Jan 23, 2007

Binary code

$0, 1, 2, \ldots, 15$ $\qquad$ $n = 4$ length

7	0111
8	1000

4 changes

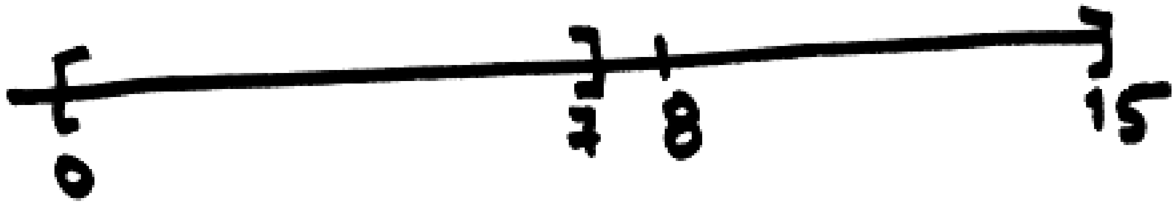
Gray code

A way to encode numbers such that any two consecutive words differ in exactly one slot.

Binary

Recursively

$0\,B_n \quad 1\,B_n$



$\underline{n=3}$ $\quad 0, 1, 2 \ldots, 7$

$\underline{n=4}$ $\quad 0, 1, 2, \ldots, 15$

$n=1$ B_1: 0, 1

$n=2$ B_2: 00, 01, 10, 11

$n=3$ B_3: $\underbrace{000, 001, 010, 011}$, $\underbrace{100, 101, 110, 111}$

Reflected gray Code

n : C_n

$n+1$: $0C_n$, $1C_n'$

$C_n' = C_n$ backwards

1: 0, 1

2: 00, 01, 11, 10

3: 000, 001, 011, 010, 110, 111, 101, 100

0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

There is only two possible moves either

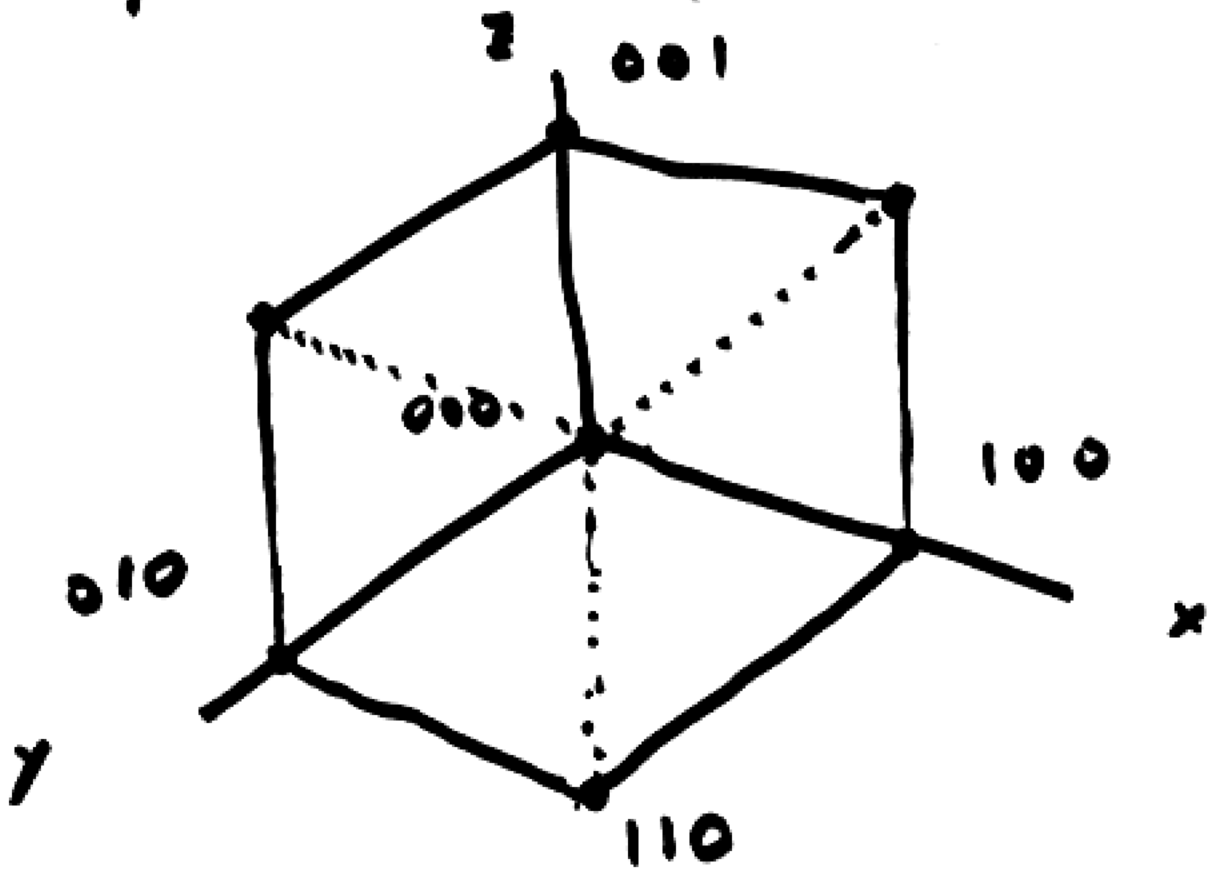
- change rightmost bit
- change the bit to the left of the rightmost 1

