UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

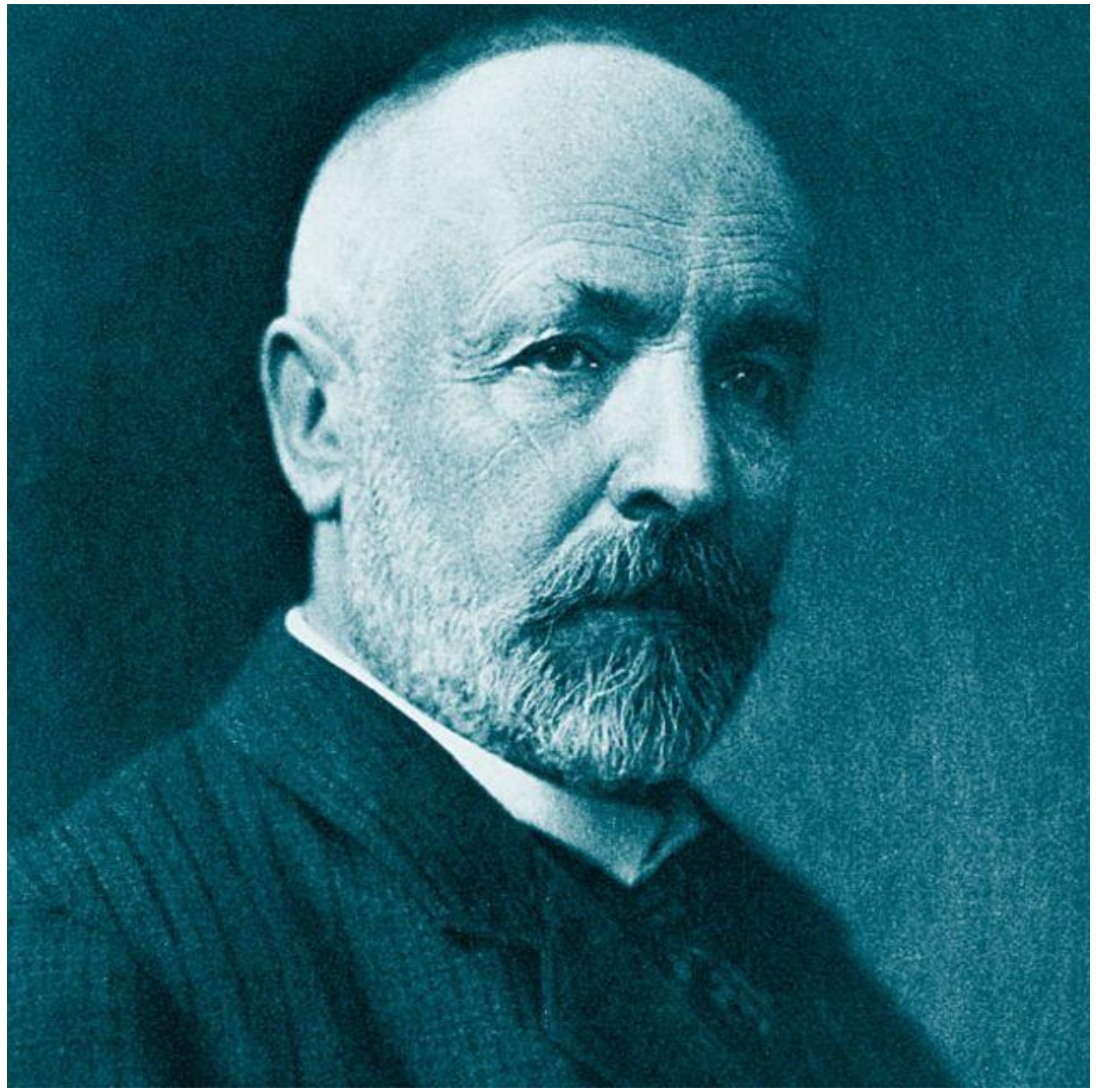
Sesión 6.2: Space-filling

CS3102 EDA

Índice

1. **Space-filling** curves

Infinitos

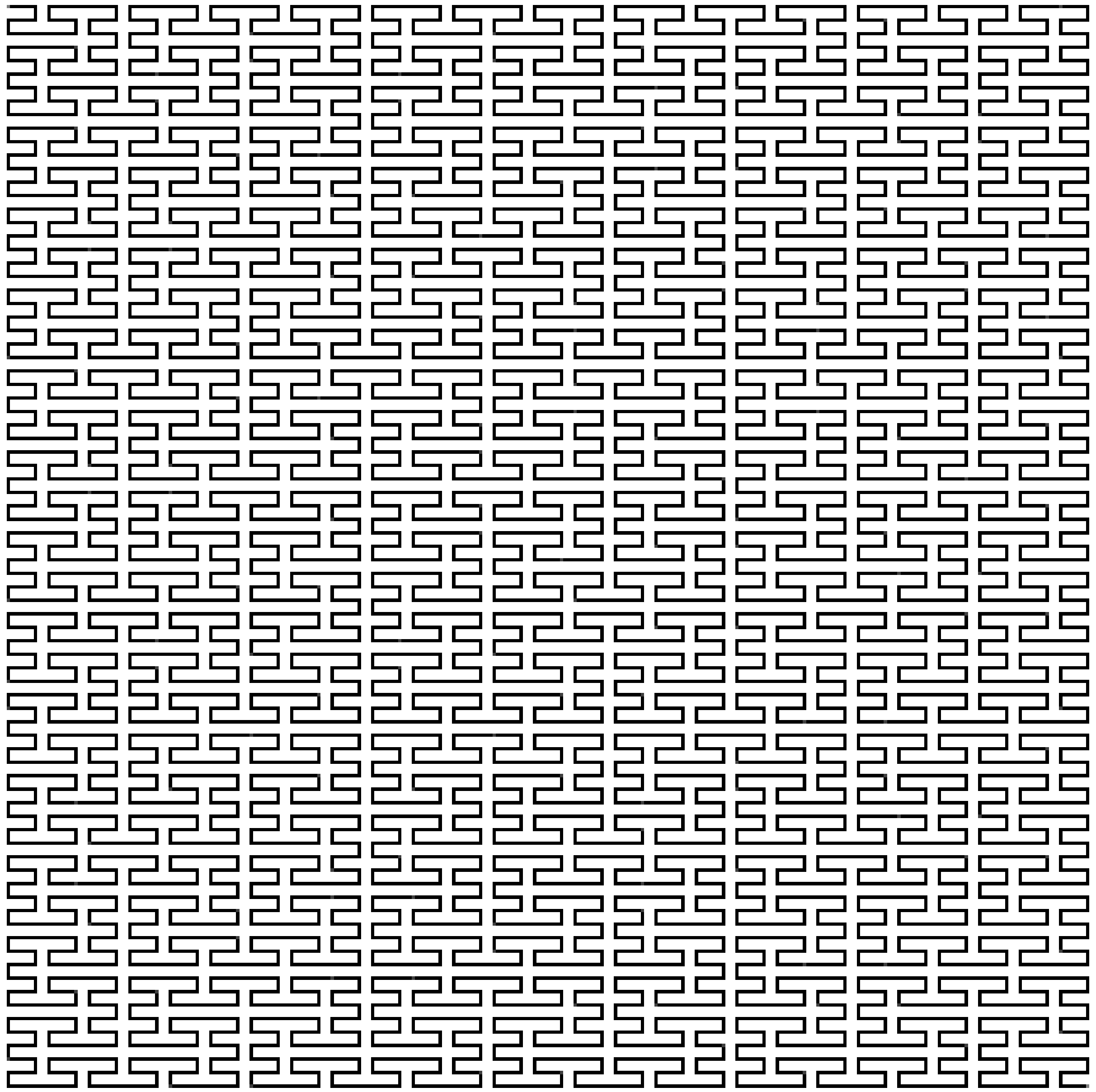


Georg Ferdinand Ludwig Philipp Cantor

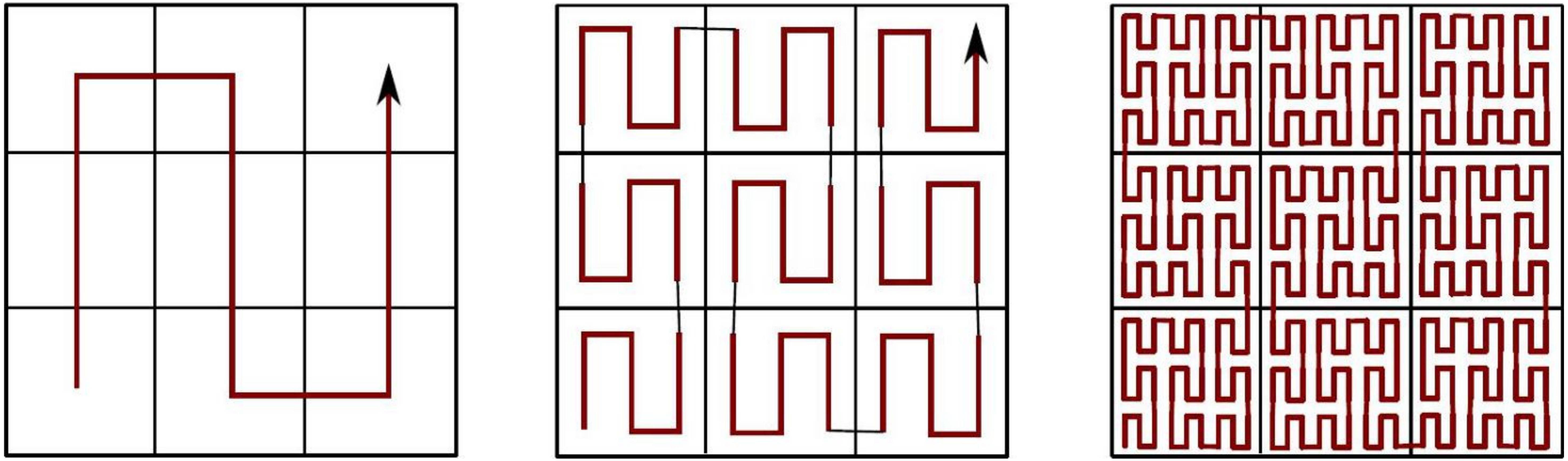
Llenar un infinito *con otro infinito*

Giuseppe Peano

Llenar un infinito *con otro infinito*



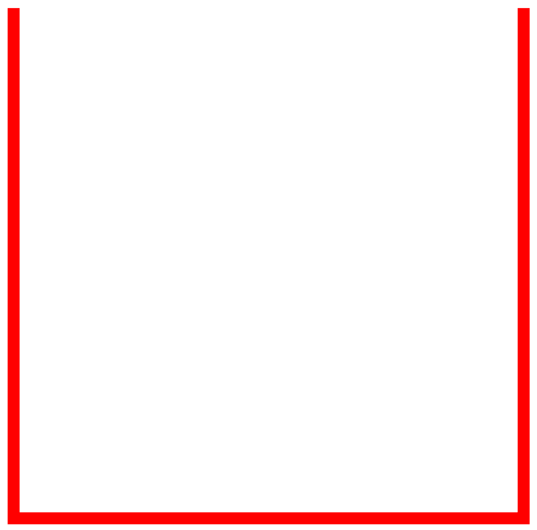
Curva *de Peano*



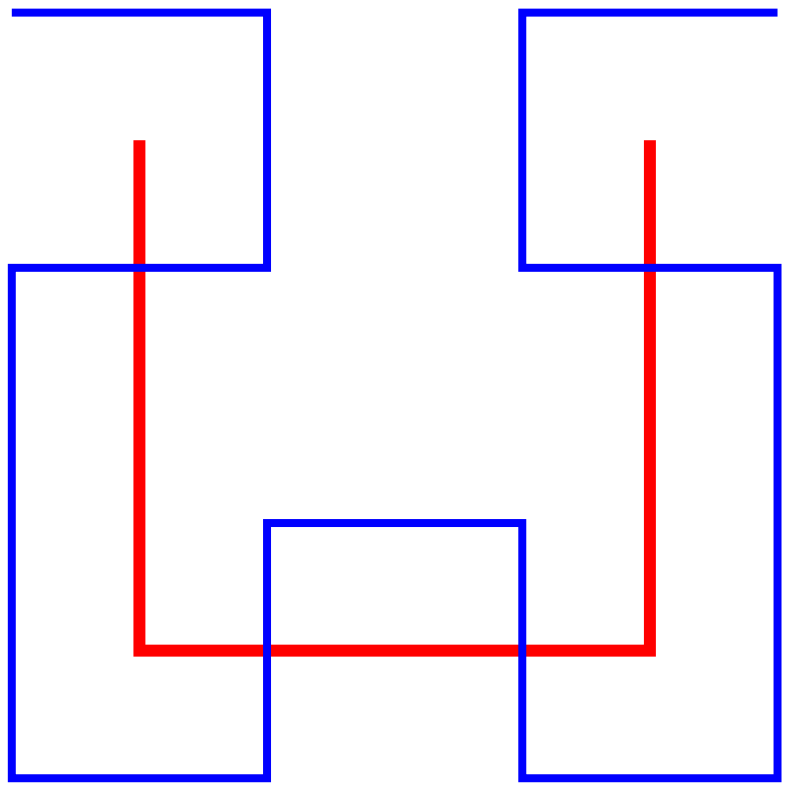
Llenar un infinito *con… ondas?*

David Hilbert

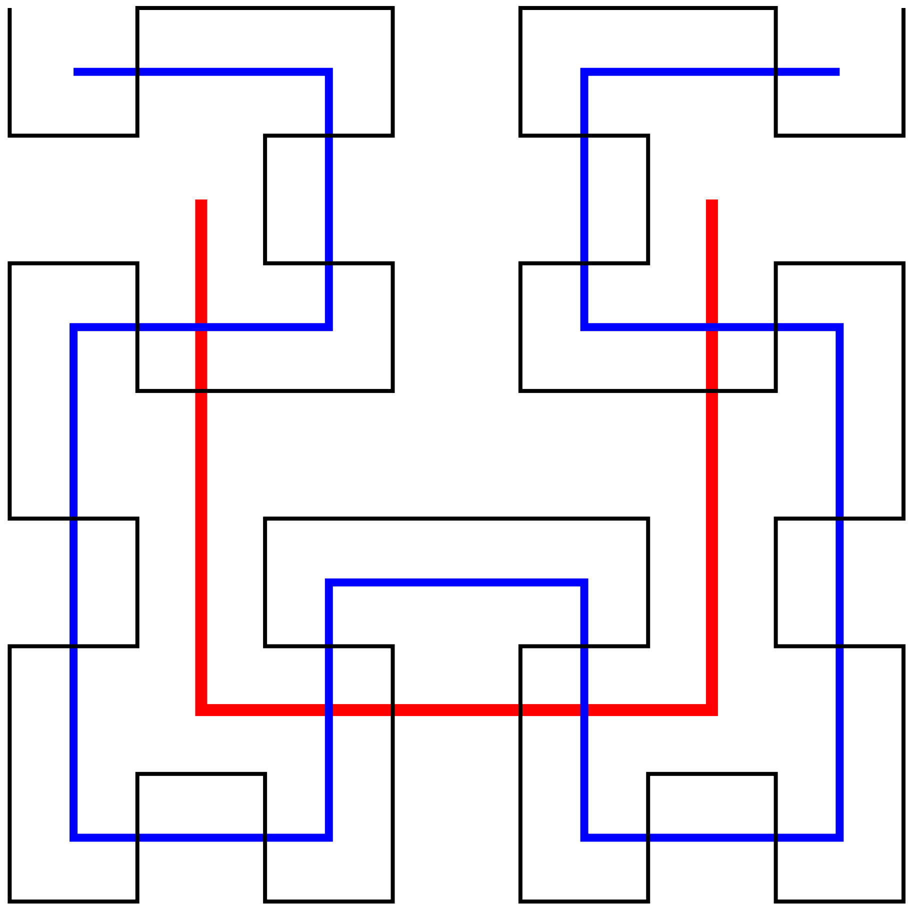
Curva *de Hilbert*



Primer orden



