

อธิบาย CRC & Hamming Code พร้อมยกตัวอย่าง

CRC (Cyclic Redundancy Checking) คือ เทคนิคการตรวจสอบหาข้อผิดพลาดในการบันทึกหรือถ่ายทอข้อมูล มักจะใช้วิธีการวนซ้ำ เพื่อทบทวนดูความแม่นยำ ถ้ามีที่ผิด การทำสองครั้งย่อมจะให้ผลไม่ตรงกัน

ตัวอย่าง

– เลขบัตรประชาชน หรือไอดีต่าง ๆ ส่วนใช้แต่หลัก CRC

- การ error checking ข้อมูลใน Hard drive, CD และ DVD เป็นต้น

CRC Example

original data from sender = 1011101

divisor = 10110 ← polynomial $G(x) = x^4 + x^3 + x$

Append = จำนวน bit divisor - 1 = 5 - 1 = 4

10110 | 1 0 1 1 1 0 1 0 0 0 0 ← append

10110		1	0	1	1	1	0	1	0	0	0	0
		1	0	1	1	0						
		<hr/>										
		0	0	0	1	0						
		<hr/>										
		0	0	0	0	0						
		<hr/>										
		0	0	1	0	1						
		<hr/>										
		0	0	0	0	0						
		<hr/>										
		0	0	1	0	1	0					
		<hr/>										
		0	0	0	0	0						
		<hr/>										
		1	0	1	0	0						

$$\begin{array}{r} 10110 \\ \underline{1000} \end{array} \text{Reminder}$$

data ที่ถูกส่งคือ 10111011000

ถ้าหากว่า data ที่ถูก encode ส่งออกไปมี error หรือ
ในระหว่างที่กับ divisor อีกที ถ้า $\text{reminder} = 0$ แสดงว่า
data ไม่มี error แต่ถ้า $\neq 0$ แสดงว่า data ที่ส่ง มี
error

Binary form: 10111011000 divided by 10110

Decimal form: 1496 divided by 22

$x^{10} + x^8 + x^7 + x^6 + x^4 + x^3$
 $x^4 + x^2 + x$

Divide operation:

Result is 1000100

Remainder is 0000

Working is

```

      1000100
-----
10111011000
10110
-----
      0001011000
      00000
-----
      001011000
      00000
-----
      01011000
      00000
-----
      1011000
      10110
-----
      000000
      00000
-----

```

```

000000
00000
-----
00000
00000
-----
0000

```

Hamming Codes

มีจุดเด่นตรงที่นอกจากเช็คได้ว่า bit ไหนเกิด error แล้ว ยังแก้ bit ที่ error ในตำแหน่งนั้นให้เป็น bit ที่ถูกต้องอีกด้วย

	p ₁	p ₂	d ₁	p ₃	d ₂	d ₃	d ₄	p ₄	d ₅	d ₆	d ₇	Parity check	Parity bit
Received data word:	1	0	0	0	1	1	0	0	1	0	0		
p ₁	1		0		1		0		1		0	Fail	1
p ₂		0	0			1	0			0	0	Fail	1
p ₃				0	1	1	0					Pass	0
p ₄								0	1	0	0	Fail	1

Checking of parity bits (switched bit highlighted)

The final step is to evaluate the value of the parity bits (remembering the bit with lowest index is the least significant bit, i.e., it goes furthest to the right). The integer value of the parity bits is 11, signifying that the 11th bit in the data word (including parity bits) is wrong and needs to be flipped.

	p ₄	p ₃	p ₂	p ₁	
Binary	1	0	1	1	
Decimal	8		2	1	Σ = 11

เลข parity bit จะอยู่ในตำแหน่ง 2^{n-1} เสมอ (ตำแหน่ง 1,2,4,8,16,32,64)

สมมติให้ส่ง data จำนวน 8 bit : 0110100

Data ที่จะส่งไปต้องมี parity bit สอดแทรกอยู่ด้วย

จะเขียนได้ดังนี้ : 011[]010[]0[][] <<< (parity bit อยู่ในตำแหน่งที่ 1,2,4 และ 8 สอดแทรก

d7-d6-d5-p4-d4-d3-d2-p3-d1-p2-p1 (ไม่แน่ใจว่าถูกไหม)

หาค่า parity แต่ละตำแหน่ง

P1 : check 1 skip 1

P2 : check 2 skip 2

P3 : check 4 skip 4

P4 : check 8 skip 8

P(n) : check 2^{n-1} skip 2^{n-1}

(hint) เริ่มเช็คจากตำแหน่ง parity bit ของตัวเอง เช่น P3 ก็เริ่มเช็คตั้งแต่ตำแหน่ง 2^{3-1} เช็ค 3 ข้าม 3

ถ้า check แล้วจำนวนเลข 1 เป็นคี่ให้ parity bit ตัวนั้นมีค่า 1

แต่ถ้า check แล้วจำนวนเลข 1 เป็นคู่ ให้ parity bit ตัวนั้นมีค่า 0

นำค่า parity bit ทุกตัวมาเรียงกัน

p4-p3-p2-p1 เช่น 1 0 1 1

1 0 1 1 แปลงเป็นฐาน 10 มีค่าเท่ากับ $2^3 + 2^1 + 2^0 = 11$

ดังนั้น bit ที่ 11 ของ data+parity bit เกิด Error ดังนั้นจึงต้องสลับเลขของ bit ตำแหน่งที่ 11 จาก 0 เป็น 1 หรือ 1 เป็น 0