$$\dots\dots\ \boxed{x}\ \boxed{1}\ 0 \dots\dots 0$$

$\uparrow$
change

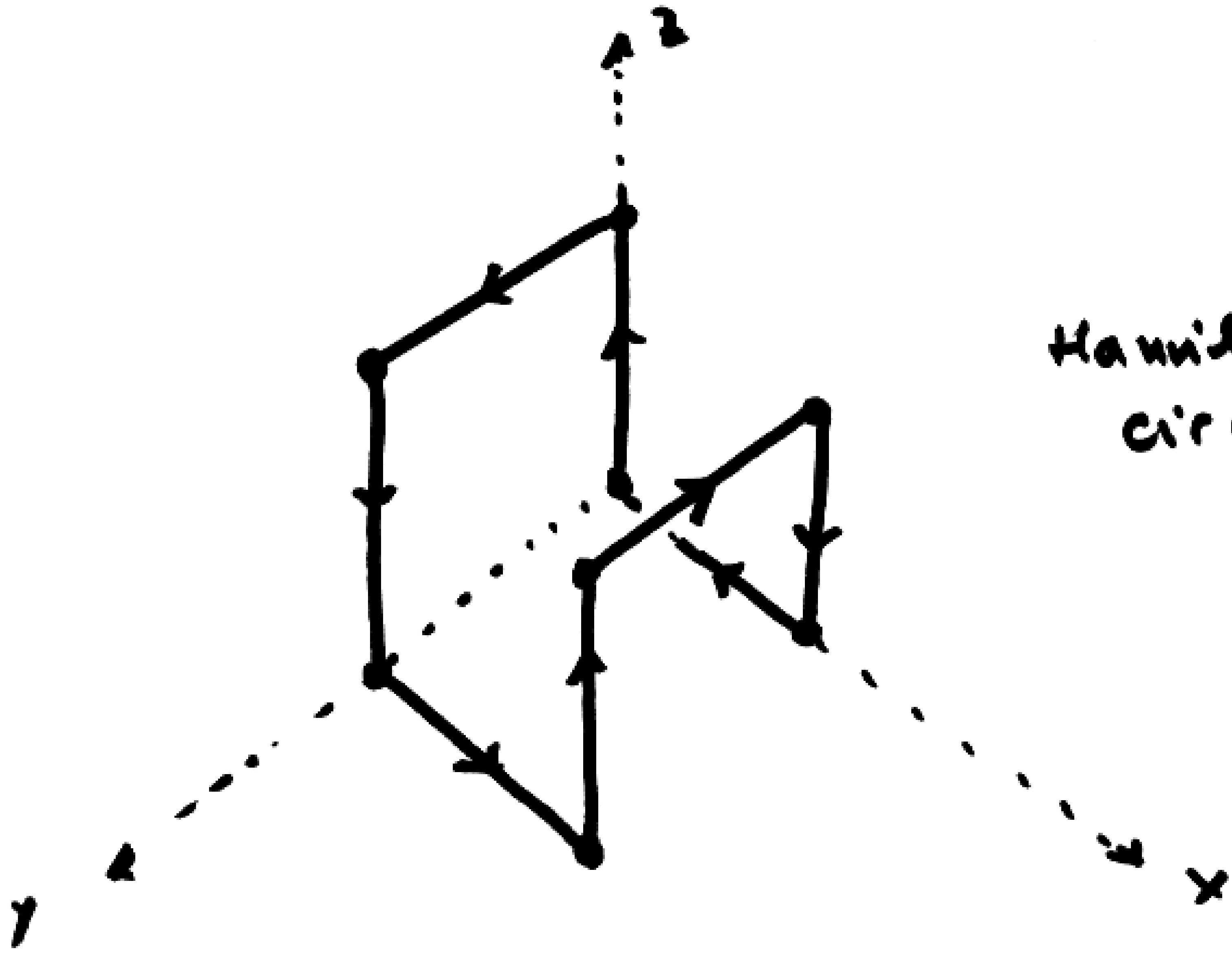
Hamiltonian circuits

Represent length 3 binary words as points on a cube



(x, y, z)

words of length 3 $\leftrightarrow$ vertices of cube.