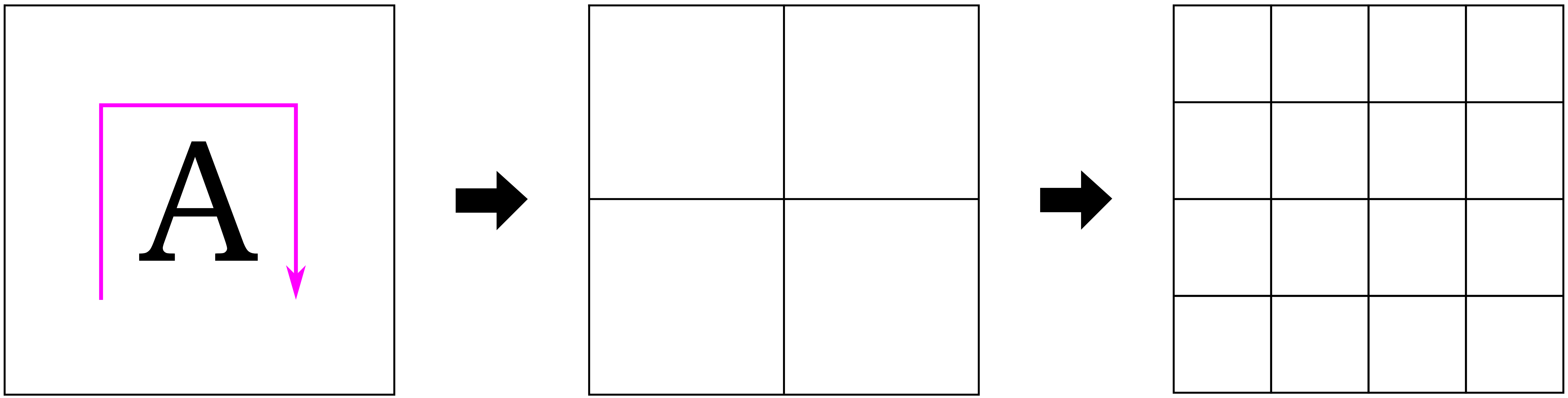
Segundo orden



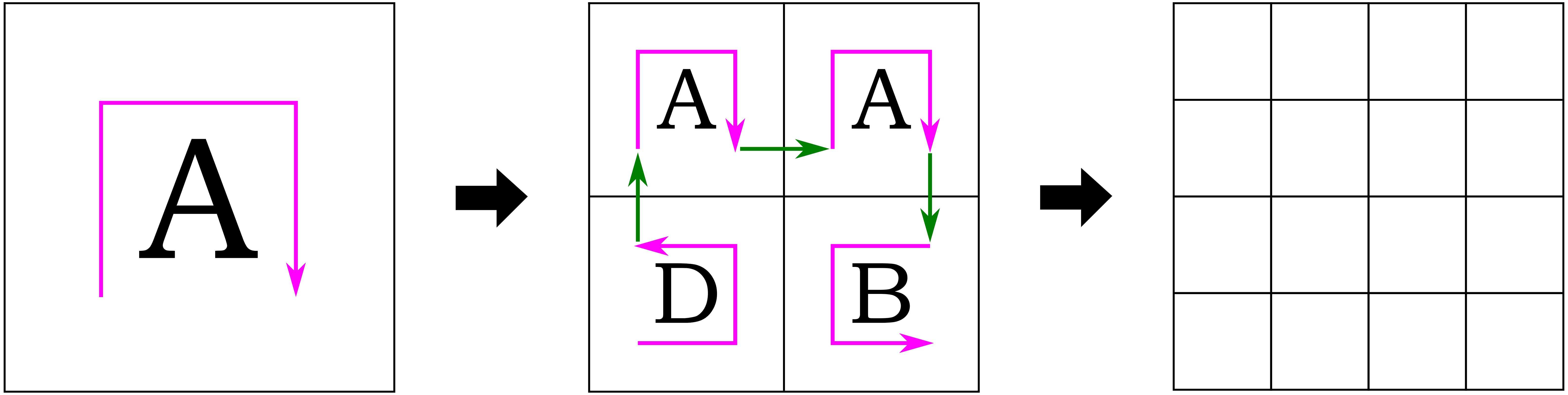
Tercer orden

Curva *de Hilbert*

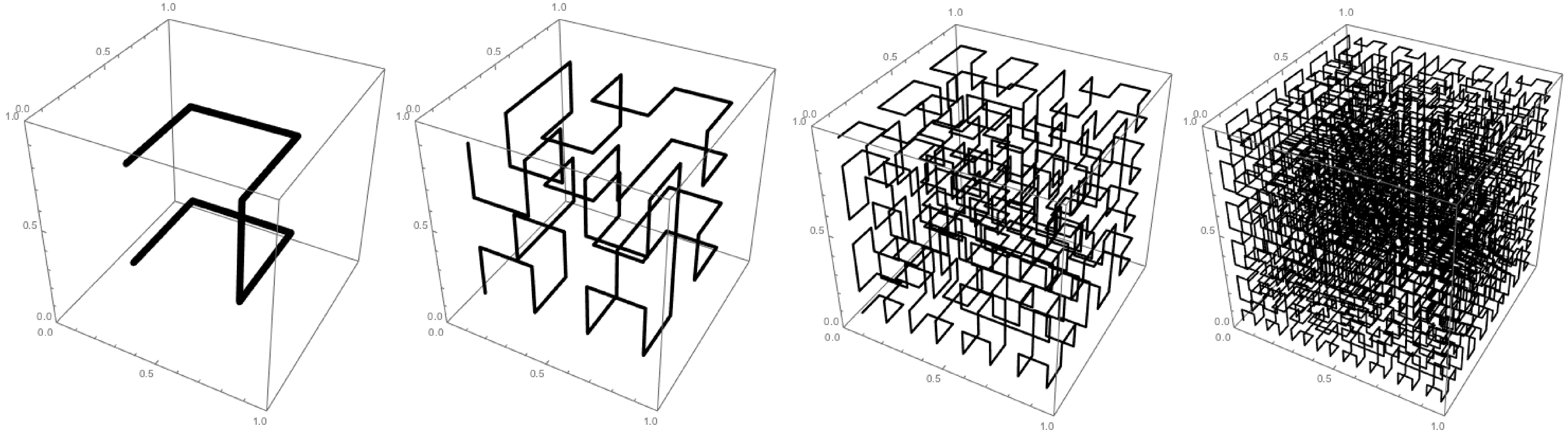
Curva *de Hilbert*



Curva *de Hilbert*

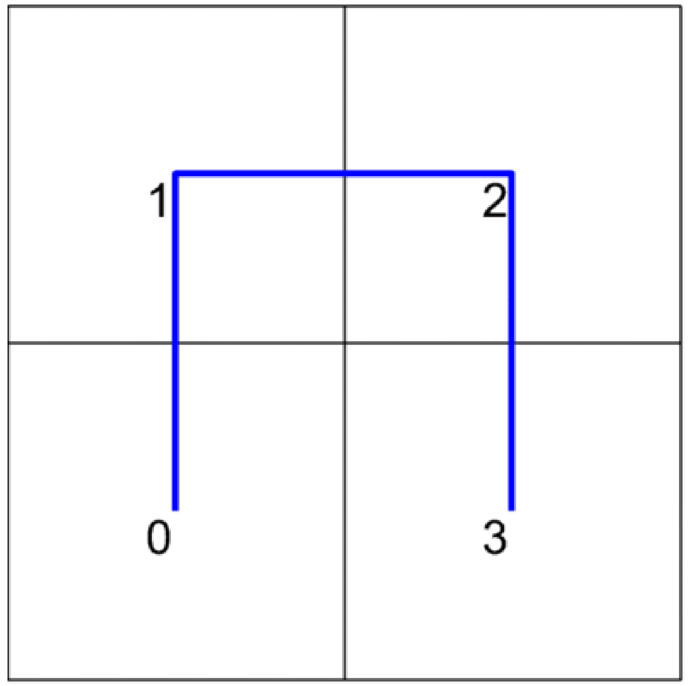


Curva *de Hilbert*

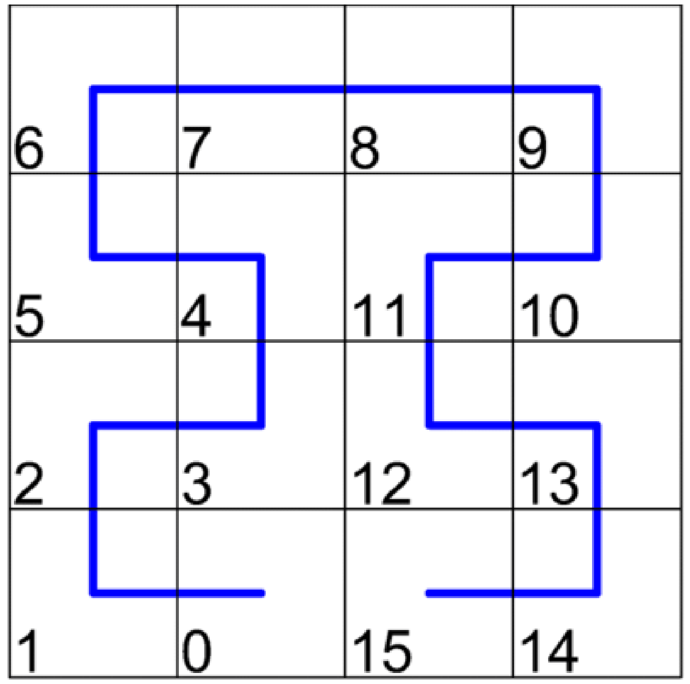


Space-filling *curves*

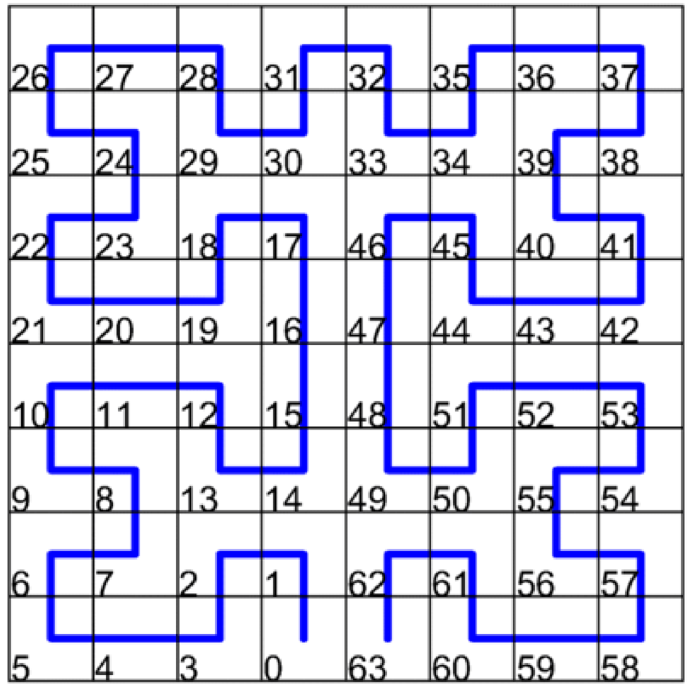
Curva *de Moore*



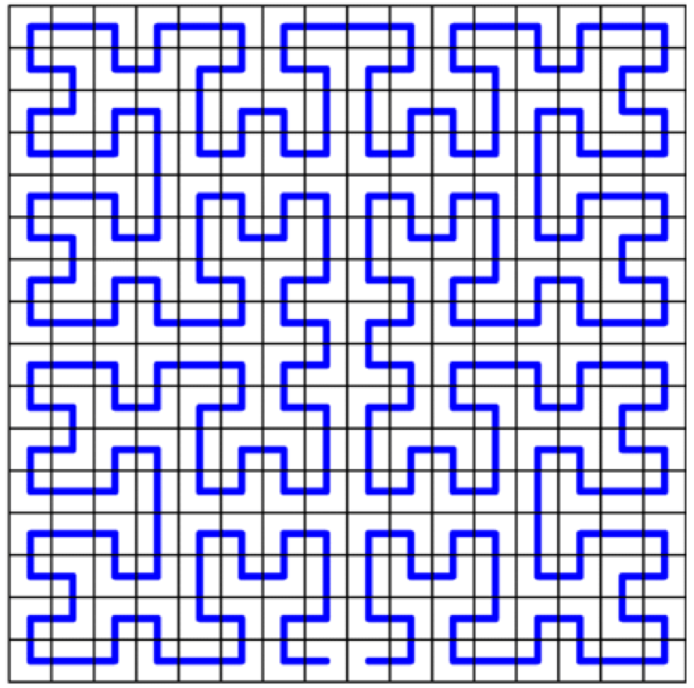
Orden 1



Orden 2

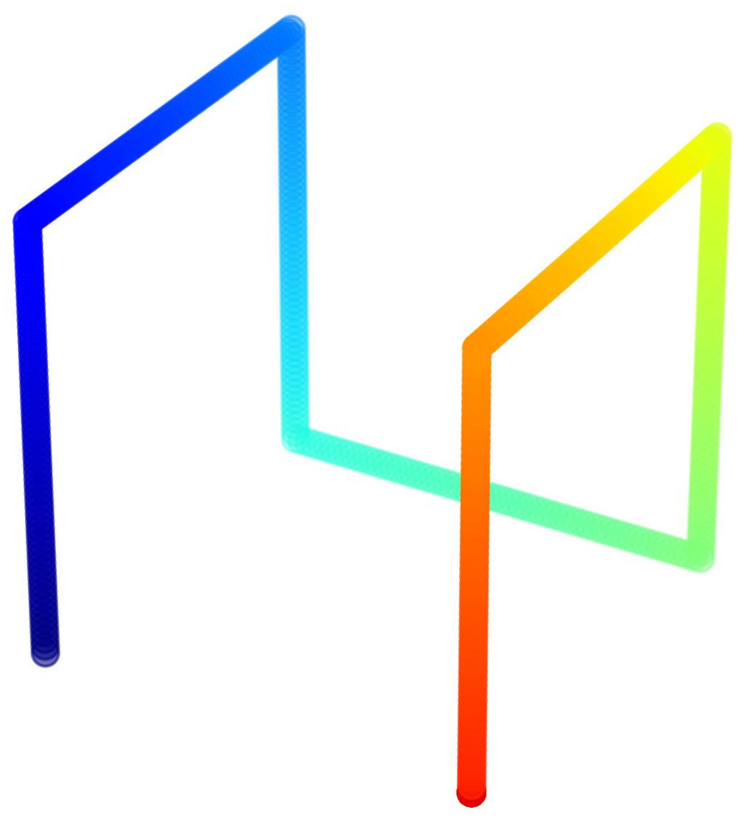


Orden 3

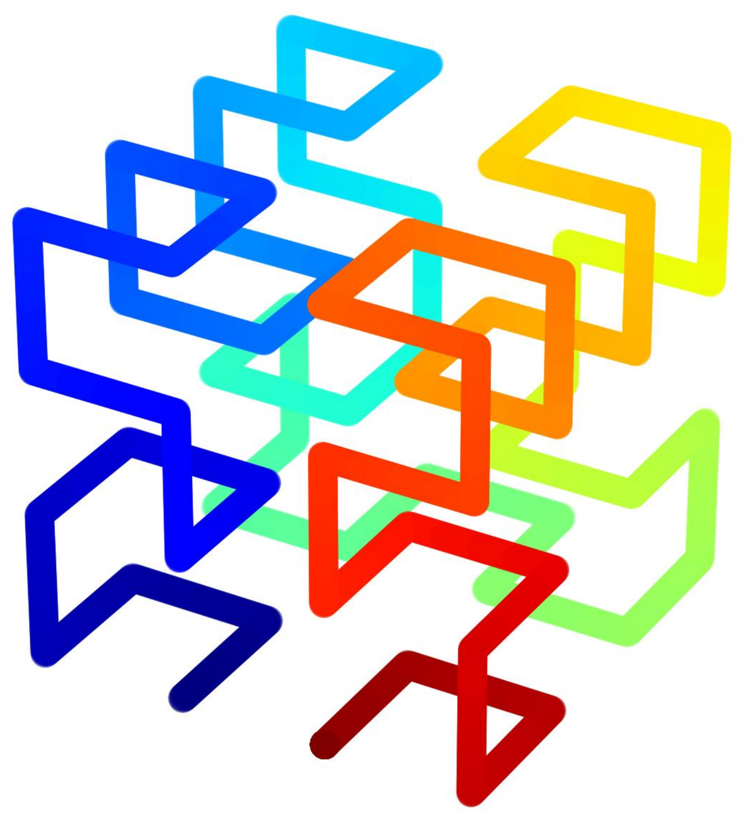


Orden 4

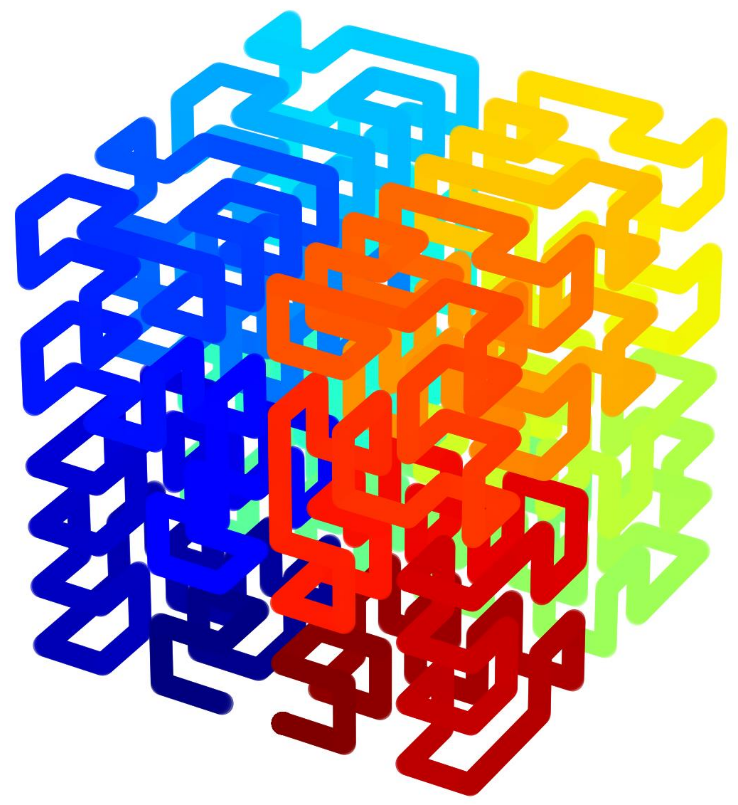
Curva *de Moore*



Orden 1



Orden 2



