UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Repaso Parcial 1

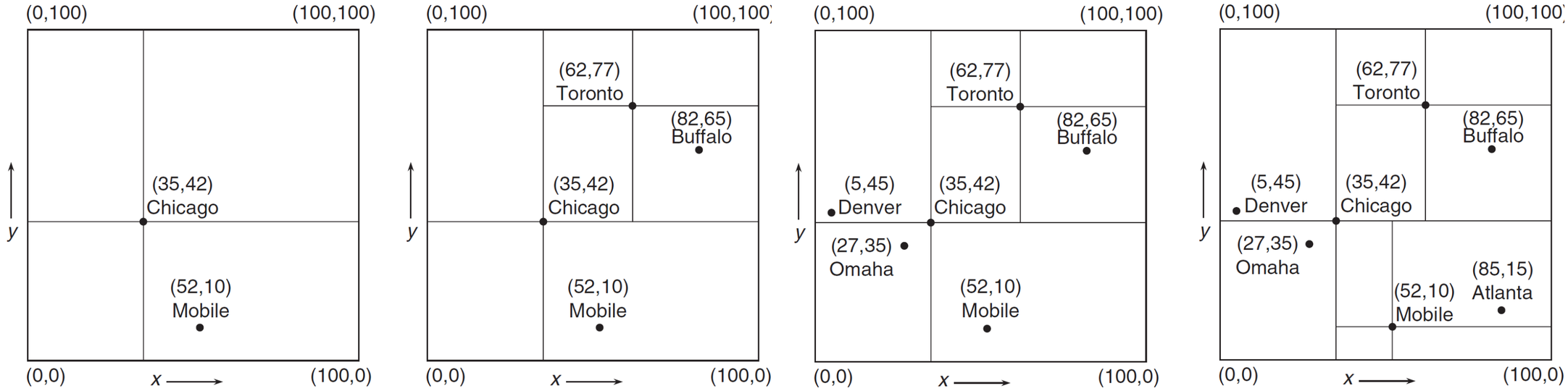
CS3102 EDA

Índice

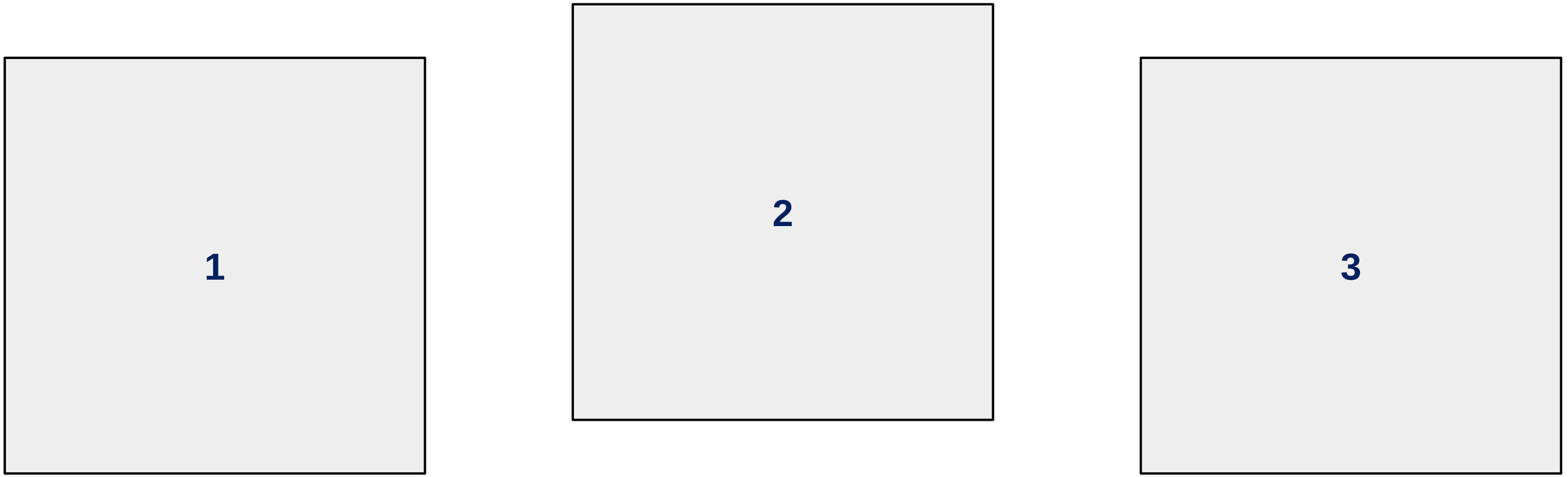
1. Point QuadTree
2. PR QuadTree
3. Region QuadTree
4. Point k-d-Tree
5. BSP-tree
6. hB-Tree
7. R-Tree