Hamiltonian circuit

Path on the cube going through every vertex only once.

Gray code of length n

$\updownarrow$

Hamiltonian circuit in the n-cube.

Binary $\longrightarrow$ Gray

$$(b_{n-1}\, b_{n-2} \cdots b_1\, b_0)_2 \leftarrow \text{Binary}$$

$$\downarrow$$

$$(c_{n-1}\, c_{n-2} \cdots c_1\, c_0) \leftarrow \text{Gray}$$

E.g. 13 $(1101)_2$

$$c_j \equiv b_j + b_{j+1} \mod 2$$

+	0	1
0	0	1
1	1	0

mod 2 addition

$0 + 0 \equiv 0 \mod 2$

$0 + 1 \equiv 1 \mod 2$

$1 + 0 \equiv 1 \mod 2$

$1 + 1 \equiv 0 \mod 2$

$1 \oplus 1 = 0$

$$c_j = b_j \oplus b_{j+1}$$

$1 \oplus 0 = 1$

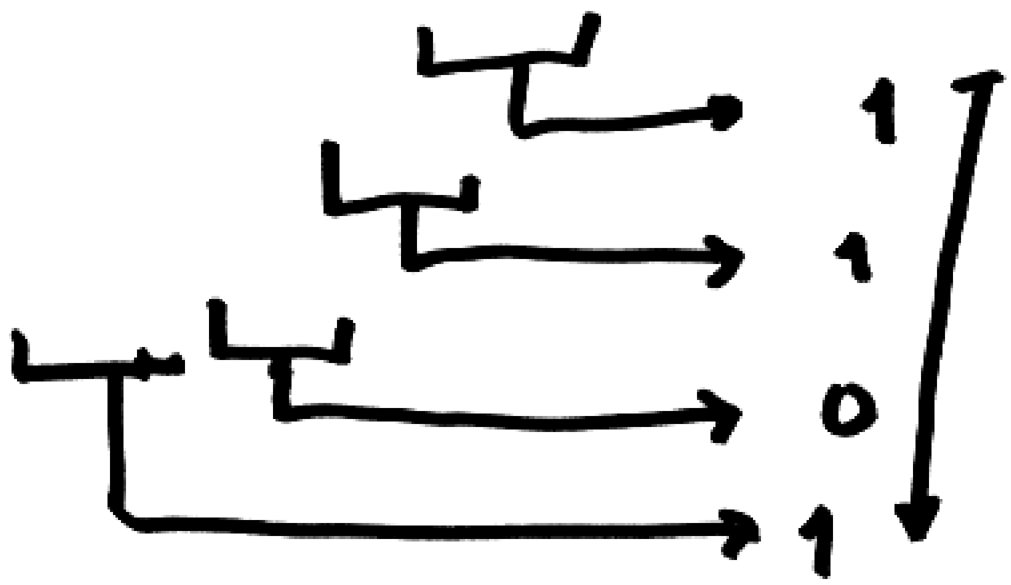
$0 \oplus 1 = 1$

$0 \oplus 0 = 0$

$1 \oplus 1 = 0$

$$c_j = b_j \oplus b_{j+1}$$

13 ...0 1 1 0 1 binary



1011 GRAY

Jan 25, 2007

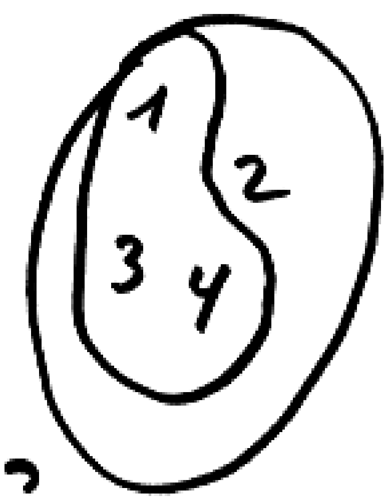
$n = 4$ $\quad\quad$ $0, 1, 2, \ldots, 15$

0 ___
1 ___
2 ___
$\vdots$
15

← strings of 4 bits
words

$\{1, 2, 3, 4\}$

subset: $\{1, 3, 4\}$



How many subsets are there?

