UCSD ECE 154C
Communications Systems III
Prof. Jack Keil Wolf

Solutions to ECE 154C Problem Set 3

1. **Solution:**
   Since the window size is 32, we need 5 bits to encode the position of the matches. Refer to following pages for the procedure.

2. **Solution:**
   We will compute the case for 2 bits. The other cases can be done in the same way. However it can also be done by using Matlab as follows. Let the initial values for $a_i's$ and $b_i's$ as the following.

$$a = [-1.66, -0.589, 0.589, 1.665]$$
$$b = [-\infty, -1.177, 0.1.177, \infty]$$

Since $f_X(x)$ is symmetric, we only need to compute one side of the region. Using the following formula for the Lloyd Algorithm, we acquire the iterated values of the $a_i's$ and $b_i's$.

$$a_j = \frac{\int_{b_{j-1}}^{b_j} x f_X(x)dx}{\int_{b_{j-1}}^{b_j} f_X x dx} = \frac{-\frac{1}{\sqrt{2\pi}} e^{-1/2x}|_{b_{j-1}}^{b_j}}{Q(b_j) - Q(b_{j-1})}$$

$$b_j = \frac{a_{j+1} + a_j}{2}$$

The converged values are

$$a = [-1.5104, -0.4528, 0.4528, 1.5104]$$
$$b = [-\infty, -0.9816, 0, 0.9816, \infty]$$

Now we compute average MSE.

$$\epsilon^2 = \sum_{j=1}^{4} \int_{b_{j-1}}^{b_j} (x - a_j)^2 f_X(x)dx$$
$$= 0.0328 + 0.0259 + 0.0259 + 0.0328 = 0.117$$

Let's check if this value satisfies the following inequality, where $R = 2$ and $\sigma^2 = 1$.

$$R \geq \frac{1}{2} log_2(\frac{\sigma^2}{\epsilon^2})$$

Then

$$2^{2R} \geq \frac{\sigma^2}{\epsilon^2}$$

$$\epsilon^2 \geq \frac{\sigma^2}{2^{2R}} = 1/16$$

Hence the computed MSE satisfies the rate distortion function. We can compute the case for 3, 4, 5 bits in the same way. The following values were found with Matlab.

For R=3,

$$a = [-2.1519, -1.3439, -0.7560, -0.2451, 0.2451, 0.7560, 1.3439, 2.1519]$$
$$b = [-\infty, -1.7479, -1.0500, -0.5005, 0, 0.5005, 1.0500, 1.7479, \infty]$$
$$MSE = 0.0345$$
$$\frac{\sigma^2}{2^{2R}} = 0.0156$$

For R=4,

$$a = [-2.7326, -2.0690, -1.6180, -1.2562, -0.9423, -0.6568, -0.3880, -0.1284, 0.1284, 0.3880, 0.6568, 0.9423, 1.2562, 1.6180, 2.0690, 2.7326]$$
$$b = [-\infty, -2.4008, -1.8435, -1.4371, -1.0993, -0.7995, -0.5224, -0.2582, 0, 0.2582, 0.5224, 0.7995, 1.0993, 1.4371, 1.8435, 2.4008, \infty]$$
$$MSE = 0.0095$$
$$\frac{\sigma^2}{2^{2R}} = 0.0039$$

For R=5,

$$a = [-3.2609, -2.6913, -2.3179, -2.0289, -1.7874, -1.5764, -1.3865, -1.2120, -1.0489, -0.8947, -0.7472, -0.6050, -0.4668, -0.3314, -0.1981, -0.0659, 0.0659, 0.1981, 0.3314, 0.4668, 0.6050, 0.7472, 0.8947, 1.0489, 1.2120, 1.3865, 1.5764, 1.7874, 2.0289, 2.3179, 2.6913, 3.2609]$$
$$b = [-\infty, -2.9761, -2.5046, -2.1734, -1.9081, -1.6819, -1.4814, -1.2992, -1.1304, -0.9718, -0.8210, -0.6761, -0.5359, -0.3991, -0.2648, -0.1320, 0, 0.1320, 0.2648, 0.3991, 0.5359, 0.6761, 0.8210, 0.9718, 1.1304, 1.2992, 1.4814, 1.6819, 1.9081, 2.1734, 2.5046, 2.9761, \infty]$$
$$MSE = 0.0025$$
$$\frac{\sigma^2}{2^{2R}} = 0.0010$$

```
function gaussian_quant
bits = 5;
M = 2^bits;
b = zeros(1, M+1);
a = zeros(1, M);
MSE = 0;
b(1) = -inf;
b(M+1) = inf;
for i = 2: (M);
    b(i) = -8 +16/M*(i-1);
end
for i = 1:M
    a(i) = (b(i) + b(i+1))/2;
end
for j=1:1000
    for k = 1:M
        a(k) = ( ( -1/sqrt(2*pi)* exp(-0.5 * b(k+1)^2) )...
         - (-1/sqrt(2*pi)* exp(-0.5 * b(k)^2) ) ) / ...
         ( normcdf(b(k+1)) - normcdf(b(k)) );
    end
    for l = 2:M
        b(l) = ( a(l-1) + a(l))/2;
    end
end
a
b
b(1) = -9999999;
b(M+1) = 99999999;
for i = 1:M
    x_1 = b(i);
    x_2 = b(i+1);
    y = a(i);
    r_1 = -1/sqrt(2*pi) * x_1 * exp(-0.5*x_1^2) + ...
    2*y/sqrt(2*pi)*exp(-0.5*x_1^2) + (y^2+1)*normcdf(x_1);
    r_2 = -1/sqrt(2*pi) * x_2 * exp(-0.5*x_2^2) + ...
    2*y/sqrt(2*pi)*exp(-0.5*x_2^2) + (y^2+1)*normcdf(x_2);
    r = r_2 - r_1;
    MSE = MSE + r;
end
MSE
```

3. **Solution:**

```
function gaussian_simulation
a = [ -2.1519    -1.3439    -0.7560    -0.2451    0.2451    0.7560
1.3439    2.1519];
b = [ -1.7479    -1.0500    -0.5005    -0.0000    0.5005    1.0500
1.7479  ];
total = 10000;
source = randn(1,total);
dist = 0;
for i=1:total
    if(source(i) < b(1))
        dist = dist + (source(i) - a(1))^2;
    elseif(source(i) < b(2) && source(i) >= b(1))
        dist = dist + (source(i) - a(2))^2;
    elseif(source(i) < b(3) && source(i) >= b(2))
        dist = dist + (source(i) - a(3))^2;
    elseif(source(i) < b(4) && source(i) >= b(3))
        dist = dist + (source(i) - a(4))^2;
    elseif(source(i) < b(5) && source(i) >= b(4))
        dist = dist + (source(i) - a(5))^2;
    elseif(source(i) < b(6) && source(i) >= b(5))
        dist = dist + (source(i) - a(6))^2;
    elseif(source(i) < b(7) && source(i) >= b(6))
        dist = dist + (source(i) - a(7))^2;
    else
        dist = dist + (source(i) - a(8))^2;
    end
end

avg = dist/total
```

4. **Solution:**
We can use the Lloyd Algorithm with the following formulas to attain the values for $a_i's$ and $b_i's$. Since the pdf is symmetric we only need to consider one side of the function.

$$a_j = \frac{\int_{b_{j-1}}^{b_j} x f_X(x) dx}{\int_{b_{j-1}}^{b_j} f_X x dx} = \frac{x^2/2 + x^3/3|_{b_{j-1}}^{b_j}}{x + x^2/2|_{b_{j-1}}^{b_j}}$$

$$b_j = \frac{a_{j+1} + a_j}{2}$$

Also we can find the MSE as the following.

$$\begin{aligned}\epsilon^2 &= \sum_{j=1}^{M} \int_{b_{j-1}}^{b_j} (x-a_j)^2 f_X(x)dx \\ &= 2\sum_{j=1}^{M/2} \int_{b_{j-1}}^{b_j} (x-a_j)^2 (1+x)dx\end{aligned}$$

The results were computed using Matlab. For R=2,

$$\begin{aligned}&a = [-0.5880, -0.1760, 0.1760, 0.5880] \\ &b = [-1.0000, -0.3820, 0, 0.3820, 1.0000] \\ &MSE = 0.0155\end{aligned}$$

For R=3,

$$\begin{aligned}&a = [-0.7495, -0.4990, -0.2851, -0.0907, 0.0907, 0.2851, 0.4990, 0.7495] \\ &b = [-1.0000, -0.6243, -0.3920, -0.1879, 0, 0.1879, 0.3920, 0.6243, 1.0000] \\ &MSE = 0.0041\end{aligned}$$

For R=4,

$$\begin{aligned}&a = [-0.8493, -0.6985, -0.5698, -0.4528, -0.3437, -0.2402, -0.1413, -0.0461, \\ &0.0461, 0.1413, 0.2402, 0.3437, 0.4528, 0.5698, 0.6985, 0.8493] \\ &b = [-1.0000, -0.7739, -0.6342, -0.5113, -0.3982, -0.2919, -0.1908, -0.0937, \\ &0, 0.0937, 0.1908, 0.2919, 0.3982, 0.5113, 0.6342, 0.7739, 1.0000] \\ &MSE = 0.0011\end{aligned}$$

For R=5,

$$\begin{aligned}&a = -0.9098, -0.8196, -0.7426, -0.6726, -0.6073, -0.5454, -0.4863, -0.4293, \\ &-0.3741, -0.3205, -0.2683, -0.2173, -0.1674, -0.1185, -0.0705, -0.0232, \\ &0.0232, 0.0705, 0.1185, 0.1674, 0.2173, 0.2683, 0.3205, 0.3741, \\ &0.4293, 0.4863, 0.5454, 0.6073, 0.6726, 0.7426, 0.8196, 0.9098 \\ &b = -1.0000, -0.8647, -0.7811, -0.7076, -0.6400, -0.5764, -0.5158, -0.4578, \\ &-0.4017, -0.3473, -0.2944, -0.2428, -0.1924, -0.1430, -0.0945, -0.0469, \\ &0, 0.0469, 0.0945, 0.1430, 0.1924, 0.2428, 0.2944, 0.3473, \\ &0.4017, 0.4578, 0.5158, 0.5764, 0.6400, 0.7076, 0.7811, 0.8647, 1.0000 \\ &MSE = 2.7000e-004\end{aligned}$$

```
function tri_quant

bits = 5;
M = 2^bits;
b = zeros(1, M+1);
a = zeros(1, M);
MSE = 0;

b(1) = -1;
b(M+1) = 1;

for i = 2: (M);
    b(i) = -1 +2/M*(i-1);
end

for i = 1:M
    a(i) = (b(i) + b(i+1))/2
end
a
b

for j=1:1000

    for k = 1:M/2
        x_1 = b(k);
        x_2 = b(k+1);

        a(k) = ((1/2* x_2^2+1/3*x_2^3 ) - ( 1/2* x_1^2 +1/3*x_1^3))/...
        ( (x_2+1/2*x_2^2) - (x_1 +1/2*x_1^2));
        a(M+1-k) = -a(k);
    end

    for l = 2:M/2
        b(l) = ( a(l-1) + a(l))/2;
        b(M+2-l) = -b(l);
    end
end
a
b

for i = 1:M/2
    x_1 = b(i);
```

```
        x_2 = b(i+1);
        y = a(i);
        r = (1/4*x_2^4 + 1/3*(1-2*y)*x_2^3+1/2*(y^2-2*y)*x_2^2+y^2*x_2) -...
            (1/4*x_1^4 + 1/3*(1-2*y)*x_1^3+1/2*(y^2-2*y)*x_1^2+y^2*x_1);

        MSE = MSE + 2*r;
    end

    MSE
```

5. **Solution:**
   We can change the a and b values in problem 3 for this problem.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	queue	numberof bits	encoded to
																																M	9	(0, M)
																															M	A	9	(0, A)
																														M	A	R	9	(0, R)
																													M	A	R	Y	9	(0, Y)
																												M	A	R	Y	_	9	(0, _)
																											M	A	R	Y	_	H	9	(0, H)
																										M	A	R	Y	_	H	A	7	(1, 4, 1)
																									M	A	R	Y	_	H	A	D	9	(0, D)
																								M	A	R	Y	_	H	A	D	_	7	(1, 3, 1)
																							M	A	R	Y	_	H	A	D	_	A	7	(1, 2, 1)
																						M	A	R	Y	_	H	A	D	_	A	_	7	(1, 1, 1)
																					M	A	R	Y	_	H	A	D	_	A	_	L	9	(0, L)
																				M	A	R	Y	_	H	A	D	_	A	_	L	I	9	(0, I)
																			M	A	R	Y	_	H	A	D	_	A	_	L	I	T	9	(0, T)
																		M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	7	(1, 0, 1)
																	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	7	(1, 3, 1)
																M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	9	(0, E)
															M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_L	8	(1, 6, 2)
													M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	7	(1, 9, 1)
												M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	9	(0, M)
											M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	9	(0, B)
										M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_LITTLE_LAMB	18	(1, 11, 12)
R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_LITTLE_LAMB_	19	(1, 23, 13)
Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	7	(1, 2, 1)
_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	7	(1, 4, 1)
H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	9	(0, R)
A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	9	(0, Y)
D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	7	(1, 4, 1)
_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	9	(0, H)
A	_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	7	(1, 4, 1)
_	L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	9	(0, D)
L	I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	7	(1, 3, 1)
I	T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	7	(1, 2, 1)
T	T	L	E	_	L	A	M	B	_	L	I	T	T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_LITTLE_LAMB_	19	(1, 22, 13)
T	L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	9	(0, W)

L	E	_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	IT	8	(1, 11, 2)
_	L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	7	(1, 20, 1)
L	A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	7	(1, 4, 1)
A	M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	9	(0, F)
M	B	_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	LE	8	(1, 13, 2)
_	M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	7	(1, 0, 1)
M	A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	9	(0, C)
A	R	Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E_	8	(1, 16, 2)
Y	_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	7	(1, 15, 1)
_	H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	9	(0, S)
H	A	D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_W	8	(1, 14, 2)
D	_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	7	(1, 12, 1)
_	A	_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	IT	8	(1, 15, 2)
_	L	I	T	T	L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E_AS_	11	(1, 8, 5)
L	E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	7	(1, 1, 1)
E	_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	9	(0, N)
_	L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	9	(0, W)
L	A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	7	(1, 11, 1)
A	M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	9	(0, .)
M	B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	7	(1, 5, 1)
B	_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	7	(1, 3, 1)
_	W	I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H_	8	(1, 27, 2)
I	T	H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	WH	8	(1, 17, 2)
H	_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	7	(1, 15, 1)
_	F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	9	(0, R)
F	L	E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E_	8	(1, 17, 2)
E	E	C	E	_	A	S	_	W	H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	OH_WHERE_	15	(1, 8, 9)
H	I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	7	(1, 4, 1)
I	T	E	_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	AS_	9	(1,27, 3)
_	A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	9	(0, M)
A	S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	9	(0, Y)
S	_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	7	(1, 2, 1)
_	S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	9	(0, L)
S	N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	9	(0, I)
N	O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	9	(0, T)
O	W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	7	(1, 0, 1)

W	.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	7	(1, 3, 1)
.	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E_	8	(1, 13, 2)
O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	9	(0, D)
H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	7	(1, 23, 1)
_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	9	(0, G)
W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	7	(1, 3, 1)
H	E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	G	7	(1, 1, 1)
E	R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	G	O	7	(1, 3, 1)
R	E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	G	O	N	9	(0, N)
E	_	O	H	_	W	H	E	R	E	_	H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	G	O	N	E_OH_WHERE_	17	(1, 31, 11)
H	A	S	_	M	Y	_	L	I	T	T	L	E	_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	OH_WHERE_	15	(1, 8, 9)
T	T	L	E	_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	9	(0, C)
T	L	E	_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	9	(0, A)
L	E	_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	7	(1, 22, 1)
E	_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	7	(1, 3, 1)
_	D	O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	HE	8	(1, 8, 2)
O	G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	H	E	_	7	(1, 2, 1)
G	_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	H	E	_	B	9	(0, B)
_	G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	H	E	_	B	E	7	(1, 2, 1)
G	O	N	E	_	O	H	_	W	H	E	R	E	_	O	H	_	W	H	E	R	E	_	C	A	N	_	H	E	_	B	E			

Total bits 780

Total characters 168

binary digits/ source letter 4,642857143