1. Point QuadTree

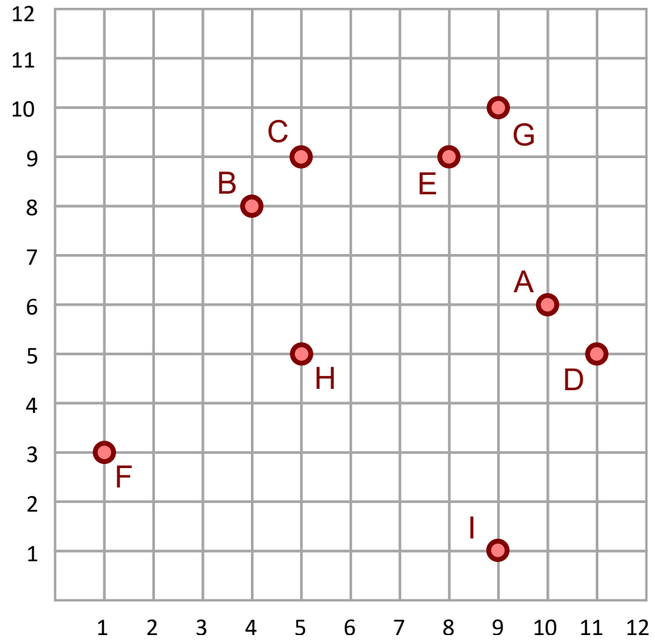
Point *QuadTree*

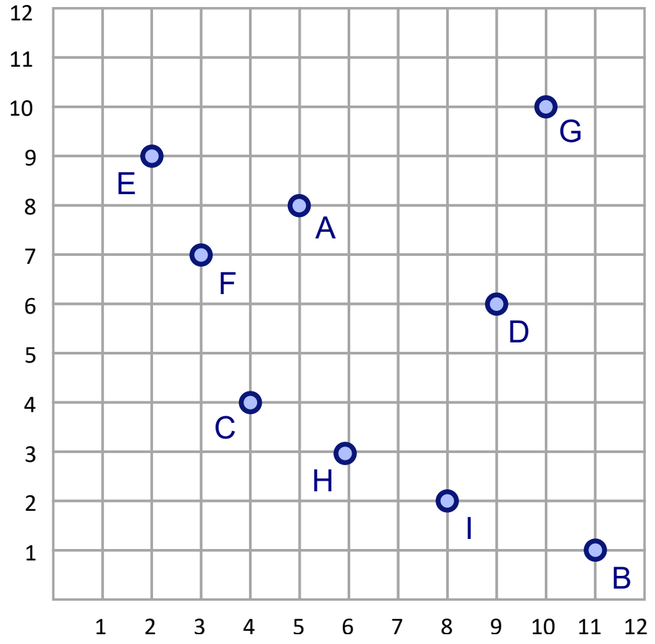


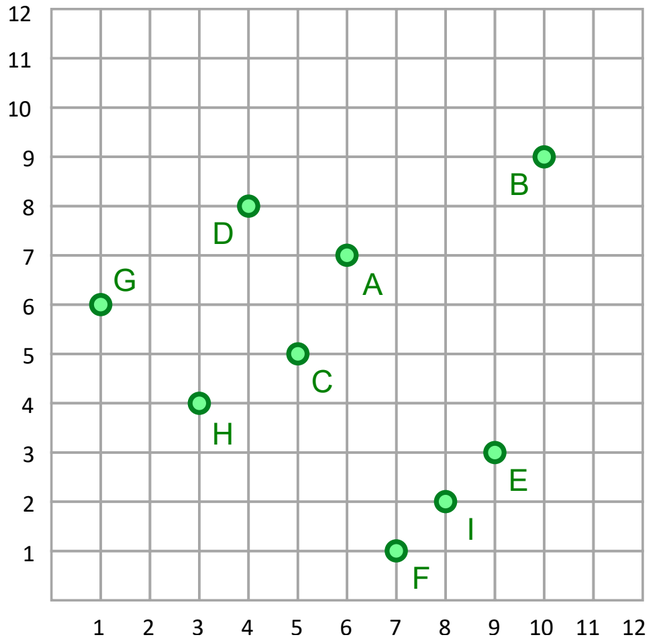
Point *QuadTree*



Point *QuadTree*

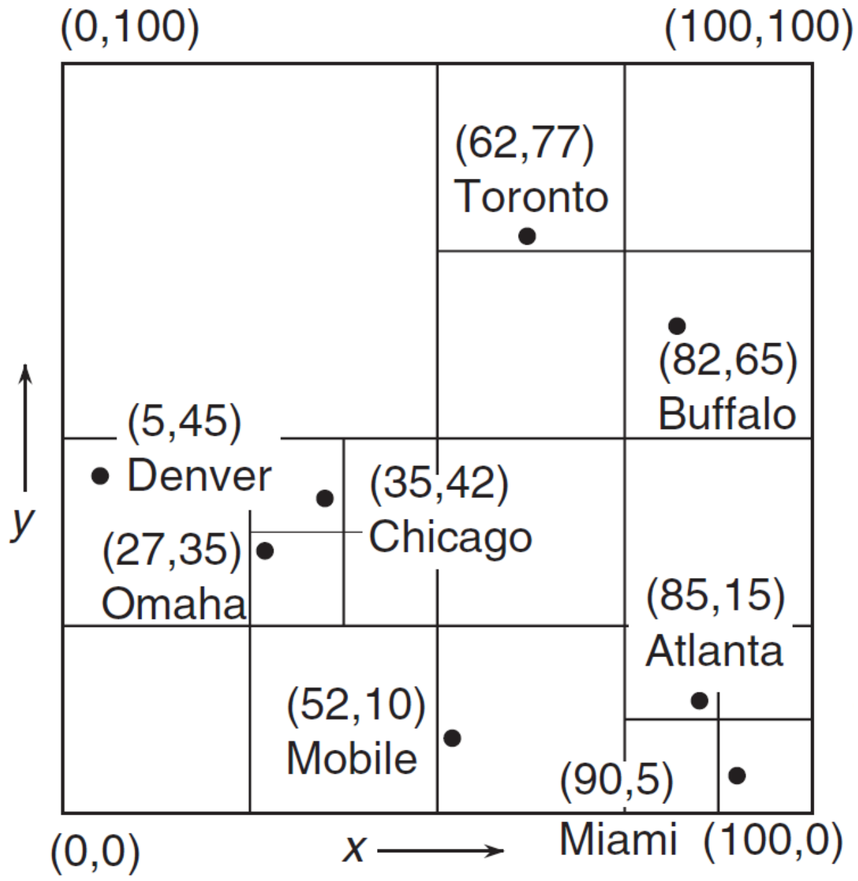


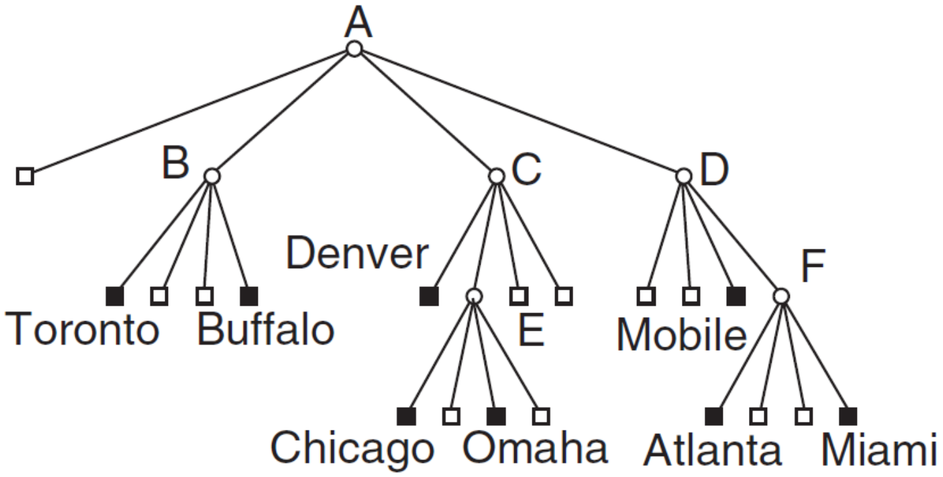




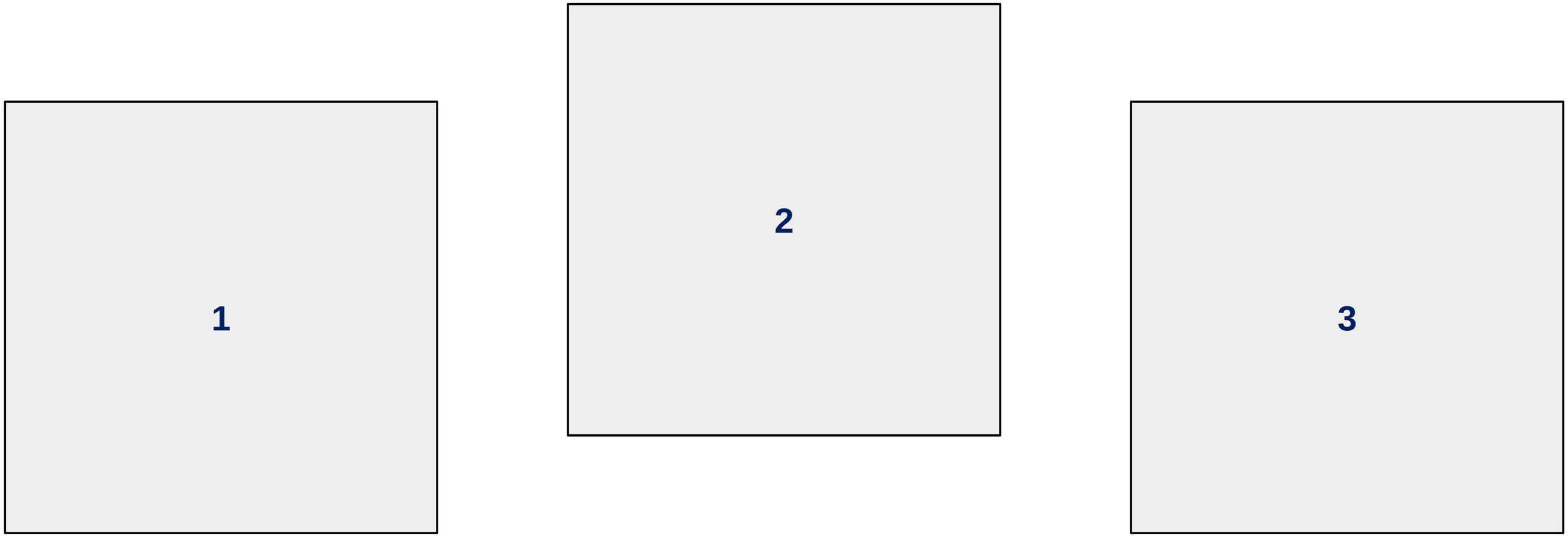
2. PR QuadTree

PR *QuadTree*

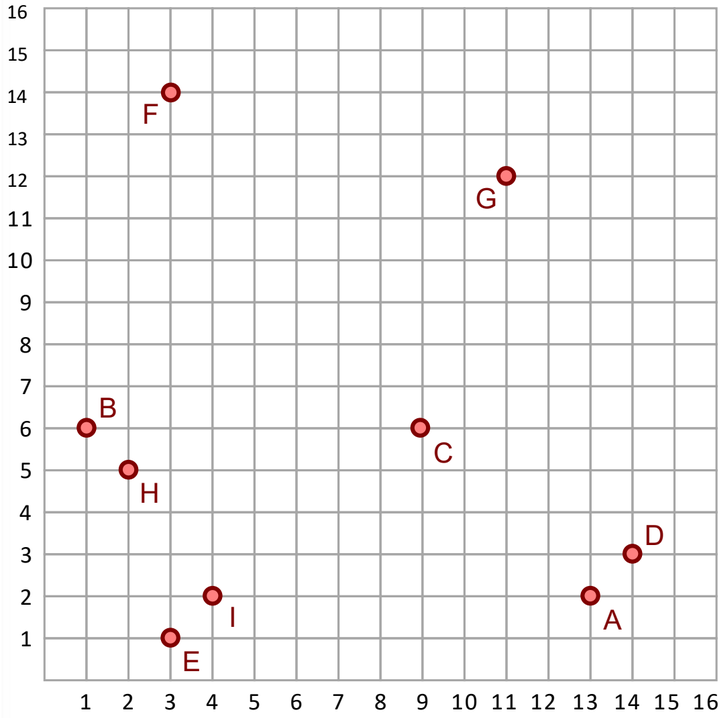

(0,100)
(100,100)
(62,77)
Toronto
(82,65)
Buffalo
(5,45)
Denver
(35,42)
Chicago
(27,35)
Omaha
(85,15)
Atlanta
(52,10)
Mobile
(90,5)
Miami
(0,0)
(100,0)
y
x

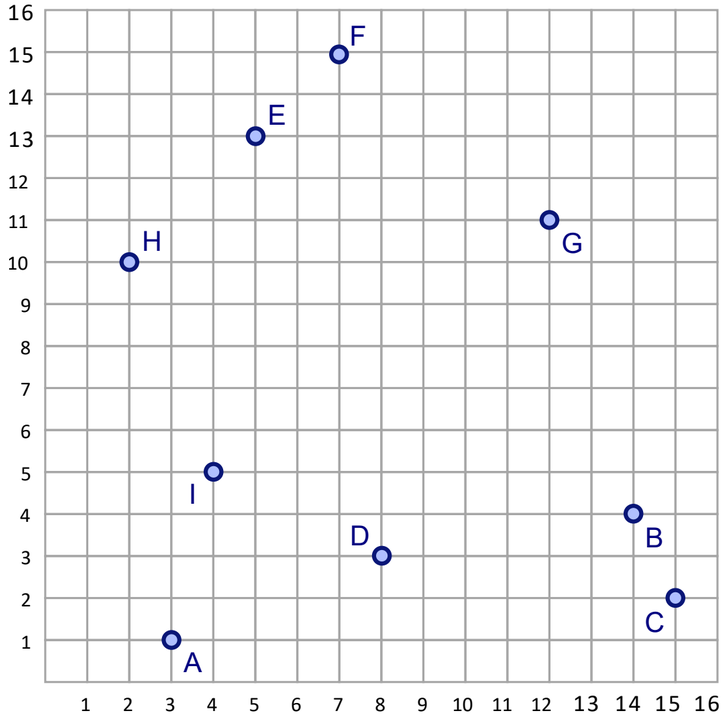

A
B
C
D
Toronto
Buffalo
Denver
E
F
Mobile
Chicago
Omaha
Atlanta
Miami

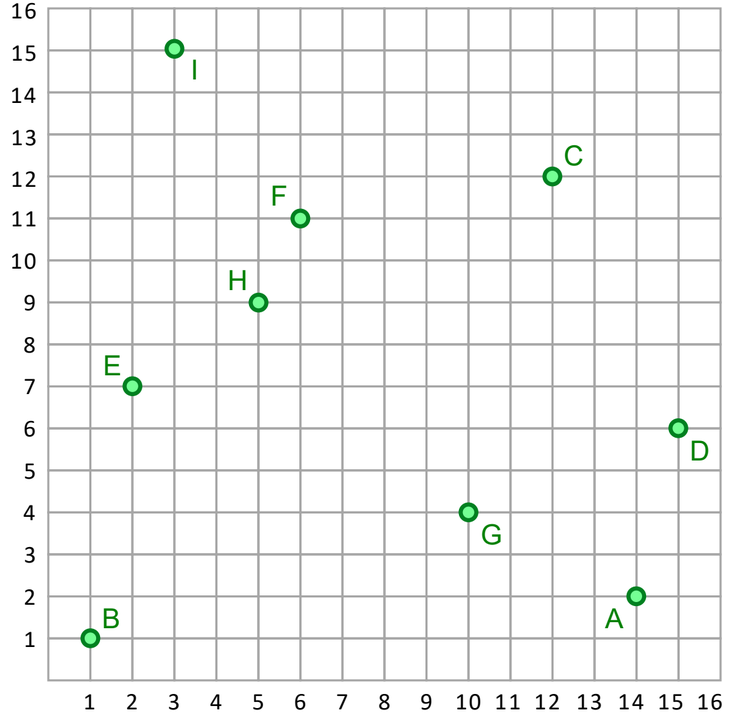
PR *QuadTree*



PR *QuadTree*

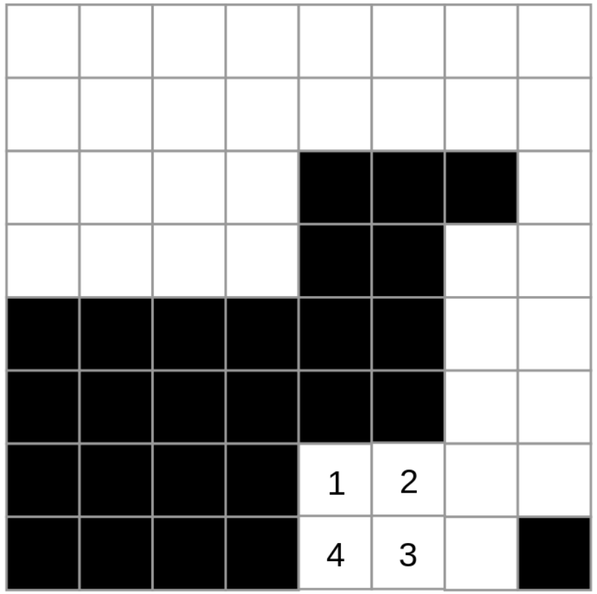


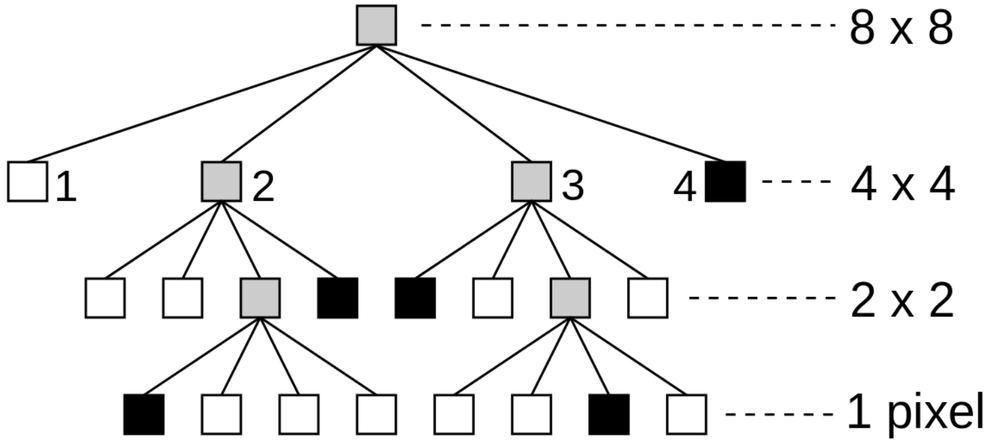




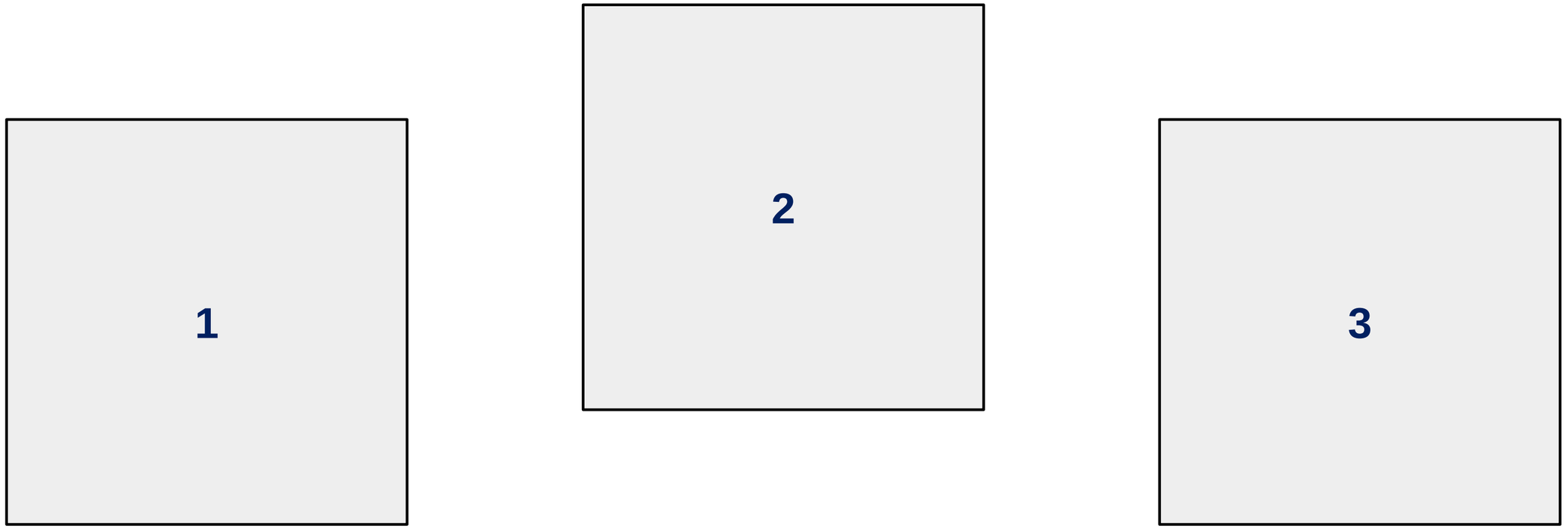
3. Region QuadTree

Region *QuadTree*

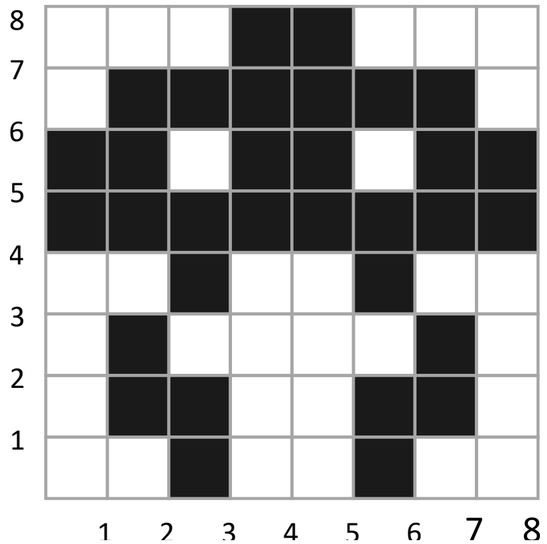


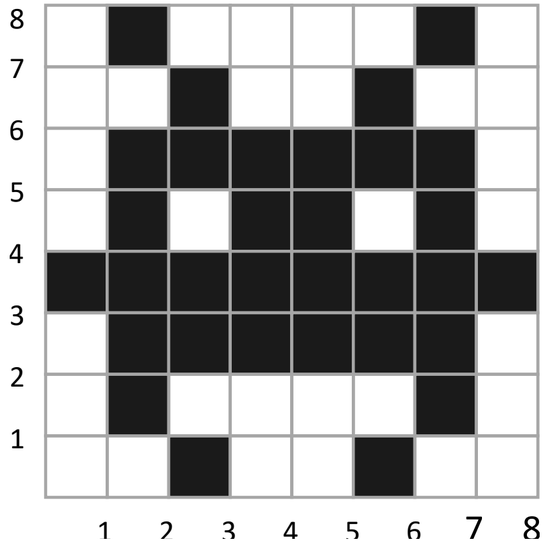


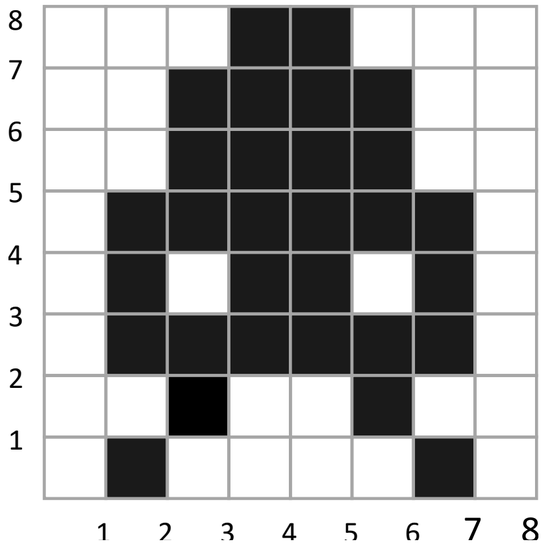
Region *QuadTree*



Region *QuadTree*

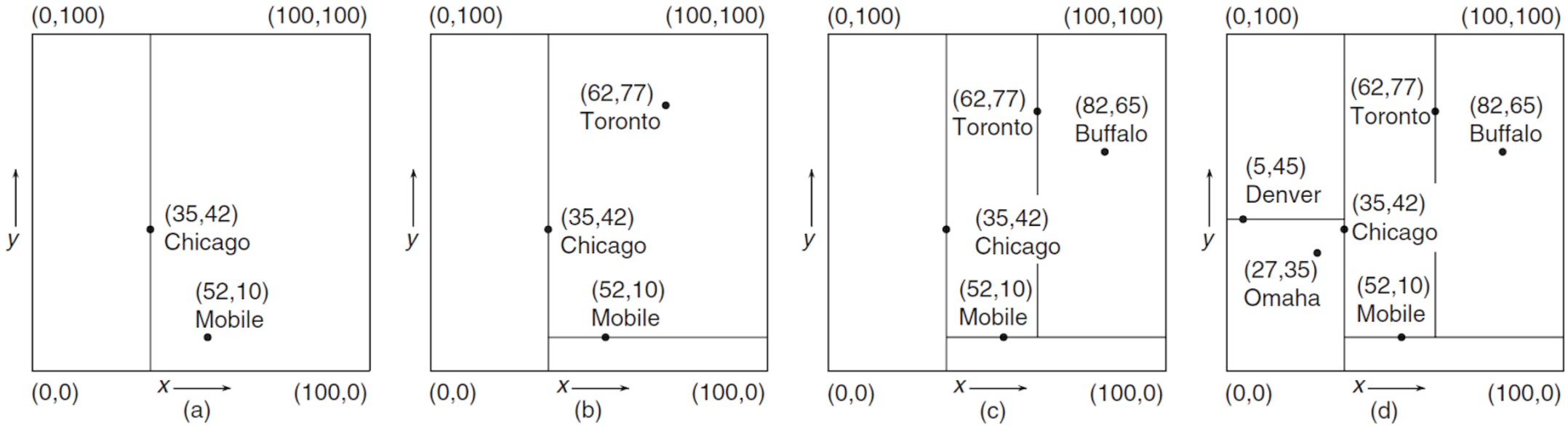




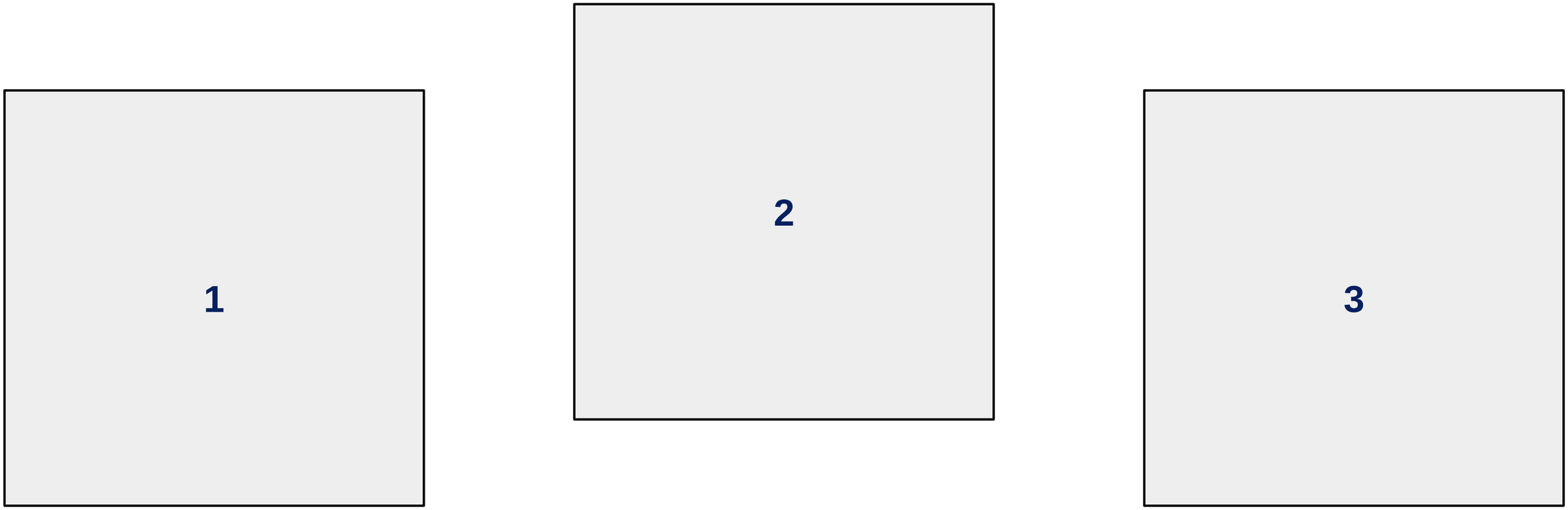


4. Point k-d-Tree

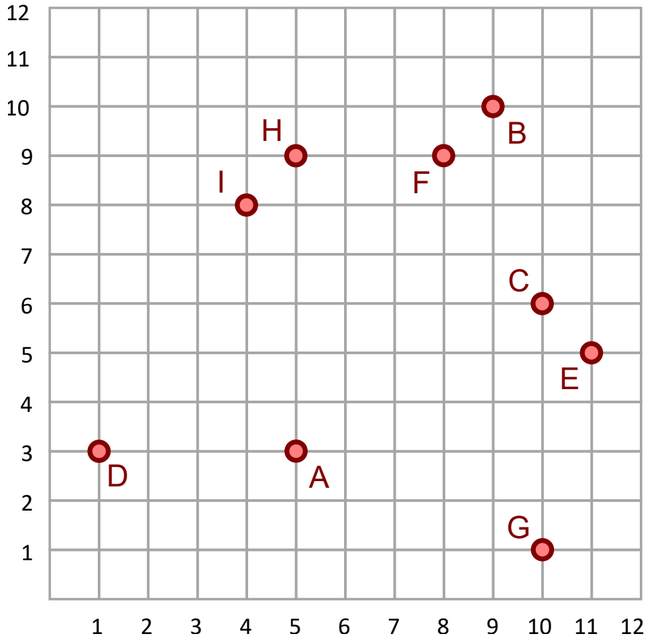
Point *k-d-Tree*

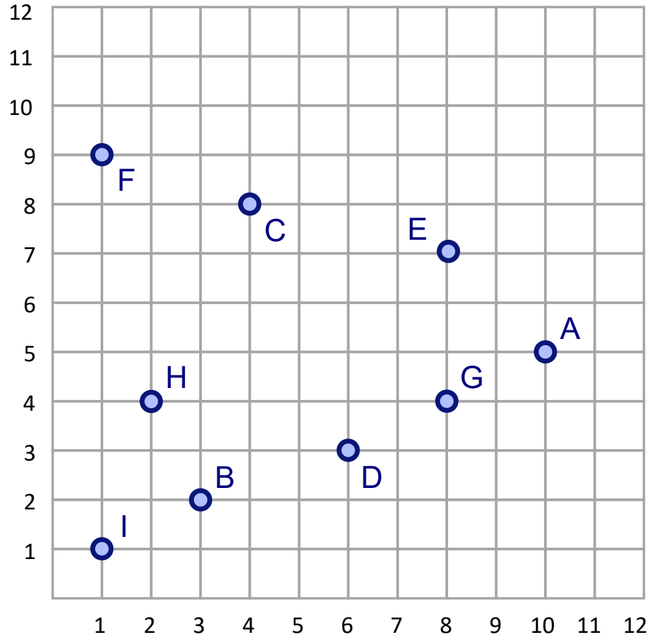


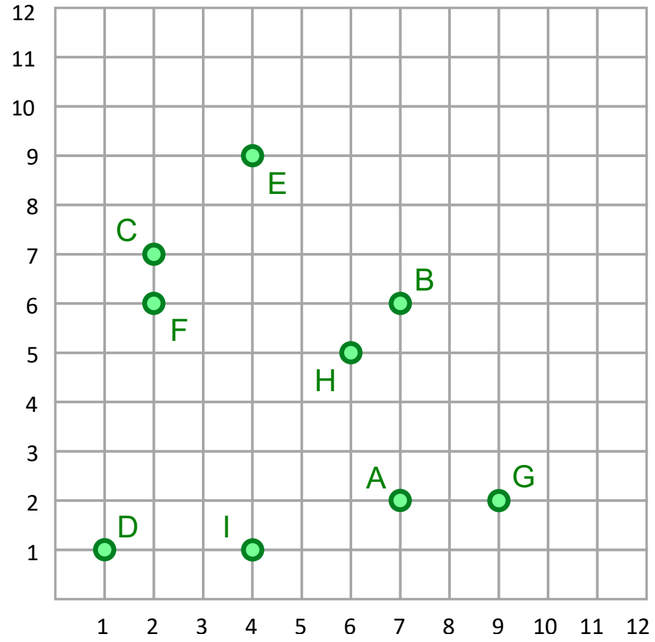
Point *k-d-Tree*



Point *k-d-Tree*





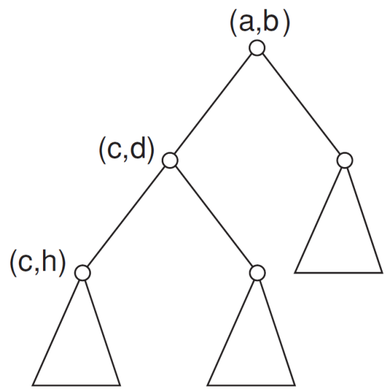


Point *k-d-Tree*

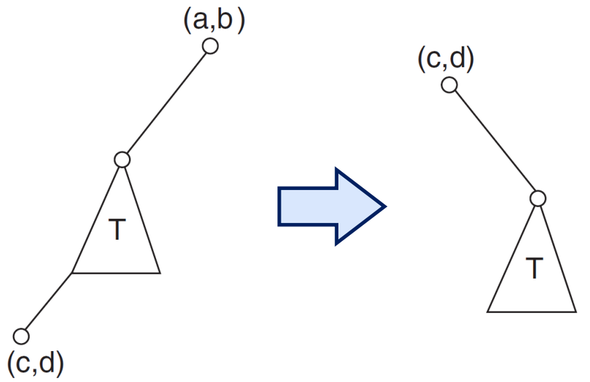
1. Si el nodo es una **hoja**: se borra

2. Si es nodo interno

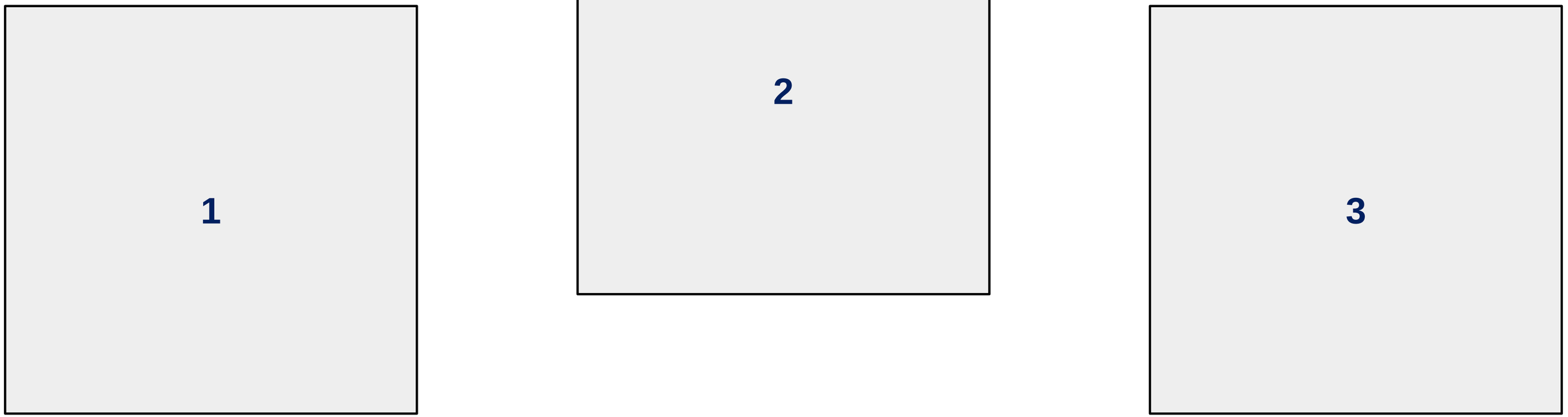
Si hay sub-árbol derecho



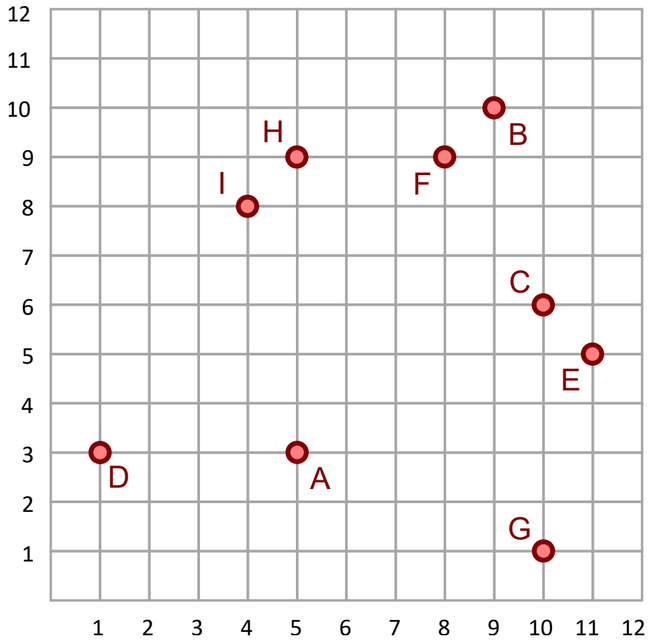
Si no hay sub-árbol derecho

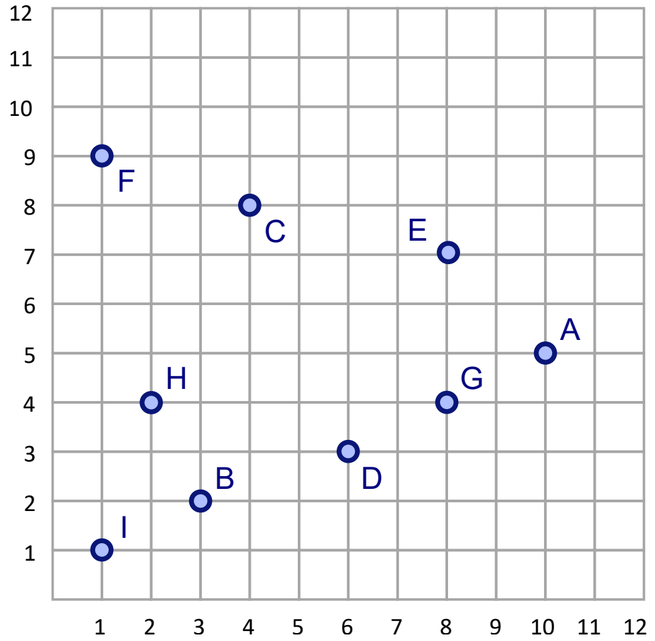