$\{1, 3, 4\} = \{1, 4, 3\}$

	1	2	3	4
$\{1,3,4\} \leftrightarrow$	1	0	1	1
$\{2,3,4\} \leftrightarrow$	0	1	1	1
$\phi \mapsto$	0	0	0	0

subset $\leftrightarrow$ word (string 4 bits)

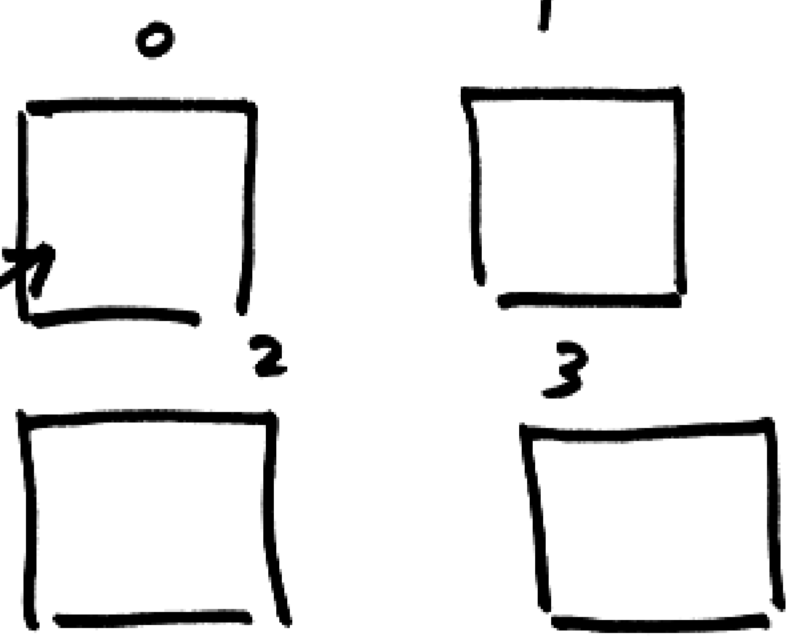
Pblm #8

000	100
001	101
011	111
010	110
110	010
111	011
101	001
100	000

change leftmost bit

Pblm #1

<u>Binary</u>



all numbers
0, 1, 2, ..., 15
that have a 1
in the 0 slot

3210|

<u>Pblm #4</u>

$x = 2^k \cdot y$

$2^x y$
(y is odd)

Highest power of 2
dividing x $= k$

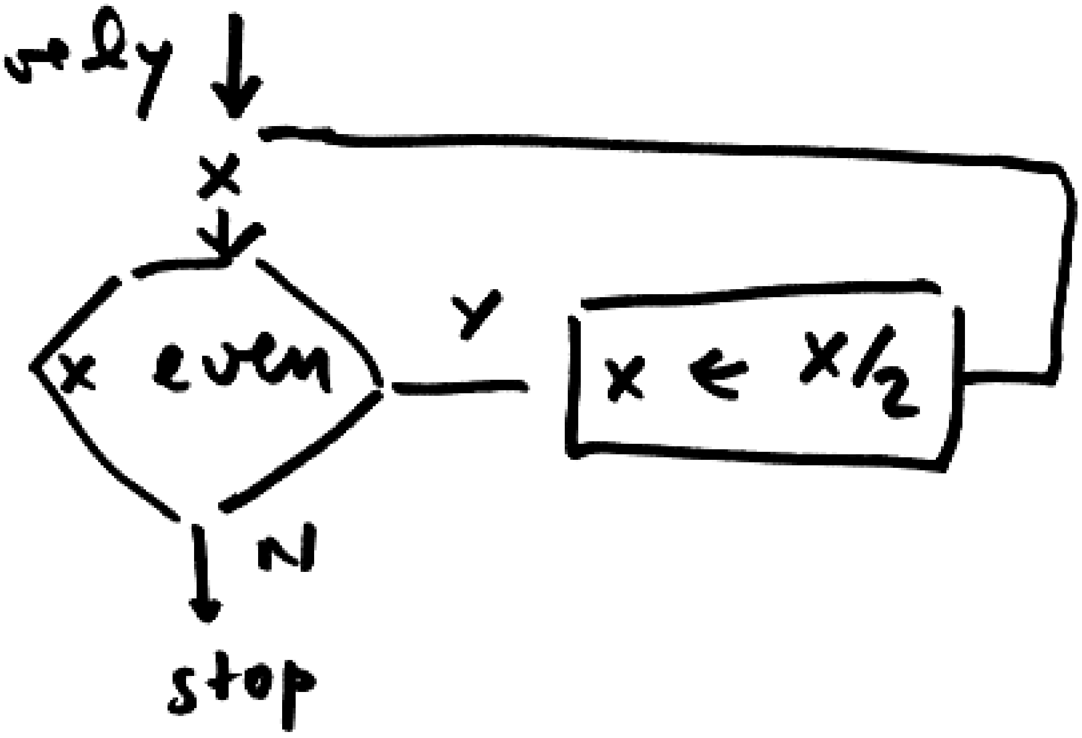
$12 = 2^2 \cdot 3$ ← odd

$\to k = 2$

$7 = 2^0 \cdot 7$

$\to k = 0$

Recursively



K = # times through this loop

$x = 36 \rightarrow 18 \rightarrow \underline{9}$

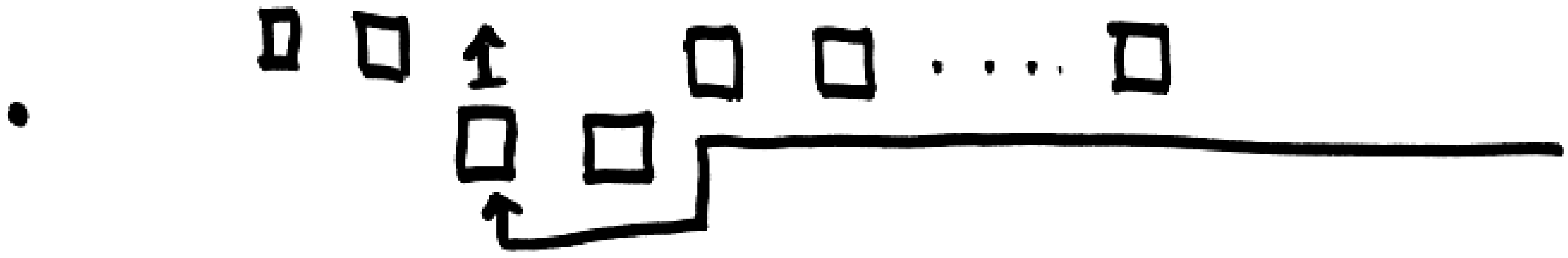
$36 = 2^2 \cdot 9$

Two possible moves in Gray code

- $\ast \ast \ast \ast \cdots \boxed{\ast}\ 1\ 0\ 0 \cdots\cdots 0$

  change

- change first (rightmost)



BINARY $\longleftrightarrow$ GRAY

good way to encode/decode

BINARY $\longmapsto$ GRAY

$$b_{n-1} \cdots b_1 \, b_0 \longmapsto c_{n-1} \cdots c_1 \, c_0$$

MOD 2 ADDITION OF BITS

$\oplus$	0	1
0	0	1
1	1	0

0 $\leftrightarrow$ FALSE

1 $\leftrightarrow$ TRUE

$a \oplus b$

exclusive OR

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

$$a \oplus b = c$$
$$a = b \oplus c$$
$$b = a \oplus c$$

$$c_j = b_j \oplus b_{j+1}$$

(Think of all binary bits being 0 to the left)

E.g. BINARY GRAY

$14 = (1110)_2 \longrightarrow$ 1001

GRAY $\longmapsto$ BINARY

$b_3\, b_2\, b_1\, b_0$

$c_0 = b_0 \oplus b_1$

$c_1 = b_1 \oplus b_2$

$c_2 = b_2 \oplus b_3$

$c_3 = b_3 \oplus b_4 = b_3$ $\quad (b_4 = 0)$

$b_3 = c_3$

$b_2 = c_2 \oplus c_3$

$b_1 = c_1 \oplus b_2 = c_1 \oplus c_2 \oplus c_3$

$b_0 = c_0 \oplus c_1 \oplus c_2 \oplus c_3$

In general

$$b_j = c_j \oplus c_{j+1} \oplus \cdots$$

E.g. GRAY $\longmapsto$ BINARY

$$1111 \longmapsto (1010)_2 = 10$$

$$11111 \longmapsto (10101)_2 = 16+4+1=21$$

$$111111 \longmapsto (101010)_2$$

$$\begin{array}{cccccc} 1 & 0 & 1 & 0 & 1 & 0 \\ 32 & & 8 & & 2 & \end{array}$$

$$32+8+2=42$$

1, 2, 5, 10, 21, 42

↑ # steps to solve spin-out or chinese rings puzzle with 4, 5, 6

$111 \mapsto 101$

$11 \mapsto 10$ 5

<u>n even</u> $\frac{2}{3}(2^{n}-1)$

<u>n odd</u> $\frac{1}{3}(2^{n+1}-1)$

Rule to get out: (right to left)

<u>n even</u> move 2^{nd} bit

<u>n odd</u> " 1^{st} bit