	0	0	0	0	0	0	0	0		X5	0.267	-0.509		
7	X1	1	0	0	0	0	0	0	0	0		X6	0.267	-0.509		
8	X8	0	0	0	0	0	0	0	1	0		X7	0.400	-0.529		
9	X3	0	0	1	0	0	0	0	0	0		X8	0.067	-0.260		
10	X2	0	1	0	0	0	0	0	0	0		X9	0.267	-0.509		
11	X6	0	0	0	0	0	1	0	0	0						
12	X9	0	0	0	0	0	0	0	0	1						
13	X1	1	0	0	0	0	0	0	0	0						
14	X5	0	0	0	0	1	0	0	0	0						
15	X7	0	0	0	0	0	0	1	0	0						
16	X2	0	1	0	0	0	0	0	0	0						
17	X2	0	1	0	0	0	0	0	0	0						
18	X9	0	0	0	0	0	0	0	0	1						
19	X7	0	0	0	0	0	0	1	0	0						
20	X7	0	0	0	0	0	0	1	0	0						
21	X7	0	0	0	0	0	0	1	0	0						
22	X5	0	0	0	0	1	0	0	0	0						
23	X6	0	0	0	0	0	1	0	0	0						
24	X6	0	0	0	0	0	1	0	0	0						
25	X1	1	0	0	0	0	0	0	0	0						
26	X6	0	0	0	0	0	1	0	0	0						
27	X1	1	0	0	0	0	0	0	0	0						
28	X9	0	0	0	0	0	0	0	0	1						
29	X5	0	0	0	0	1	0	0	0	0						
30	X7	0	0	0	0	0	0	1	0	0						
31	X1	1	0	0	0	0	0	0	0	0						
32																
33	n_x=	6	3	2	0	4	4	6	1	4						

	A	B	C	D
29	X5	=IF(A29="X1",1,0)	=IF(A29="X2",1,0)	=IF(A29="X3",1,0)
30	X7	=IF(A30="X1",1,0)	=IF(A30="X2",1,0)	=IF(A30="X3",1,0)
31	X1	=IF(A31="X1",1,0)	=IF(A31="X2",1,0)	=IF(A31="X3",1,0)
32				
33	n_x=	=SUM(B2:B31)	=SUM(C2:C31)	=SUM(D2:D31)

	L	M	N	O	P
1	X_i	$P_e[X_i]$	P_e Log_2 P_e	H_e (b/sym)	H_f (b)
2	X1	=B33/SUM($B$33:$F$33)	=IF(M2>0,M2*LOG(M2,2),0)	=-SUM(N2:N10)	=O2*SUM(B33:J33)
3	X2	=C33/SUM($B$33:$F$33)	=IF(M3>0,M3*LOG(M3,2),0)		
4	X3	=D33/SUM($B$33:$F$33)	=IF(M4>0,M4*LOG(M4,2),0)		
5	X4	=E33/SUM($B$33:$F$33)	=IF(M5>0,M5*LOG(M5,2),0)		
6	X5	=F33/SUM($B$33:$F$33)	=IF(M6>0,M6*LOG(M6,2),0)		
7	X6	=G33/SUM($B$33:$F$33)	=IF(M7>0,M7*LOG(M7,2),0)		
8	X7	=H33/SUM($B$33:$F$33)	=IF(M8>0,M8*LOG(M8,2),0)		
9	X8	=I33/SUM($B$33:$F$33)	=IF(M9>0,M9*LOG(M9,2),0)		
10	X9	=J33/SUM($B$33:$F$33)	=IF(M10>0,M10*LOG(M10,2),0)		

)

Project 9.3 (Huffman Code) *Using Example 13.48 as a guide, generate a Huffman Code of the symbols in Project 9.2.*

