UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Repaso Parcial 2

CS3102 EDA

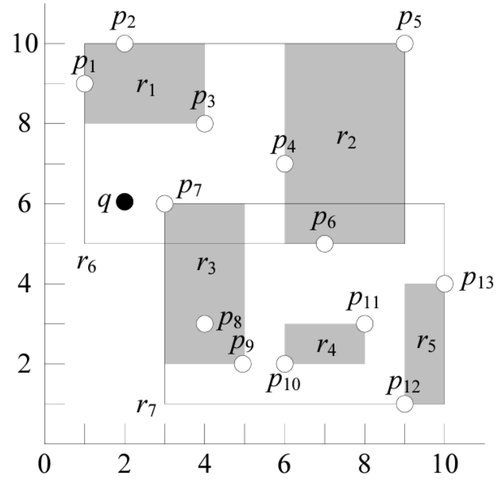
Índice

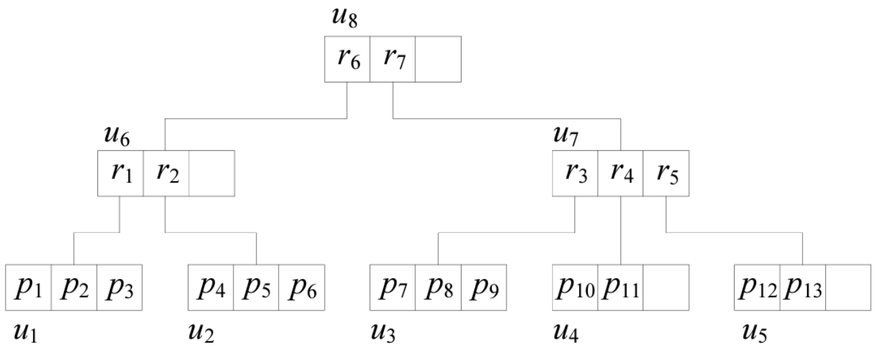
1. **Branch** and bound

Branch *and bound*

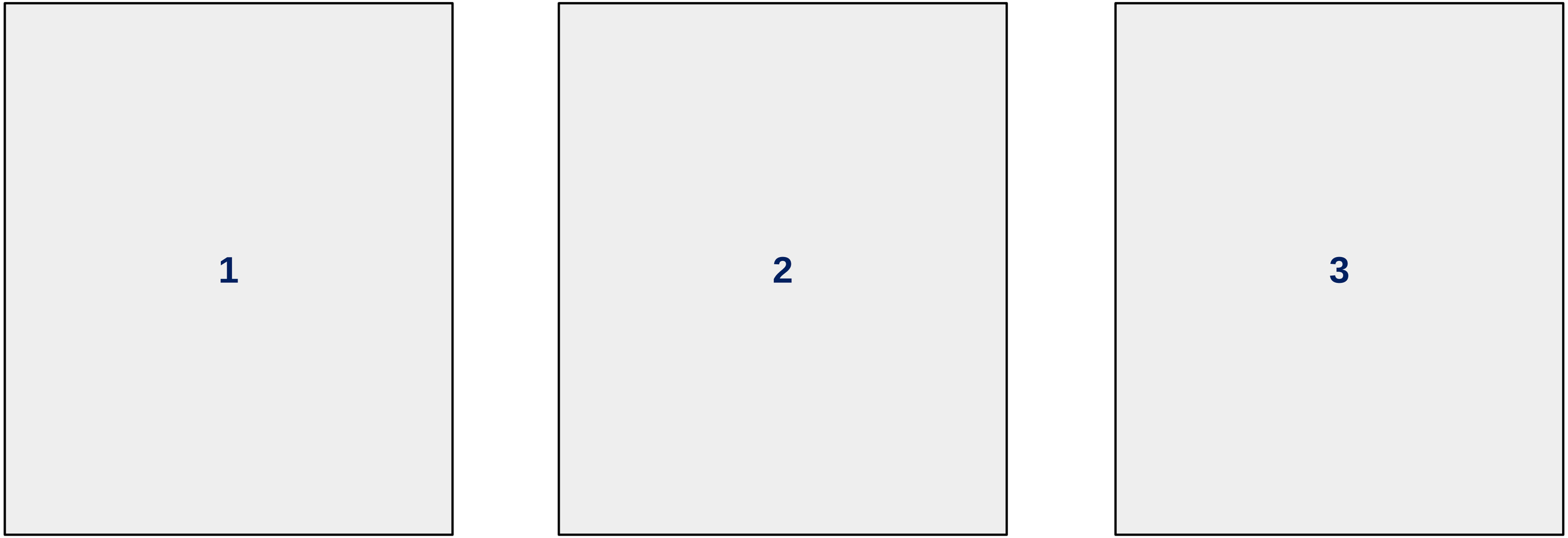
algorithm BaB(u, q)
/* u is the node being accessed, q is the query point;
p_{best} is a global variable that keeps the NN found so far;
the algorithm should be invoked by setting u to the root */

1. **if** u is a leaf node **then**
2. **if** the NN of q in u is closer to q than p_{best} **then**
3. p_{best} = the NN of q in u
4. **else**
5. sort the MBRs in u in ascending order of their mindists to q
/* let $r_1, ..., r_f$ be the sorted order */
6. **for** $i = 1$ to f
7. **if** $mindist(q, r_i) < \|q, p_{best}\|$ **then**
8. Bab(u_i, q)
/* u_i is child node of r_i */

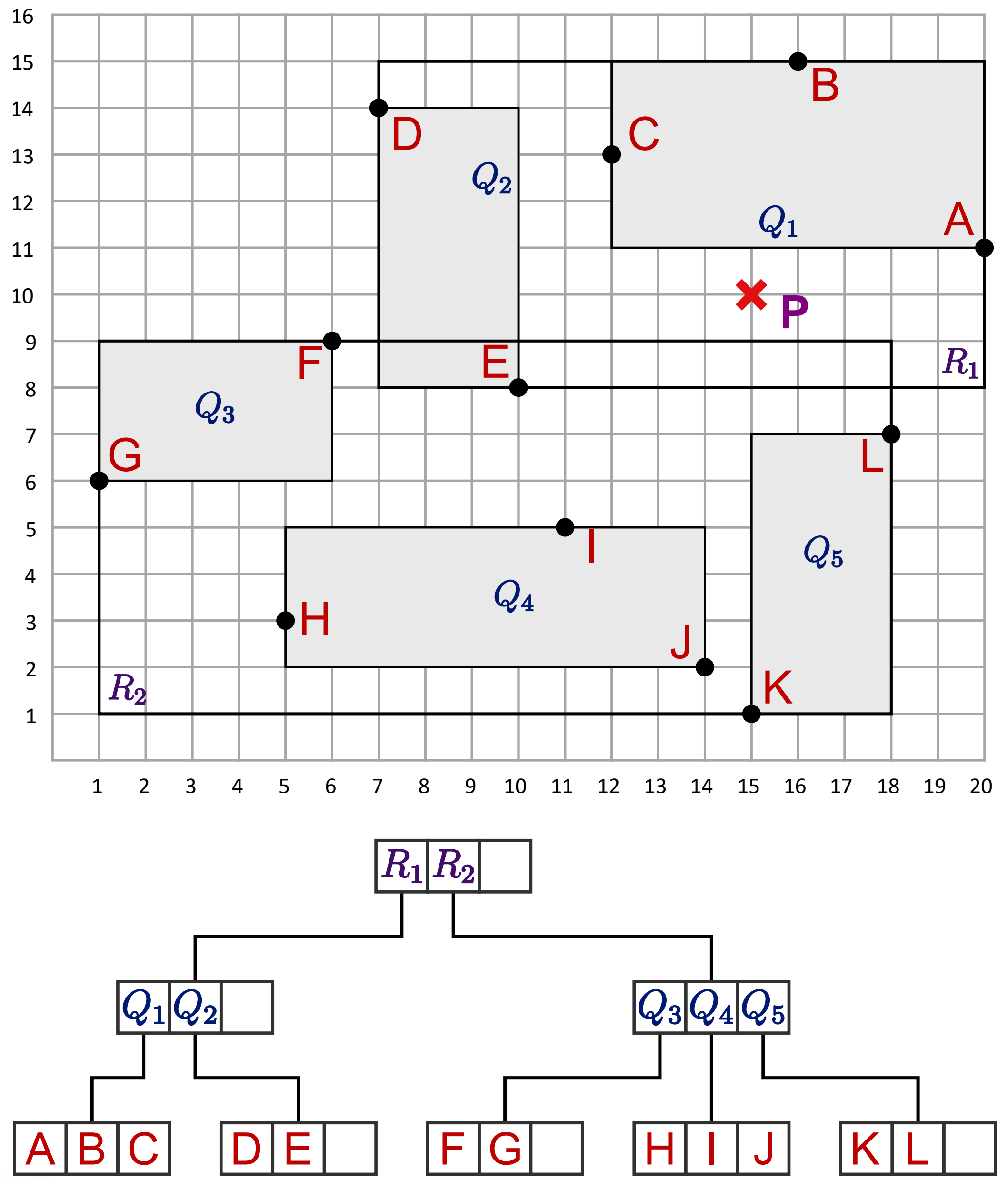


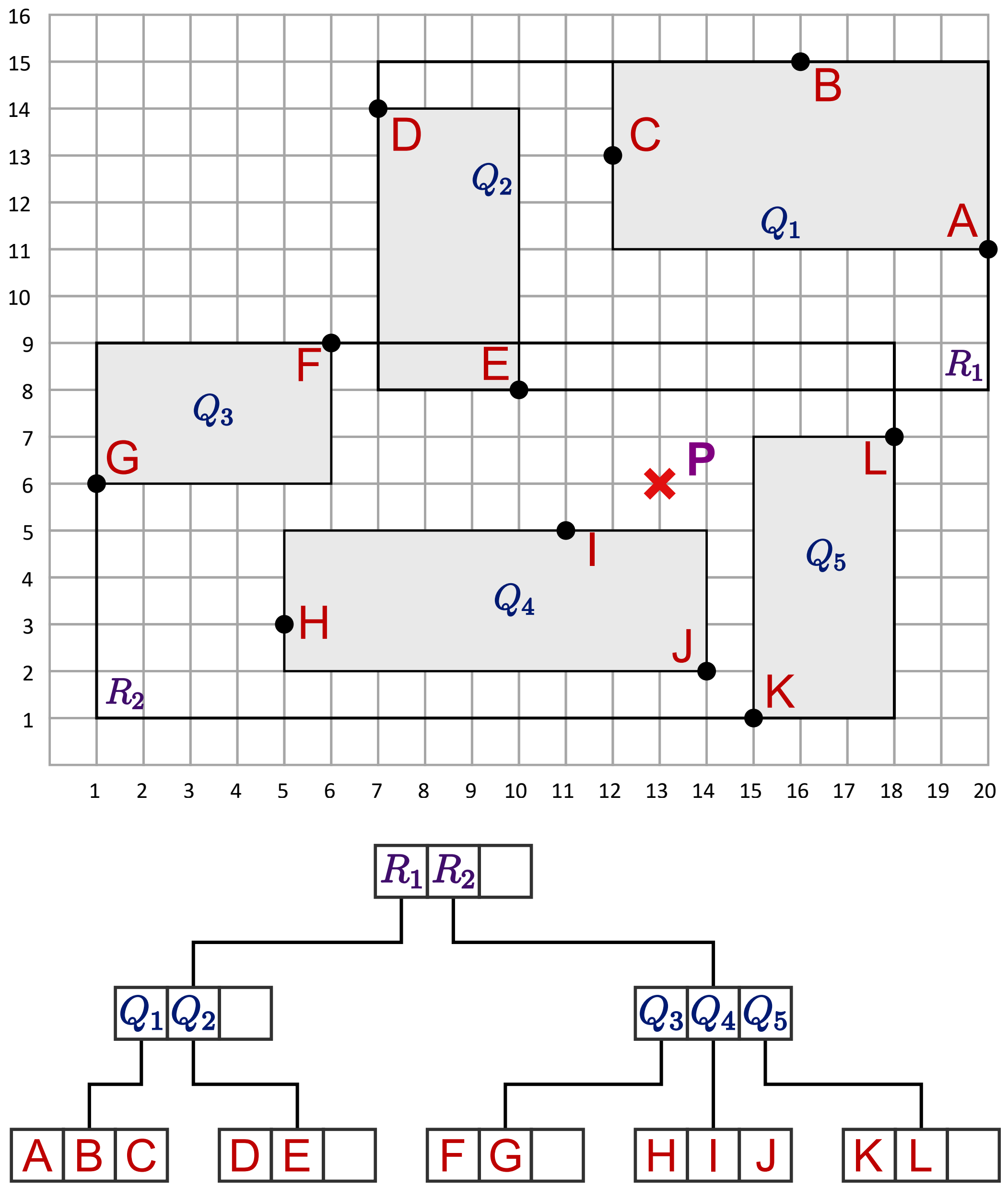


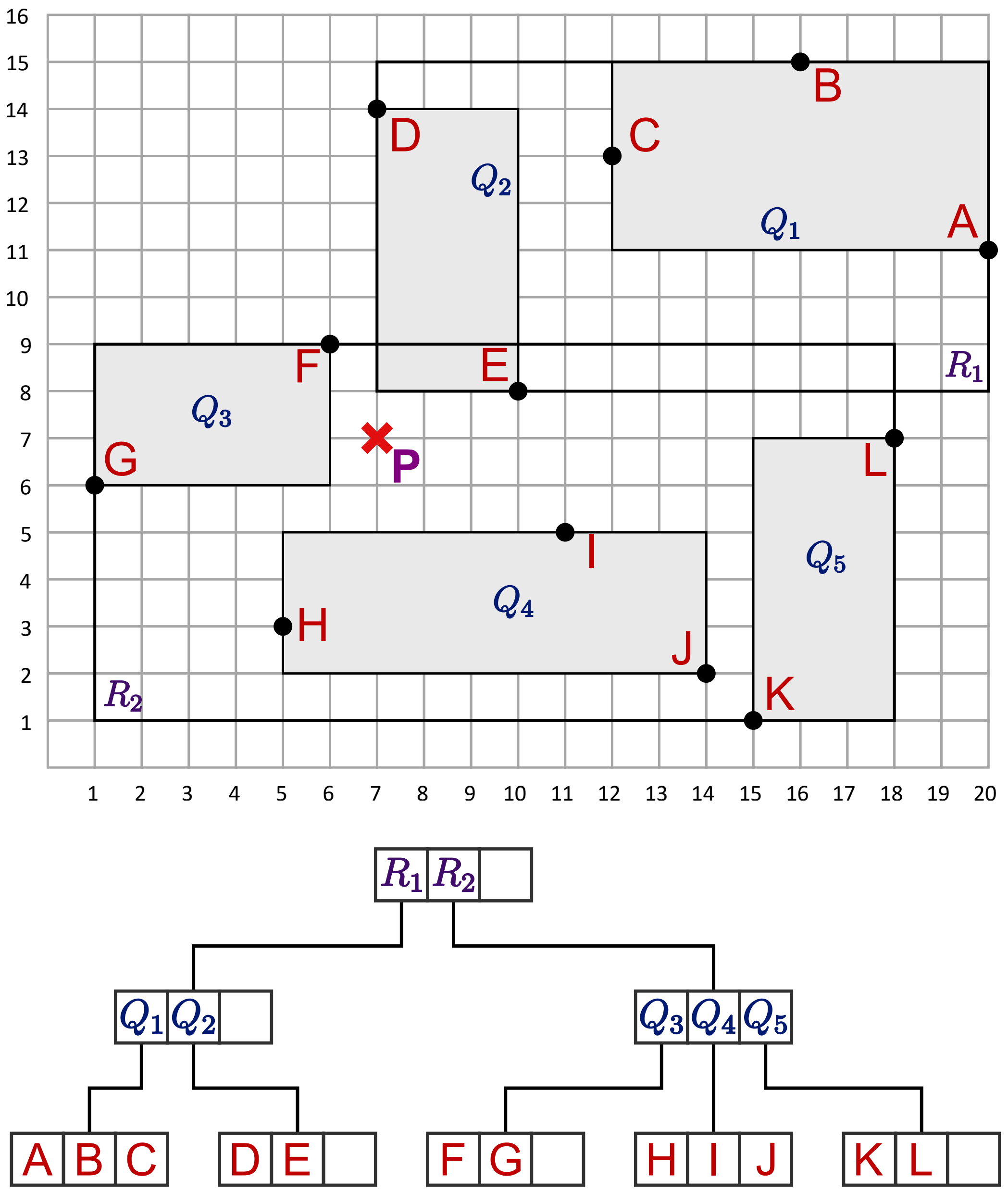
Branch *and bound*



Branch *and bound*





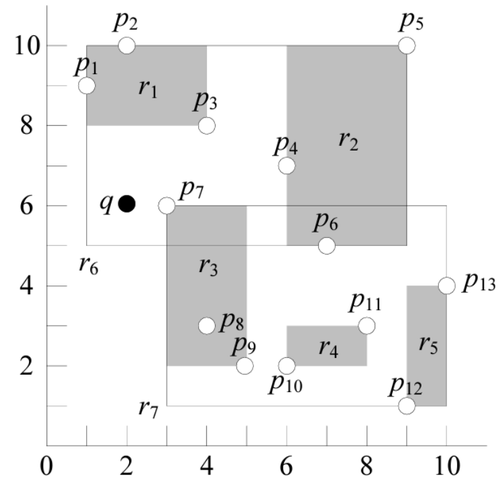


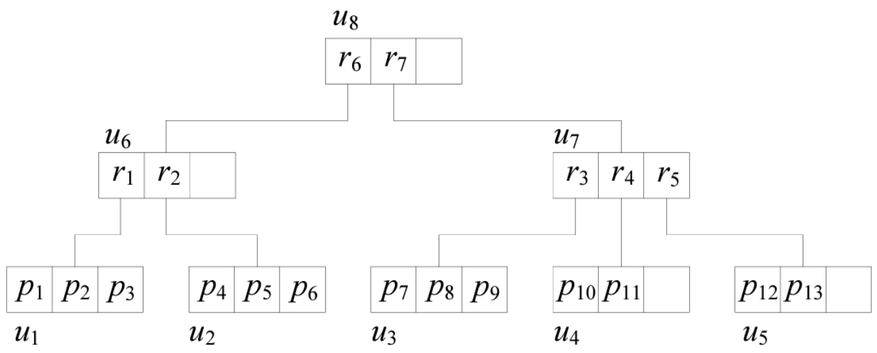
2. Best First

Best *First*

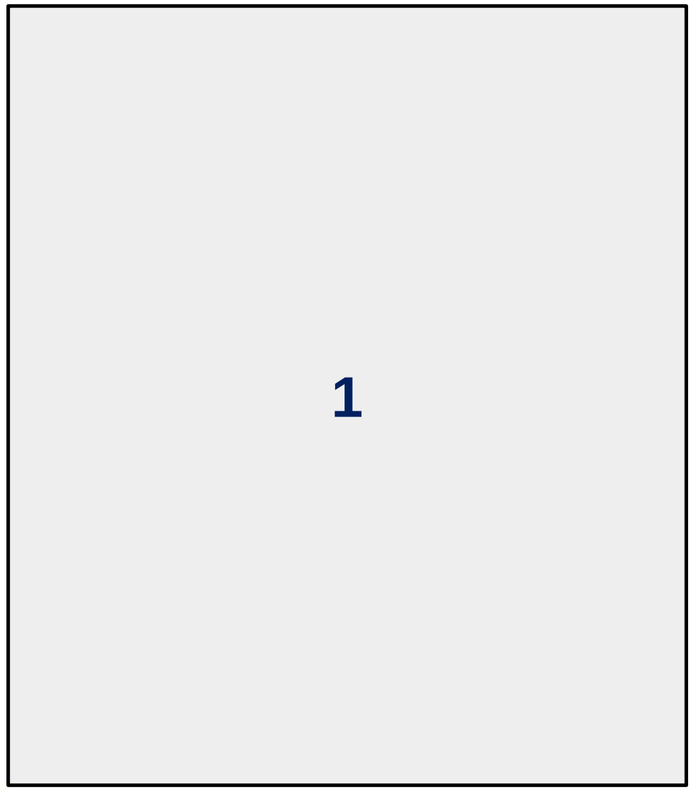
algorithm BF(q)
/* in the following H is a sorted list where each entry is an MBR whose sorting key in H is its mindist to q;
p_{best} is a global variable that keeps the NN found so far. */

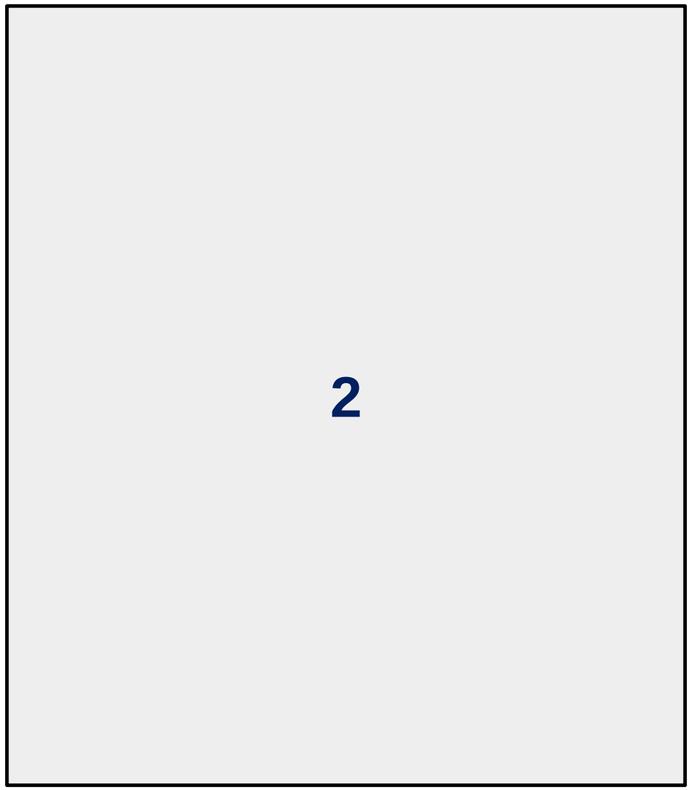
1. insert the MBR of the root in H
2. **while** $\|q, p_{best}\|$ is greater than the smallest mindist in H
   /* if $p_{best} = \emptyset$, $\|q, p_{best}\| = \infty$ */
3. remove from H the MBR r with the smallest mindist
4. access the child node u of r
5. **if** u is an intermediate node **then**
6. insert all the MBRs in u into H
7. **else**
8. **if** the NN of q in u is closer to q than p_{best} **then**
9. p_{best} = the NN of q in u

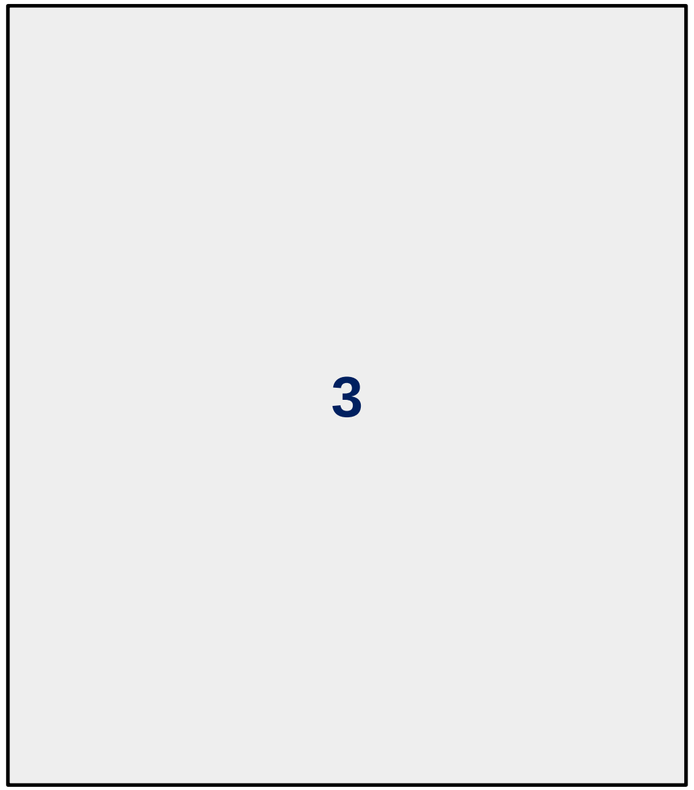




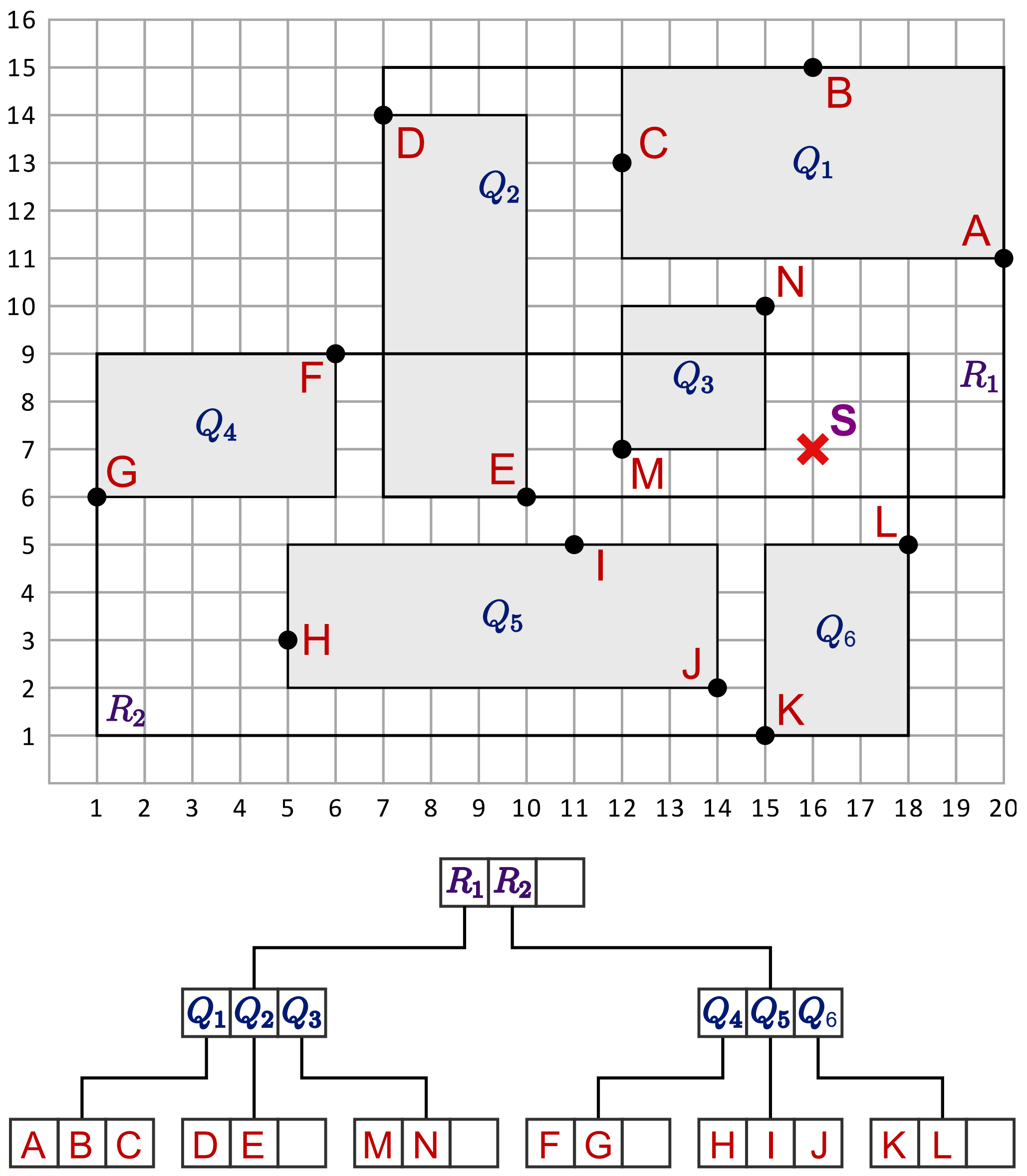
Best *First*

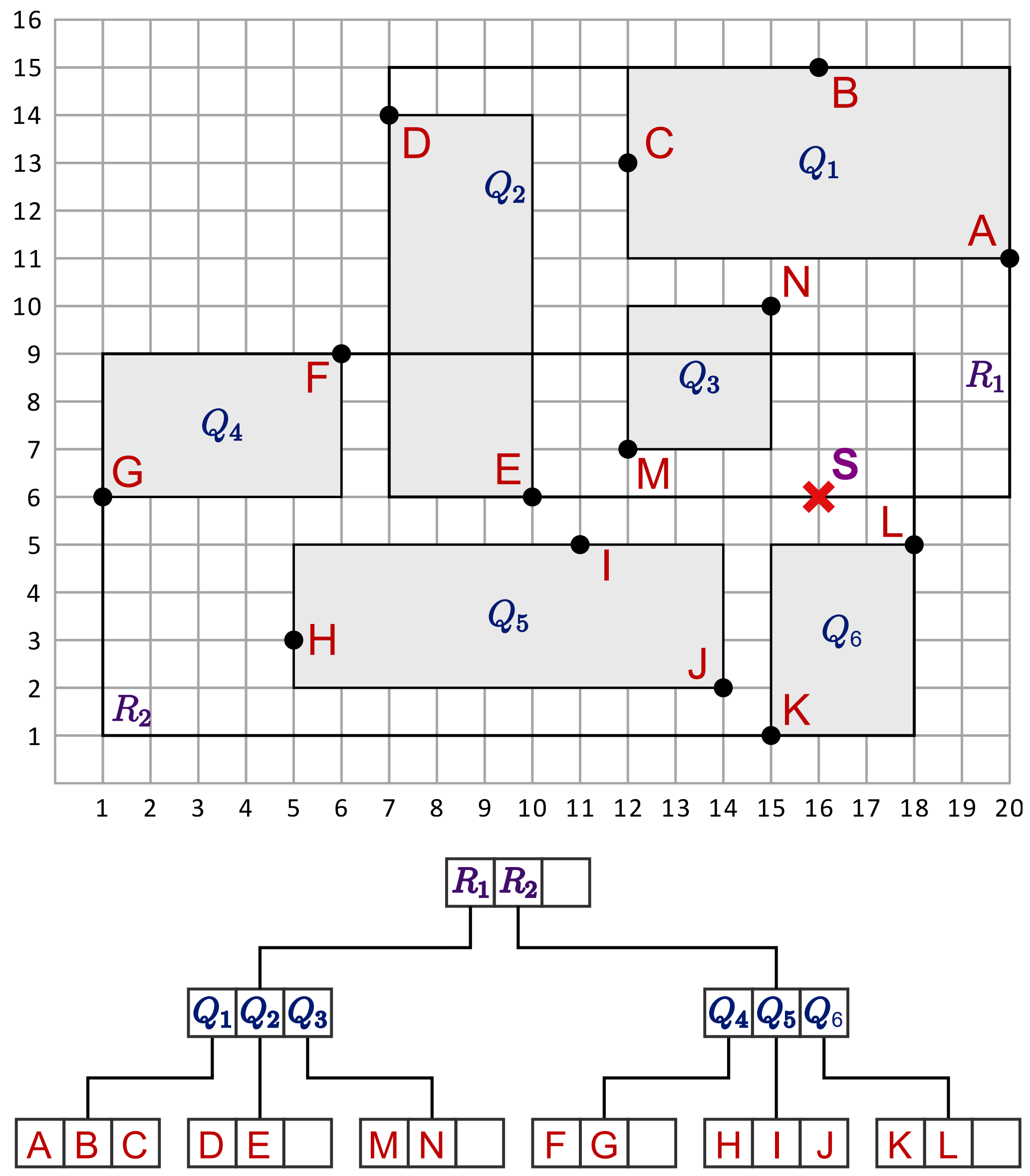


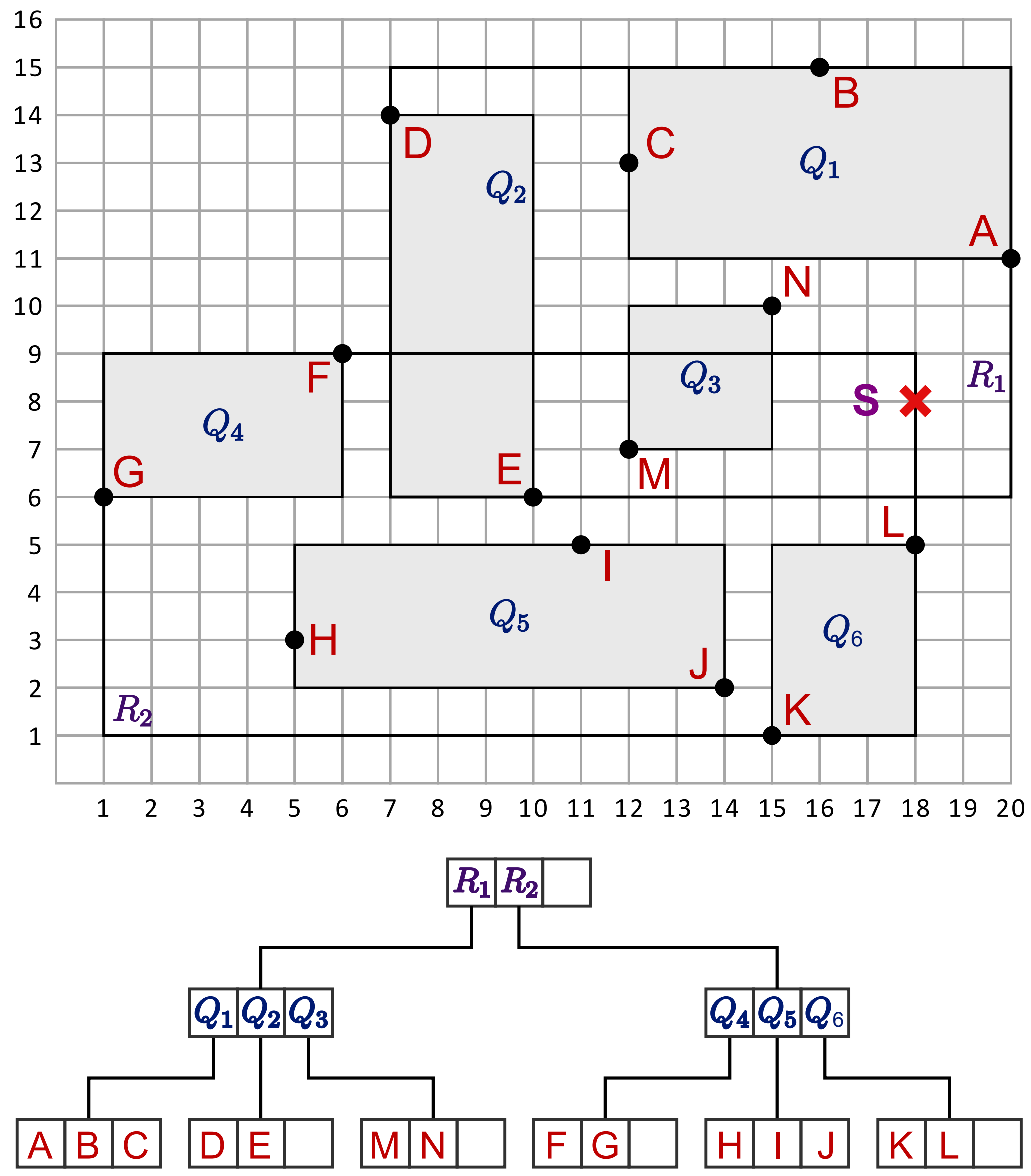




Best *First*


A B C D E F G H I J K L M N S
Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 R_1 R_2


A B C D E F G H I J K L M N S
Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 R_1 R_2


A B C D E F G H I J K L M N S
Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 R_1 R_2

3. R*-Tree: split

R*-Tree: *split*

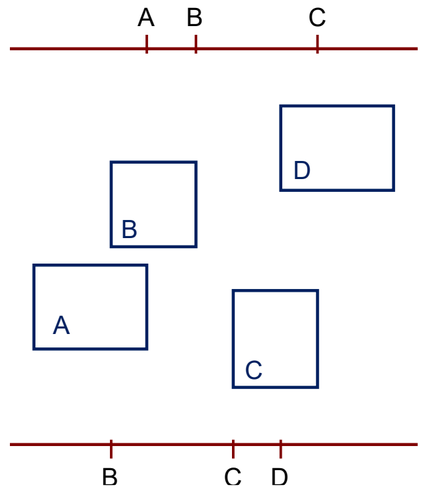
Split

- Seleccionar eje

a) Eje x

- Limite inferior
- Limite superior

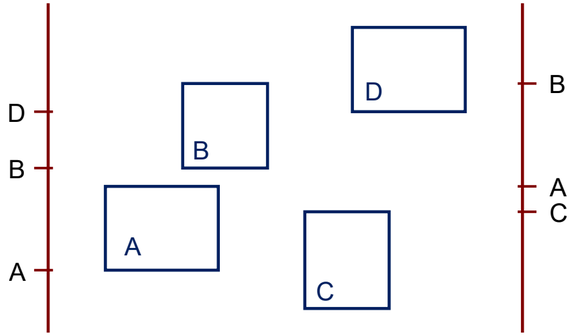
Suma de semiperímetros de las regiones



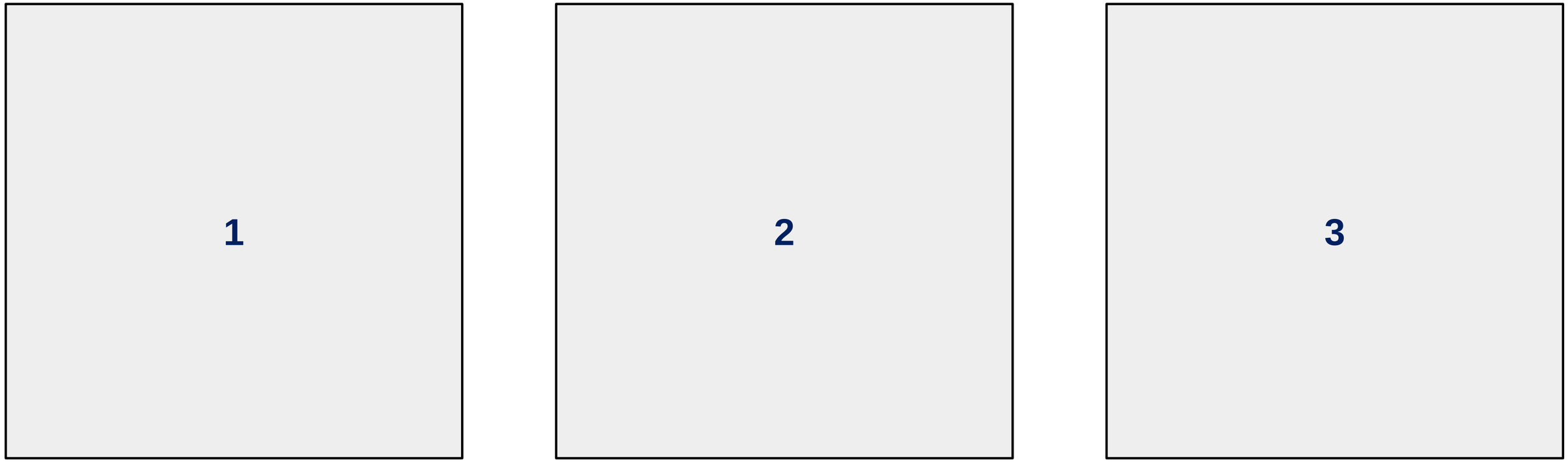
b) Eje y

- Limite inferior
- Limite superior

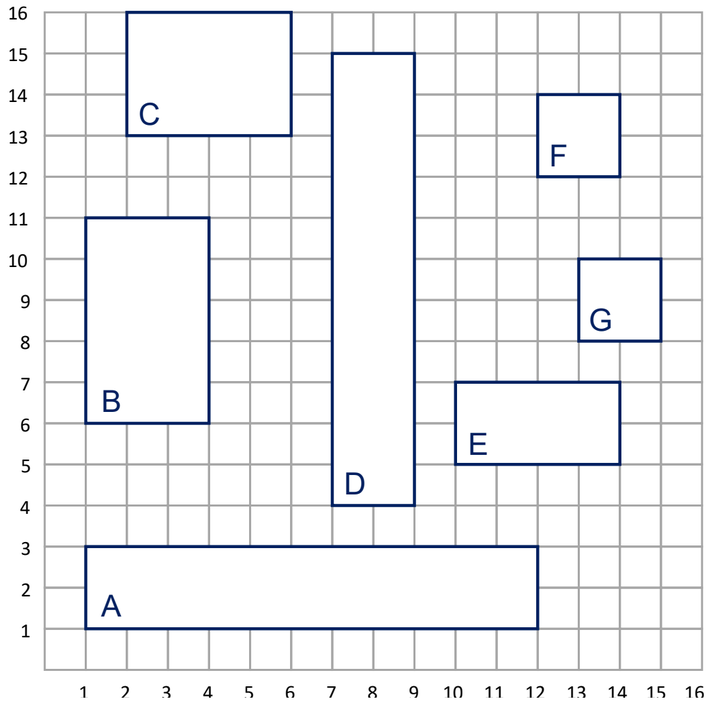
Suma de semiperímetros de las regiones

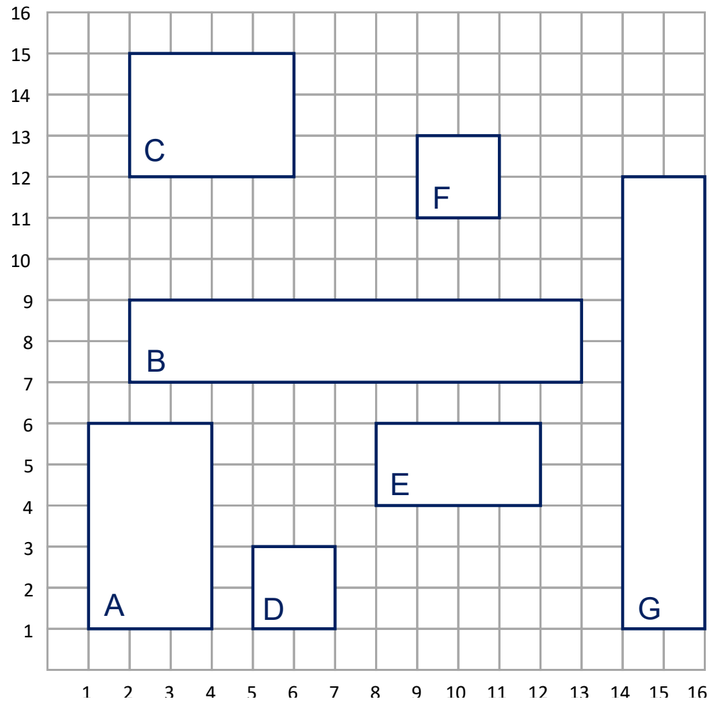


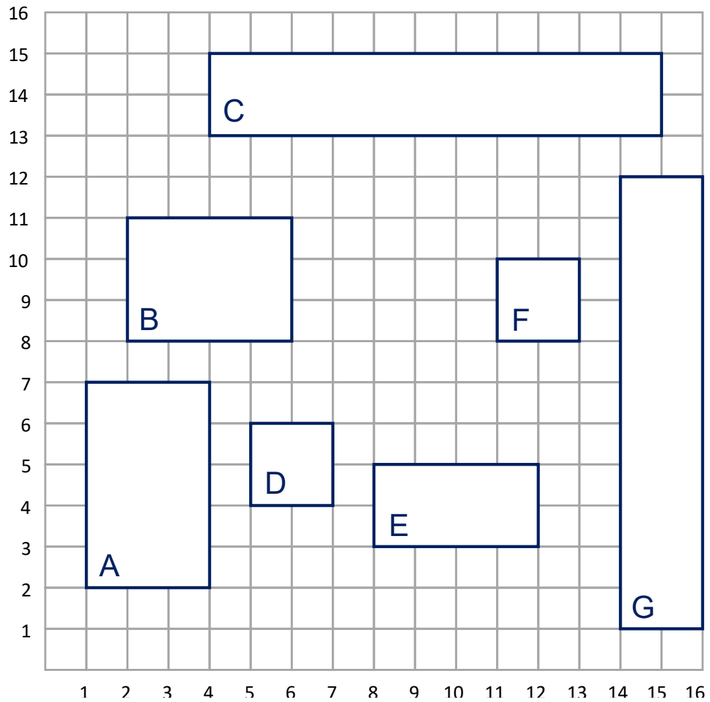
R*-Tree: *split*



R*-Tree: *split*

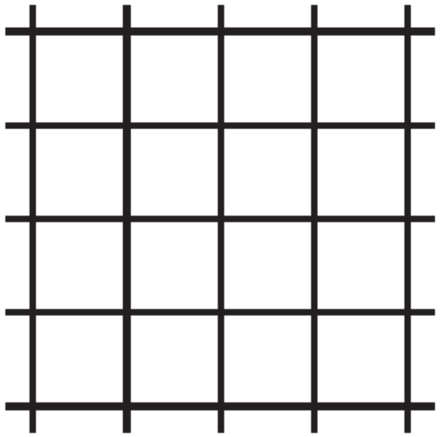






4. Celdas/mosaico

Celdas/*mosaico*



Notación $[4^4]$

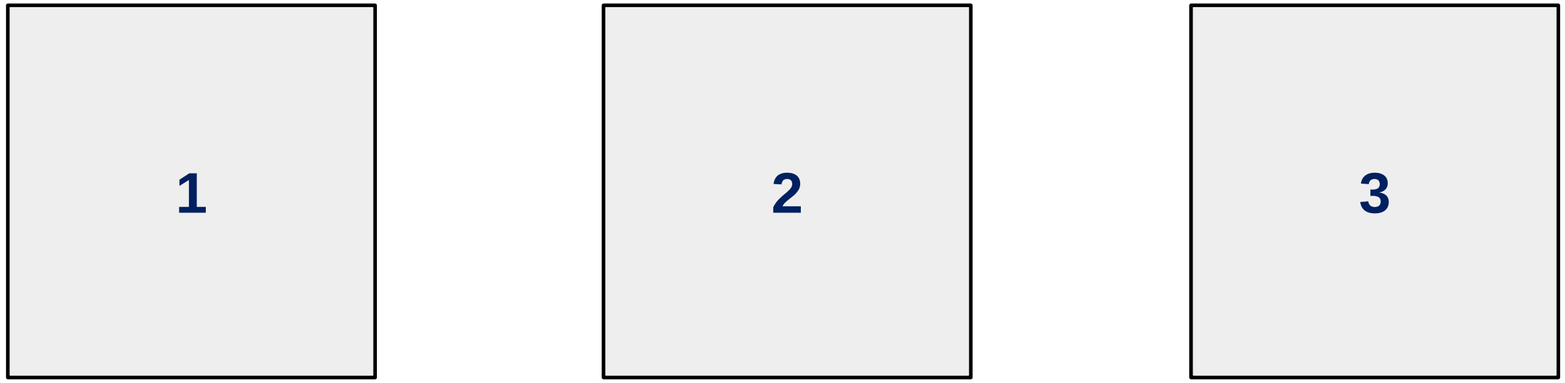
Regular tiling Similar tiling

Mosaico ilimitado La celda atómica se puede descomponer

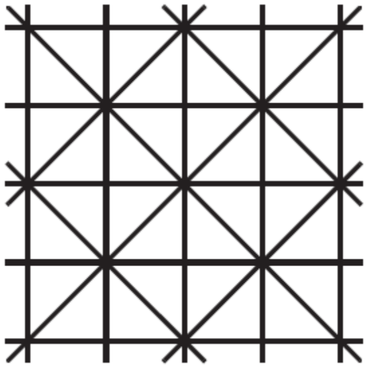
Número de adyacencia: 2

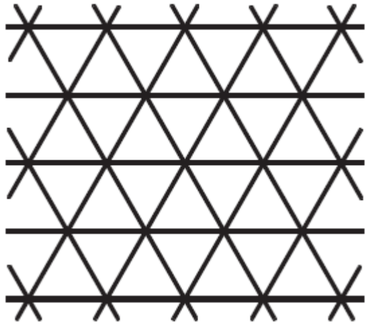
Orientación uniforme

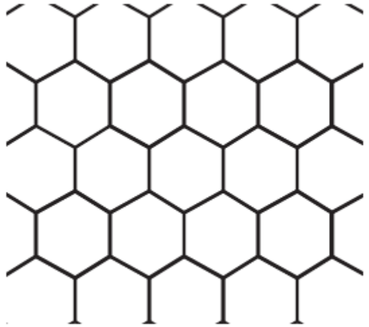
Celdas/*mosaico*



Celdas/*mosaico*







Notación

Similar tiling?

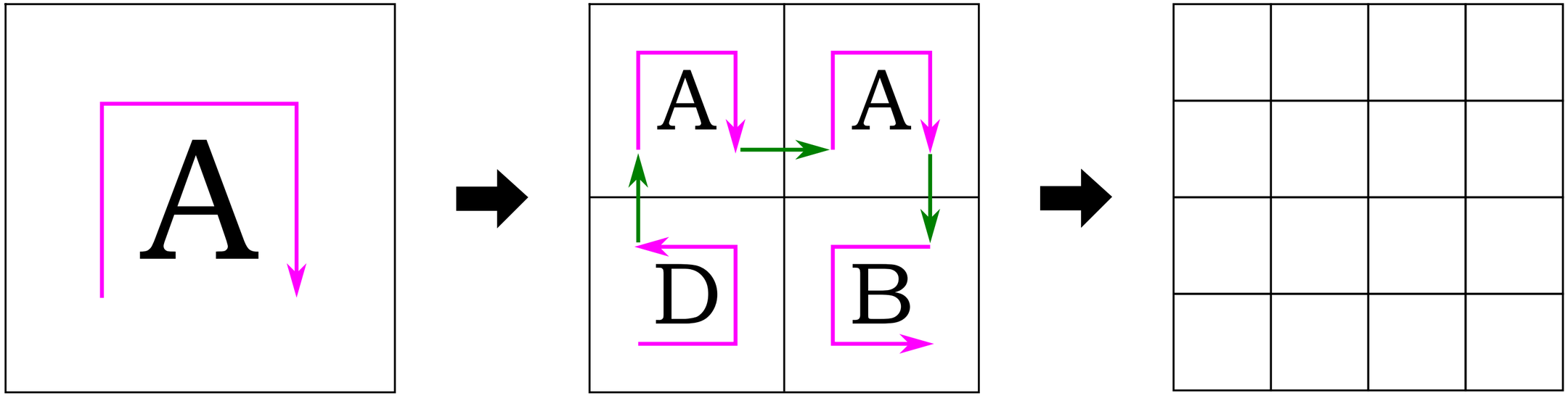
Mosaico limitado?

Número de adyacencia:

Orientación uniforme?:

5. Hilbert-curve

Hilbert-*curve*



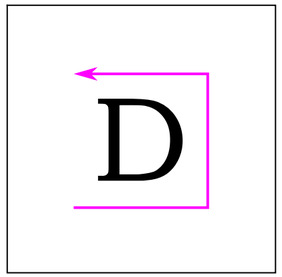
Hilbert-*curve*

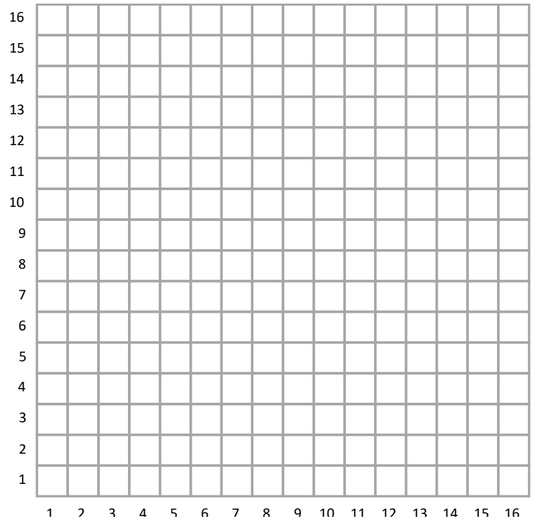
1

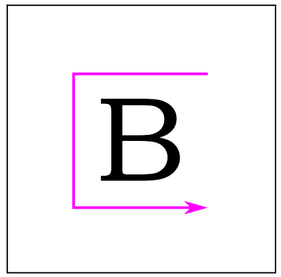
2

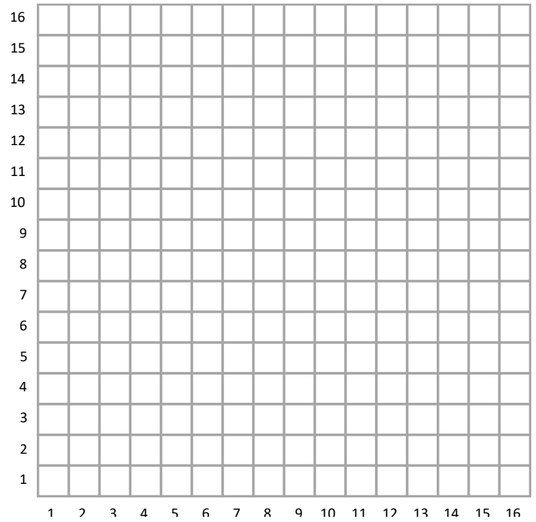
3

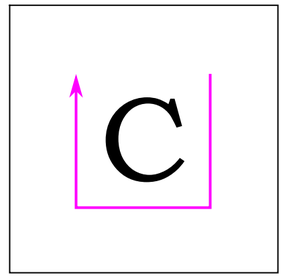
Hilbert-*curve*

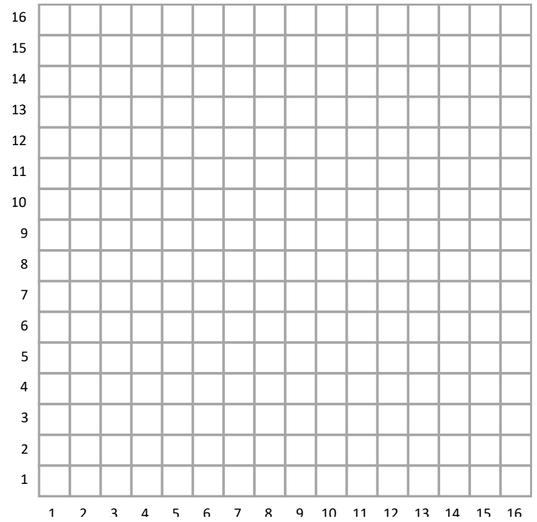






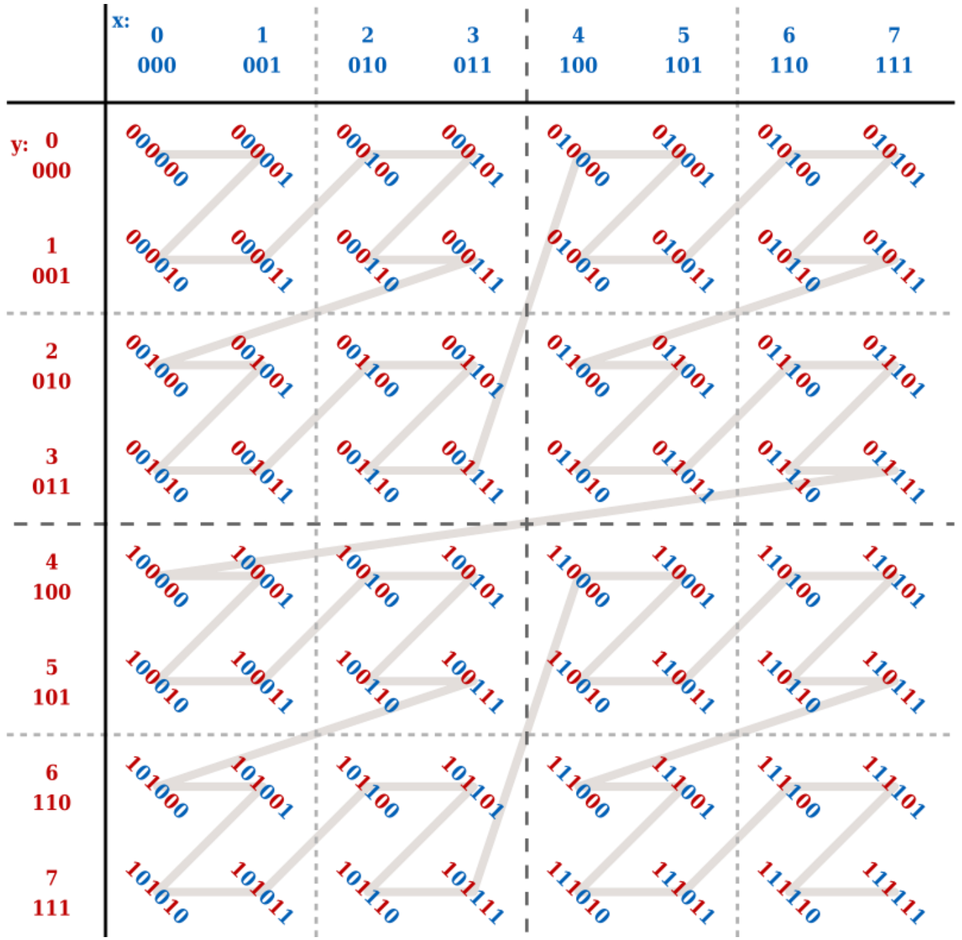




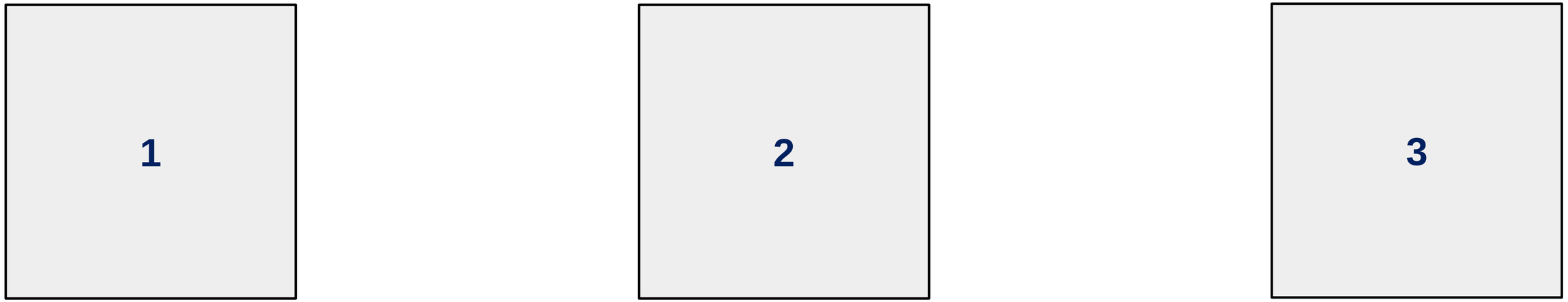


6. z-curve

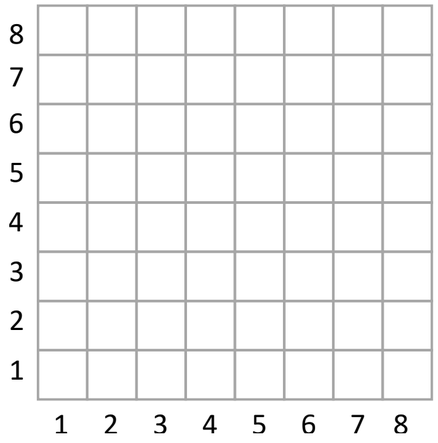
Z-curve

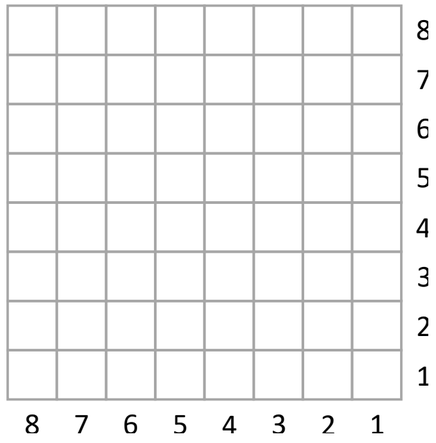

x:
0 000
1 001
2 010
3 011
4 100
5 101
6 110
7 111
y:
0 000
1 001
2 010
3 011
4 100
5 101
6 110
7 111
000000 000001 000100 000101 010000 010001 010100 010101
000010 000011 000110 000111 010010 010011 010110 010111
001000 001001 001100 001101 011000 011001 011100 011101
001010 001011 001110 001111 011010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