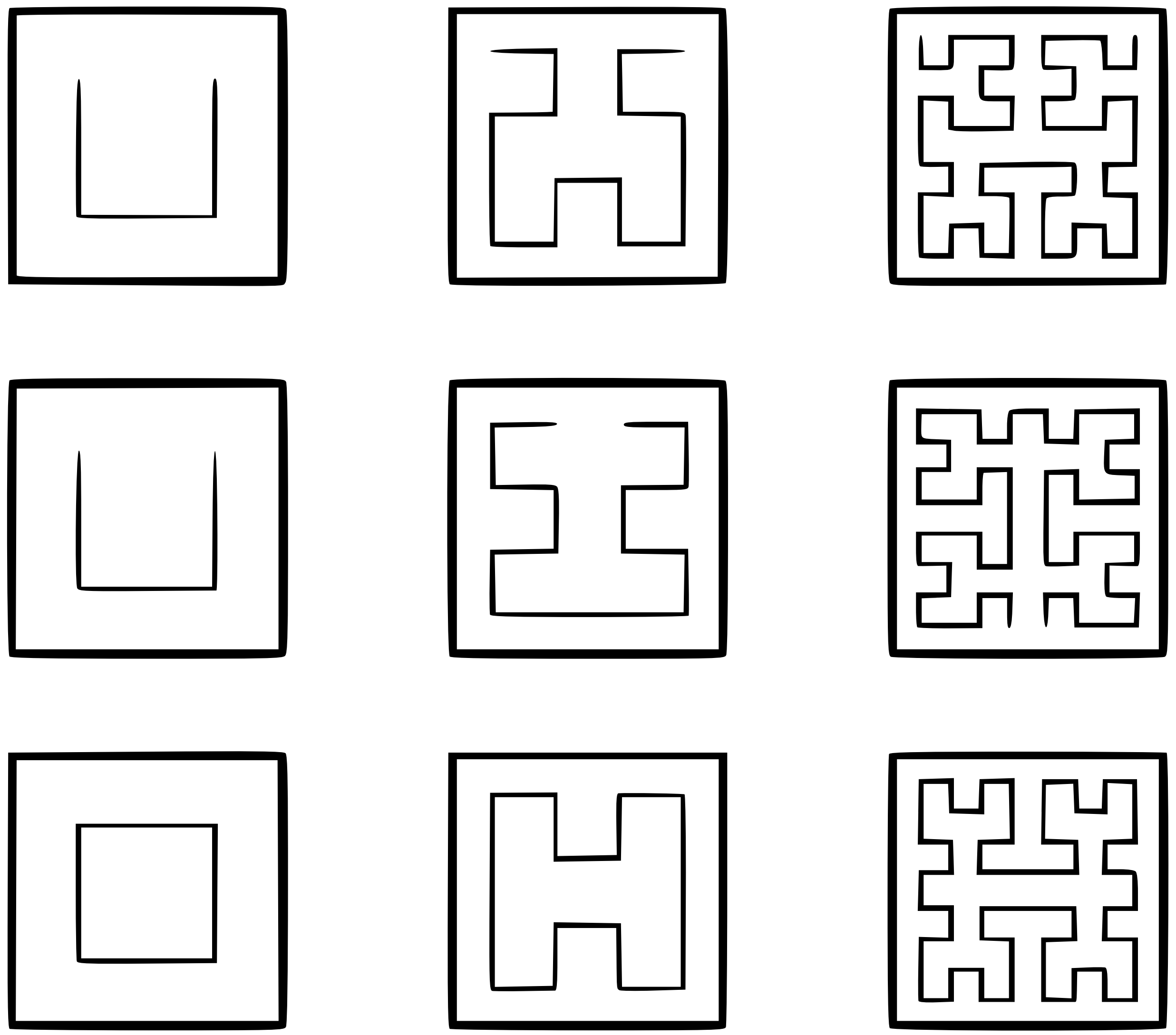
Orden 3

Curva *de Moore*

Hilbert curve

Moore curve

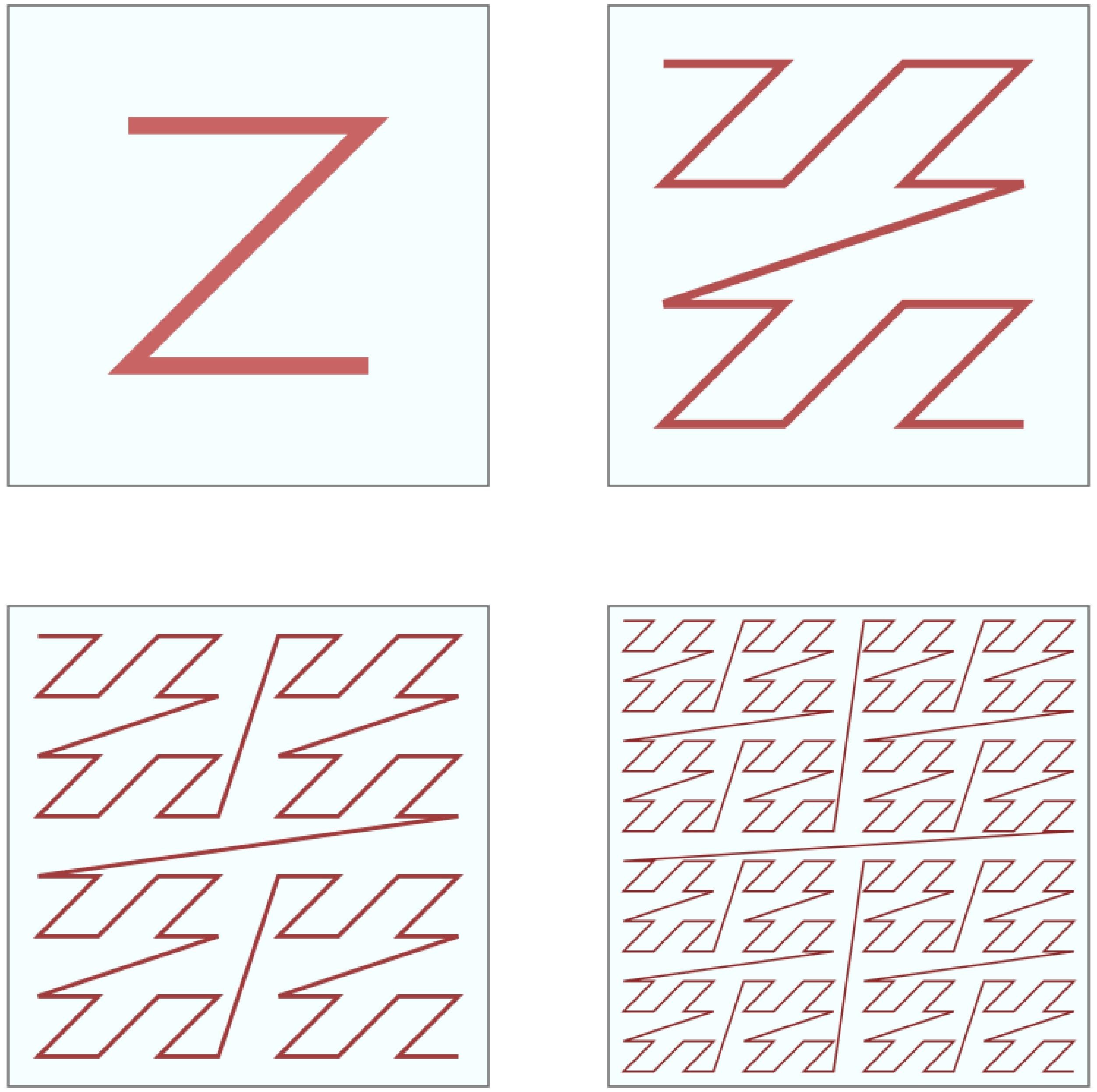
Modified Moore curve



Nivel 0 Nivel 1 Nivel 2

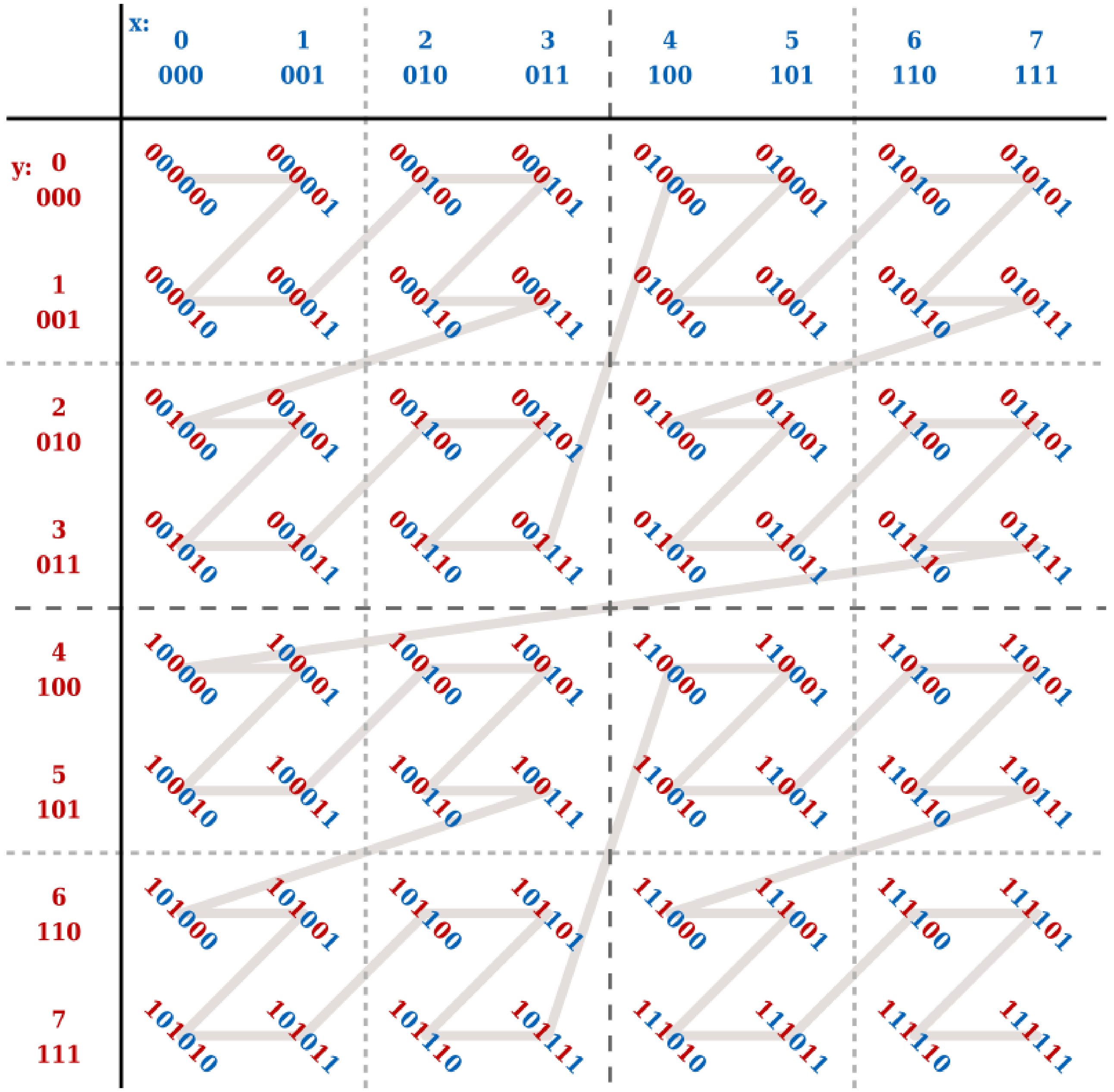
Curva *de Morton*

(Z-order, Lebesgue curve)



Curva *de Morton*

(Z-order, Lebesgue curve)



Curva *de Morton*

(Z-order, Lebesgue curve)

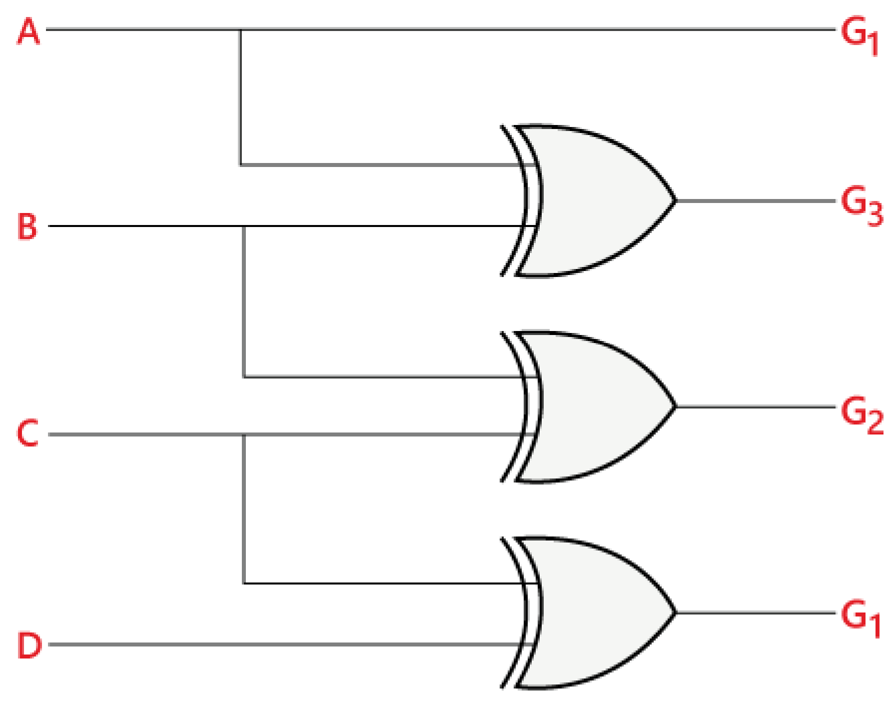
Algoritmo de combinación de bits

Curva *Gray*

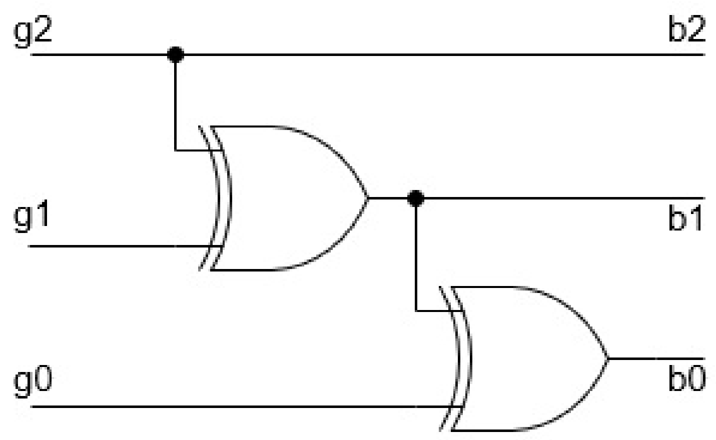
y \ x:	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000	000000	000001	000100	000101	010000	010001	010100	010101
1 001	000010	000011	000110	000111	010010	010011	010110	010111
2 010	001000	001001	001100	001101	011000	011001	011100	011101
3 011	001010	001011	001110	001111	011010	011011	011110	011111
4 100	100000	100001	100100	100101	110000	110001	110100	110101
5 101	100010	100011	100110	100111	110010	110011	110110	110111
6 110	101000	101001	101100	101101	111000	111001	111100	111101
7 111	101010	101011	101110	101111	111010	111011	111110	111111

Decimal	Binary	Gray	Decimal of Gray
0	0000	0000	0
1	0001	0001	1
2	0010	0011	3
3	0011	0010	2
4	0100	0110	6
5	0101	0111	7
6	0110	0101	5
7	0111	0100	4
8	1000	1100	12
9	1001	1101	13
10	1010	1111	15
11	1011	1110	14
12	1100	1010	10
13	1101	1011	11
14	1110	1001	9
15	1111	1000	8

Curva *Gray*

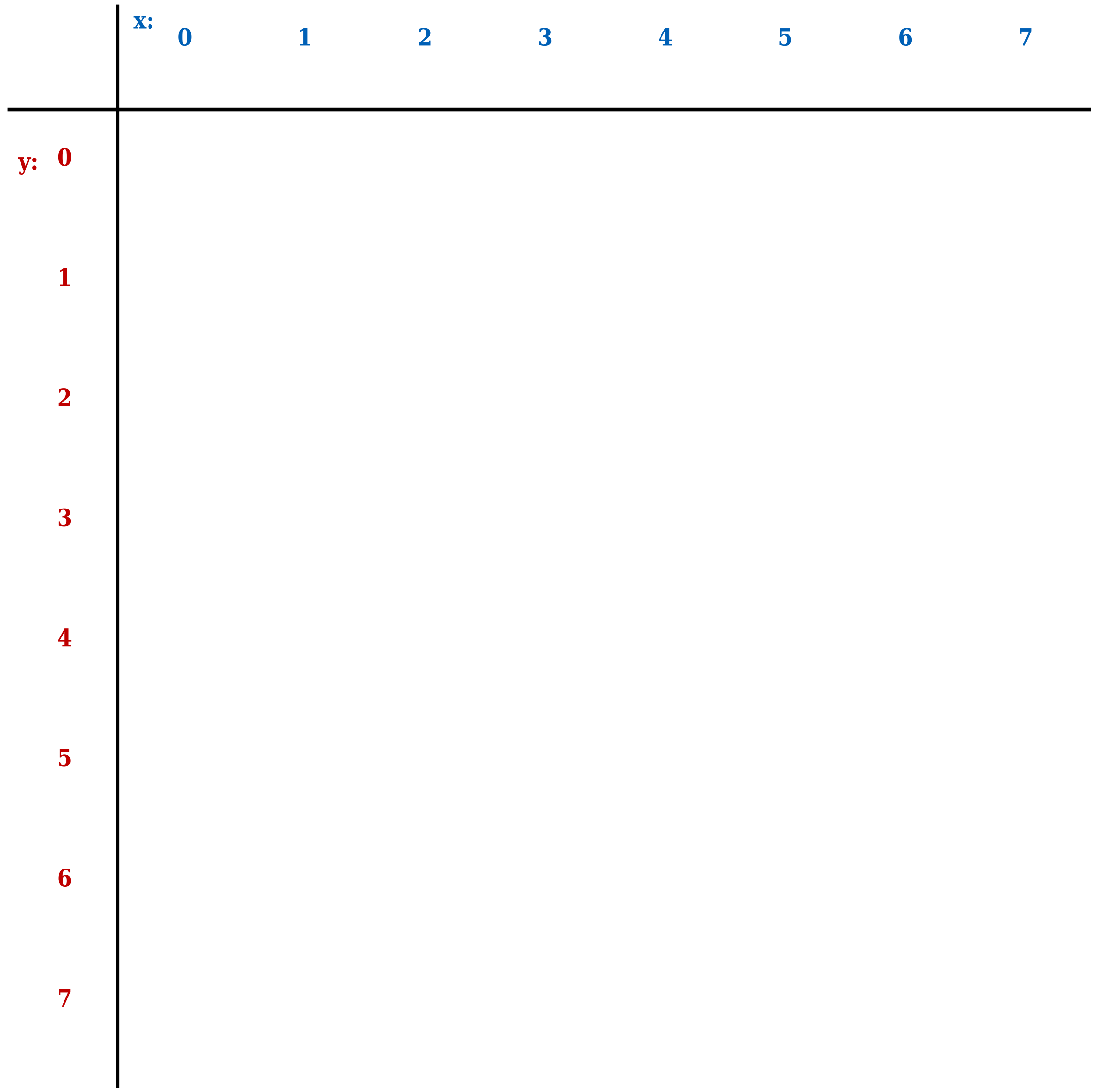


Gray-code



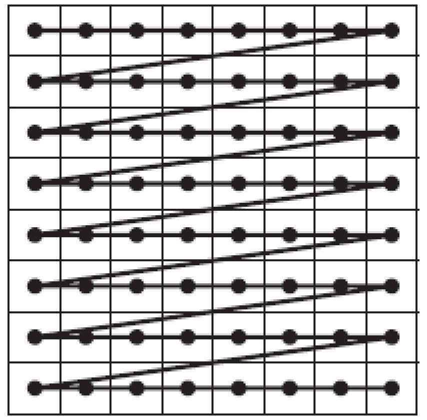
Inverse Gray-code

Curva *Double Gray*

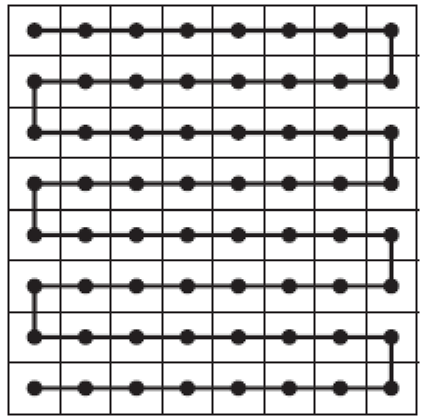


Decimal	Binary	Gray	Decimal of Gray
0	0000	0000	0
1	0001	0001	1
2	0010	0011	3
3	0011	0010	2
4	0100	0110	6
5	0101	0111	7
6	0110	0101	5
7	0111	0100	4
8	1000	1100	12
9	1001	1101	13
10	1010	1111	15
11	1011	1110	14
12	1100	1010	10
13	1101	1011	11
14	1110	1001	9
15	1111	1000	8

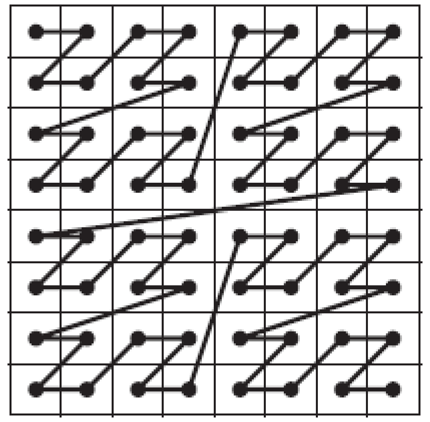
Space-filling *curves*



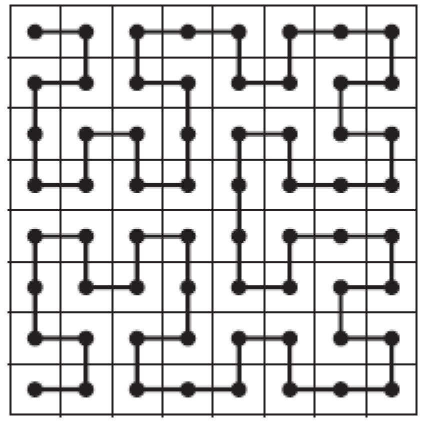
Row order



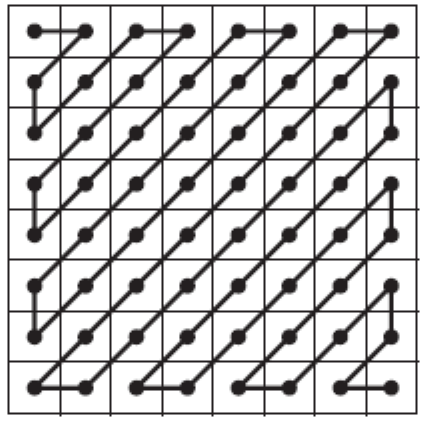
Row-prime order



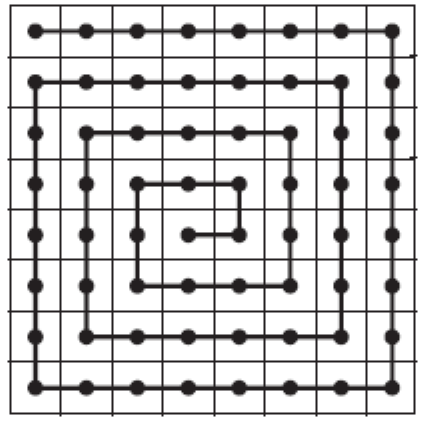
Morton order



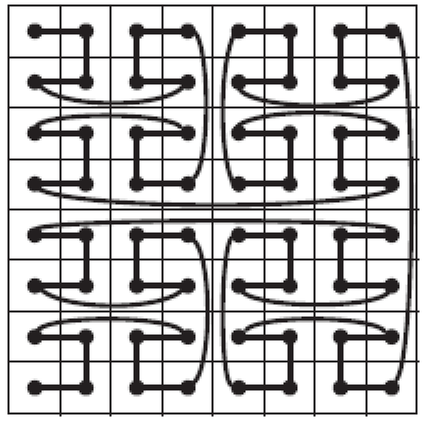
Peano-Hilbert order



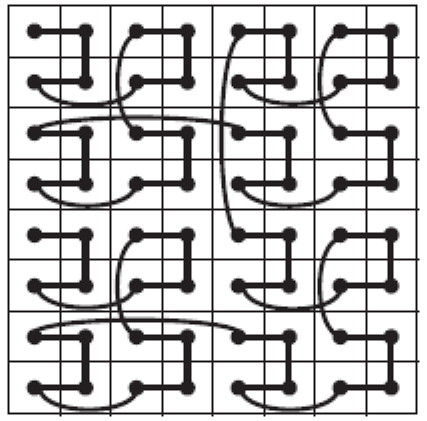
Cantor-diagonal order



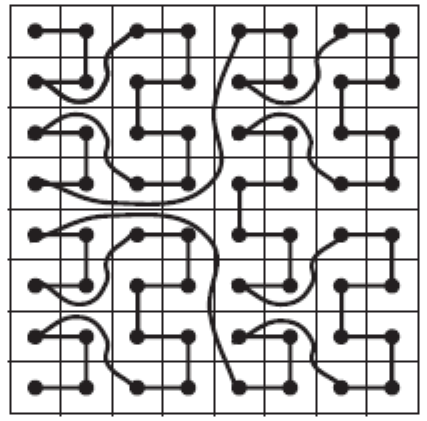
Spiral order



Gray order



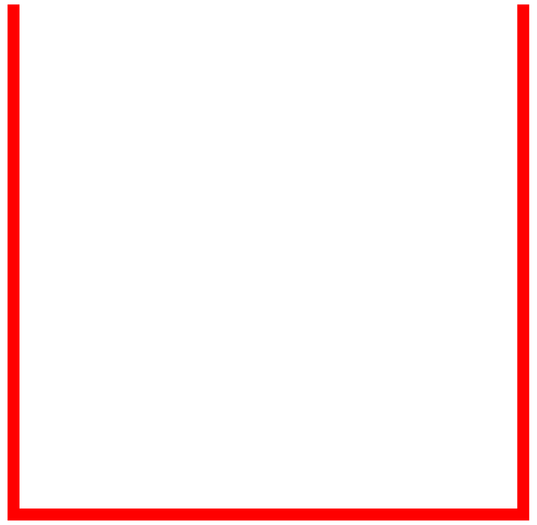
Double Gray order



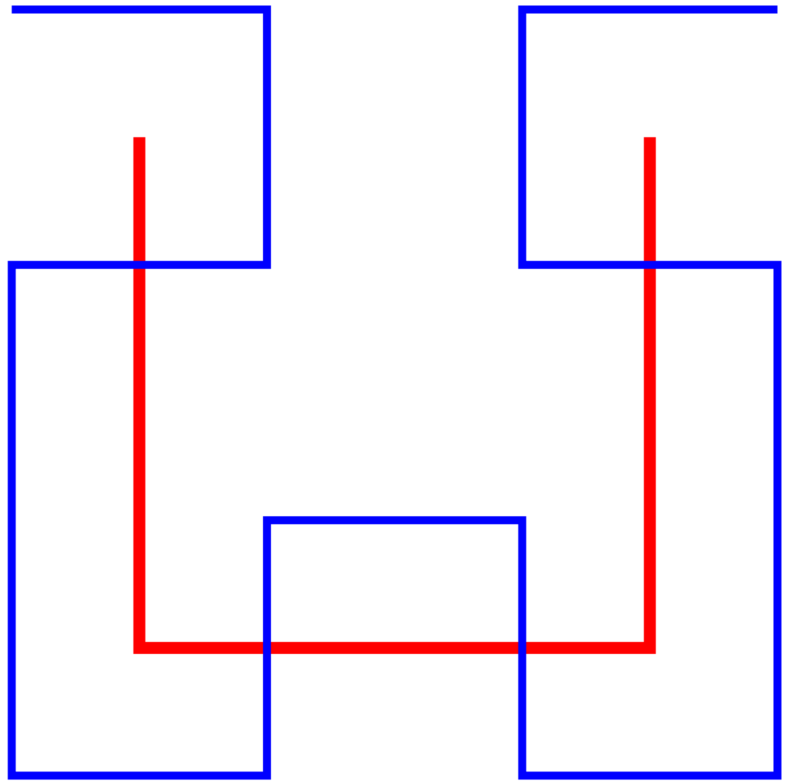
U order

2. Hilbert curves

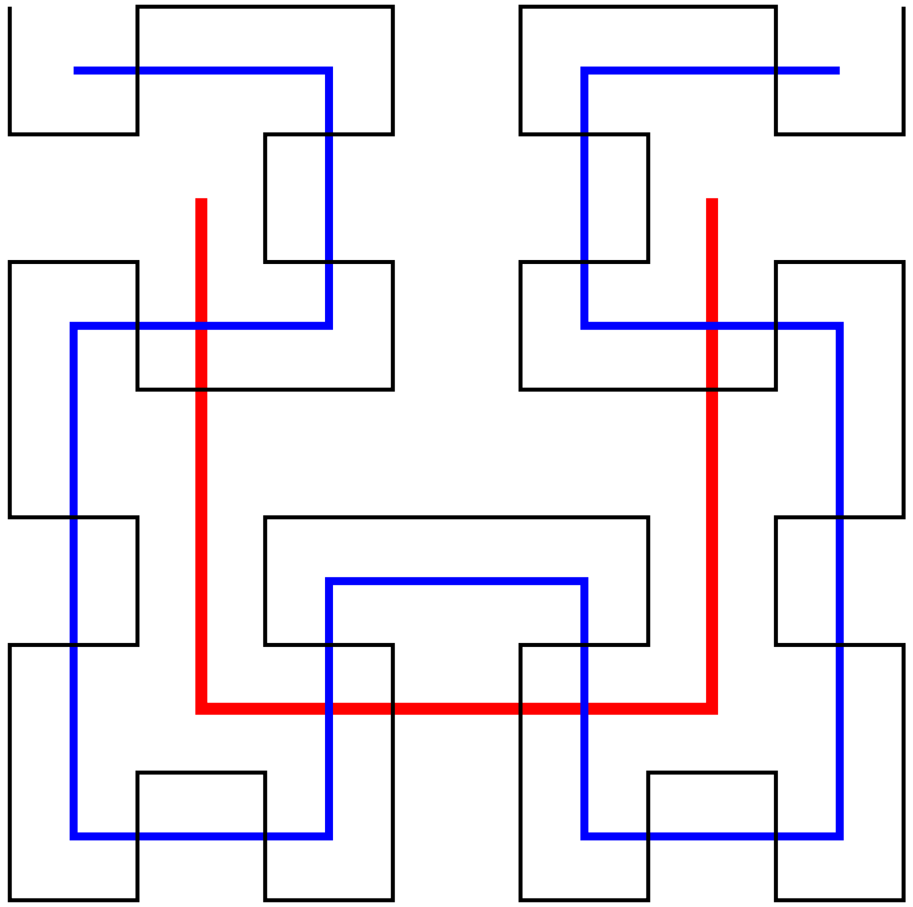
Curva *de Hilbert*



Primer orden

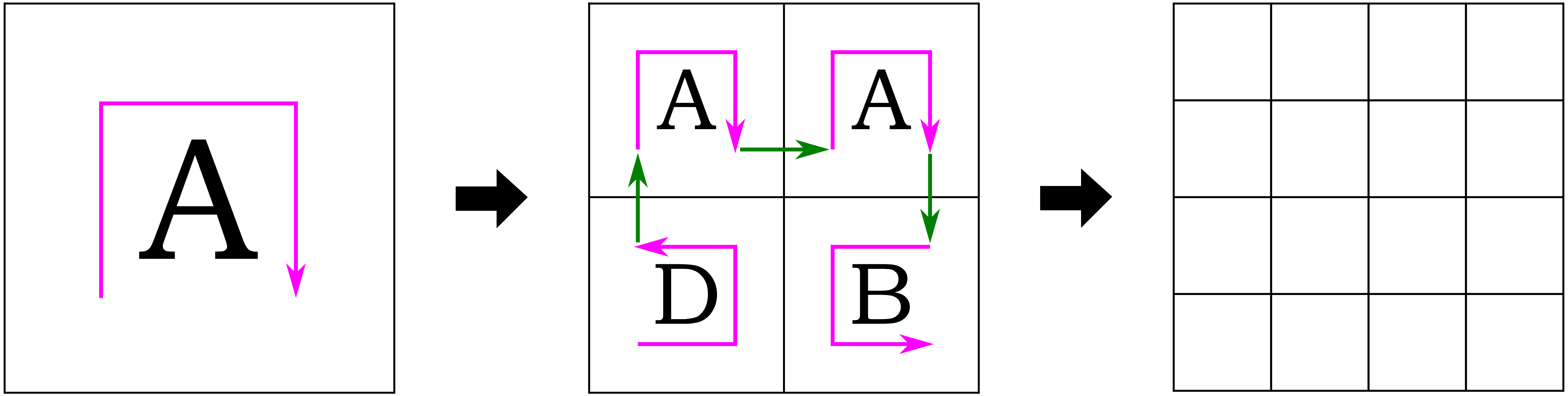


Segundo orden

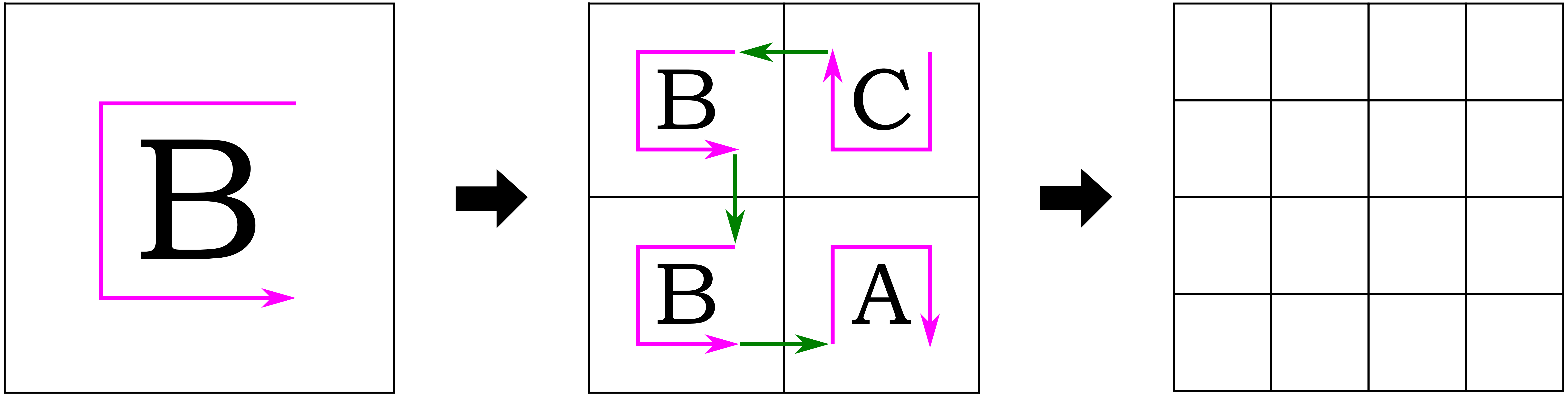


Tercer orden

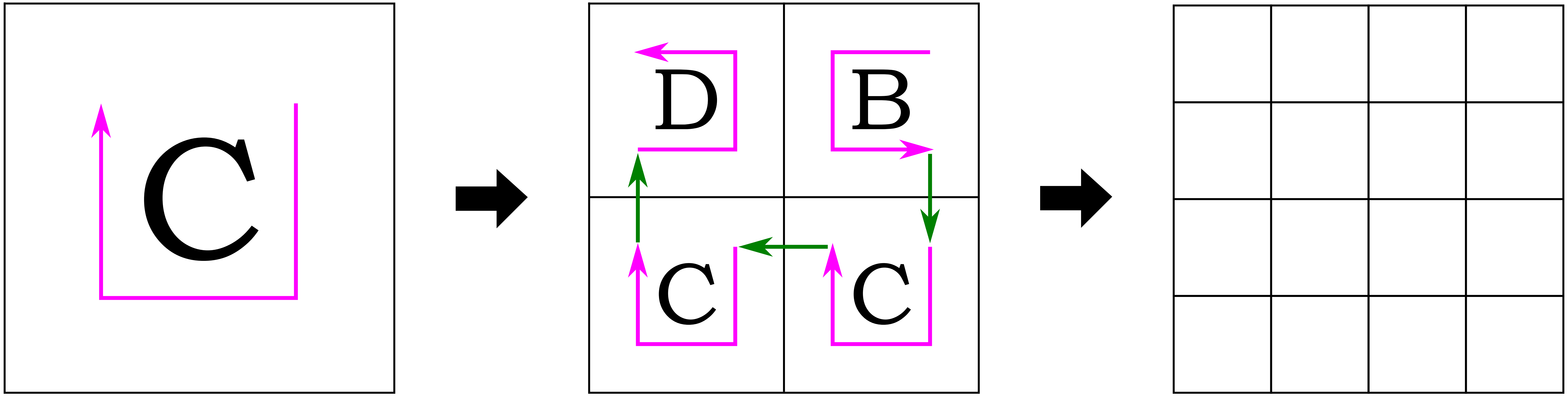
Curva *de Hilbert*



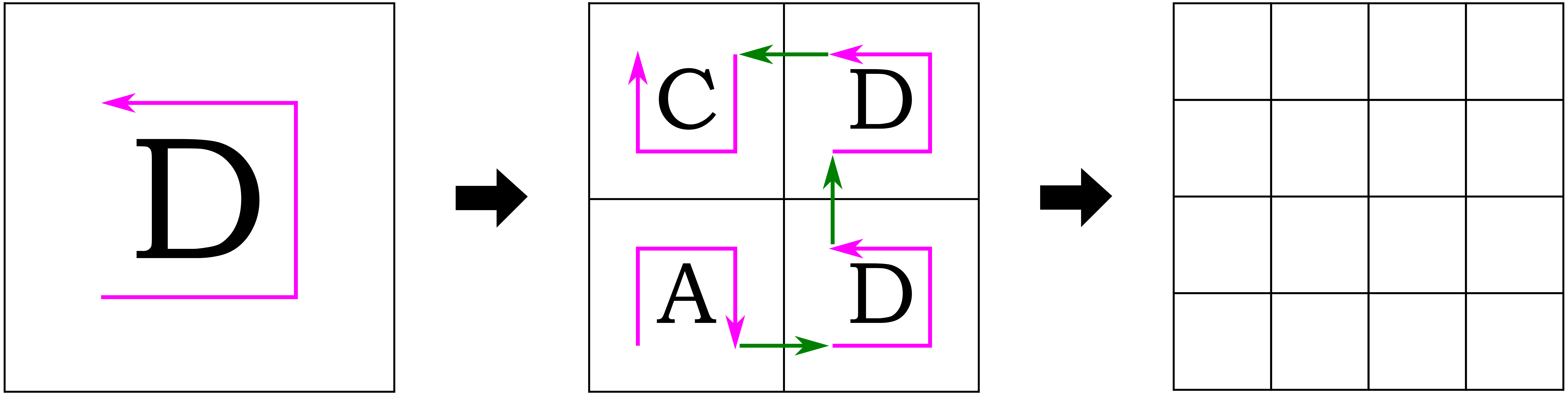
Curva *de Hilbert*



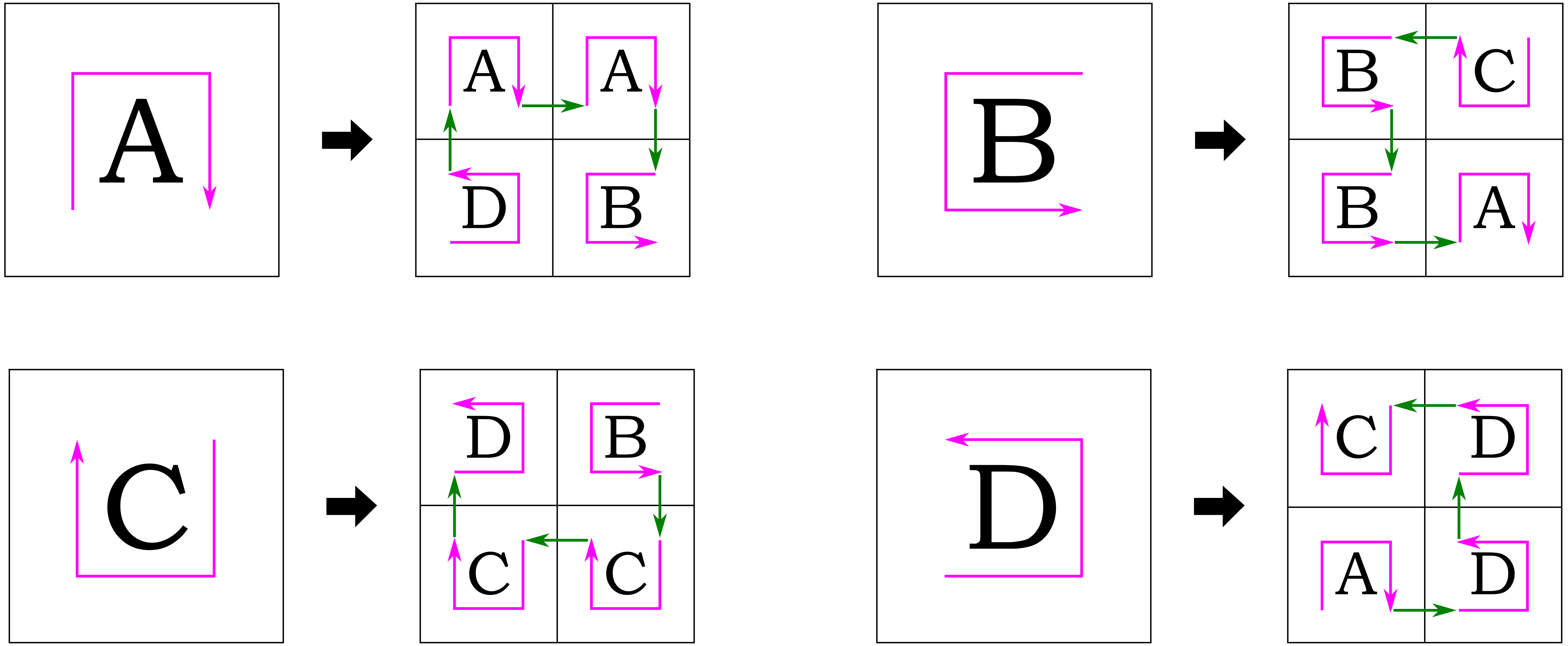
Curva *de Hilbert*



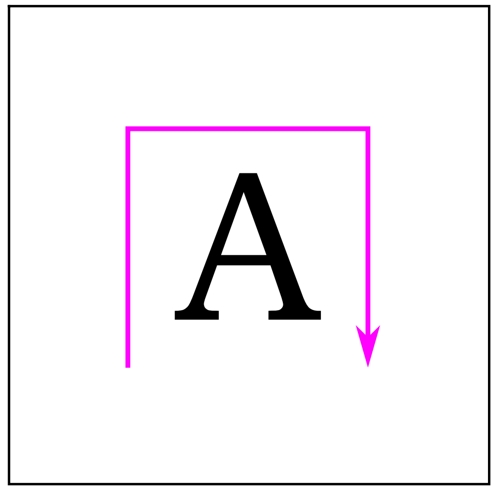
Curva *de Hilbert*

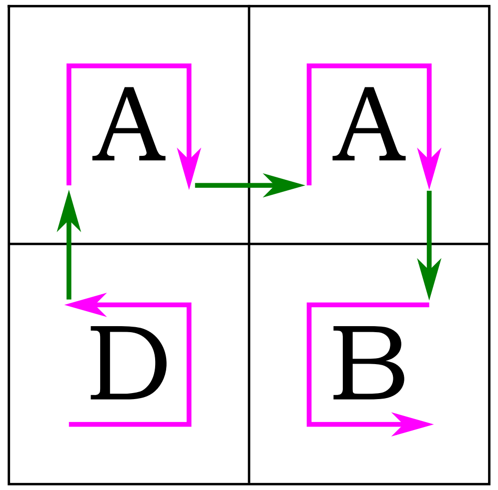


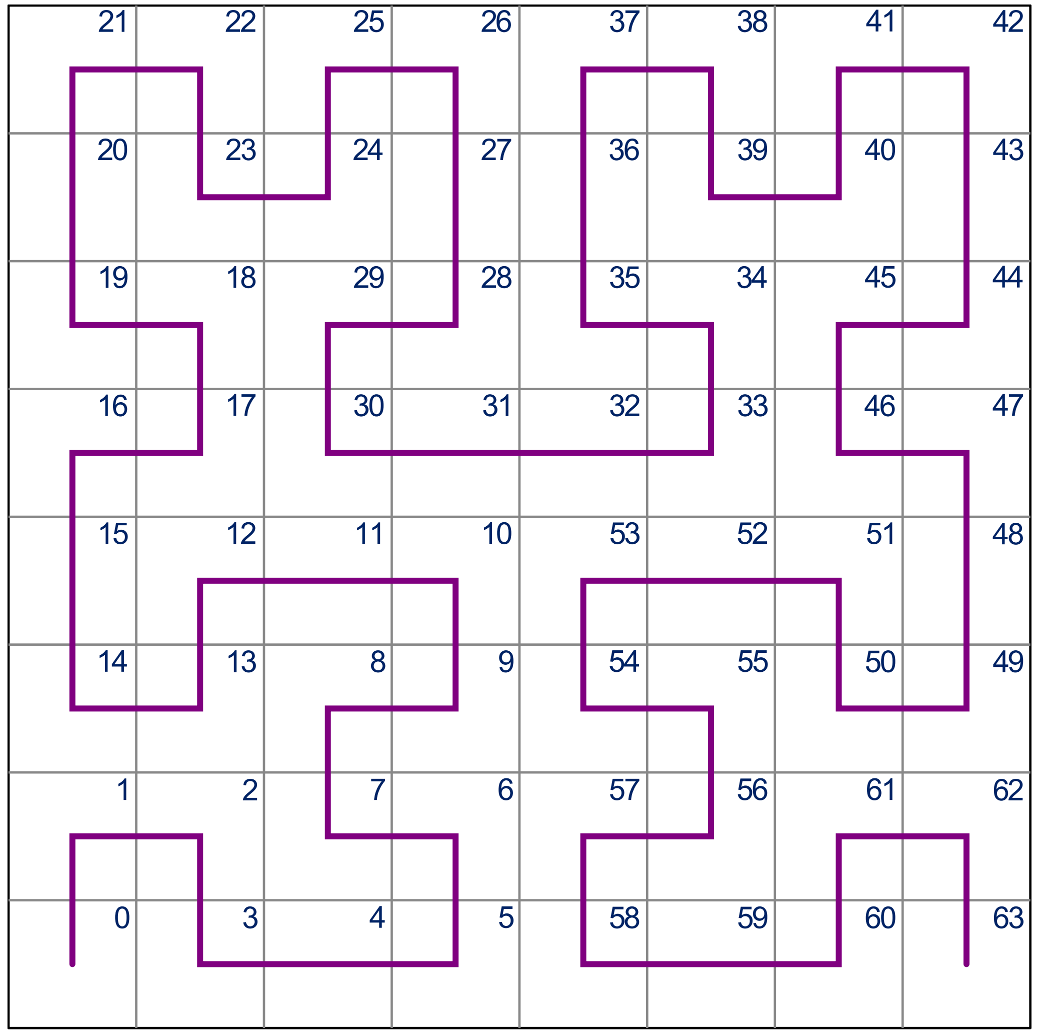
Curva *de Hilbert*



Índice *de Hilbert*



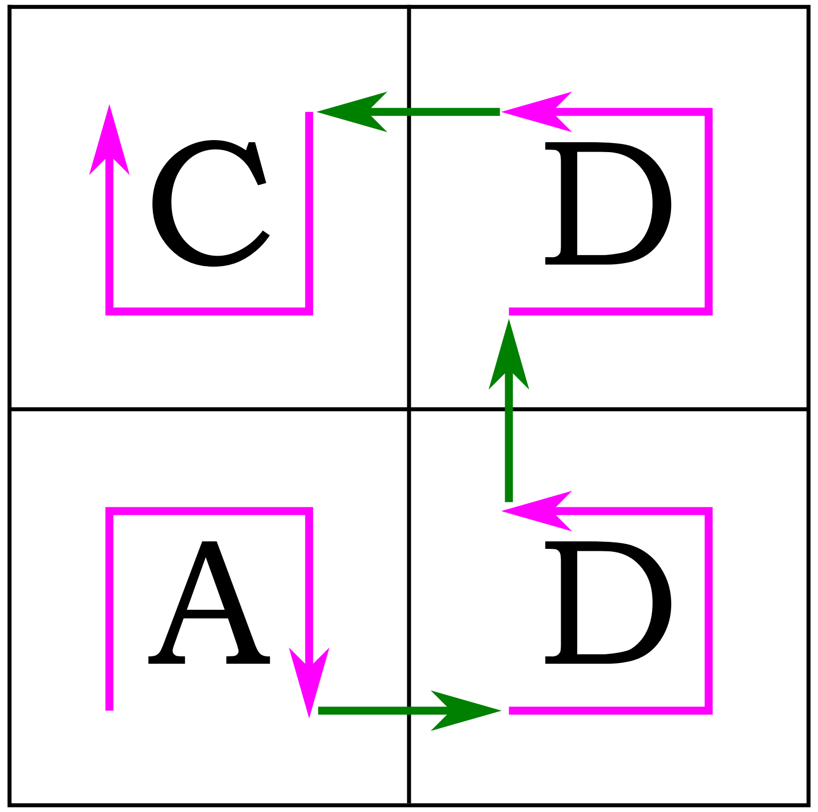




Curva *de Hilbert*

Origen de coordenadas   Eje $\vec{x}$ Eje $\vec{y}$

```
procedure hilbert(x, y, xi, xj, yi, yj, n)
/* x and y are the coordinates of the bottom left corner */
/* xi & xj are the i & j components of the unit x vector of the frame */
/* similarly yi and yj */
if (n <= 0) then
   LineTo(x + (xi + yi)/2, y + (xj + yj)/2);
else
   {
   hilbert(x,           y,           yi/2, yj/2,  xi/2,  xj/2, n-1);
   hilbert(x+xi/2,      y+xj/2 ,     xi/2, xj/2,  yi/2,  yj/2, n-1);
   hilbert(x+xi/2+yi/2, y+xj/2+yj/2, xi/2, xj/2,  yi/2,  yj/2, n-1);
   hilbert(x+xi/2+yi,   y+xj/2+yj,  -yi/2,-yj/2, -xi/2, -xj/2, n-1);
   }
end procedure;
```




UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA
MECATRÓNICA
BIOINGENIERÍA
CIENCIA DE
LA COMPUTACIÓN