(ans: Symbols with zero probability are not assigned a code. The process of generating a Huffman code is manually done and is simplified by using the sum function (as shown in the code below), cut and paste/special/values, and the sort function.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Original			First Sort				Second Sort				Third Sort				Fourth Sort		
2	X	P_e[X]		X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code
3	X1	0.200		X1	0.200			X1	0.200			X1	0.200			X6-X9	0.267	
4	X2	0.100		X7	0.200			X7	0.200			X7	0.200			X1	0.200	
5	X3	0.067		X5	0.133			X5	0.133			X2-X3-X8	0.200			X7	0.200	
6	X4	0.000		X6	0.133			X6	0.133			X5	0.133			X2-X3-X8	0.200	1
7	X5	0.133		X9	0.133			X9	0.133			X6	0.133	1		X5	0.133	0
8	X6	0.133		X2	0.100			X2	0.100	1		X9	0.133	0				
9	X7	0.200		X3	0.067	1		X3-X8	0.100	0								
10	X8	0.033		X8	0.033	0										Code Table		
11	X9	0.133		X4	0.000											X1	01	
12																X2	1111	
13				Fifth Sort				Sixth Sort				Seventh Sort				X3	11101	
14				X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code		X4	x	
15				X2-X3-X8-X5	0.333			X1-X7	0.400			X2-X3-X8-X5-X6-X9	0.600	1		X5	110	
16				X6-X9	0.267			X2-X3-X8-X5	0.333	1		X1-X7	0.400	0		X6	101	
17				X1	0.200	1		X6-X9	0.267	0						X7	00	
18				X7	0.200	0										X8	11100	
19																X9	100	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Original			First Sort				Second Sort				Third Sort				Fourth Sort		
2	X	P_e[X]		X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code
3	X1	0.2		X1	0.2			X1	0.2			X1	0.2			X6-X9	=SUM(M7:M8)	
4	X2	0.1		X7	0.2			X7	0.2			X7	0.2			X1	0.2	
5	X3	0.06666666		X5	0.133333333333			X5	0.133333333333			X2-X3-X8	=SUM(I8:I9)			X7	0.2	
6	X4	0		X6	0.133333333333			X6	0.133333333333			X5	0.13333333333			X2-X3-X8	0.2	1
7	X5	0.13333333		X9	0.133333333333			X9	0.133333333333			X6	0.13333333333	1		X5	0.133333333333	0
8	X6	0.13333333		X2	0.1			X2	0.1	1		X9	0.13333333333	0				
9	X7	0.2		X3	0.066666666666	1		X3-X8	=SUM(E9:E10)	0								
10	X8	0.03333333		X8	0.033333333333	0										Code Table		
11	X9	0.13333333		X4	0											X1	01	
12																X2	1111	
13				Fifth Sort				Sixth Sort				Seventh Sort				X3	11101	
14				X	P_e[X]	code		X	P_e[X]	code		X	P_e[X]	code		X4	x	
15				X2-X3-X8-X5	=SUM(Q6:Q7)			X1-X7	=SUM(E17:E18)			X2-X3-X8-X5-X6-X9	=SUM(I16:I17)	1		X5	110	
16				X6-X9	0.266666666666			X2-X3-X8-X5	0.333333333333	1		X1-X7	0.4	0		X6	101	
17				X1	0.2	1		X6-X9	0.266666666666	0						X7	00	
18				X7	0.2	0										X8	11100	
19																X9	100	

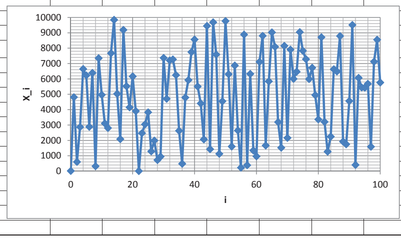
)

Project 9.4 (Good and Bad Pseudo-Random Number Generator) *Using Example 13.49 as a guide, use your own values for A, B, h (=10,000) and seed X_O to generate 100 pseudo-random numbers that generate 100 pseudo-random bits. Plot the X_i values to observe any periodic behavior. Choose four-digit values of A and B that do not show a periodic X_i and close-by values of A and B that do, forming not so-random numbers.*

(ans: The PRNG produces at most h unique values. An unfortunate choice for A and B causes the PRN sequence to degenerate into a periodic sequence with period $<<$ h or one that is constant, as shown below. A good choice produces a period that has a period $>$ 100 or more.