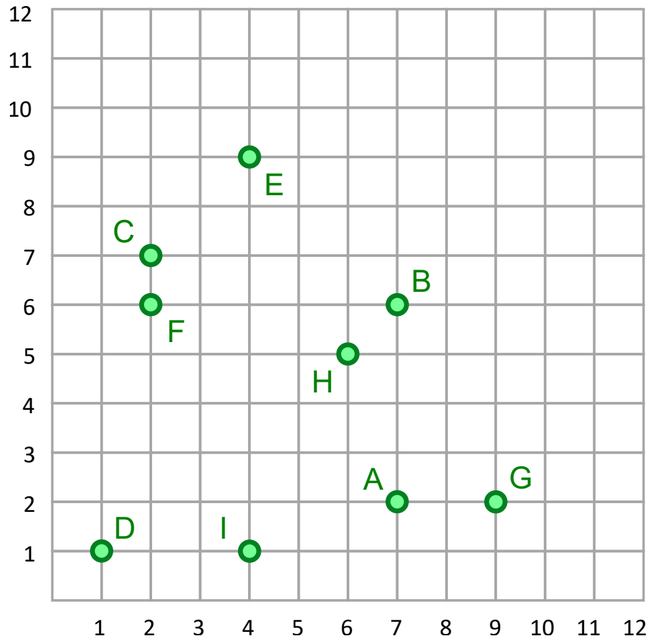
Point *k-d-Tree*



Point *k-d-Tree*

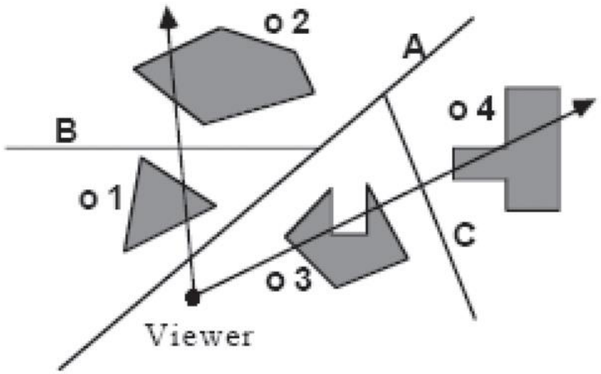


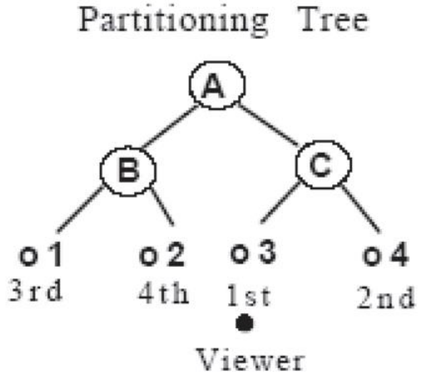


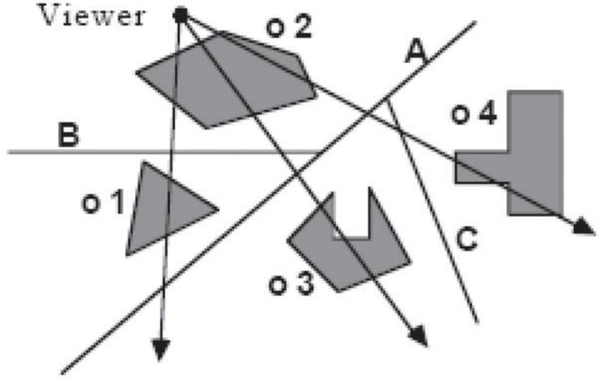


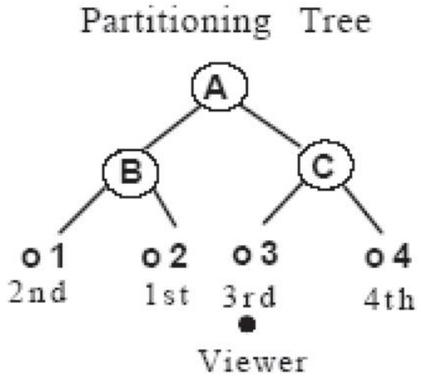
5. BSP-Tree

BSP-*Tree*

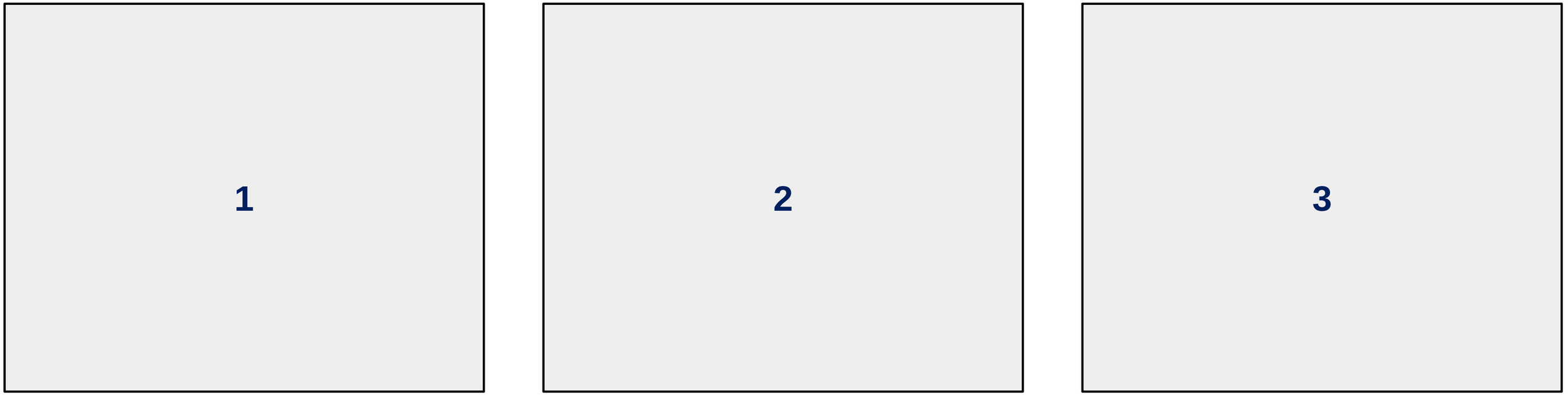

o 2
A
B
o 4
o 1
C
o 3
Viewer


Partitioning Tree
A
B
C
o 1
3rd
o 2
4th
o 3
1st
o 4
2nd
Viewer

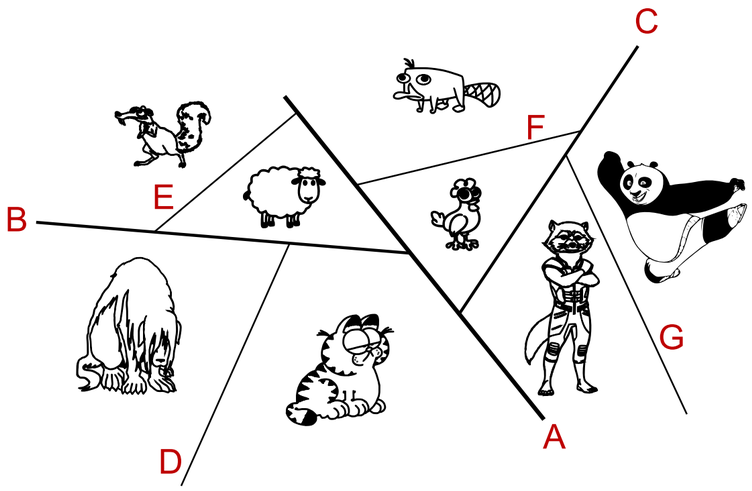

Viewer
o 2
A
B
o 4
o 1
C
o 3

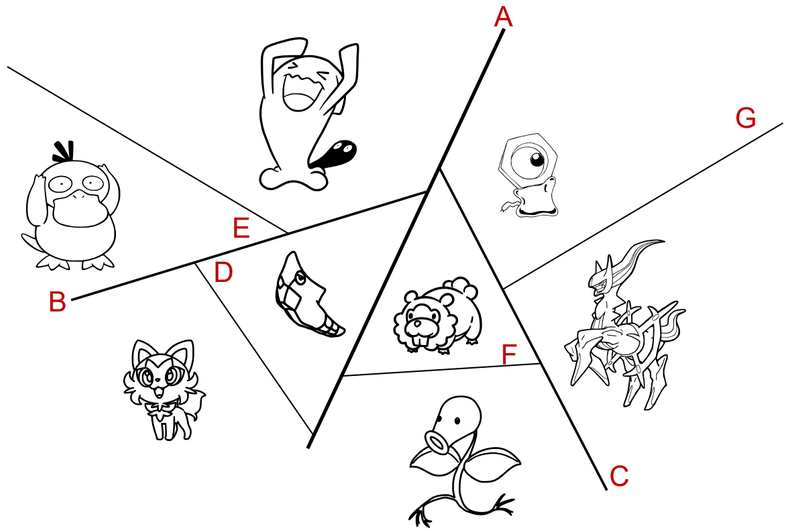

Partitioning Tree
A
B
C
o 1
2nd
o 2
1st
o 3
3rd
o 4
4th
Viewer

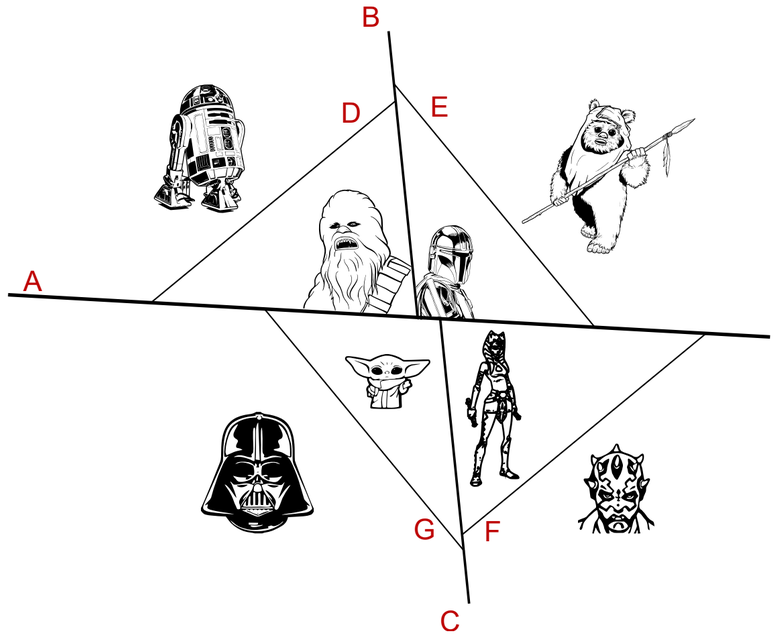
BSP-*Tree*



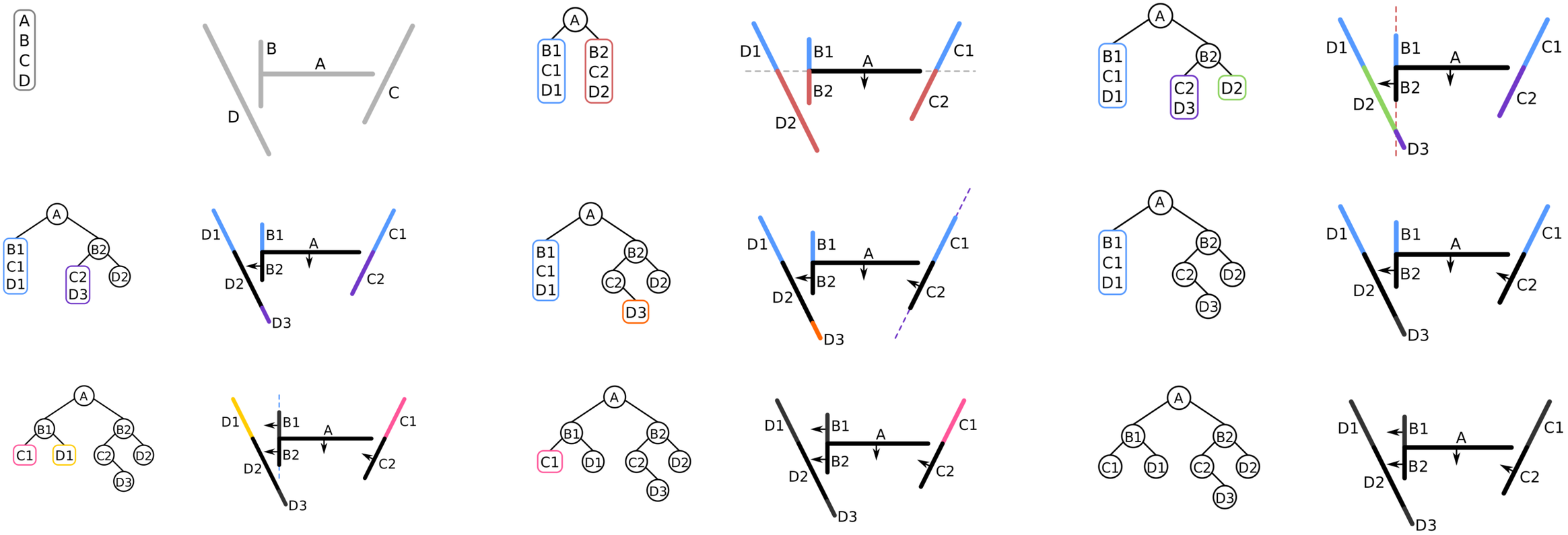
BSP-*Tree*



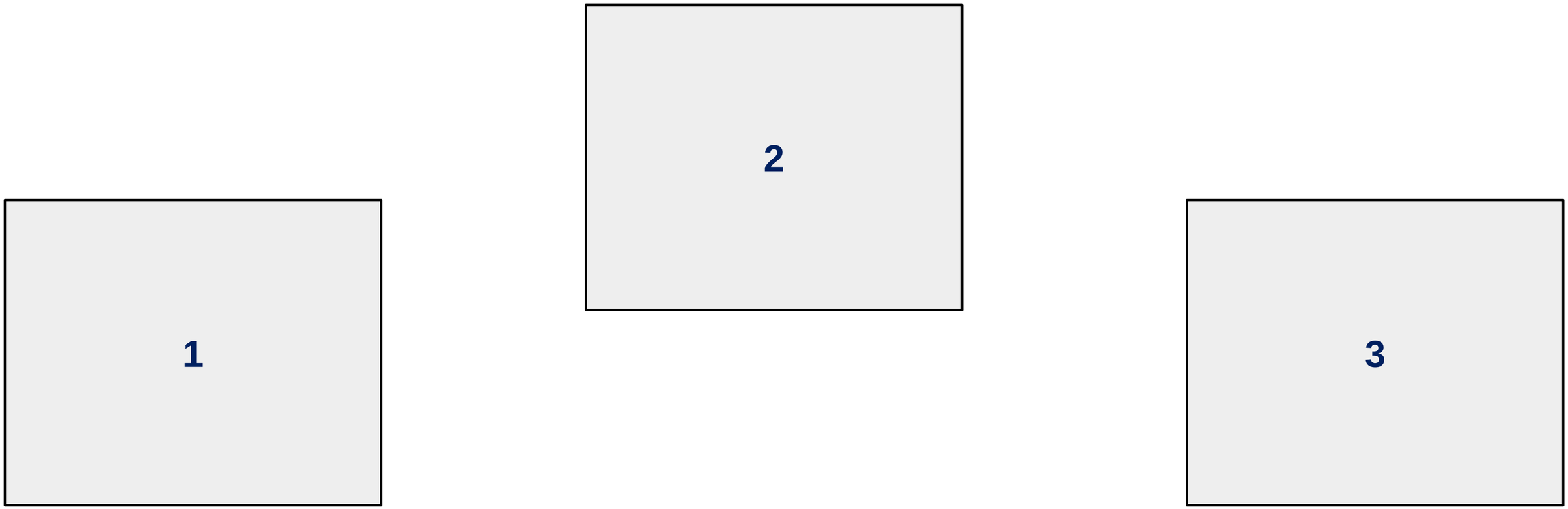




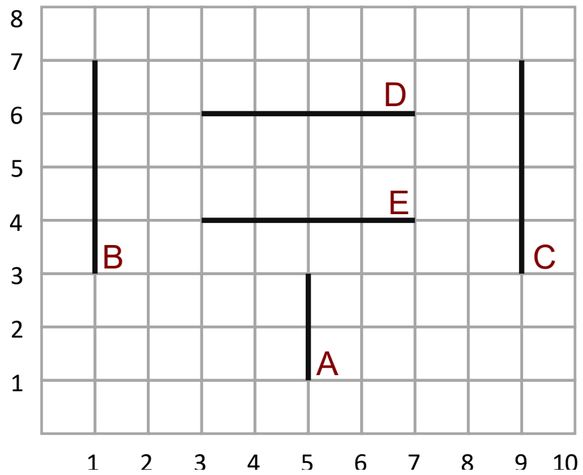
BSP-*Tree*

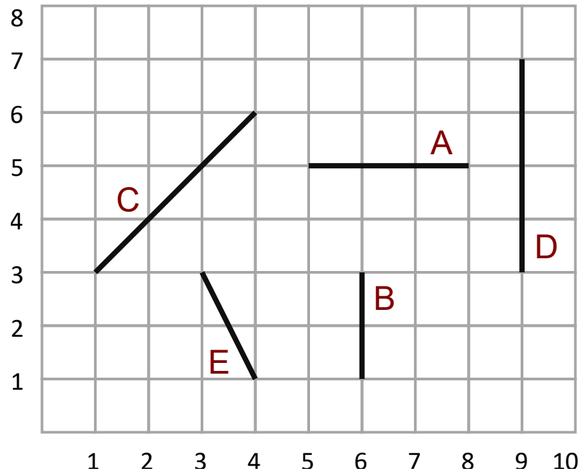

A
B
C
D
B
A
C
D
A
B1
C1
D1
B2
C2
D2
D1
B1
A
C1
B2
C2
D2
A
B1
C1
D1
B2
C2
D3
D2
D1
B1
A
C1
B2
C2
D2
D3
A
B1
C1
D1
B2
C2
D2
D3
D1
B1
A
C1
B2
C2
D2
D3
A
B1
C1
D1
B2
C2
D3
D1
B1
A
C1
B2
C2
D2
D3
A
B1
C1
D1
B2
C2
D2
D3
D1
B1
A
C1
B2
C2
D2
D3
A
B1
C1
D1
B2
C2
D3
D1
B1
A
C1
B2
C2
D2
D3

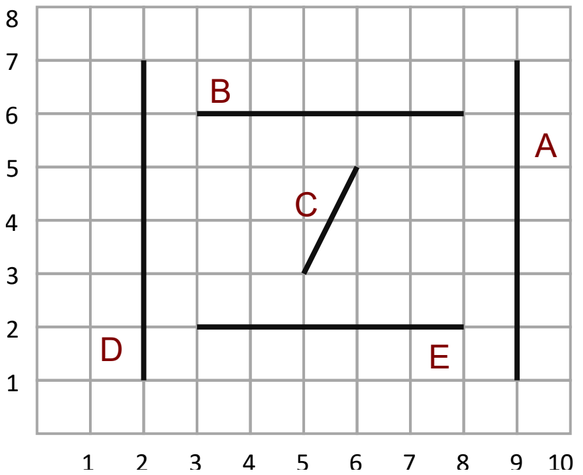
BSP-*Tree*



BSP-*Tree*

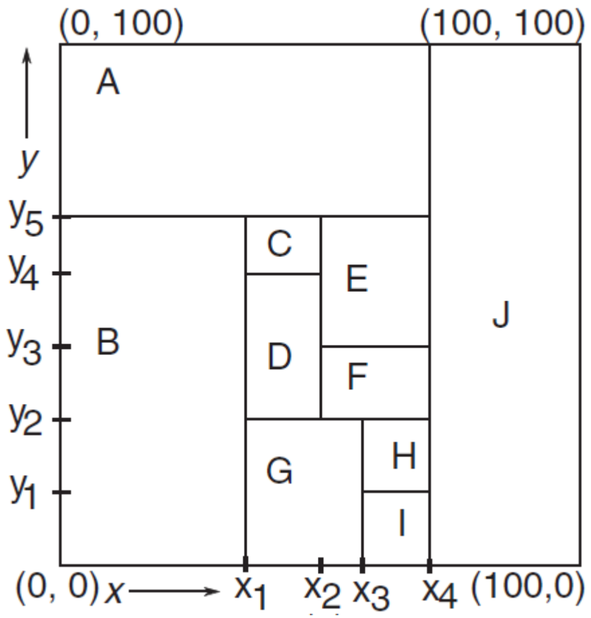

8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10
D
E
B
C
A

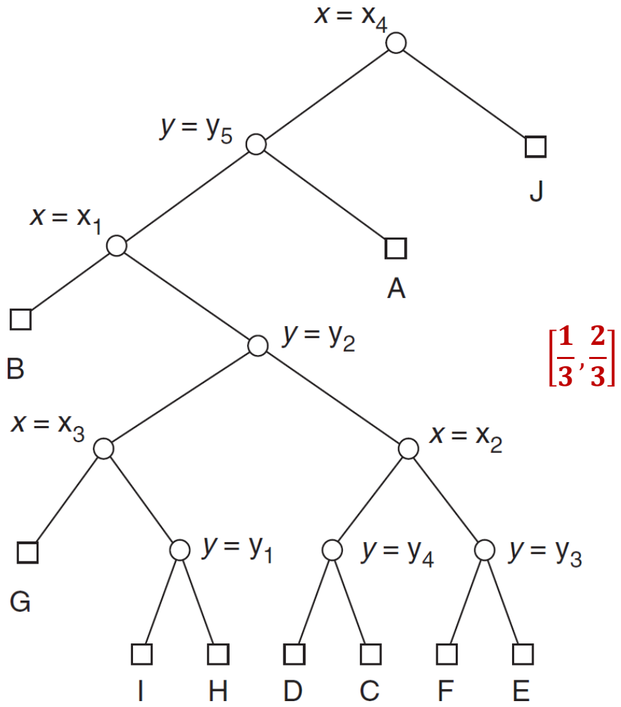

8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10
A
C
D
B
E


8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10
B
A
C
D
E

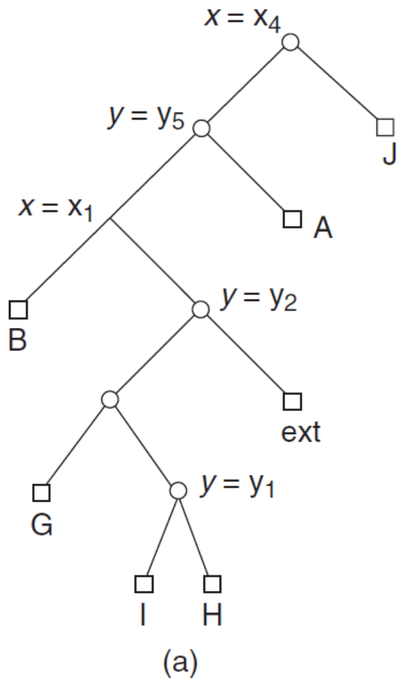
6. hB-Tree

hB-*Tree*

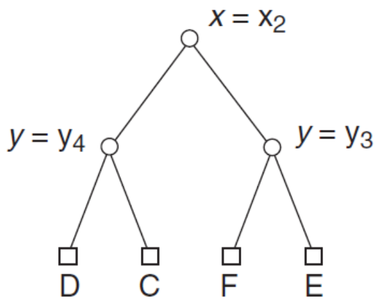




$\left[\frac{1}{3}, \frac{2}{3}\right]$

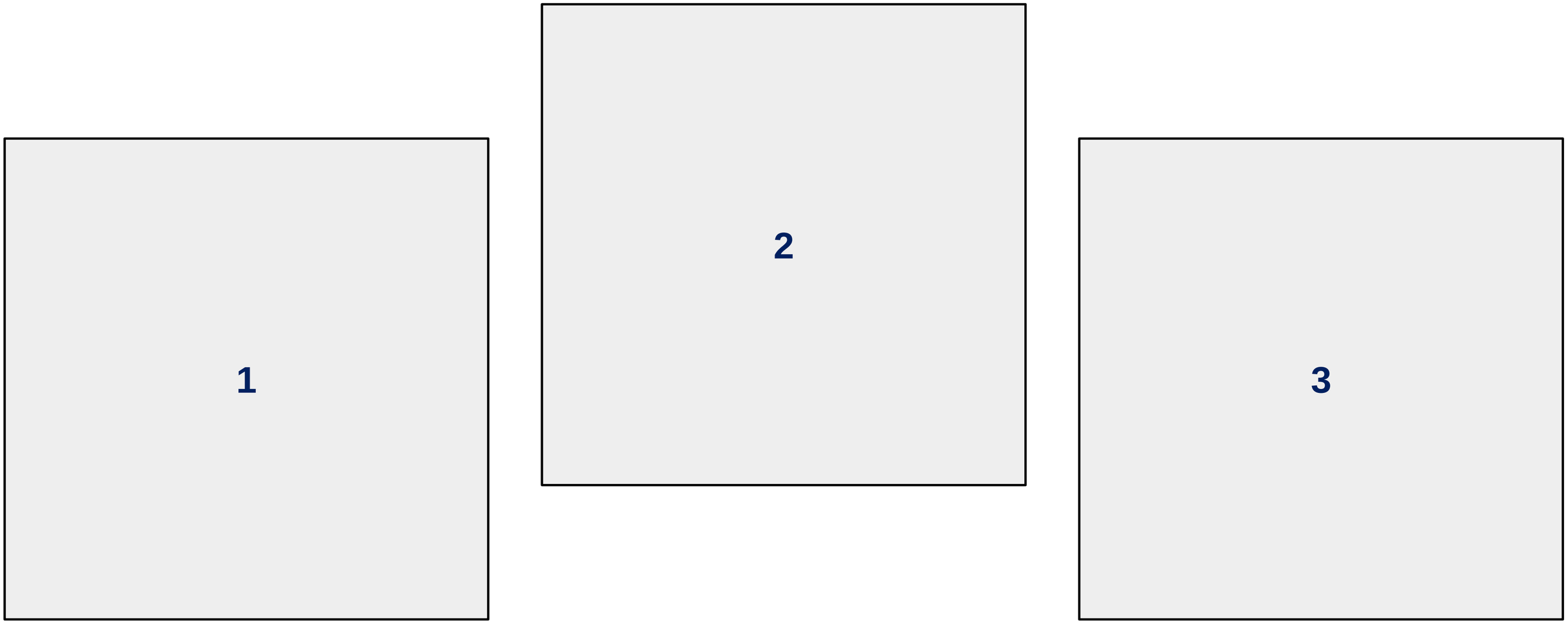


(a)

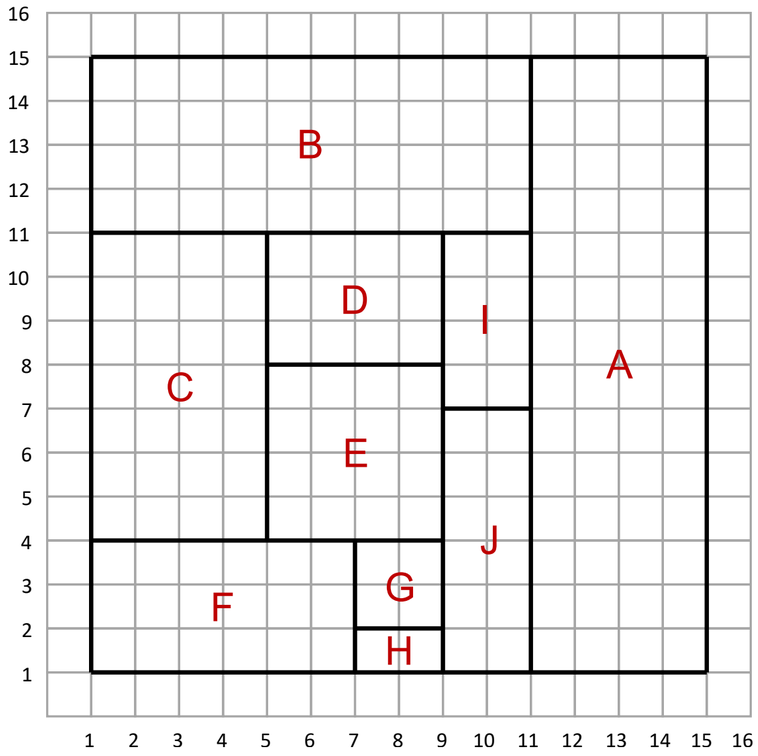


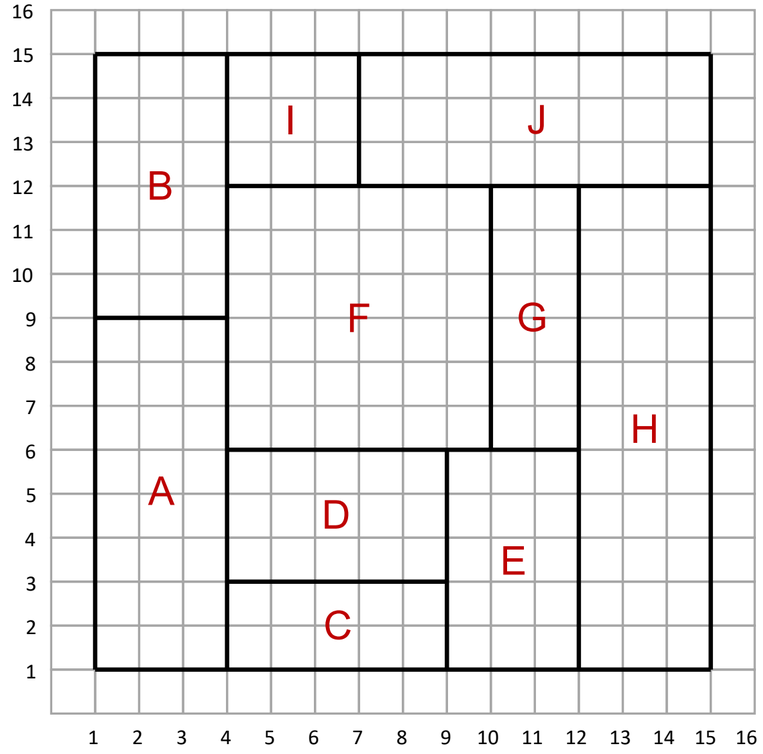
(b)

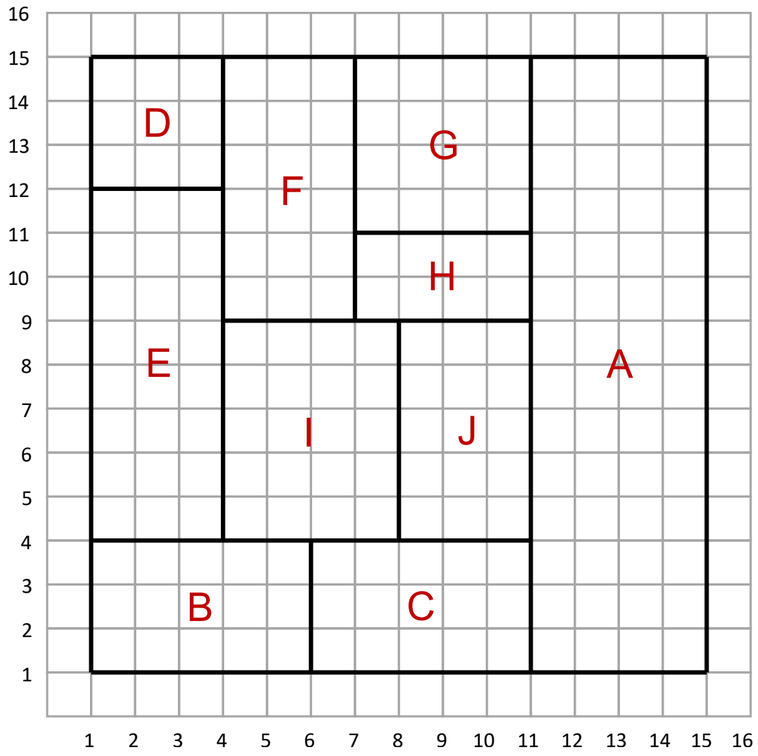
hB-*Tree*



hB-*Tree*







7. R-Tree

R-*Tree*

Algorithm `insert`(u, p)

1. **if** u is a leaf node **then**
2. add p to u
3. **if** u overflows **then**
   /* namely, u has $B + 1$ points */
4. `handle-overflow`(u)
5. **else**
6. $v \leftarrow$ `choose-subtree`(u, p)
   /* which subtree under u should we insert p into? */
7. `insert`(v, p)

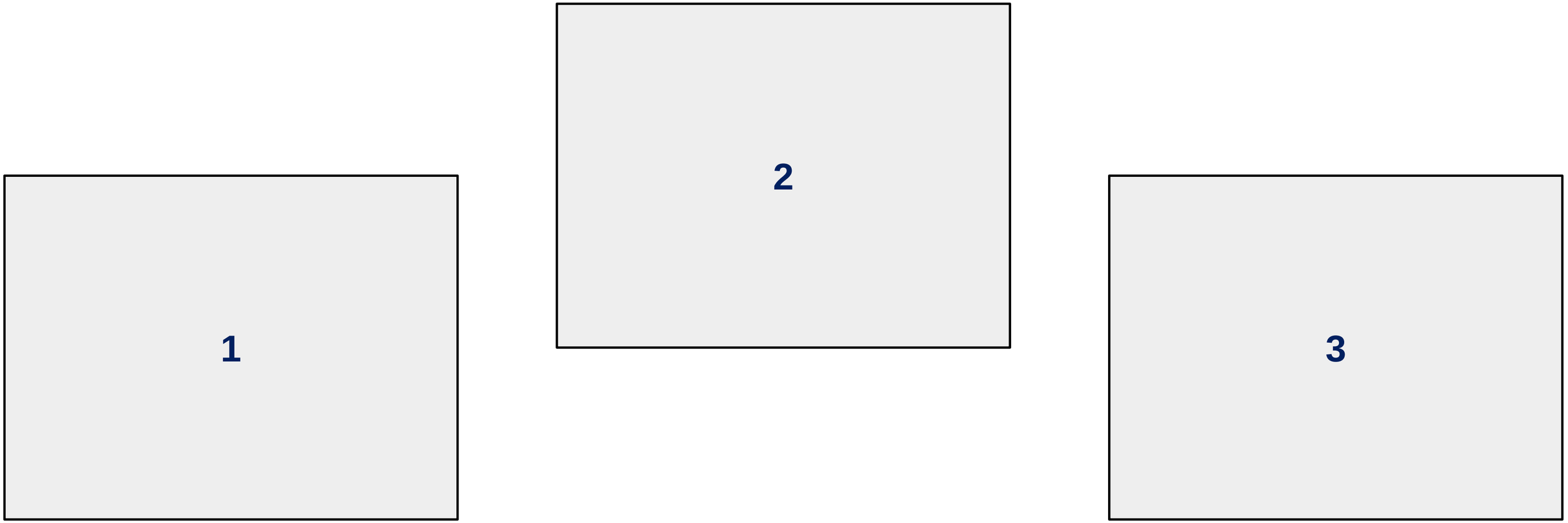
Algorithm `handle-overflow`(u)

1. `split`(u) into u and u'
2. **if** u is the root **then**
3. create a new root with u and u' as its child nodes
4. **else**
5. $w \leftarrow$ the parent of u
6. update $MBR(u)$ in w
7. add u' as a child of w
8. **if** w overflows **then**
9. `handle-overflow`(w)

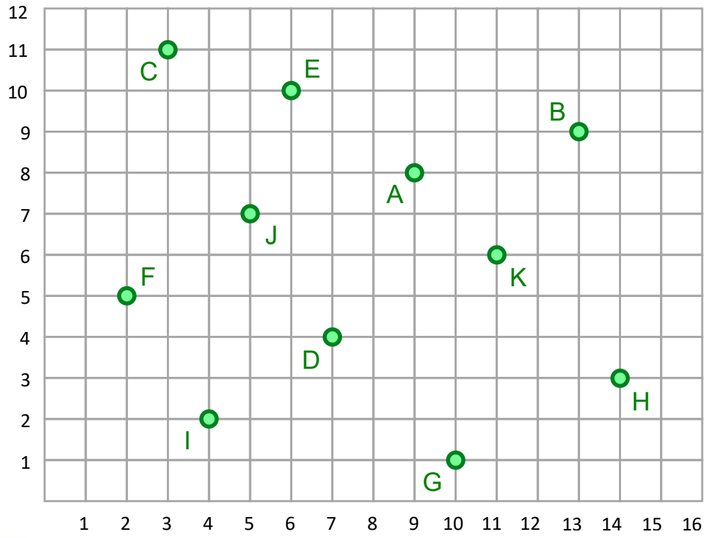
Linear Split

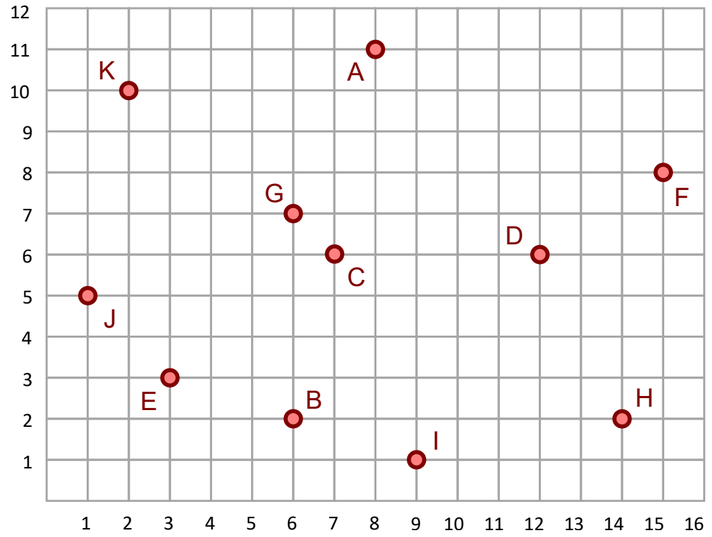
Elija dos objetos como semillas para los dos nodos, donde estos objetos estén lo más separados posible. A continuación, considere cada objeto restante en un orden aleatorio y asígnelo al nodo que requiera la menor ampliación de su respectiva MBR.

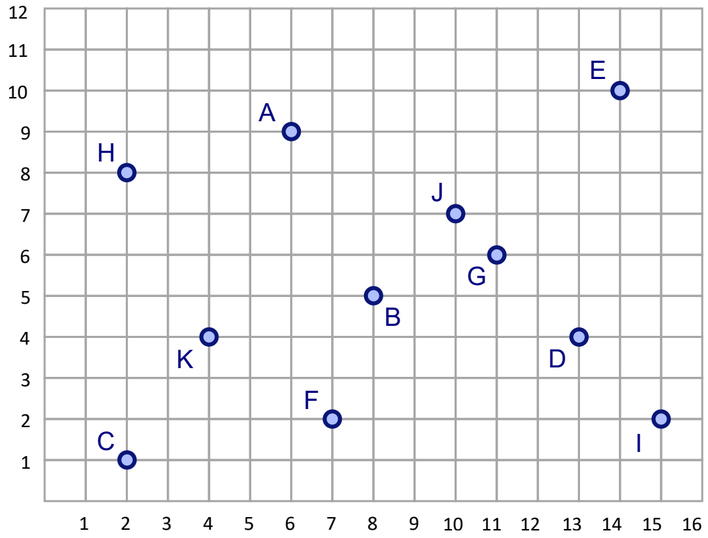
R-*Tree*



R-*Tree*







UTEC

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA