

Sesión 7.1: Hilbert R-Tree

CS3102 EDA

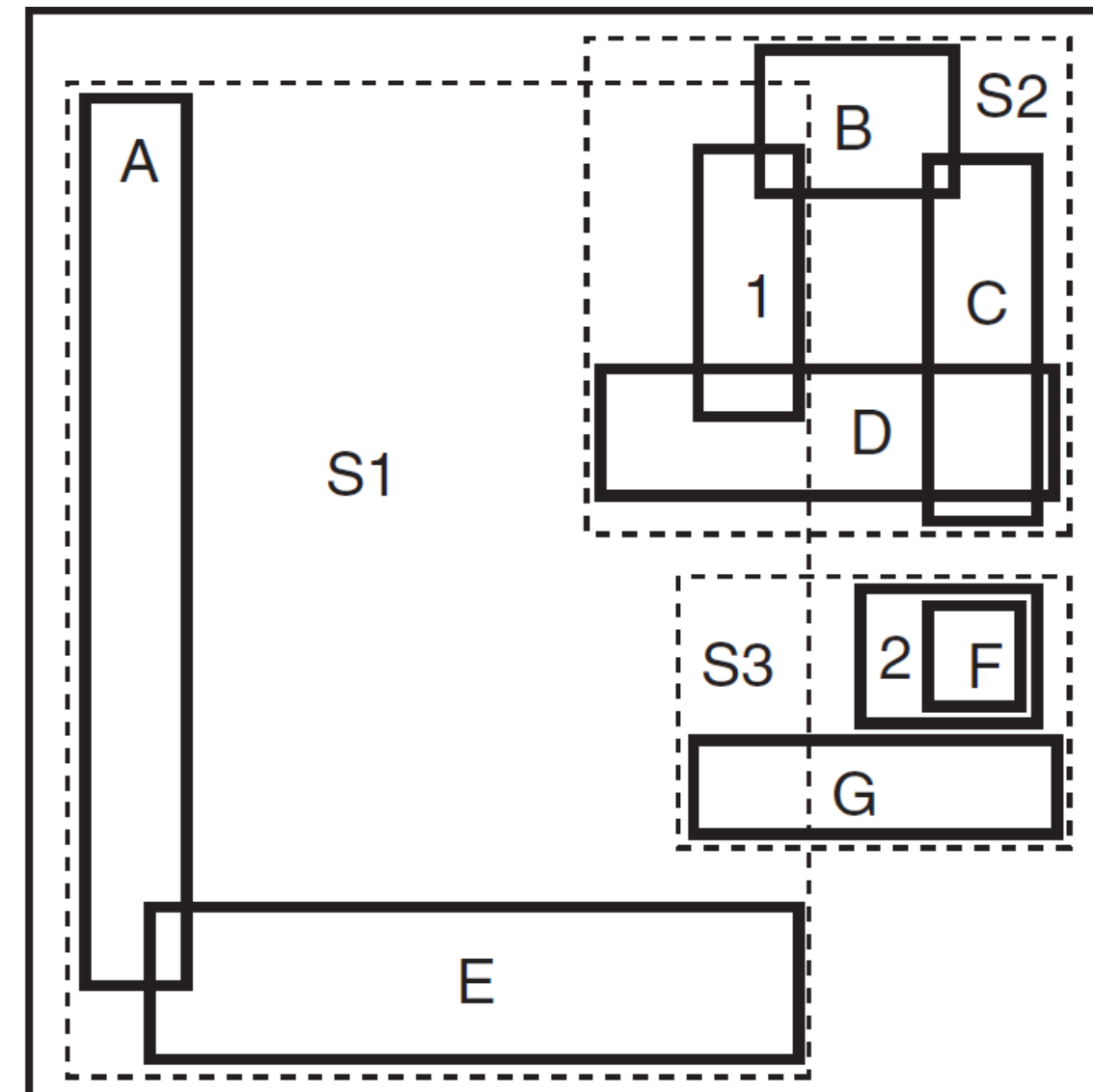
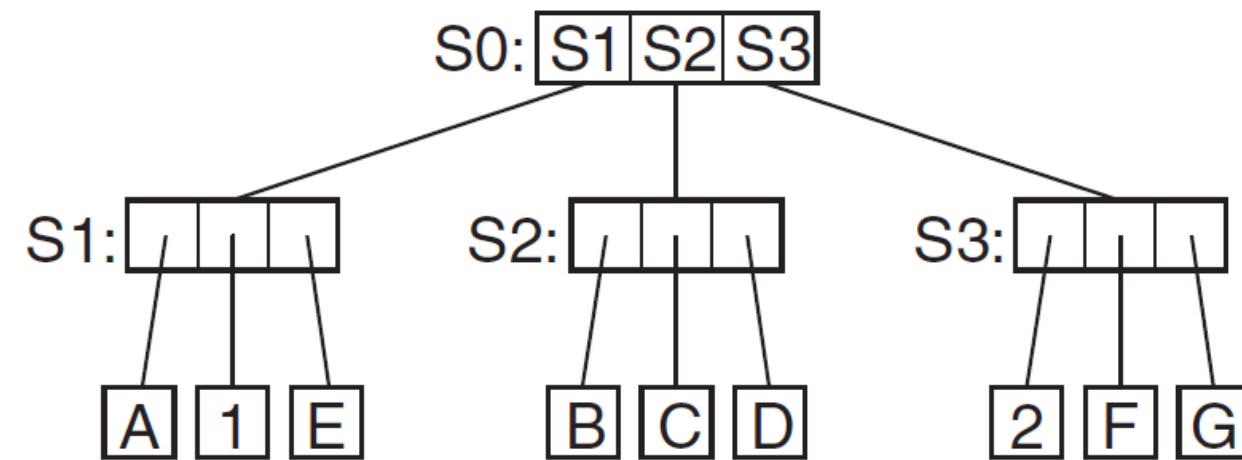
Índice

1. Space-filling curves
2. Hilbert curves

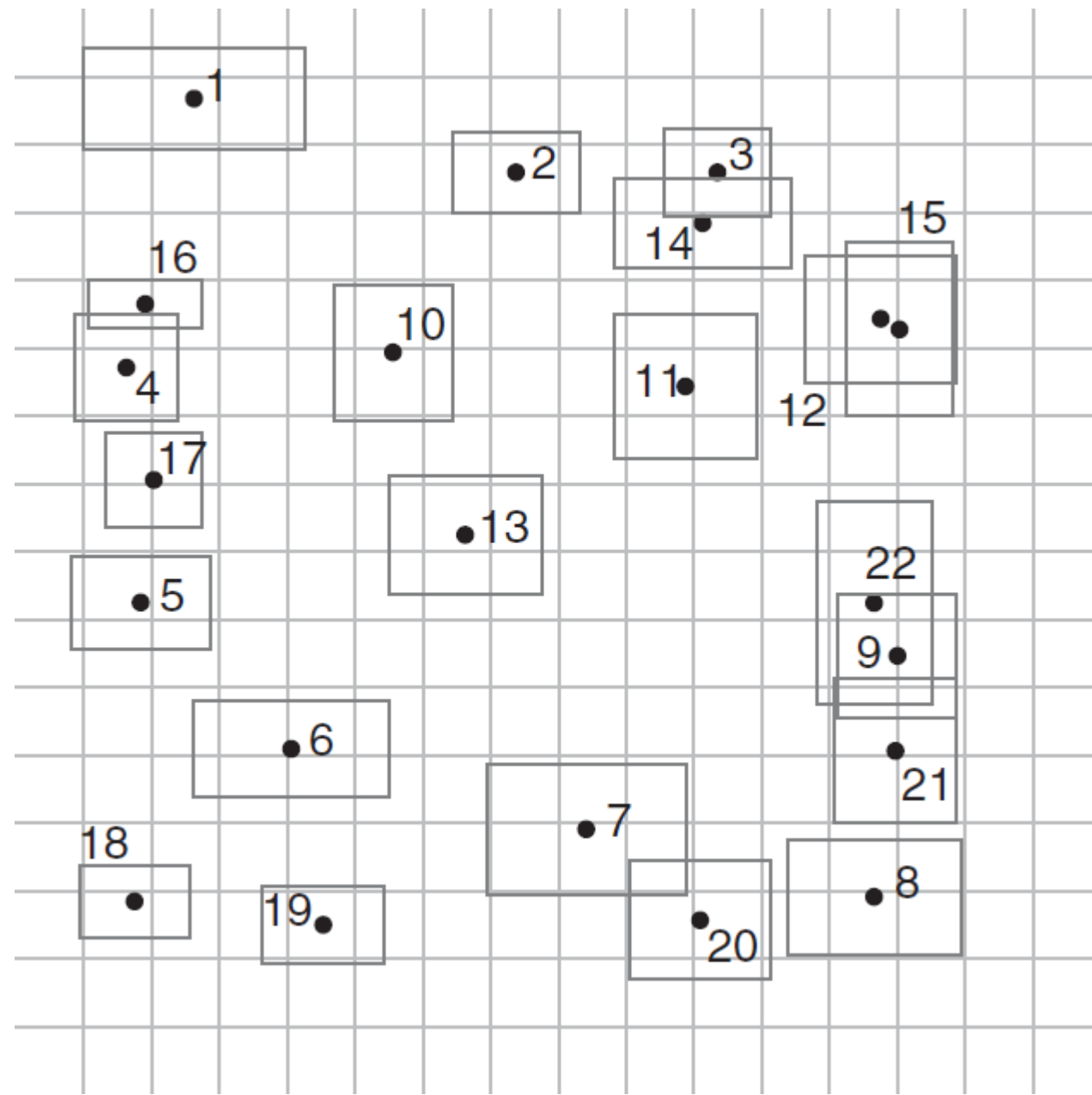


1. Hilbert R-Tree

Hilbert *R-Tree*

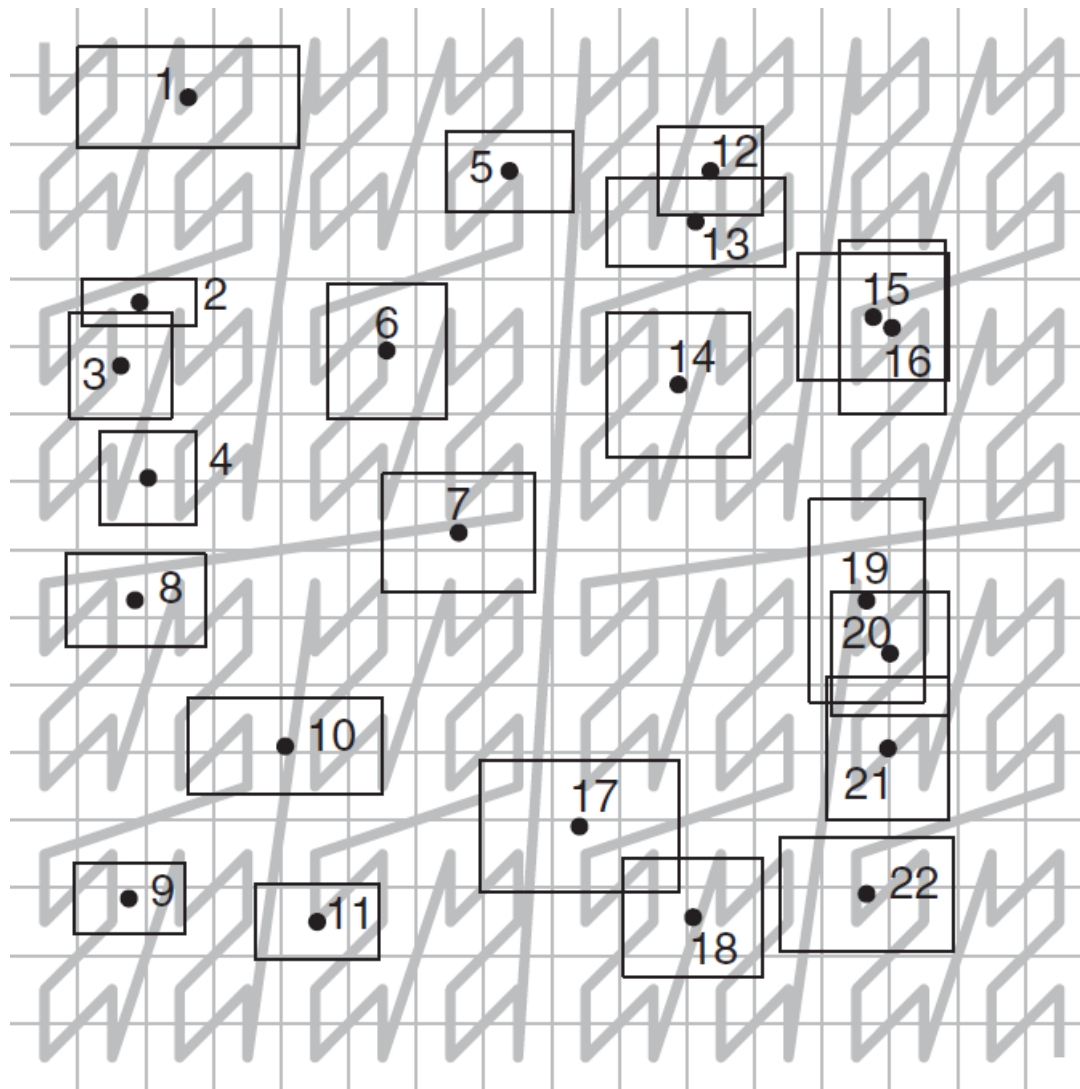


Hilbert *R-Tree*

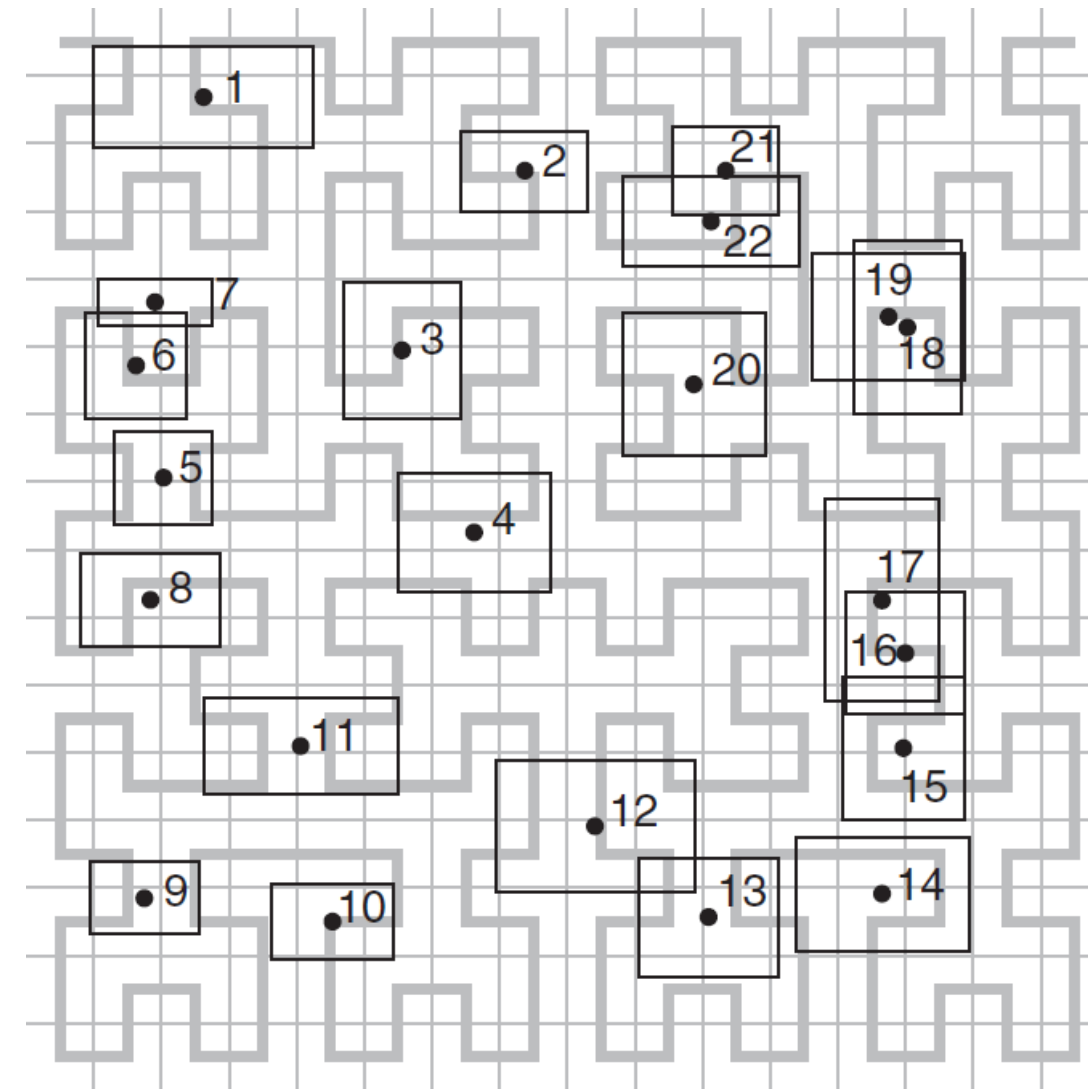


Los MBB son “**representados**” por sus centroides
 Cada MBB es enumerado por el H-index de su centroide

Hilbert *R-Tree*

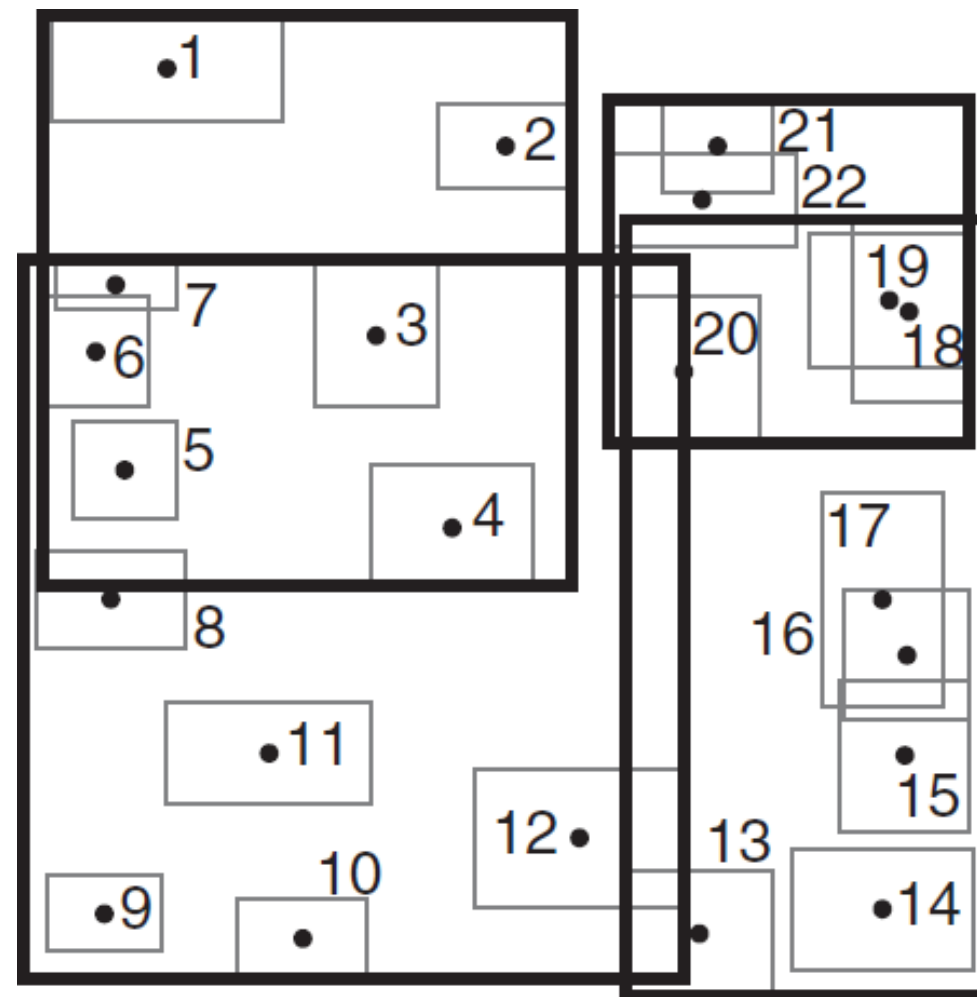


Z-order



Peano-Hilbert order

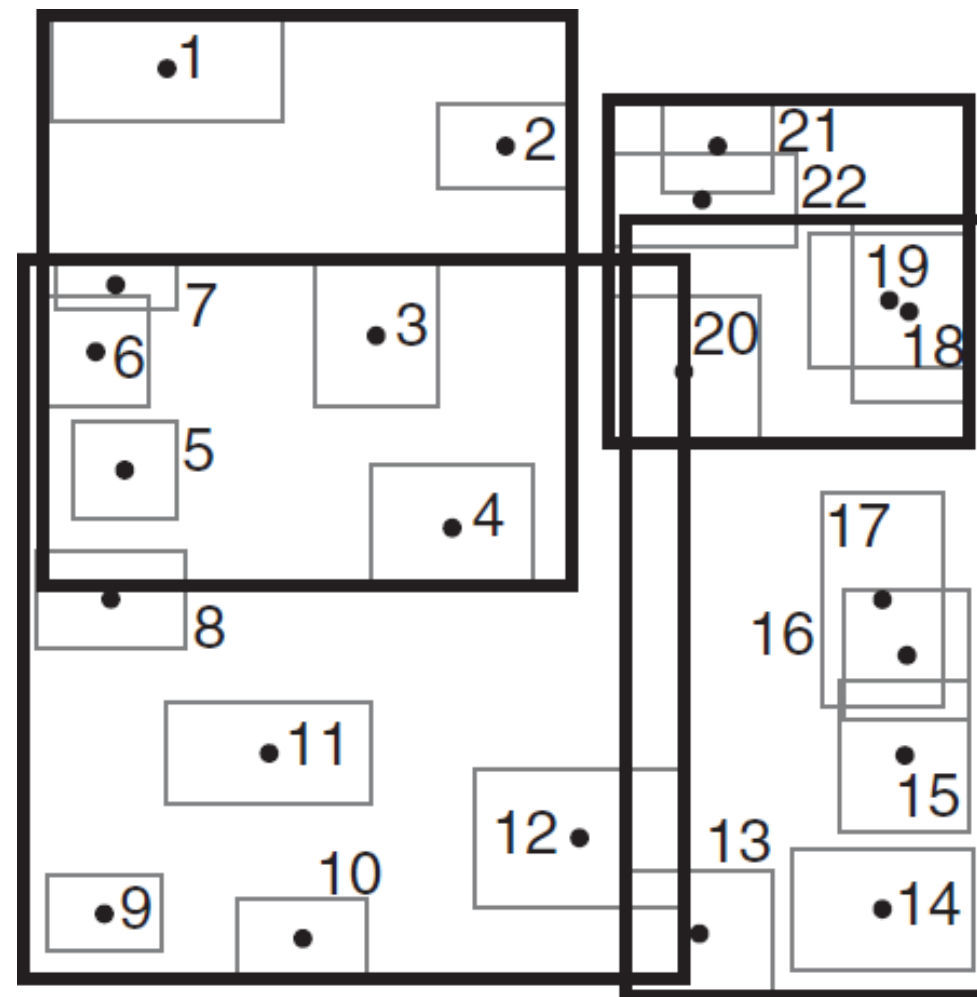
Hilbert *R-Tree*



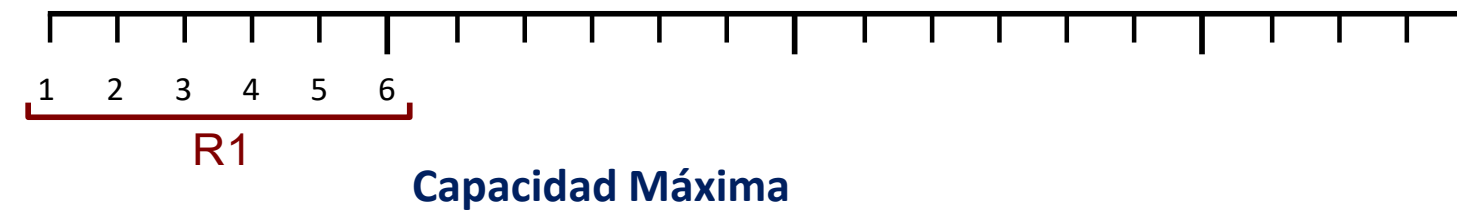
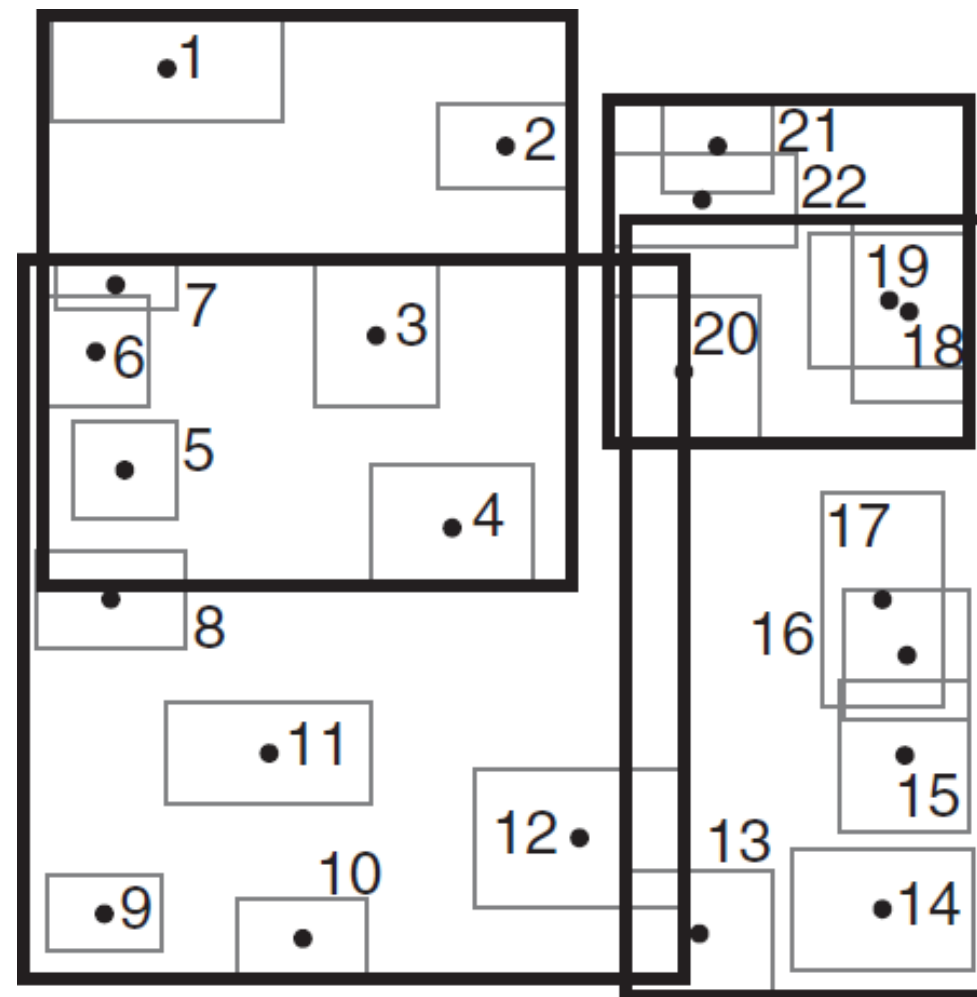
R-Tree se llena empaquetando $n = 6$ bloques, en orden del H-index

Esta variante solo funciona como estructura estática

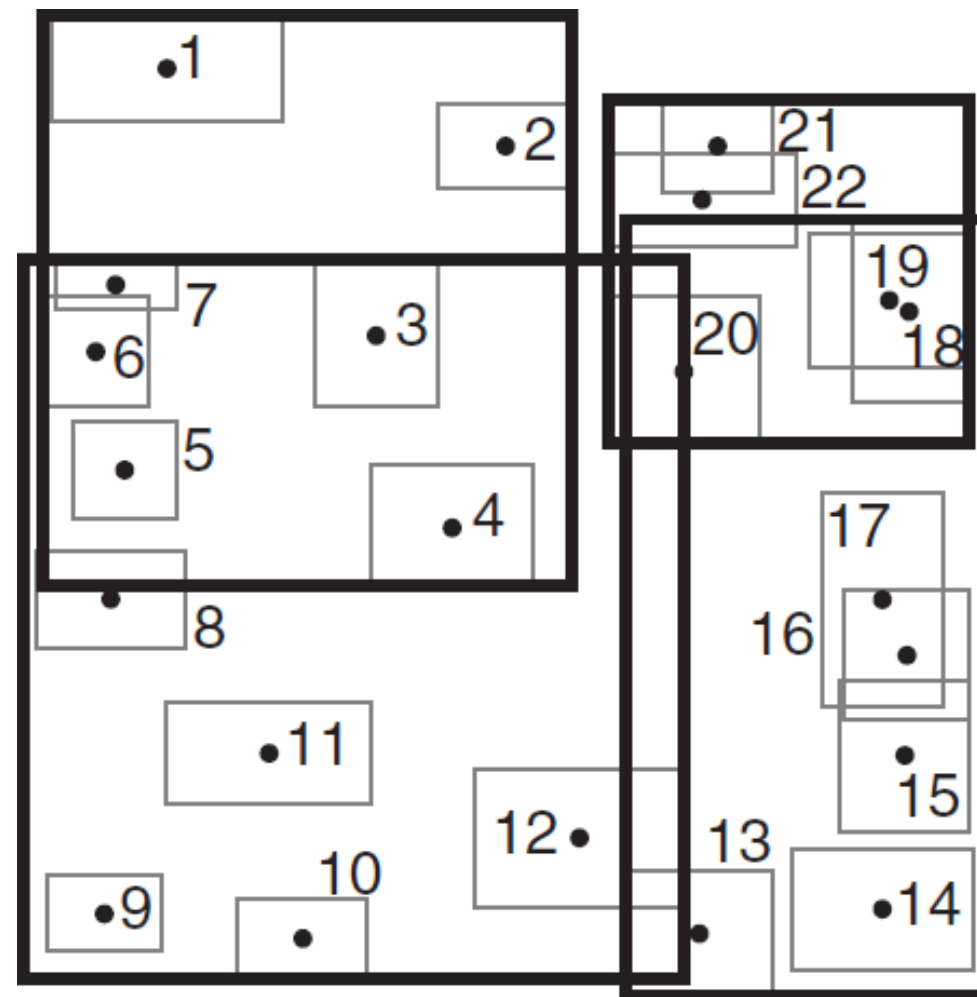
Hilbert *R-Tree*



Hilbert *R-Tree*

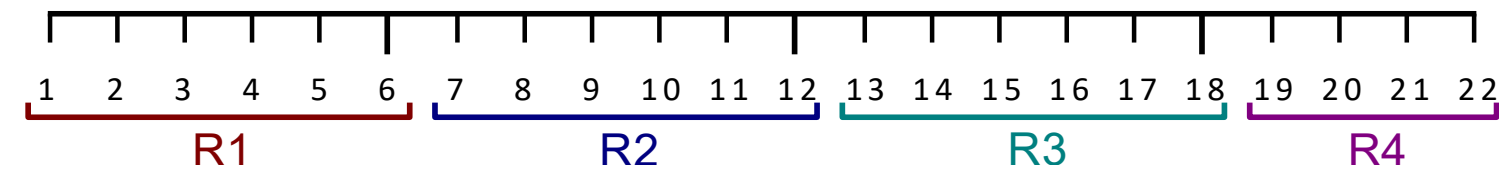
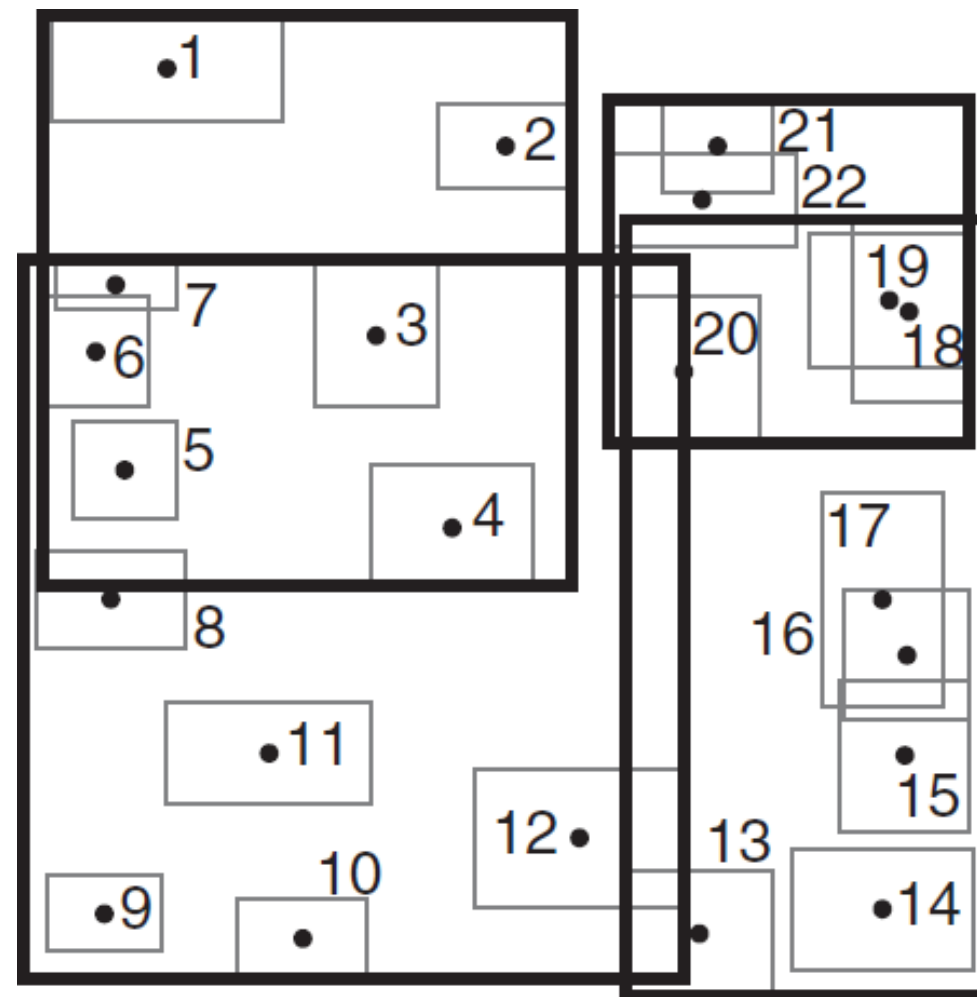


Hilbert *R-Tree*

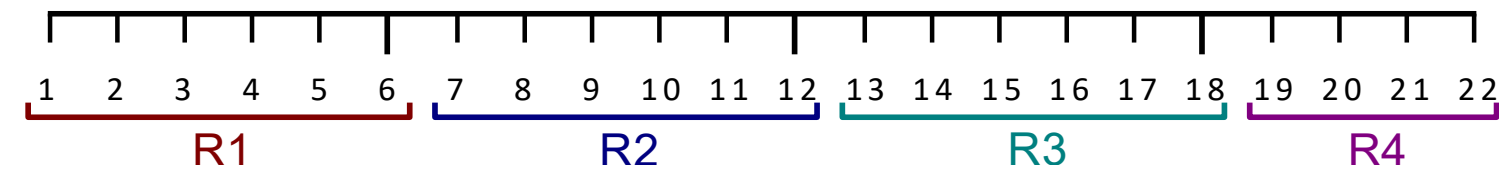
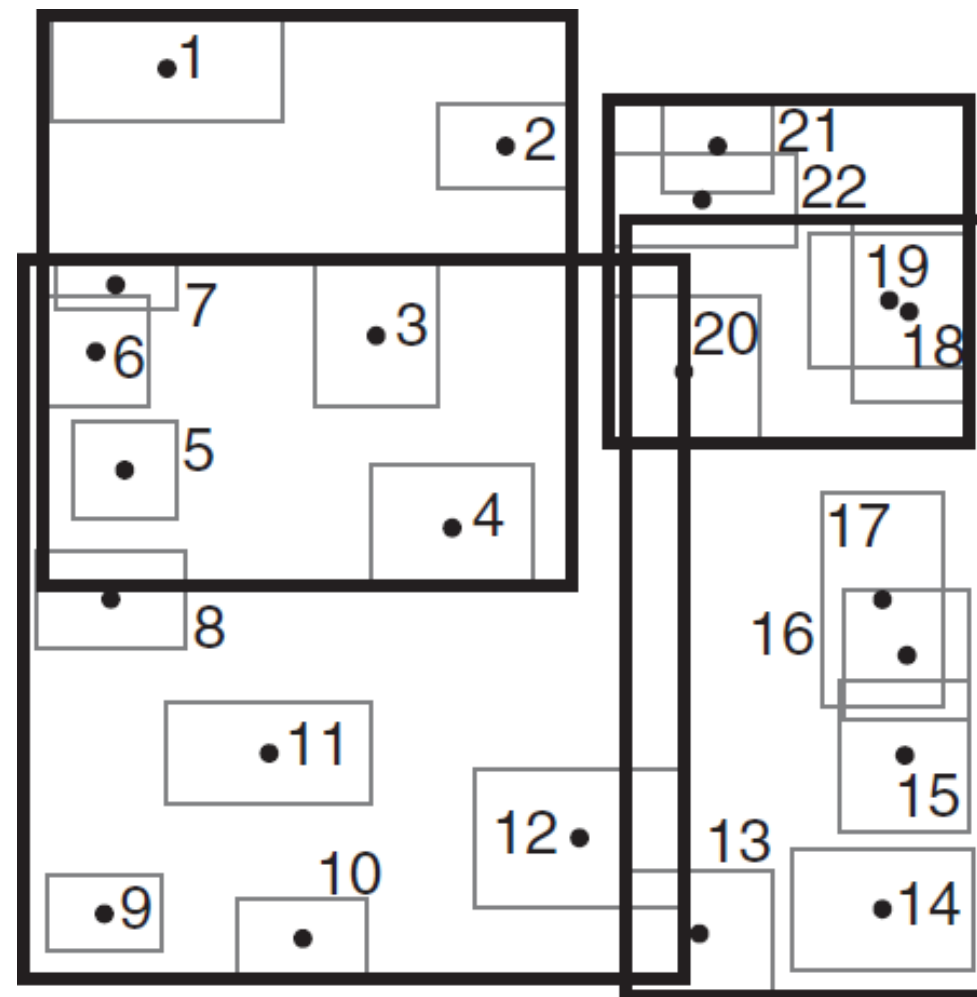


AQUÍ HAY UN ERROR
 la repartición debería ser equitativa

Hilbert *R-Tree*

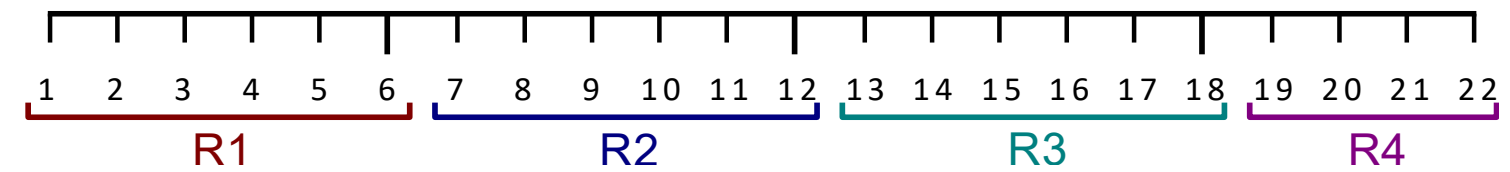
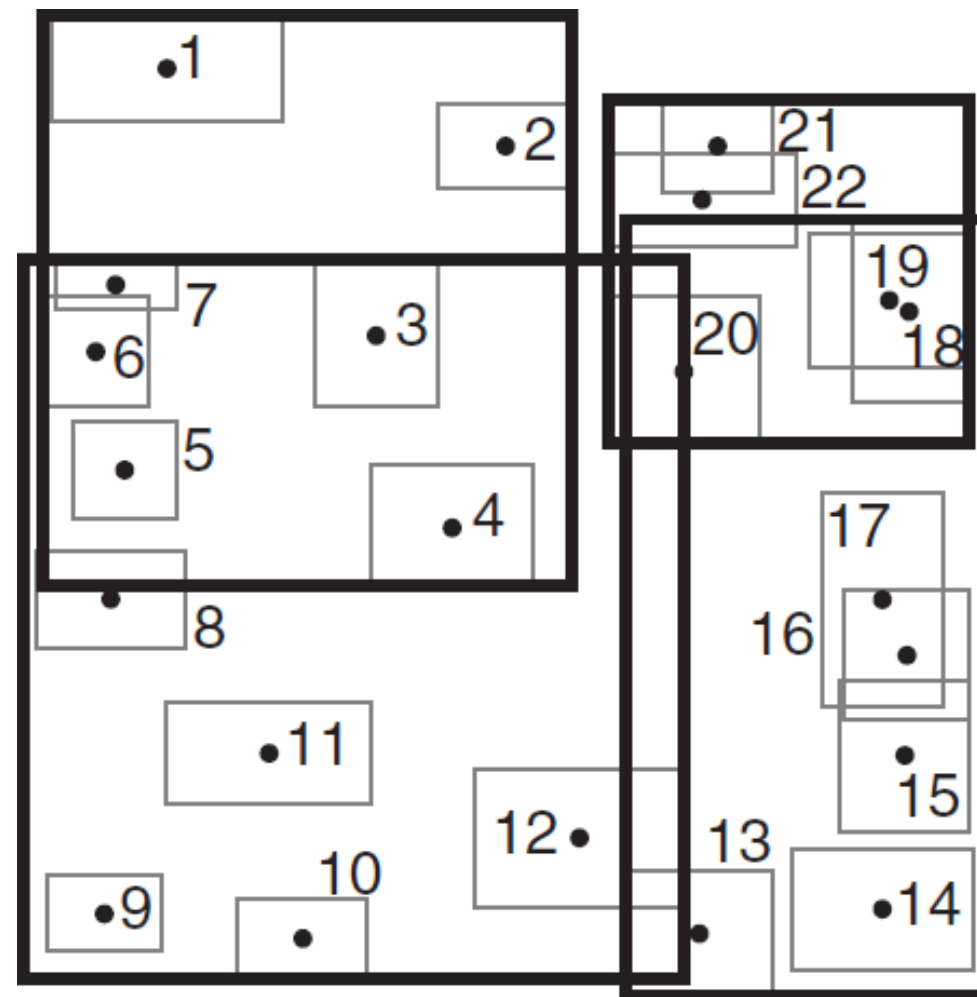


Hilbert *R-Tree*



Que hacer si el nodo se sobrecarga?

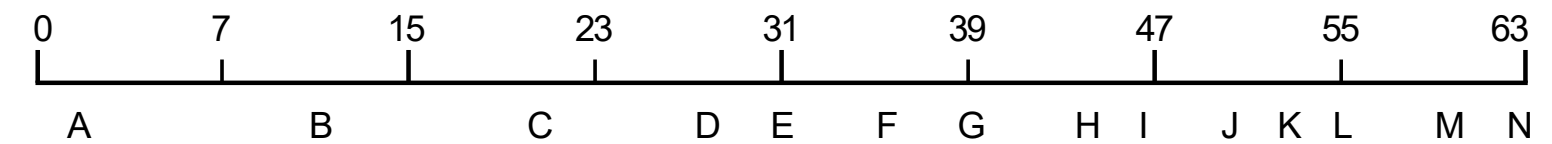
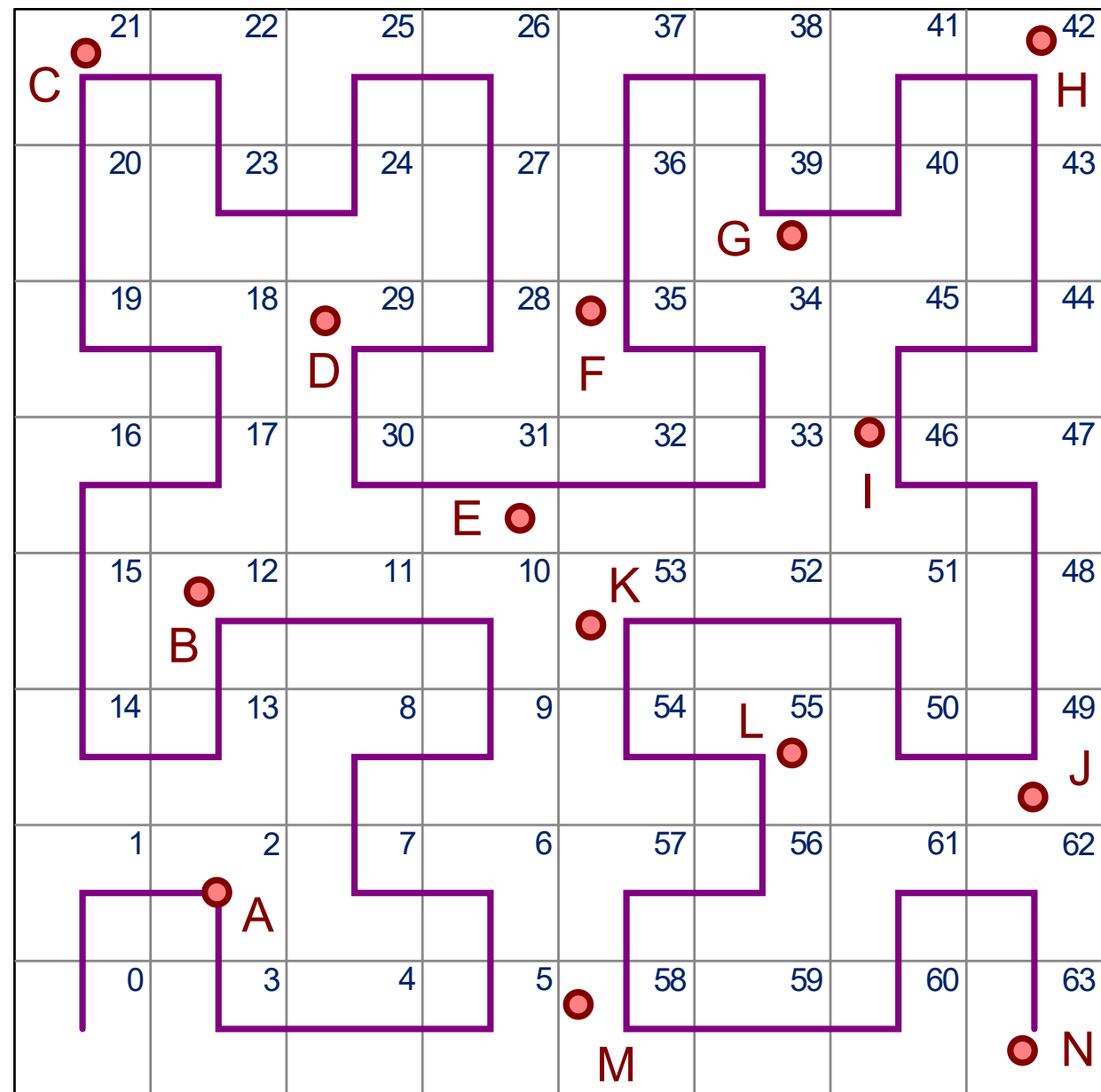
Hilbert *R-Tree*



Que hacer si el nodo se sobrecarga?

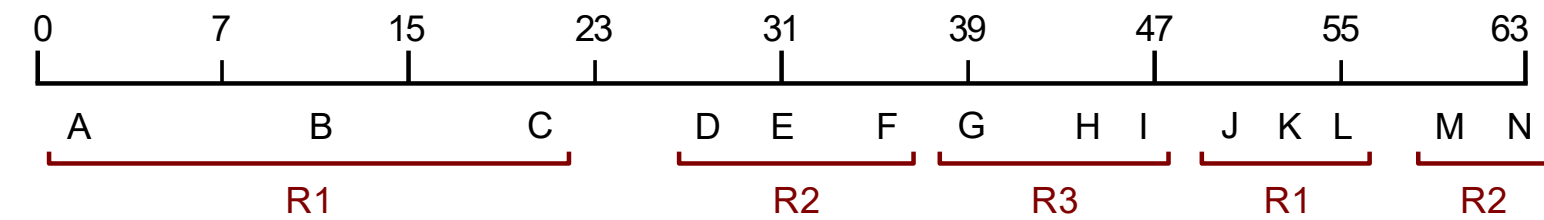
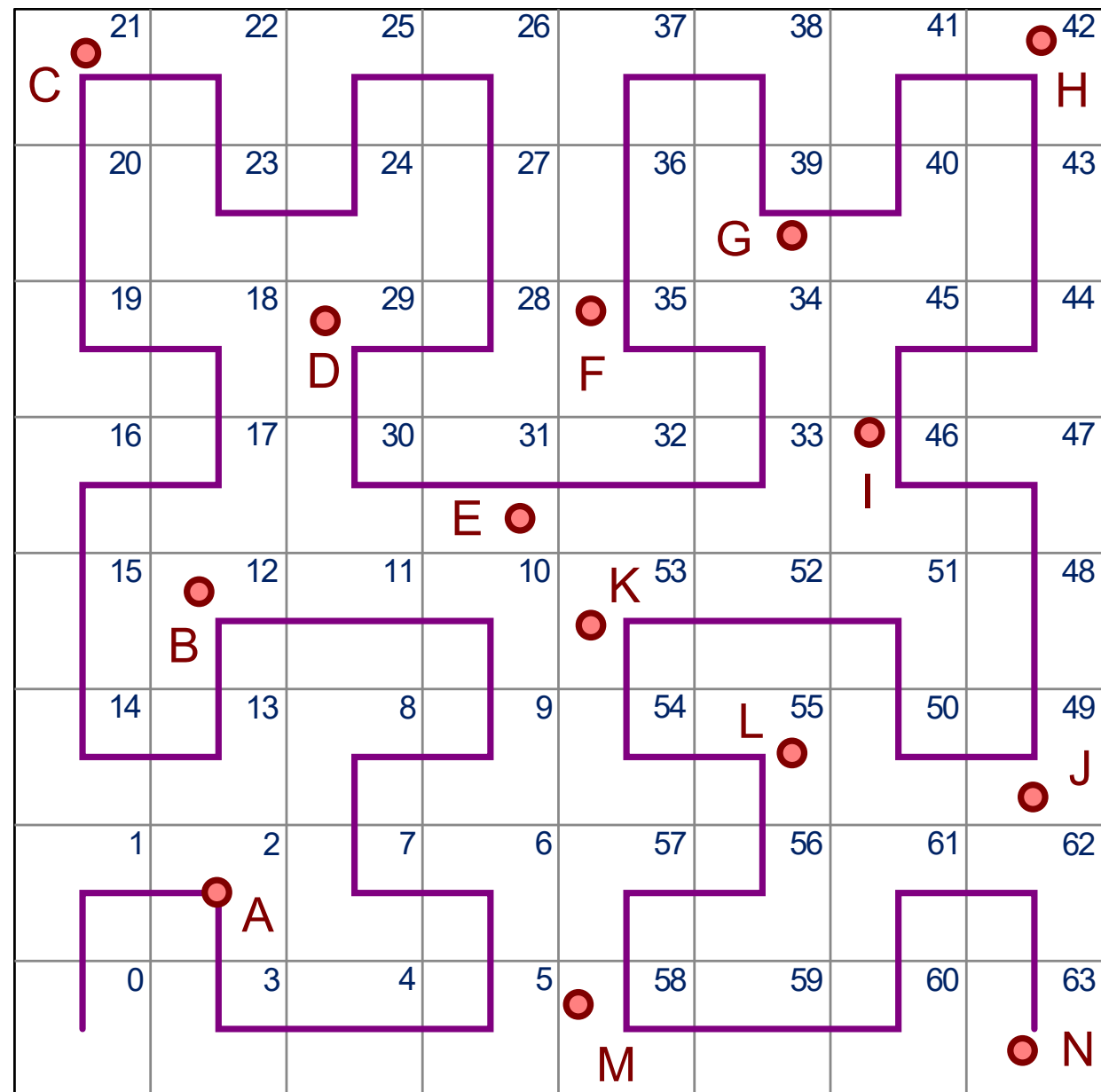
Creamos un nodo padre

Hilbert *R-Tree*



Agrupamos los puntos en grupos de 3 (la capacidad del nodo) en orden del H-index.

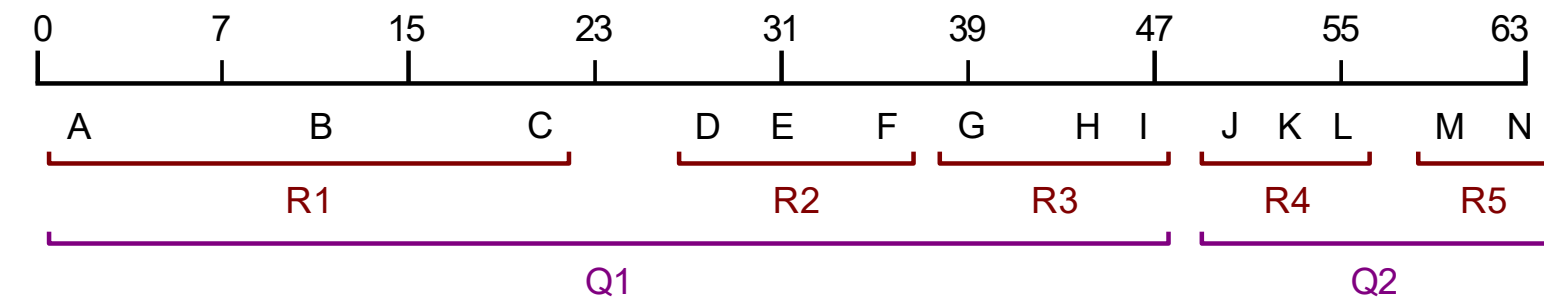
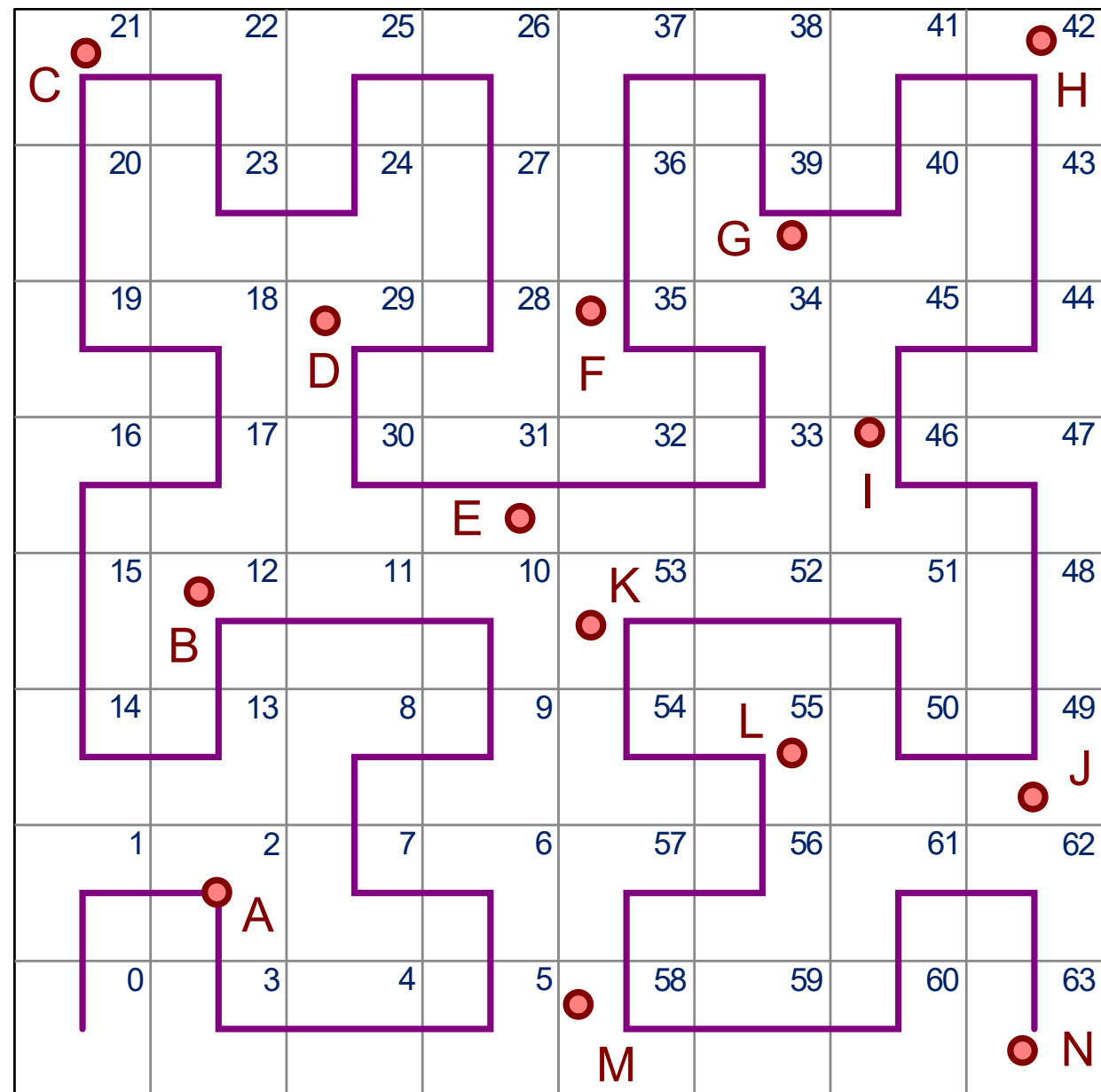
Hilbert *R-Tree*



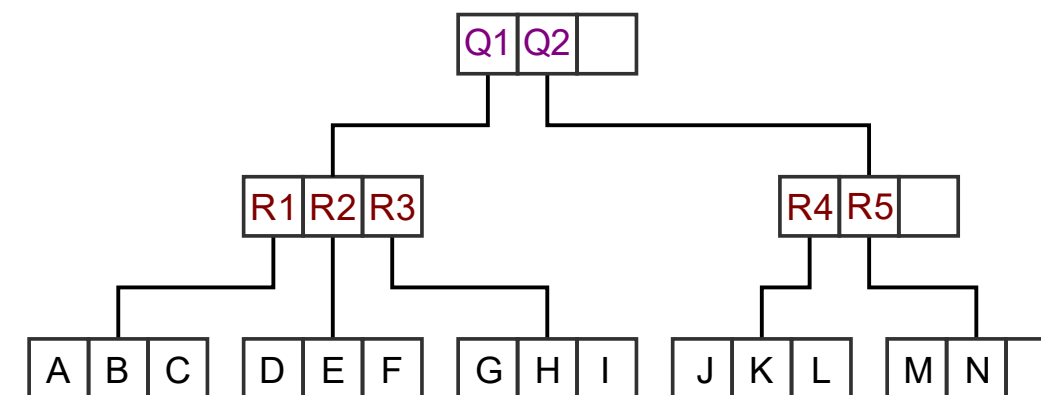
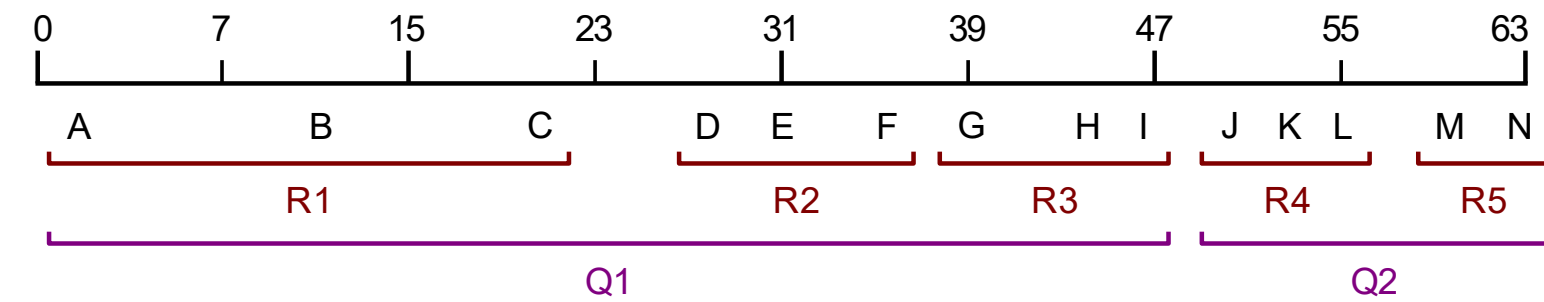
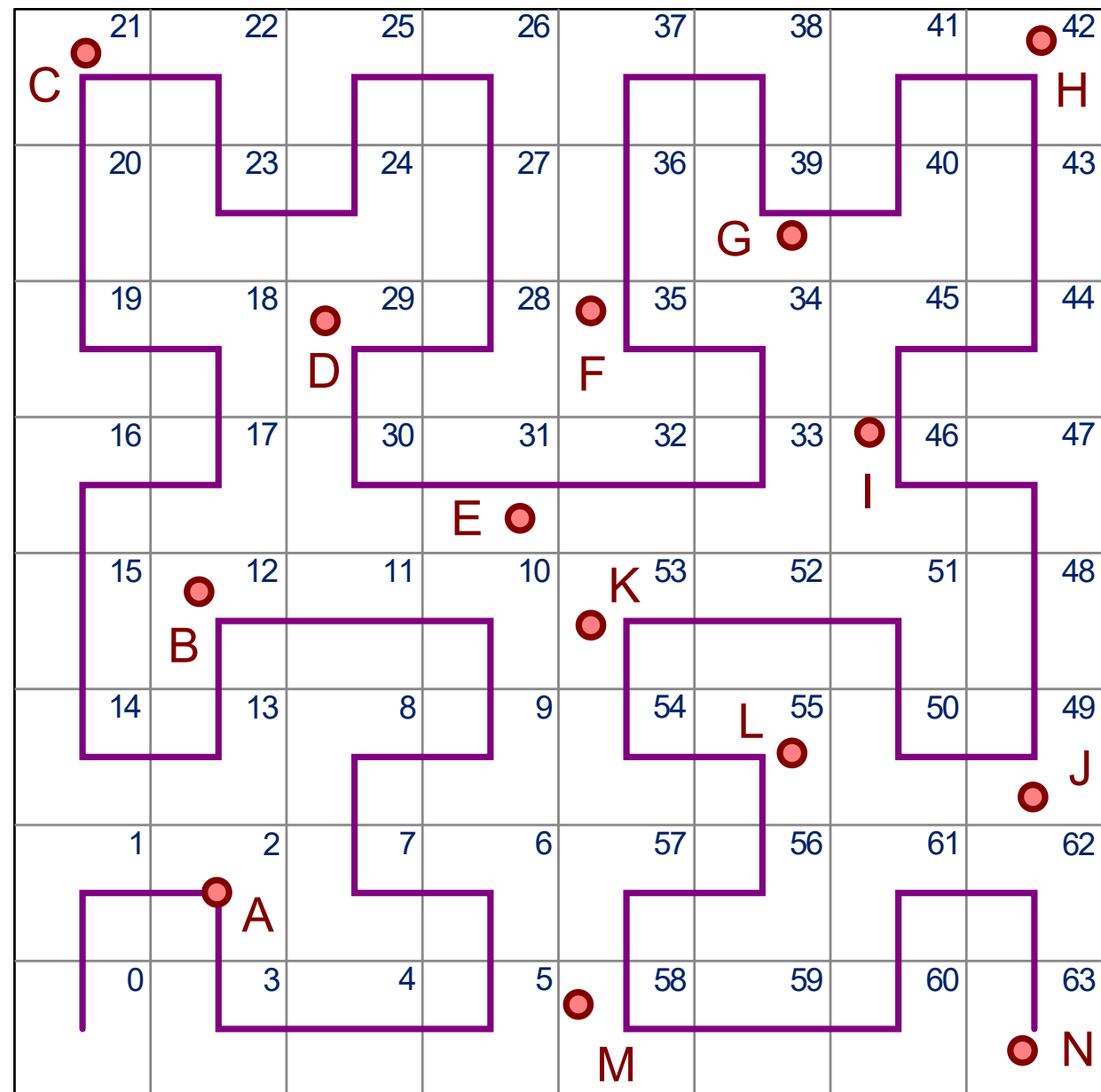
La cantidad de grupos es mayor a 3 (la capacidad del nodo padre).

Entonces, creamos otro nivel del árbol

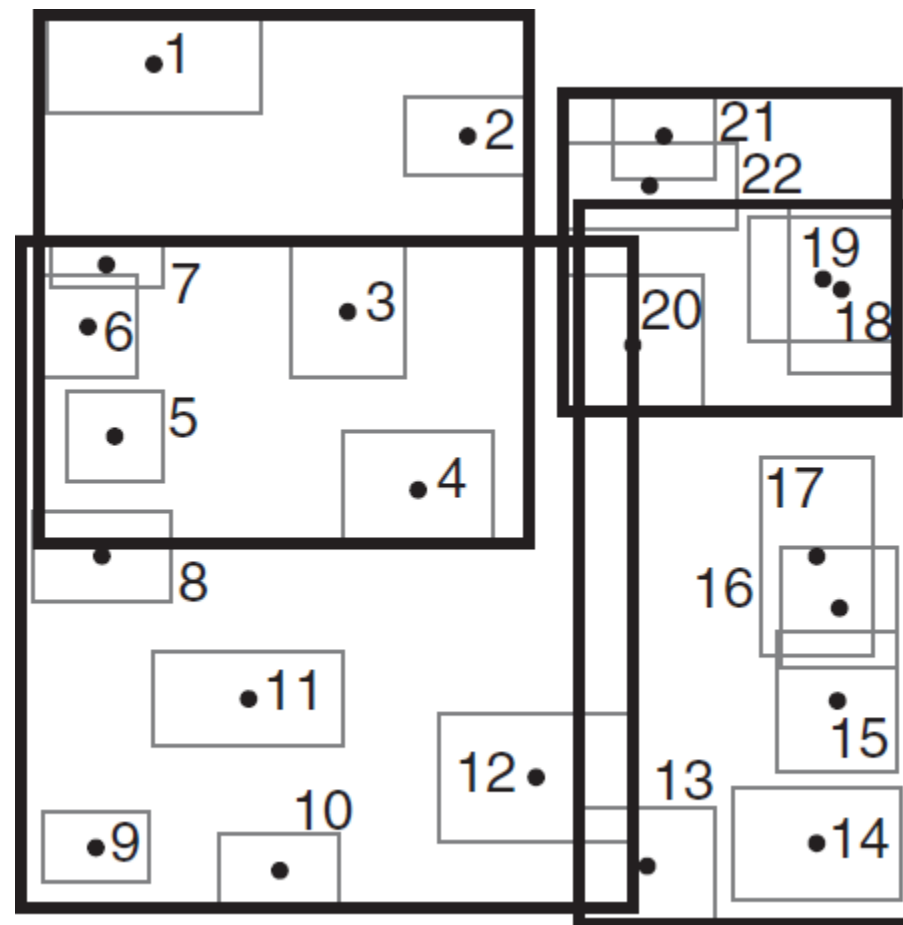
Hilbert *R-Tree*



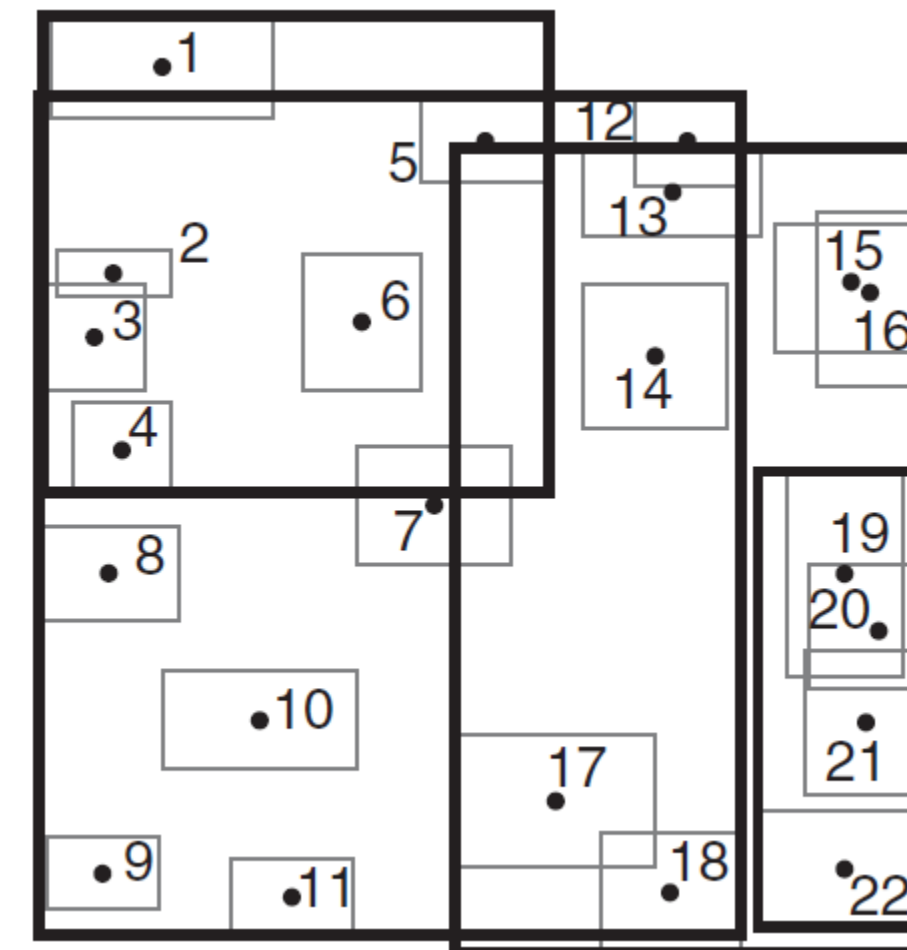
Hilbert *R-Tree*



Hilbert *R-Tree*

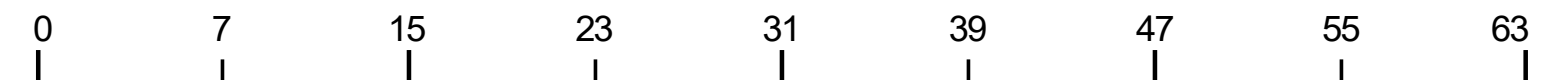
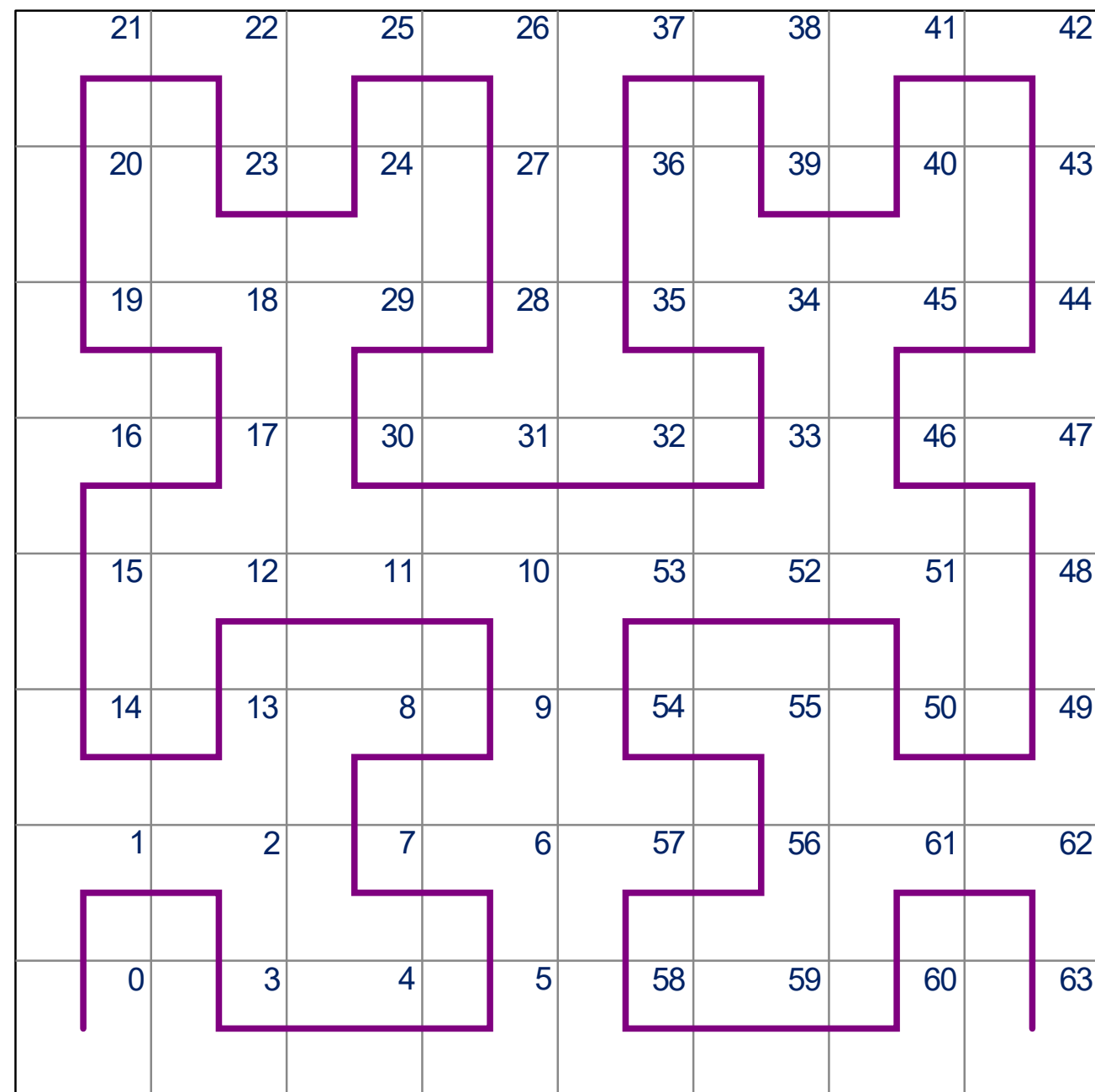


Hilbert packed R-tree



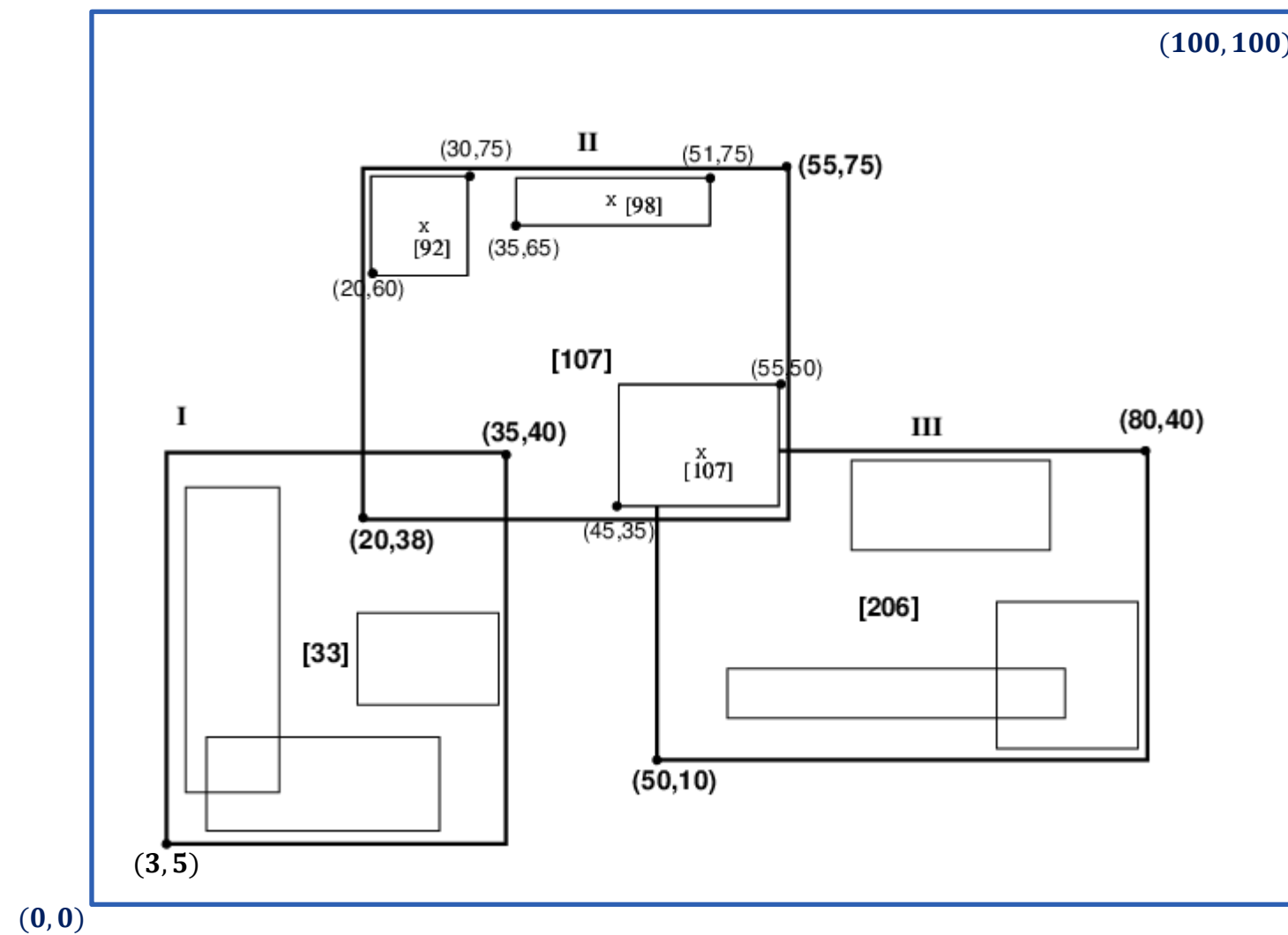
Morton packed R-tree

Hilbert *R-Tree*

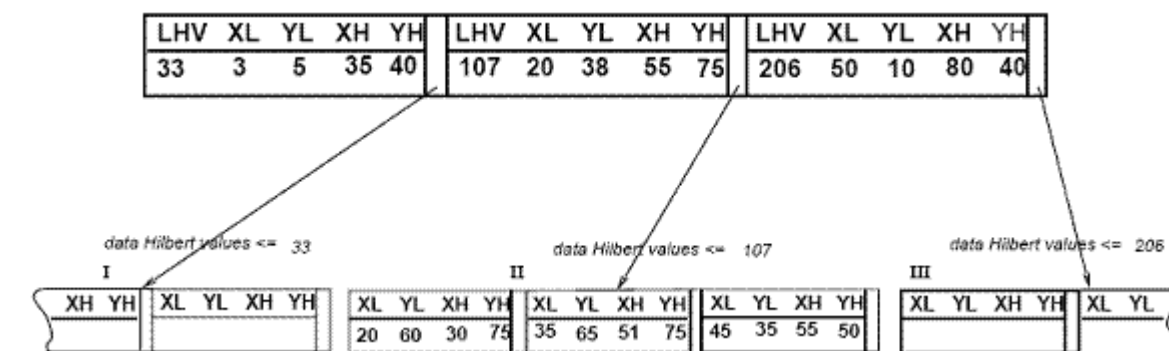


2. Dynamic Hilbert R-Tree

Dynamic Hilbert *R-Tree*



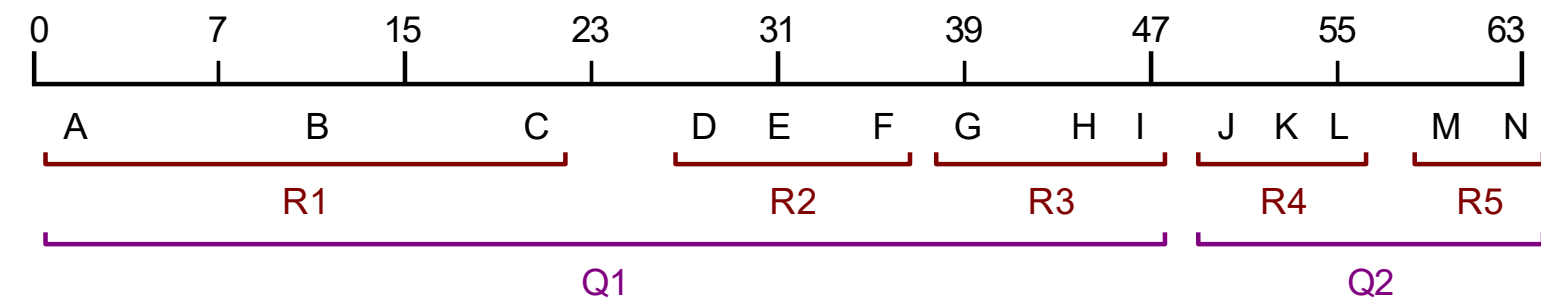
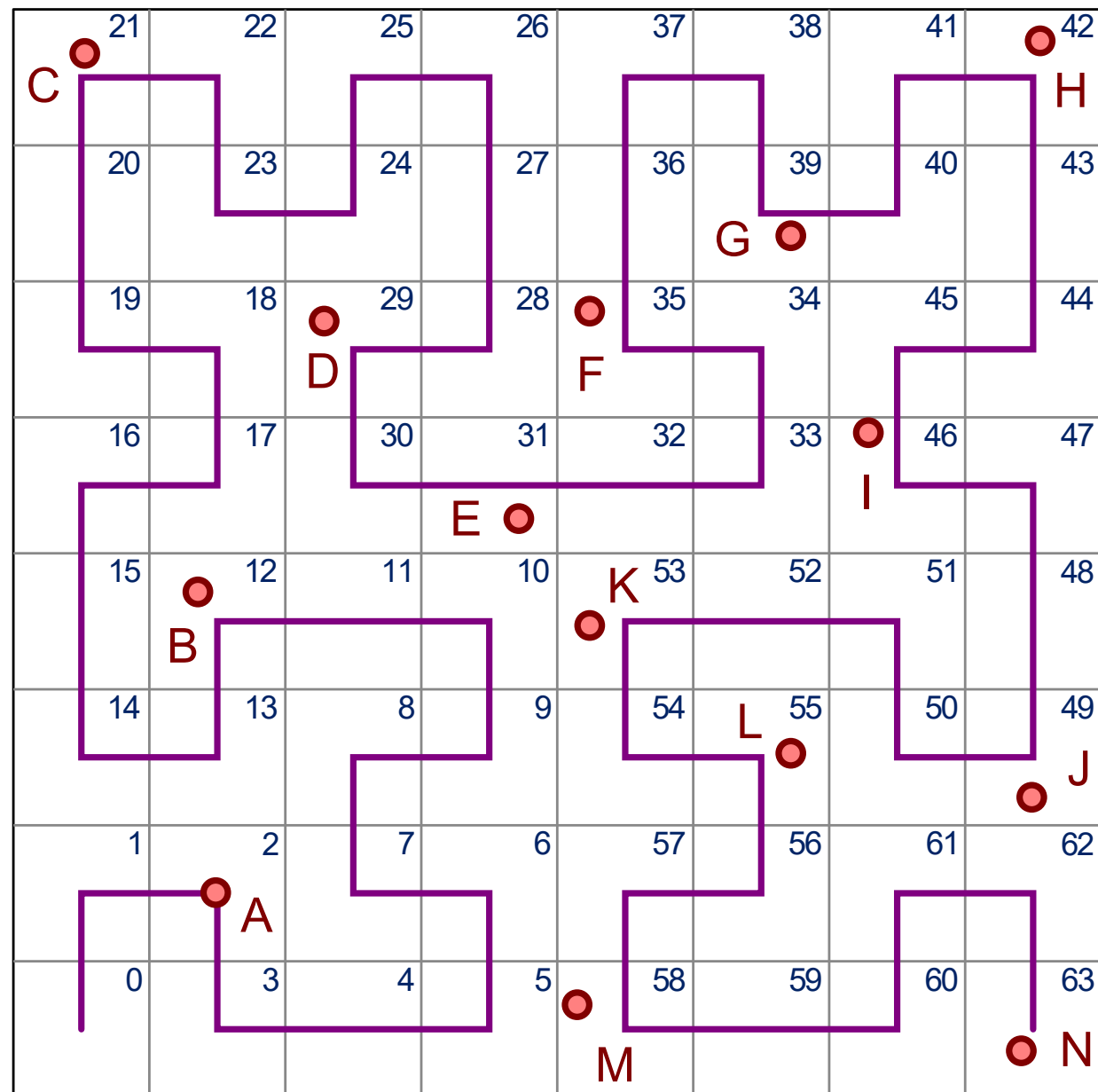
LHV: Largest Hilbert value



Los nodos internos almacenan **LHV**

Esto nos ayudará a ordenar los nodos, y seleccionar una hoja durante la inserción.

Dynamic Hilbert *R-Tree*



Por ejemplo:

LHV de **R1** es 21, ya que es el h-index de C

LHV de **R2** es 35, ya que es el h-index de F

LHV de **R3** es 46, ya que es el h-index de I

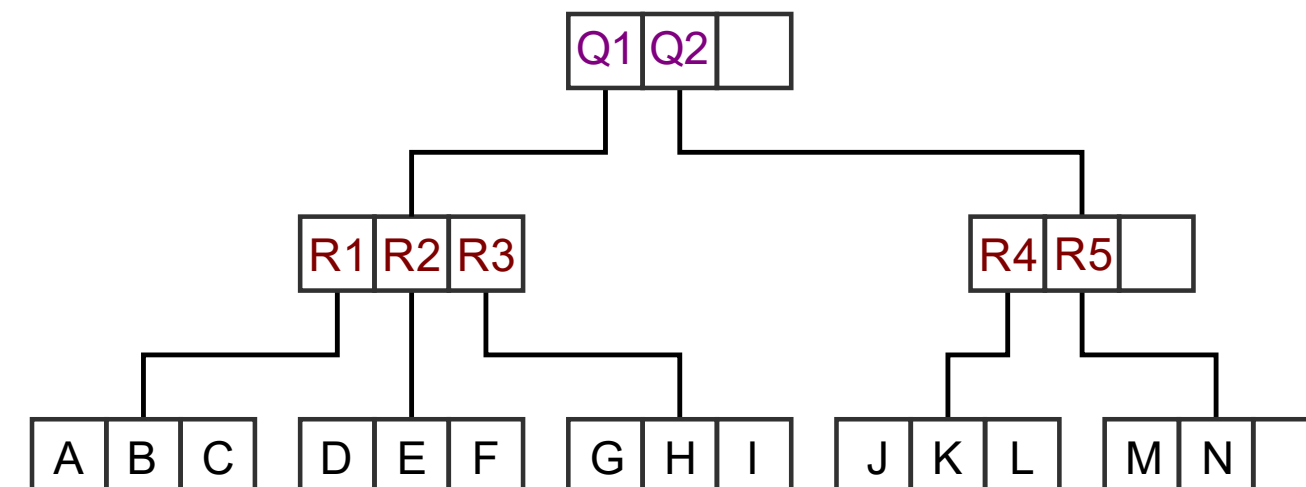
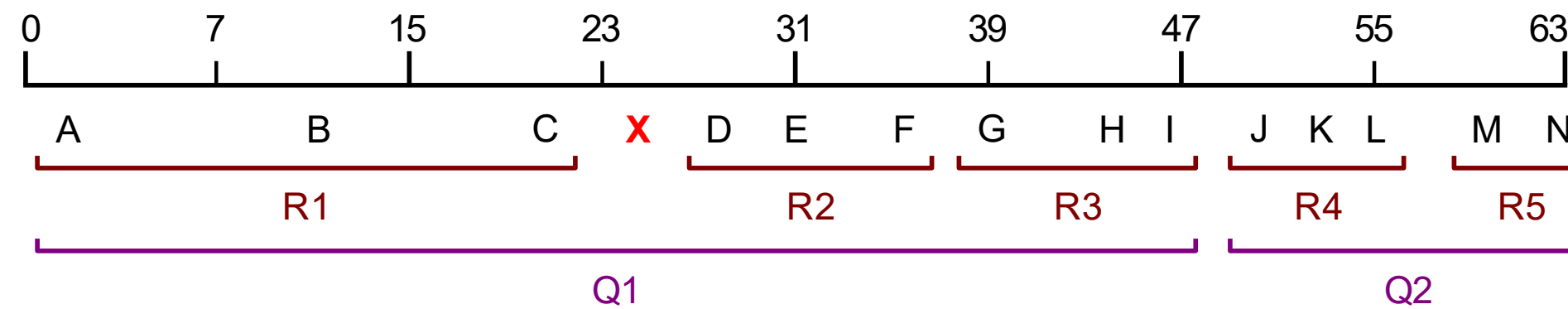
LHV de **Q1** es 46, ya que es el LHV de **R3**

Dynamic Hilbert *R-Tree*

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

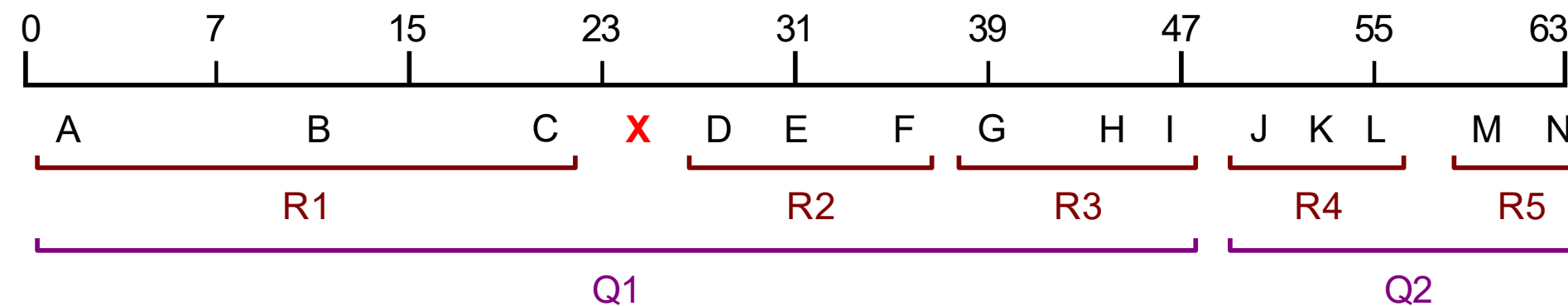
Dynamic Hilbert *R-Tree*

Por ejemplo: Agregamos el punto **x**, el cual tiene h-index 25.



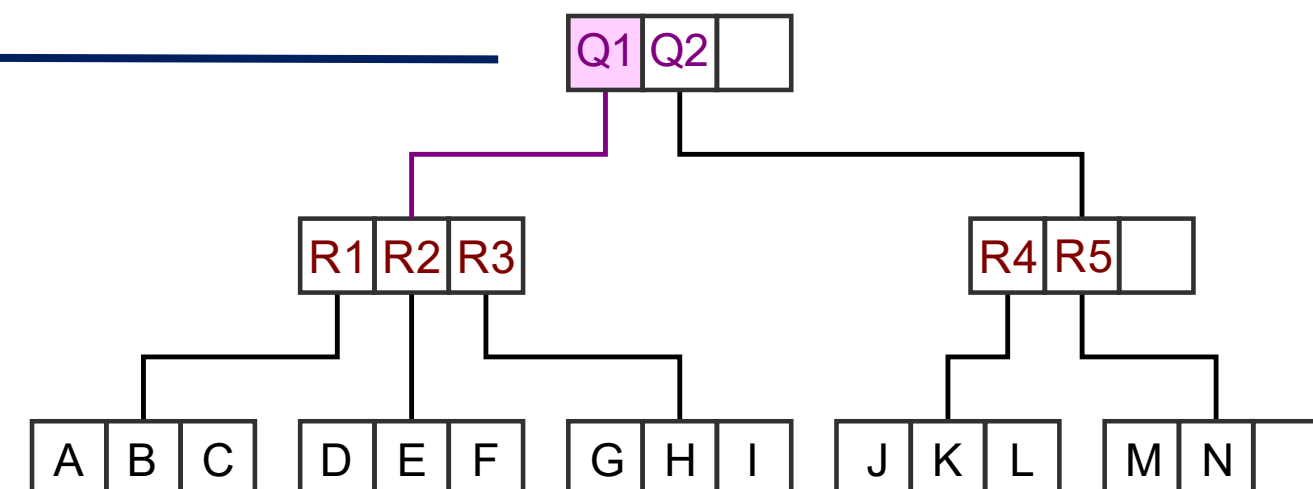
Dynamic Hilbert *R-Tree*

Por ejemplo: Agregamos el punto **x**, el cual tiene h-index 25.



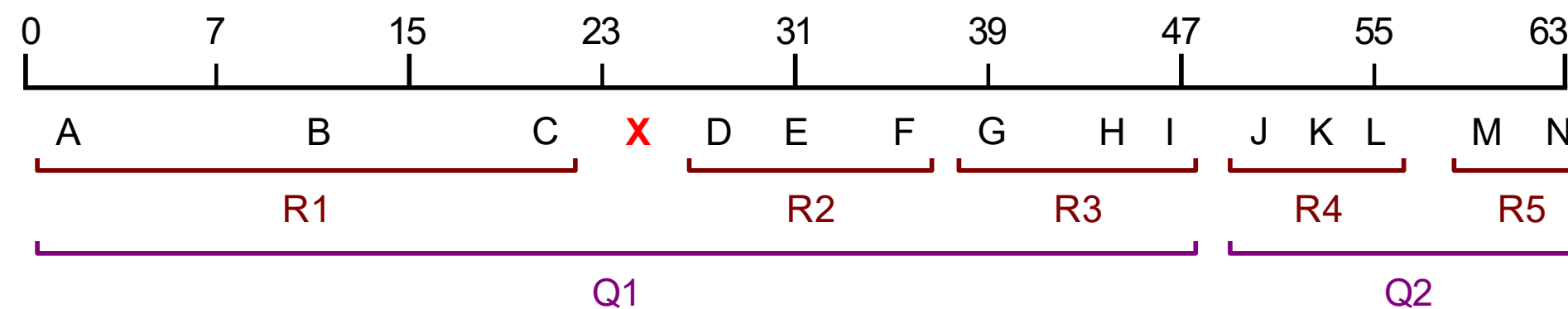
$LHV(Q1) > h\text{-index}(x)$
 $LHV(Q2) > h\text{-index}(x)$
 $LHV(Q1) < LHV(Q2)$

Elegimos **Q1** (es el menor)



Dynamic Hilbert *R-Tree*

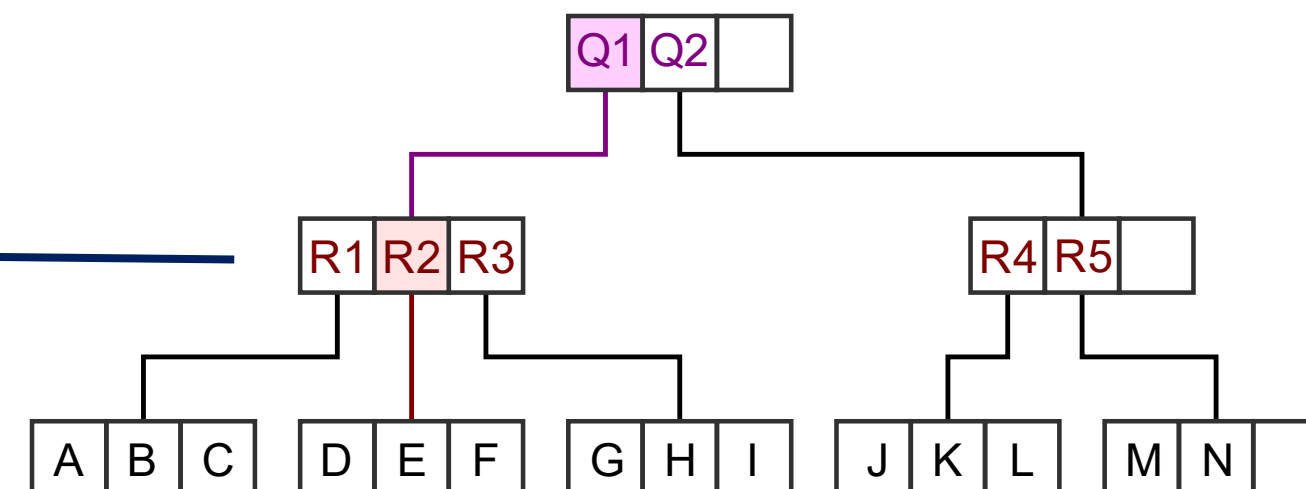
Por ejemplo: Agregamos el punto **x**, el cual tiene h-index 25.



$LHV(R1) < h\text{-index}(x)$
 $LHV(R2) > h\text{-index}(x)$
 $LHV(R3) > h\text{-index}(x)$

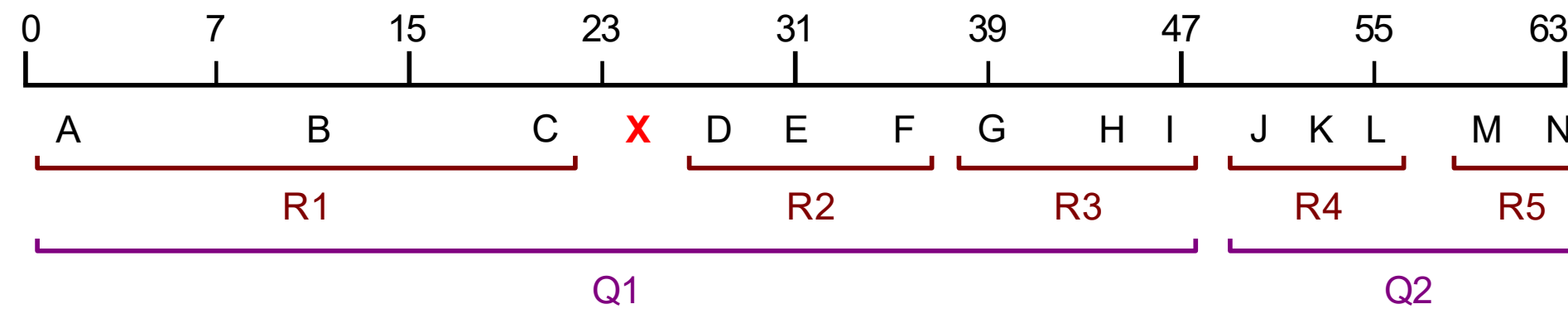
$LHV(R1) < LHV(R2)$
 $LHV(R1) < LHV(R2)$

Elegimos **R1** (es el menor que supera)



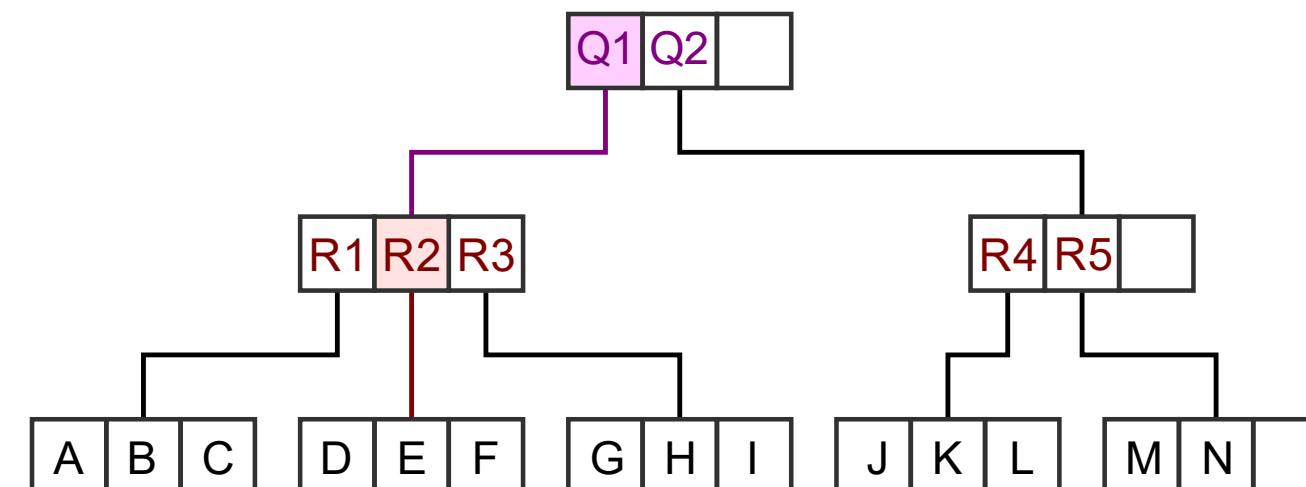
Dynamic Hilbert *R-Tree*

Por ejemplo: Agregamos el punto **x**, el cual tiene h-index 25.



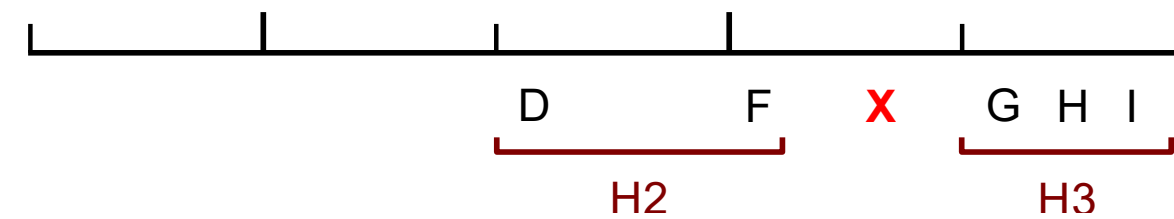
Note que el nodo R2 se sobrecarga

:C



Dynamic Hilbert *R-Tree*

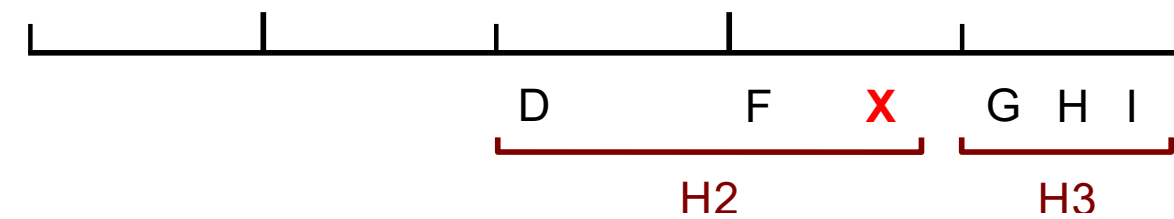
Sobrecarga Buscamos apoyo del hermano **izquierdo**.



Repartimos todos los datos entre los nodos

Dynamic Hilbert *R-Tree*

Sobrecarga Buscamos apoyo del hermano **izquierdo**.

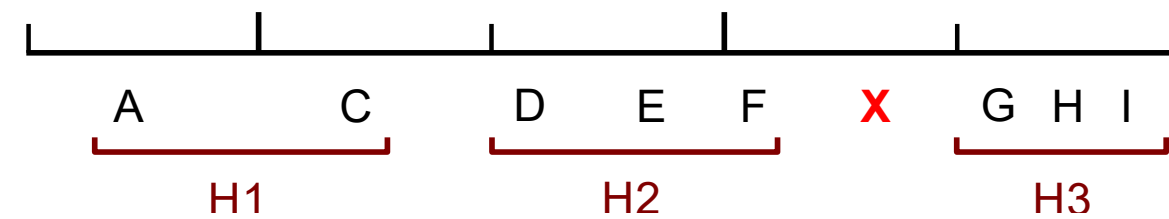


Repartimos todos los datos entre los nodos

Pero, que hacemos si el nodo izquierdo está lleno?

Dynamic Hilbert *R-Tree*

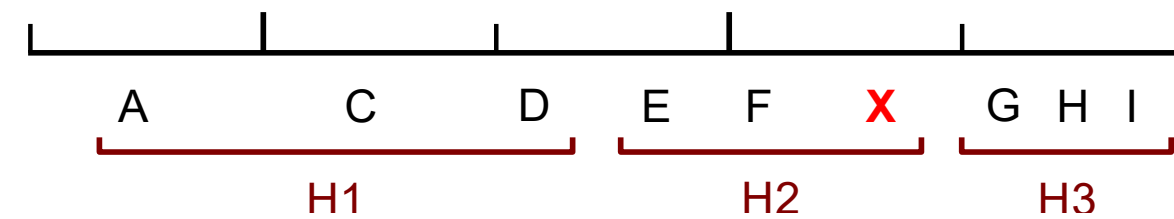
Sobrecarga Buscamos apoyo del hermano **izquierdo**.



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

Dynamic Hilbert *R-Tree*

Sobrecarga Buscamos apoyo del hermano **izquierdo**.

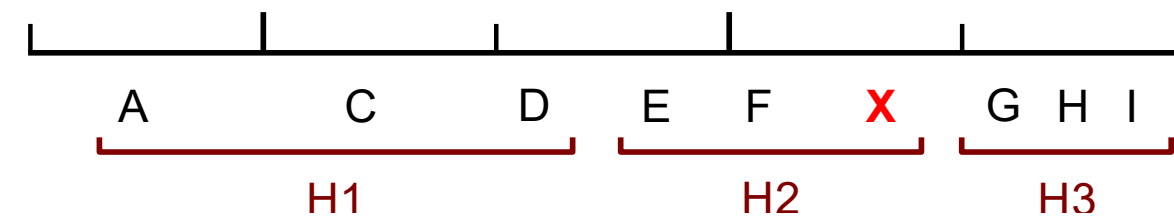


Repartimos todos los datos entre **todos** los nodos

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

Dynamic Hilbert *R-Tree*

Sobrecarga Buscamos apoyo del hermano **izquierdo**.



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

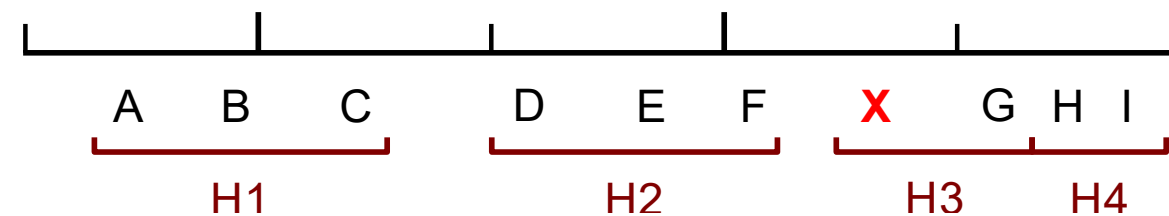
Pero, que hacemos si **todos** los nodos izquierdos están llenos?

Creamos nuevo nodo

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

Dynamic Hilbert *R-Tree*

Sobrecarga Buscamos apoyo del hermano **izquierdo**.



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

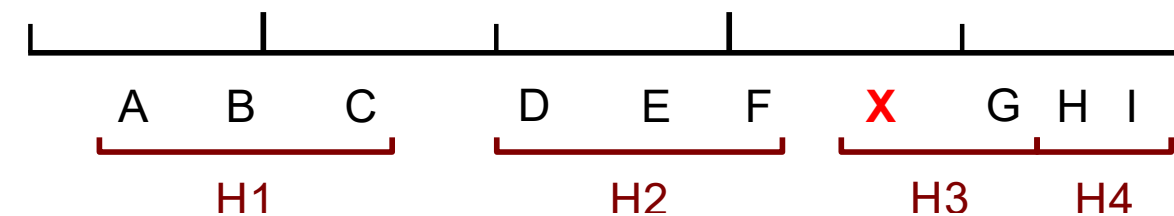
Pero, que hacemos si **todos** los nodos izquierdos están llenos?

Creamos nuevo nodo

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

Dynamic Hilbert *R-Tree*

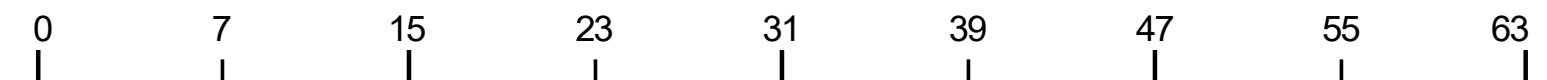
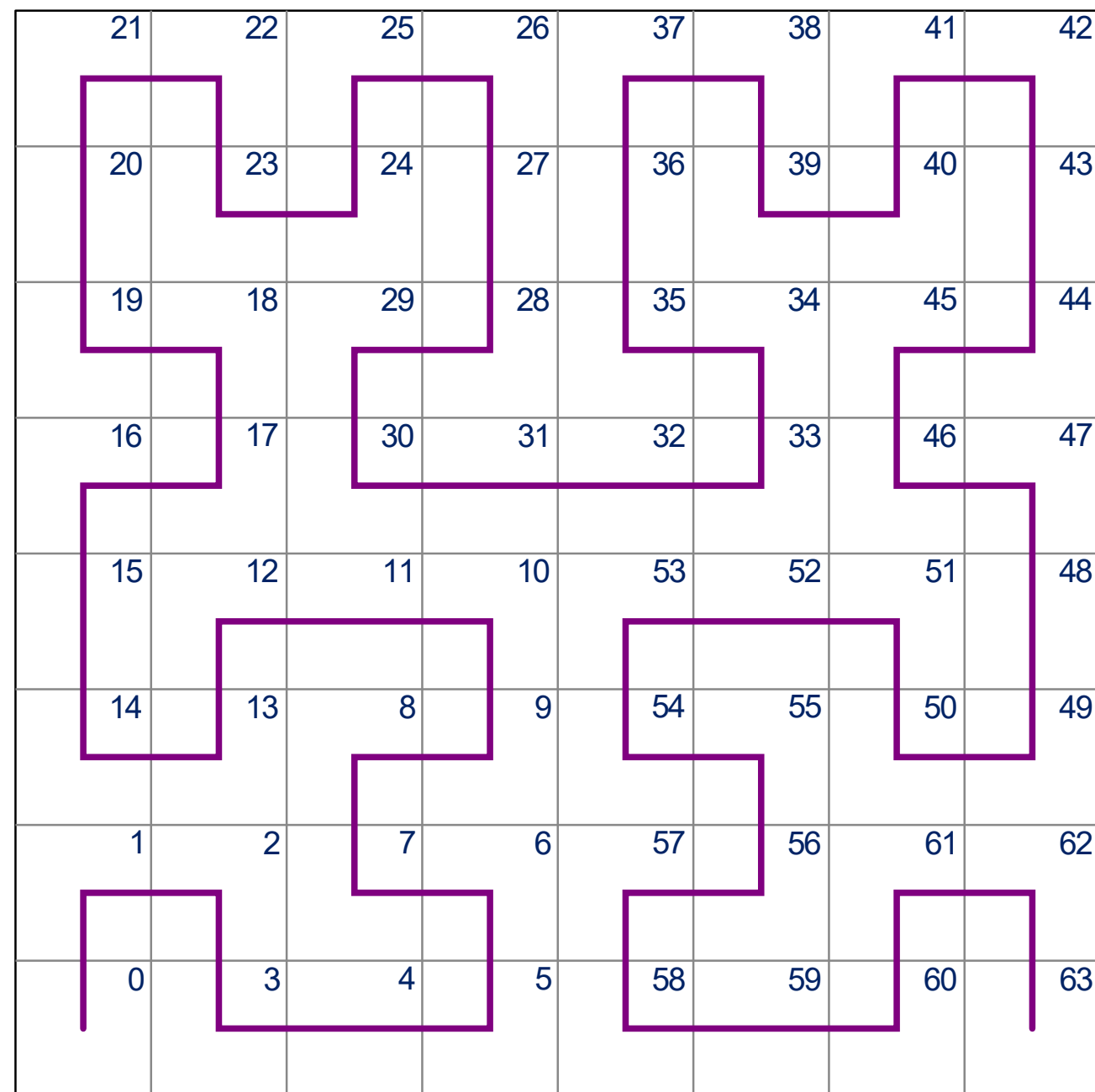
Sobrecarga Buscamos apoyo del hermano **izquierdo**.



Note que el nodo padre se **sobrecarga**

Entonces, repetimos este proceso en los nodos internos, donde tomamos como referencia el LHV.

Dynamic Hilbert *R-Tree*





INGENIERIA
MECATRÓNICA

BIÓINGENIERÍA

INGENIERIA
CIENCIA DE
LA COMPUTACIÓN

INGENIERIA
AMBIENTAL

INGENIERIA
ENERGÍA

INGENIERIA
INDUSTRIAL

INGENIERIA
ELECTRÓNICA



UTECH
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