Good choice of A and B:

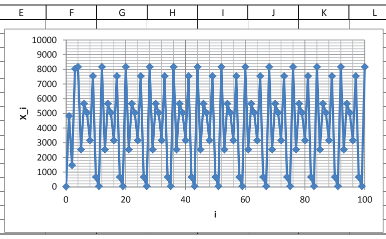
	A	B	C	D
1	A=	B=	h=	seed=
2	1234	8765	10000	13
3				
4	i	X_i	Rand Bit	
5	0	13		
6	1	4807	0	
7	2	603	0	
8	3	2867	0	
9	4	6643	1	
10	5	6227	1	
11	6	2883	0	
12	7	6387	1	
13	8	323	0	
14	9	7347	1	
15	10	4963	0	
16	11	3107	0	



	A	B	C	D
1	A=	B=	h=	seed=
2	1234	8765	10000	13
3				
4	i	X_i	Rand Bit	
5	0	=D2		
6	=1+A5	=MOD($A$2*B5+$B$2,$C$2)	=IF(B6<$C$2/2,0,1)	
7	=1+A6	=MOD($A$2*B6+$B$2,$C$2)	=IF(B7<$C$2/2,0,1)	
8	=1+A7	=MOD($A$2*B7+$B$2,$C$2)	=IF(B8<$C$2/2,0,1)	
9	=1+A8	=MOD($A$2*B8+$B$2,$C$2)	=IF(B9<$C$2/2,0,1)	
10	=1+A9	=MOD($A$2*B9+$B$2,$C$2)	=IF(B10<$C$2/2,0,1)	

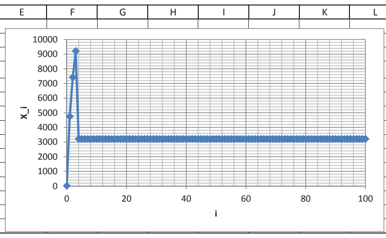
Poor choice of close-by A and B causes periodic RN sequence:

	A	B	C	D
1	A=	B=	h=	seed=
2	1235	8765	10000	13
3				
4	i	X_i	Rand Bit	
5	0	13		
6	1	4820	0	
7	2	1465	0	
8	3	8040	1	
9	4	8165	1	
10	5	2540	0	
11	6	5665	1	
12	7	5040	1	
13	8	3165	0	
14	9	7540	1	
15	10	665	0	



Poor choice of close-by A and B causes constant RN sequence:

	A	B	C	D
1	A=	B=	h=	seed=
2	1230	8765	10000	13
3				
4	i	X_i	Rand Bit	
5	0	13		
6	1	4755	0	
7	2	7415	1	
8	3	9215	1	
9	4	3215	0	
10	5	3215	0	
11	6	3215	0	
12	7	3215	0	
13	8	3215	0	
14	9	3215	0	
15	10	3215	0	

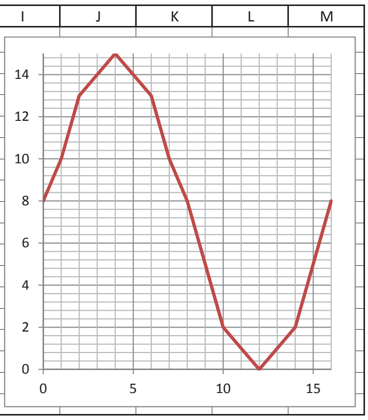


)

Project 9.5 (Encryption of sinusoidal data) *Using Example 13.50 as a guide, quantize $n_x = 17$ samples (period = 16) of sinusoidal waveform to 4-bit data and your PRNG to generate a random bit sequence having the same number of bits to encrypt the data. Compare plots of the original waveform and encrypted data. At the receiver de-encrypt the data and plot the de-encrypted data.*