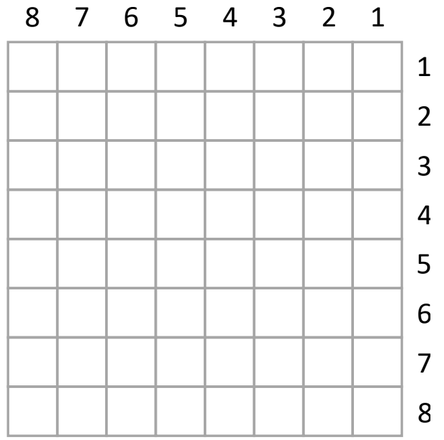
Z-curve



Z-curve

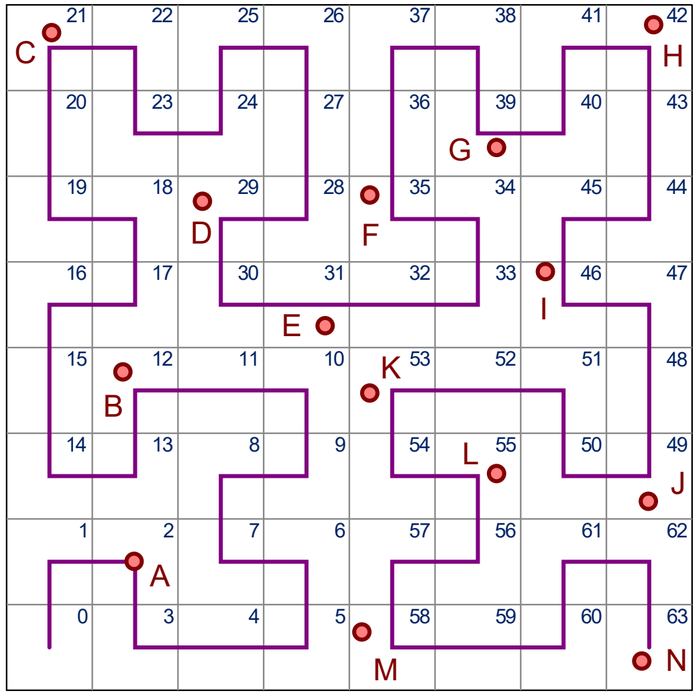


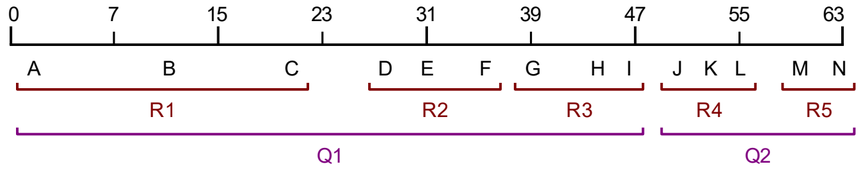


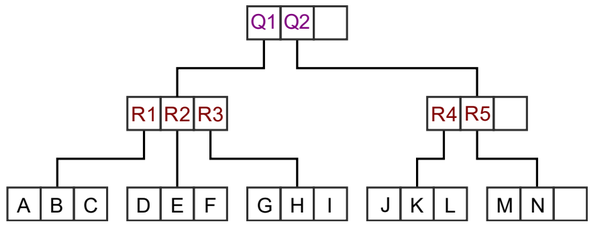


7. Hilbert R-Tree

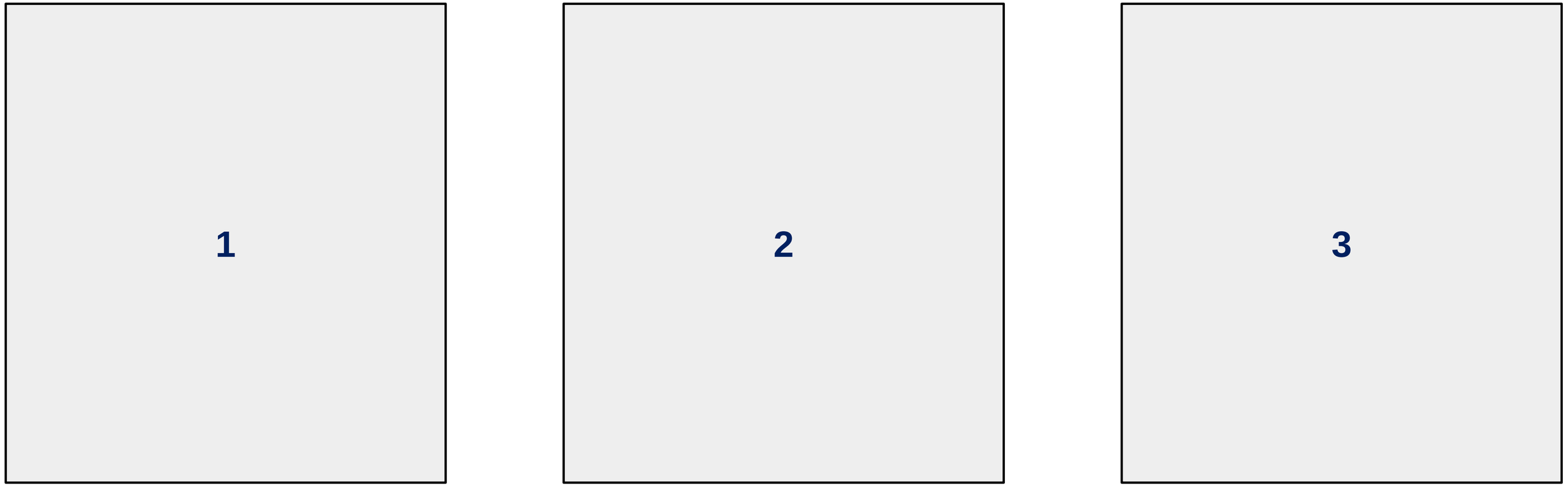
Hilbert R-*Tree*





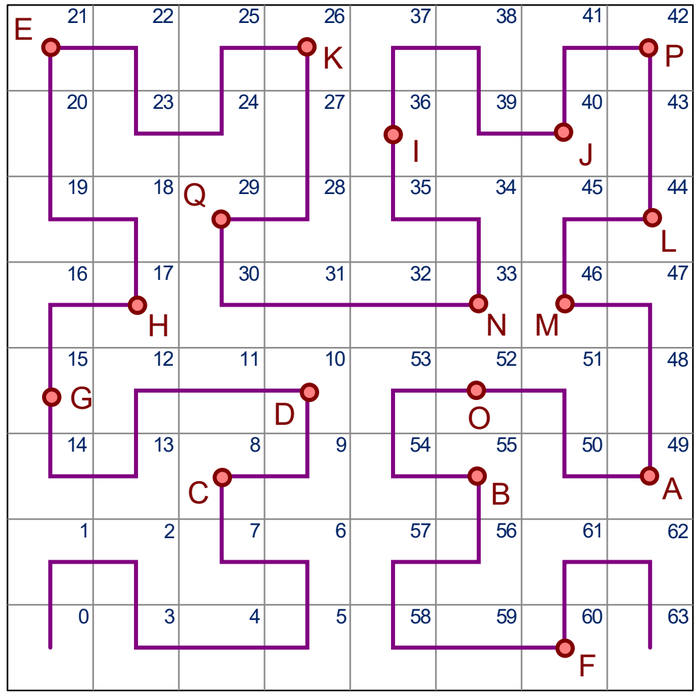


Hilbert R-*Tree*

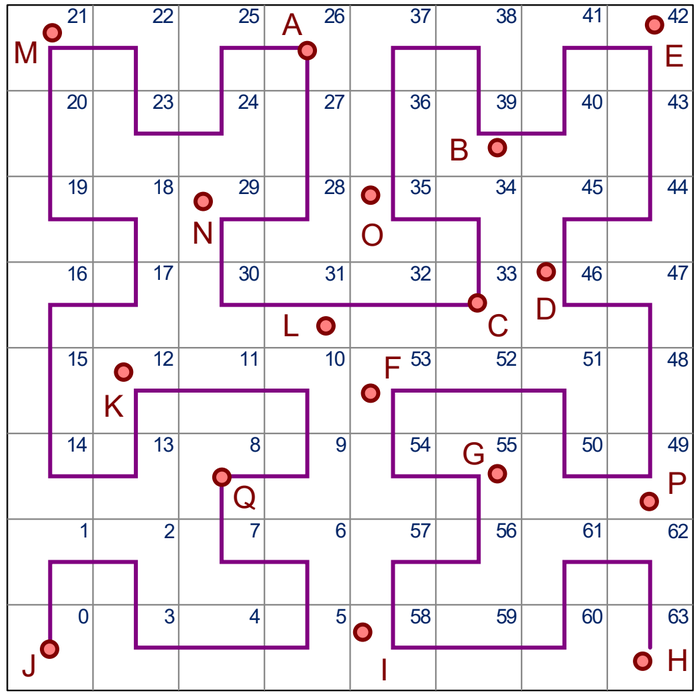


Hilbert R-*Tree*

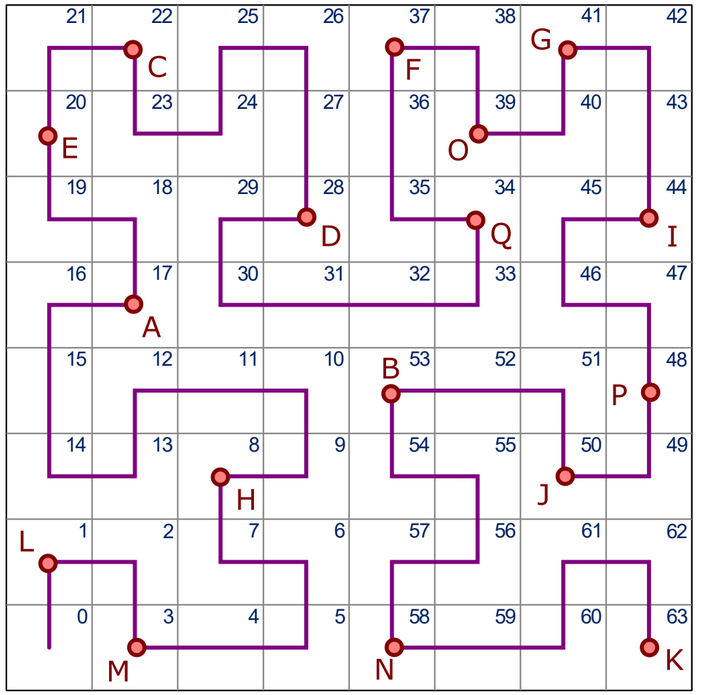
$M = 4$



$M = 3$



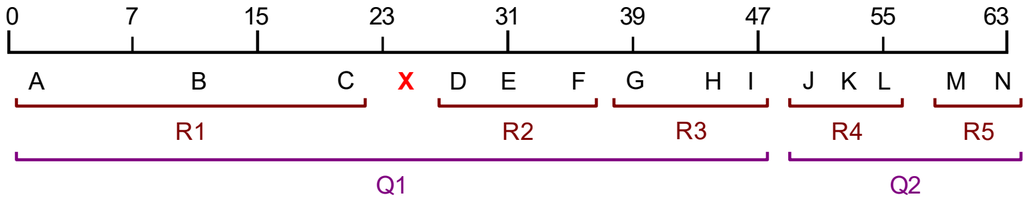
$M = 2$

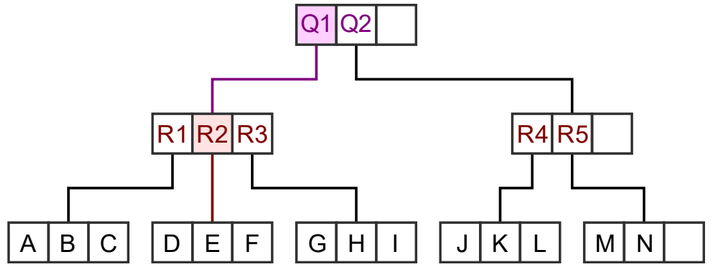


8. Dynamic Hilbert R-Tree

Dynamic Hilbert *R-Tree*

Durante la inserción, se selecciona el nodo con **LHV mínimo que supera al h-index** del **nuevo dato**.





Sobrecarga Buscamos apoyo del hermano **izquierdo**.

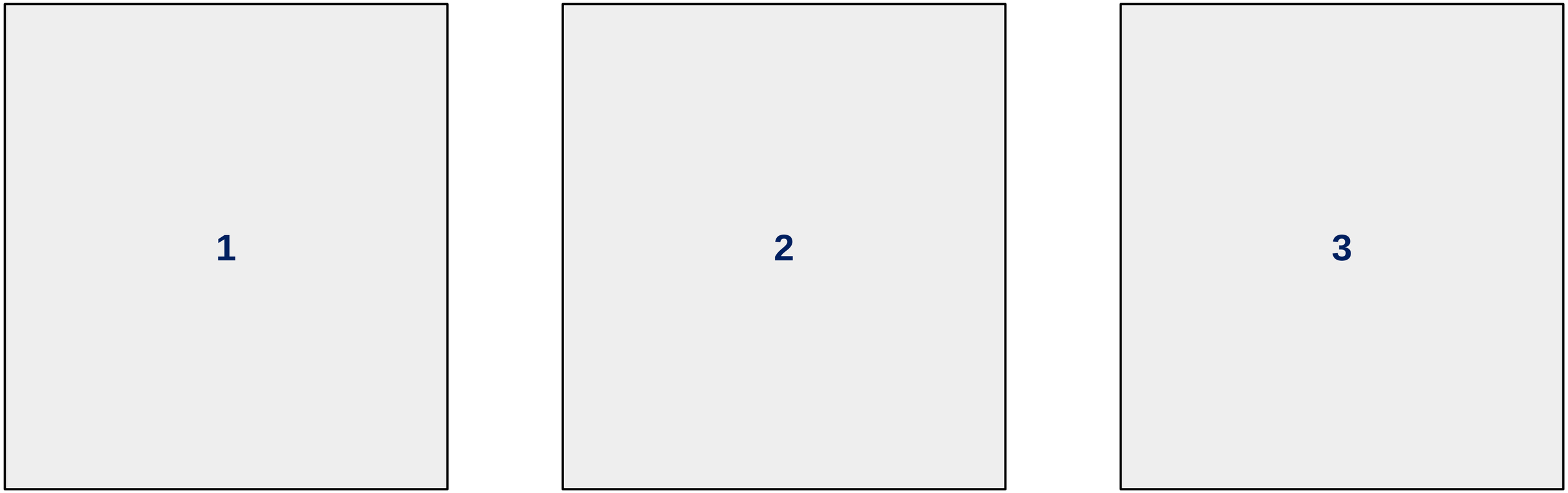
Repartimos todos los datos entre **todos** los nodos

Pero, que hacemos si **todos** los nodos izquierdos están llenos?

Creamos nuevo nodo

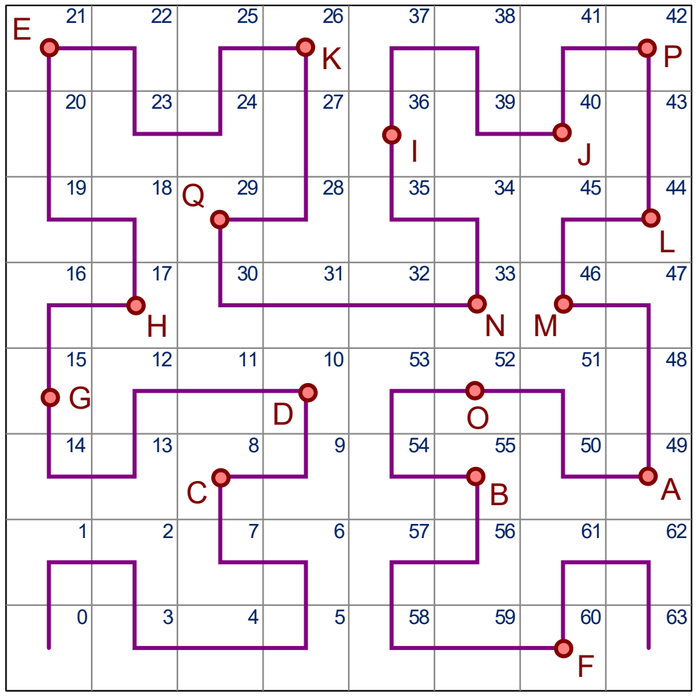
… y repartimos todos los datos entre **todos** los nodos

Dynamic Hilbert *R-Tree*

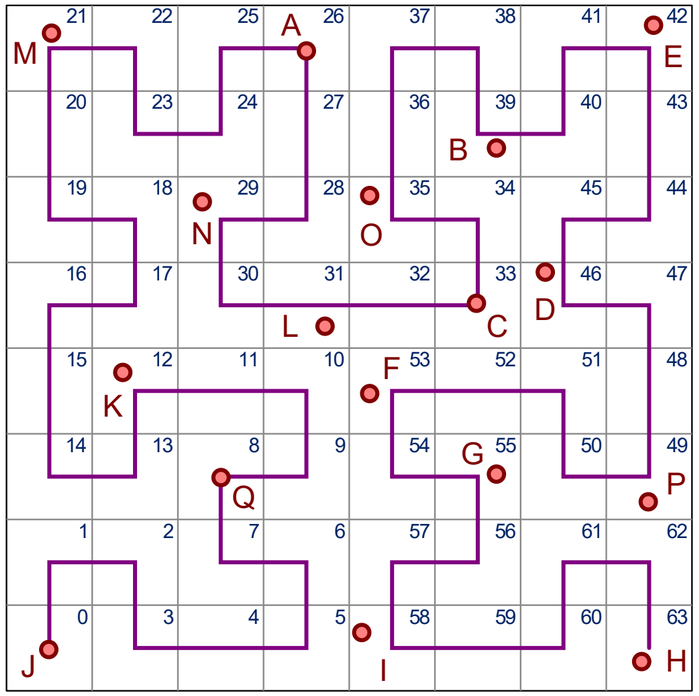


Dynamic Hilbert *R-Tree*

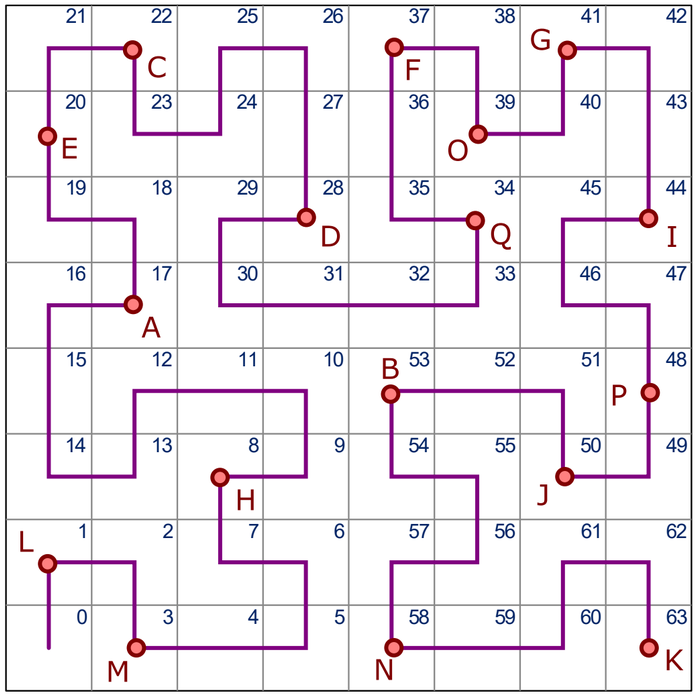
$M = 3$



$M = 3$



$M = 3$



UTEC

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA