

Hamming Code

(ハミング符号)

1950年にベル研究所のリチャード・ハミングによって考案された。知られている誤り訂正符号の中では最も古く、ブロックあたり1ビットの誤りを訂正できる。リード・ソロモン符号などに比べると、ある程度高速で処理できるが、訂正力は高くない。このためエラー発生率が低く、速度が要求される用途に使う。ECCメモリやRAID 2などに使用される。(Wikipedia)

符号長： $n = 2^m - 1$

情報ビット数： $k = n - m$

つまり、情報 k ビットに対し、
 $n - k$ ビット付加して 1 誤りの訂正を行う。
 通常 (n, k) ハミング符号と呼ぶ。

例 $(7, 4)$ ハミング符号 (符号長 7、情報長 4)

29年 III 27 の例は、 $(7, 4)$ ハミング符号で、
 $n = 7, k = 4$ である。

情報ビットに付加した $n - k = 3$ ビットの内容は、

$$c_1 = (x_1 + x_2 + x_3) \bmod 2$$

$$c_2 = (x_2 + x_3 + x_4) \bmod 2$$

$$c_3 = (x_1 + x_2 + x_4) \bmod 2$$

表 2 の検査行列 H は、表 1 の右 3 ビットで、
 チェック bit c_1, c_2, c_3 (左半分 3 行 4 列)
 と単位行列 (右半分 3 行 3 列) から表 2 の
 ように作成される。受信符号 7 ビットの
 y から $S = Hy^T$ を $\bmod 2$ で計算して、シンδροームを求め、どのビットが誤りかを知る。

表 1 行列 $G = I^4 A^T$, ($H = AI^3$, $HG^T = GH^T = 0$)

誤りがある位置 (1がある点)				シンδροーム		
e1	e2	e3	e4	c1	c2	c3
1	0	0	0	1	0	1
0	1	0	0	1	1	1
0	0	1	0	1	1	0
0	0	0	1	0	1	1

誤りが第 3 ビットであるとき、シンδροームは 110 となる。(赤線)
 誤りがないときはシンδροームは 000 となる。

表 2 検査行列 $H = AI^3$ シンδροーム $S = Hy^T$, y は 7 ビット受信符号

誤りビット	1	2	3	4	5	6	7
シンδροーム $Hy^T \bmod 2$	1	1	1	0	1	0	0
	0	1	1	1	0	1	0
	1	1	0	1	0	0	1

シンδροームには $[0 0 0]$ が含まれず、かつ、同じものが 2 か所以上出現しないようにしなければならない。

使用例 1

$y = [1\ 001001]$ を受信、正誤を調べる。

$$S = Hy^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \text{mod} 2 \Rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{から誤りは第2ビットと判定}$$

正しい $y = [1\ 1\ 0\ 1\ 0\ 0\ 1]$

使用例 2

$y = [1101101]$ を受信、正誤を調べる。

$$S = Hy^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} \text{mod} 2 \Rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{から誤りは第5ビットと判定}$$

正しい $y = [1\ 1\ 0\ 1\ 0\ 0\ 1]$

$HG^T = 0$ の計算

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \text{mod}2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$GH^T = 0$ の計算

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{mod}2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

c_1, c_2, c_3 の順序を変えても成り立つか?、 c_2, c_3 を入れ替えてみる。

Cの順序を変えた場合 G, Hが変わる

$HG^T = 0$ の計算

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \text{ mod } 2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ OK}$$

$GH^T = 0$ の計算

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ mod } 2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ OK}$$

1. 符号化の方法, 送信したいデータ 4 ビットとG との mod2 乗算を行う

送信したいデータ = [1 1 0 1]とする。

$$[1 \ 1 \ 0 \ 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [\ 1 \ 1 \ 0 \ 1 \ \underline{0 \ 1 \ 0}]$$

情報ビット 検査ビット

参考 送信したいデータ = [1 1 0 1]とG とのmod2乗算によって、

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

データを左4ビットに取り出す

c1 c2 c3
を計算し
右3ビットに。

2. チェックの方法, Hと受信したデータ7ビットのmod2乗算を行いシンドロームを求める。

受信データを、 $y=[1\ 1\ 0\ 0\ 0\ 1\ 0]$ とする。

表2 検査行列H シンドローム $S=Hy^T$, y は7ビット受信符号

誤りビット	1	2	3	4	5	6	7
シンドローム	1	1	1	0	1	0	0
$Hy^T \text{ mod } 2$	1	1	0	1	0	1	0
	0	1	1	1	0	0	1

Hの作り方

Gの右4行3列を転置してHの左4列3行に配置する。

次にその右に3行3列の単位行列を付加する。

検査例1 $y=[1\ 1\ 0\ 0\ 0\ 1\ 0]$ の場合

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ mod } 2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

誤りは
第4ビット

$w=[1\ 1\ 0\ 1\ 0\ 1\ 0]$ が正解, OK!

検査例2 $w=[1\ 1\ 0\ 1\ 0\ 1\ 0]$ の場合

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ mod } 2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

誤りなし