(ans: Generating 4-bit sine sequence:

	A	B	C	D	E	F	G	H
1	i	s_i	sq_i		Data			
2	0	0.000	8		1	0	0	0
3	1	0.383	10		1	0	1	0
4	2	0.707	13		1	1	0	1
5	3	0.924	14		1	1	1	0
6	4	1.000	15		1	1	1	1
7	5	0.924	14		1	1	1	0
8	6	0.707	13		1	1	0	1
9	7	0.383	10		1	0	1	0
10	8	0.000	8		1	0	0	0
11	9	-0.383	5		0	1	0	1
12	10	-0.707	2		0	0	1	0
13	11	-0.924	1		0	0	0	1
14	12	-1.000	0		0	0	0	0
15	13	-0.924	1		0	0	0	1
16	14	-0.707	2		0	0	1	0
17	15	-0.383	5		0	1	0	1
18	16	0.000	8		1	0	0	0



	A	B	C	D	E	F	G	H
1	i	s_i	sq_i			Data		
2	0	=SIN(2*PI()*A2/16)	=ROUND(7.5*(1+B2),0)		=MOD(FLOOR(C2/8,1),2)	=MOD(FLOOR(C2/4,1),2)	=MOD(FLOOR(C2/2,1),2)	=MOD(C2,2)
3	=1+A2	=SIN(2*PI()*A3/16)	=ROUND(7.5*(1+B3),0)		=MOD(FLOOR(C3/8,1),2)	=MOD(FLOOR(C3/4,1),2)	=MOD(FLOOR(C3/2,1),2)	=MOD(C3,2)
4	=1+A3	=SIN(2*PI()*A4/16)	=ROUND(7.5*(1+B4),0)		=MOD(FLOOR(C4/8,1),2)	=MOD(FLOOR(C4/4,1),2)	=MOD(FLOOR(C4/2,1),2)	=MOD(C4,2)
5	=1+A4	=SIN(2*PI()*A5/16)	=ROUND(7.5*(1+B5),0)		=MOD(FLOOR(C5/8,1),2)	=MOD(FLOOR(C5/4,1),2)	=MOD(FLOOR(C5/2,1),2)	=MOD(C5,2)
6	=1+A5	=SIN(2*PI()*A6/16)	=ROUND(7.5*(1+B6),0)		=MOD(FLOOR(C6/8,1),2)	=MOD(FLOOR(C6/4,1),2)	=MOD(FLOOR(C6/2,1),2)	=MOD(C6,2)
7	=1+A6	=SIN(2*PI()*A7/16)	=ROUND(7.5*(1+B7),0)		=MOD(FLOOR(C7/8,1),2)	=MOD(FLOOR(C7/4,1),2)	=MOD(FLOOR(C7/2,1),2)	=MOD(C7,2)
8	=1+A7	=SIN(2*PI()*A8/16)	=ROUND(7.5*(1+B8),0)		=MOD(FLOOR(C8/8,1),2)	=MOD(FLOOR(C8/4,1),2)	=MOD(FLOOR(C8/2,1),2)	=MOD(C8,2)
9	=1+A8	=SIN(2*PI()*A9/16)	=ROUND(7.5*(1+B9),0)		=MOD(FLOOR(C9/8,1),2)	=MOD(FLOOR(C9/4,1),2)	=MOD(FLOOR(C9/2,1),2)	=MOD(C9,2)
10	=1+A9	=SIN(2*PI()*A10/16)	=ROUND(7.5*(1+B10),0)		=MOD(FLOOR(C10/8,1),2)	=MOD(FLOOR(C10/4,1),2)	=MOD(FLOOR(C10/2,1),2)	=MOD(C10,2)

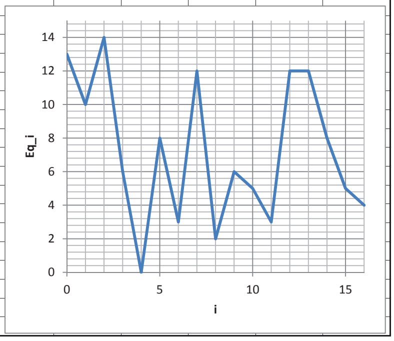
Generating 4-bit random binary sequence. Note that the last random digit value in each column is the seed for the next column.

	A	B	C	D	E	F	G	H	I	J
1	A=	B=	h=	seed=						
2	1234	8765	10000	13						
3										
4	i	Rand Int					RBS			
5	0	13	7267	8083	8547					
6	1	4807	6243	3187	5763		0	1	0	1
7	2	603	2627	1523	307		0	0	0	0
8	3	2867	483	8147	7603		0	0	1	1
9	4	6643	4787	2163	867		1	0	0	0
10	5	6227	5923	7907	8643		1	1	1	1
11	6	2883	7747	6003	4227		0	1	1	0
12	7	6387	8563	6467	4883		1	1	1	0
13	8	323	5507	9043	4387		0	1	1	0
14	9	7347	4403	7827	2323		1	0	1	0
15	10	4963	2067	7283	5347		0	0	1	1
16	11	3107	9443	5987	6963		0	1	1	1
17	12	2803	1427	6723	1107		0	0	1	0
18	13	7667	9683	4947	4803		1	1	0	0
19	14	9843	7587	3363	5667		1	1	0	1
20	15	5027	1123	8707	1843		1	0	1	0
21	16	2083	4547	3203	3027		0	0	0	0
22	17	9187	9763	1267	4083		1	1	0	0
23	18	5523	6307	2243	7187		1	1	0	1
24	19	4147	1603	6627	7523		0	0	1	1
25	20	6163	6867	6483	2147		1	1	1	0
26	21	3907	2643	8787	8163		0	0	1	1
27	22	3	227	1923	1907		0	0	0	0
28	23	2467	8883	1747	2003		0	1	0	0
29	24	3043	387	4563	467		0	0	0	0
30	25	3827	6323	9507	5043		0	1	1	1
31	26	1283	1347	403	1827		0	0	0	0
32	27	1987	963	6067	3283		0	0	1	0
33	28	723	7107	5443	9987		0	1	1	1
34	29	947	8803	5427	2723		0	1	1	0
35	30	7363	1667	5683	8947		1	0	1	1
36	31	4707	5843	1587	9363		0	1	0	1
37	32	7203	9027	7123	2707		1	1	1	0
38	33	7267	8083	8547	9203		1	1	1	1

	A	B	C	D	E	F	G	H	I	J
1	A=	B=	h=	seed=						
2	1234	8765	10000	13						
3										
4	i	Rand Int					RBS			
5	0	=D2	=B38	=C38	=D38					
6	=1+A5	=MOD($A$2*B5+$B$2,$C$2)	=MOD($A$2*C5+$B$2,$	=MOD($A$2*D5+$B$2,$	=MOD($A$2*E5+$B$2,$		=IF(B6<$C$2/2,0,1)	=IF(C6<$C$2/2,0,1)	=IF(D6<$C$2/2,0,1)	=IF(E6<$C$2/2,0,1)
7	=1+A6	=MOD($A$2*B6+$B$2,$C$2)	=MOD($A$2*C6+$B$2,$	=MOD($A$2*D6+$B$2,$	=MOD($A$2*E6+$B$2,$		=IF(B7<$C$2/2,0,1)	=IF(C7<$C$2/2,0,1)	=IF(D7<$C$2/2,0,1)	=IF(E7<$C$2/2,0,1)
8	=1+A7	=MOD($A$2*B7+$B$2,$C$2)	=MOD($A$2*C7+$B$2,$	=MOD($A$2*D7+$B$2,$	=MOD($A$2*E7+$B$2,$		=IF(B8<$C$2/2,0,1)	=IF(C8<$C$2/2,0,1)	=IF(D8<$C$2/2,0,1)	=IF(E8<$C$2/2,0,1)
9	=1+A8	=MOD($A$2*B8+$B$2,$C$2)	=MOD($A$2*C8+$B$2,$	=MOD($A$2*D8+$B$2,$	=MOD($A$2*E8+$B$2,$		=IF(B9<$C$2/2,0,1)	=IF(C9<$C$2/2,0,1)	=IF(D9<$C$2/2,0,1)	=IF(E9<$C$2/2,0,1)
10	=1+A9	=MOD($A$2*B9+$B$2,$C$2)	=MOD($A$2*C9+$B$2,$	=MOD($A$2*D9+$B$2,$	=MOD($A$2*E9+$B$2,$		=IF(B10<$C$2/2,0,1)	=IF(C10<$C$2/2,0,1)	=IF(D10<$C$2/2,0,1)	=IF(E10<$C$2/2,0,1)

Encryption:

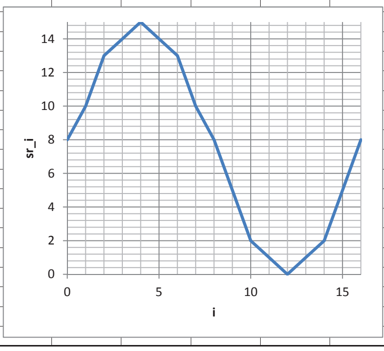
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	D					RBS					E					i	Eq_i
2	1	0	0	0		0	1	0	1		1	1	0	1		0	13
3	1	0	1	0		0	0	0	0		1	0	1	0		1	10
4	1	1	0	1		0	0	1	1		1	1	1	0		2	14
5	1	1	1	0		1	0	0	0		0	1	1	0		3	6
6	1	1	1	1		1	1	1	1		0	0	0	0		4	0
7	1	1	1	0		0	1	1	0		1	0	0	0		5	8
8	1	1	0	1		1	1	1	0		0	0	1	1		6	3
9	1	0	1	0		0	1	1	0		1	1	0	0		7	12
10	1	0	0	0		1	0	1	0		0	0	1	0		8	2
11	0	1	0	1		0	0	1	1		0	1	1	0		9	6
12	0	0	1	0		0	1	1	1		0	1	0	1		10	5
13	0	0	0	1		0	0	1	0		0	0	1	1		11	3
14	0	0	0	0		1	1	0	0		1	1	0	0		12	12
15	0	0	0	1		1	1	0	1		1	1	0	0		13	12
16	0	0	1	0		1	0	1	0		1	0	0	0		14	8
17	0	1	0	1		0	0	0	0		0	1	0	1		15	5
18	1	0	0	0		1	1	0	0		0	1	0	0		16	4



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	D					RBS					E					i	Eq_i
2	1	0	0	0		0	1	0	1		=IF(OR(AND(A2,NOT(F2)),AND(NOT(A2),F2)),1,0)	=IF(OR(AND(B2,NOT(G2)),AND(NOT(B2),G2)),1,0)	=IF(	=IF(		0	=8*K2+4*L2+2*M2+N2
3	1	0	1	0		0	0	0	0		=IF(OR(AND(A3,NOT(F3)),AND(NOT(A3),F3)),1,0)	=IF(OR(AND(B3,NOT(G3)),AND(NOT(B3),G3)),1,0)	=IF(	=IF(		=1+P2	=8*K3+4*L3+2*M3+N3
4	1	1	0	1		0	0	1	1		=IF(OR(AND(A4,NOT(F4)),AND(NOT(A4),F4)),1,0)	=IF(OR(AND(B4,NOT(G4)),AND(NOT(B4),G4)),1,0)	=IF(	=IF(		=1+P3	=8*K4+4*L4+2*M4+N4
5	1	1	1	0		1	0	0	0		=IF(OR(AND(A5,NOT(F5)),AND(NOT(A5),F5)),1,0)	=IF(OR(AND(B5,NOT(G5)),AND(NOT(B5),G5)),1,0)	=IF(	=IF(		=1+P4	=8*K5+4*L5+2*M5+N5
6	1	1	1	1		1	1	1	1		=IF(OR(AND(A6,NOT(F6)),AND(NOT(A6),F6)),1,0)	=IF(OR(AND(B6,NOT(G6)),AND(NOT(B6),G6)),1,0)	=IF(	=IF(		=1+P5	=8*K6+4*L6+2*M6+N6
7	1	1	1	0		0	1	1	0		=IF(OR(AND(A7,NOT(F7)),AND(NOT(A7),F7)),1,0)	=IF(OR(AND(B7,NOT(G7)),AND(NOT(B7),G7)),1,0)	=IF(	=IF(		=1+P6	=8*K7+4*L7+2*M7+N7
8	1	1	0	1		1	1	1	0		=IF(OR(AND(A8,NOT(F8)),AND(NOT(A8),F8)),1,0)	=IF(OR(AND(B8,NOT(G8)),AND(NOT(B8),G8)),1,0)	=IF(	=IF(		=1+P7	=8*K8+4*L8+2*M8+N8
9	1	0	1	0		0	1	1	0		=IF(OR(AND(A9,NOT(F9)),AND(NOT(A9),F9)),1,0)	=IF(OR(AND(B9,NOT(G9)),AND(NOT(B9),G9)),1,0)	=IF(	=IF(		=1+P8	=8*K9+4*L9+2*M9+N9
10	1	0	0	0		1	0	1	0		=IF(OR(AND(A10,NOT(F10)),AND(NOT(A10),F10)),1,0)	=IF(OR(AND(B10,NOT(G10)),AND(NOT(B10),G10)),1,0)	=IF(	=IF(		=1+P9	=8*K10+4*L10+2*M10+N10

Decryption:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	E					RBS					$D_{R,i}$					i	sr_i
2	1	1	0	1		0	1	0	1		1	0	0	0		0	8
3	1	0	1	0		0	0	0	0		1	0	1	0		1	10
4	1	1	1	0		0	0	1	1		1	1	0	1		2	13
5	0	1	1	0		1	0	0	0		1	1	1	0		3	14
6	0	0	0	0		1	1	1	1		1	1	1	1		4	15
7	1	0	0	0		0	1	1	0		1	1	1	0		5	14
8	0	0	1	1		1	1	1	0		1	1	0	1		6	13
9	1	1	0	0		0	1	1	0		1	0	1	0		7	10
10	0	0	1	0		1	0	1	0		1	0	0	0		8	8
11	0	1	1	0		0	0	1	1		0	1	0	1		9	5
12	0	1	0	1		0	1	1	1		0	0	1	0		10	2
13	0	0	1	1		0	0	1	0		0	0	0	1		11	1
14	1	1	0	0		1	1	0	0		0	0	0	0		12	0
15	1	1	0	0		1	1	0	1		0	0	0	1		13	1
16	1	0	0	0		1	0	1	0		0	0	1	0		14	2
17	0	1	0	1		0	0	0	0		0	1	0	1		15	5
18	0	1	0	0		1	1	0	0		1	0	0	0		16	8



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	E					RBS					$D_{R,i}$					i	sr_i
2	1	1	0	1		0	1	0	1		=IF(OR(AND(A2,NOT(F2)),AND(NOT(A2),F2)),1,0)	=IF(OR(AND(B2,NOT(G2)),AND(NOT(B2),G2)),1,0)	=IF(	=IF(		0	=8*K2+4*L2+2*M2+N2
3	1	0	1	0		0	0	0	0		=IF(OR(AND(A3,NOT(F3)),AND(NOT(A3),F3)),1,0)	=IF(OR(AND(B3,NOT(G3)),AND(NOT(B3),G3)),1,0)	=IF(	=IF(		=1+P2	=8*K3+4*L3+2*M3+N3
4	1	1	1	0		0	0	1	1		=IF(OR(AND(A4,NOT(F4)),AND(NOT(A4),F4)),1,0)	=IF(OR(AND(B4,NOT(G4)),AND(NOT(B4),G4)),1,0)	=IF(	=IF(		=1+P3	=8*K4+4*L4+2*M4+N4
5	0	1	1	0		1	0	0	0		=IF(OR(AND(A5,NOT(F5)),AND(NOT(A5),F5)),1,0)	=IF(OR(AND(B5,NOT(G5)),AND(NOT(B5),G5)),1,0)	=IF(	=IF(		=1+P4	=8*K5+4*L5+2*M5+N5
6	0	0	0	0		1	1	1	1		=IF(OR(AND(A6,NOT(F6)),AND(NOT(A6),F6)),1,0)	=IF(OR(AND(B6,NOT(G6)),AND(NOT(B6),G6)),1,0)	=IF(	=IF(		=1+P5	=8*K6+4*L6+2*M6+N6
7	1	0	0	0		0	1	1	0		=IF(OR(AND(A7,NOT(F7)),AND(NOT(A7),F7)),1,0)	=IF(OR(AND(B7,NOT(G7)),AND(NOT(B7),G7)),1,0)	=IF(	=IF(		=1+P6	=8*K7+4*L7+2*M7+N7
8	0	0	1	1		1	1	1	0		=IF(OR(AND(A8,NOT(F8)),AND(NOT(A8),F8)),1,0)	=IF(OR(AND(B8,NOT(G8)),AND(NOT(B8),G8)),1,0)	=IF(	=IF(		=1+P7	=8*K8+4*L8+2*M8+N8
9	1	1	0	0		0	1	1	0		=IF(OR(AND(A9,NOT(F9)),AND(NOT(A9),F9)),1,0)	=IF(OR(AND(B9,NOT(G9)),AND(NOT(B9),G9)),1,0)	=IF(	=IF(		=1+P8	=8*K9+4*L9+2*M9+N9
10	0	0	1	0		1	0	1	0		=IF(OR(AND(A10,NOT(F10)),AND(NOT(A10),F10)),1,0)	=IF(OR(AND(B10,NOT(G10)),AND(NOT(B10),G10)),1,0)	=IF(	=IF(		=1+P9	=8*K10+4*L10+2*M10+N10

)

Project 9.6 (Secure Key Transmission) *Using Example 13.51 as a guide, illustrate transmission of the key. If x and y are both ≤ 15, what is the maximum value of N that still produces successful key transmission?*

(ans: N=21 works:

	A	B	C	D	E	F
1	a	N		T		R
2	11	21				
3				x		y
4				9		11
5						
6				a^x		a^y
7				2357947691		285311670611
8						
9				X		Y
10				8		2
11						
12				Y^x		X^y
13				512		8589934592
14						
15				K_T		K_R
16				8		8

N=22 does work even though `F13` *is displayed in scientific notation, the key values match:*

	A	B	C	D	E	F
1	a	N		T		R
2	11	22				
3				x		y
4				9		11
5						
6				a^x		a^y
7				2357947691		285311670611
8						
9				X		Y
10				11		11
11						
12				Y^x		X^y
13				2357947691		2.85312E+11
14						
15				K_T		K_R
16				11		11

N≥45 does not work because it exceeds Excel's number system, as indicated in row 16.:

	A	B	C	D	E	F
1	a	N		T		R
2	11	45				
3				x		y
4				9		11
5						
6				a^x		a^y
7				2357947691		285311670611
8						
9				X		Y
10				26		41
11						
12				Y^x		X^y
13				327381934393961		3.67034E+15
14						
15				K_T		K_R
16				#NUM!		#NUM!

)

Additional project:

Project 9.7 (Encrypting images) *Use Project 2.3 to form a 4-pixel image defined by 12 bytes. Form* $8 \times 12 = 96$ *random bits to encrypt the image colors and display the encrypted image below the original image. Use the same random bits to decrypt the encrypted image and display the decrypted image below the encrypted image.*