

# Árvores R

**Prof. Dr. Lucas C. Ribas**

**Disciplina:** Estrutura de Dados II

Departamento de Ciências de Computação e Estatística



UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”



**IBILCE / UNESP - CÂMPUS DE SÃO JOSÉ DO RIO PRETO**

# Agenda



- Conceitos introdutórios
- Estrutura da Árvore R
- Consulta
- Inserção
- Split

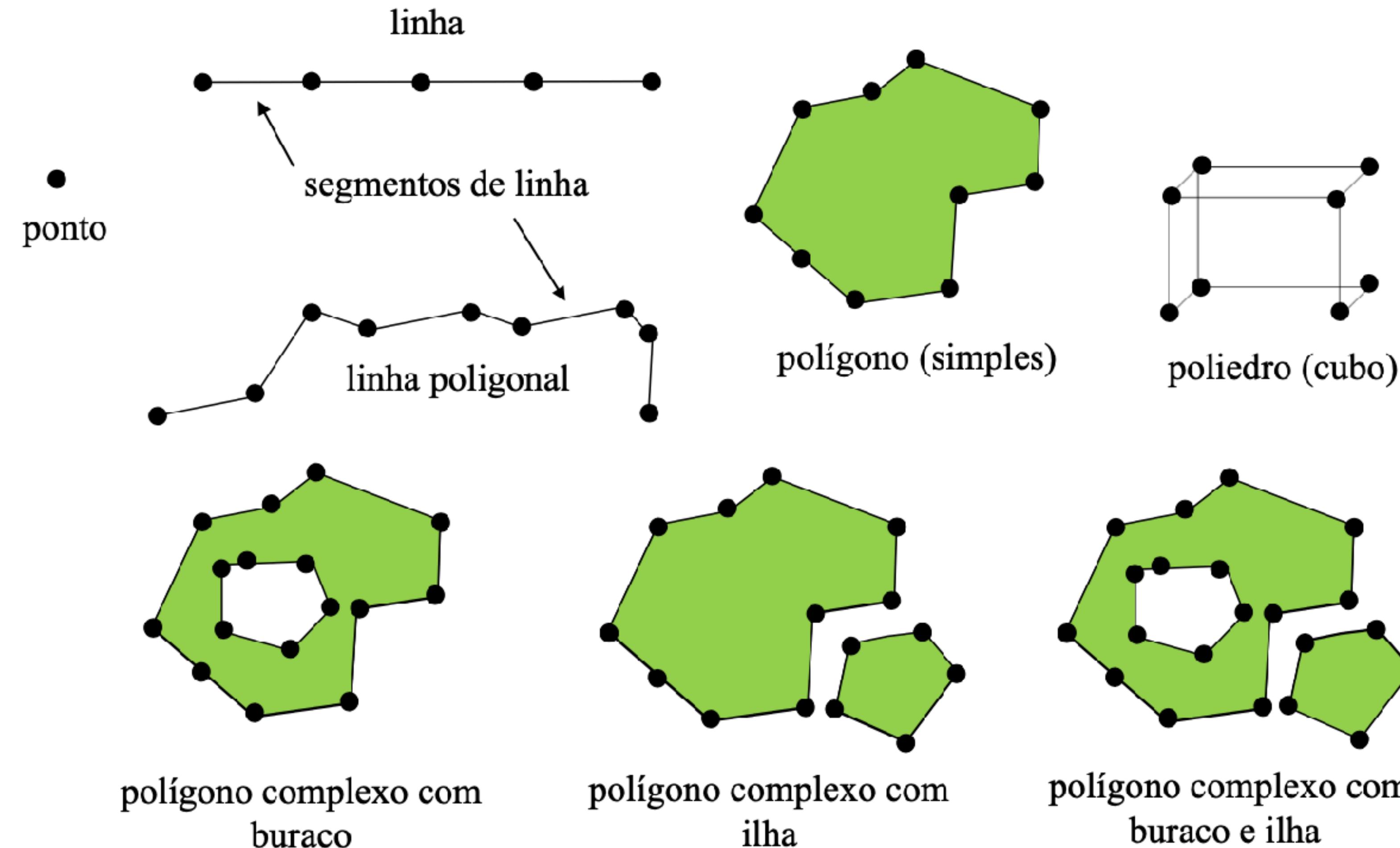
# Introdução

# Tipos de dados espaciais



- **Ponto:** unidade mínima representativa de um objeto espacial
- **Linha:** sequência de pontos retilíneos
- **Linha poligonal:** sequência de pontos não retilíneos
- **Polígono:** sequência fechada de linhas ou linhas poligonais
- **Polígono complexo:** polígono com buracos e/ou partes disjuntas
- **Poliedro:** sólido composto por um número finito de faces.

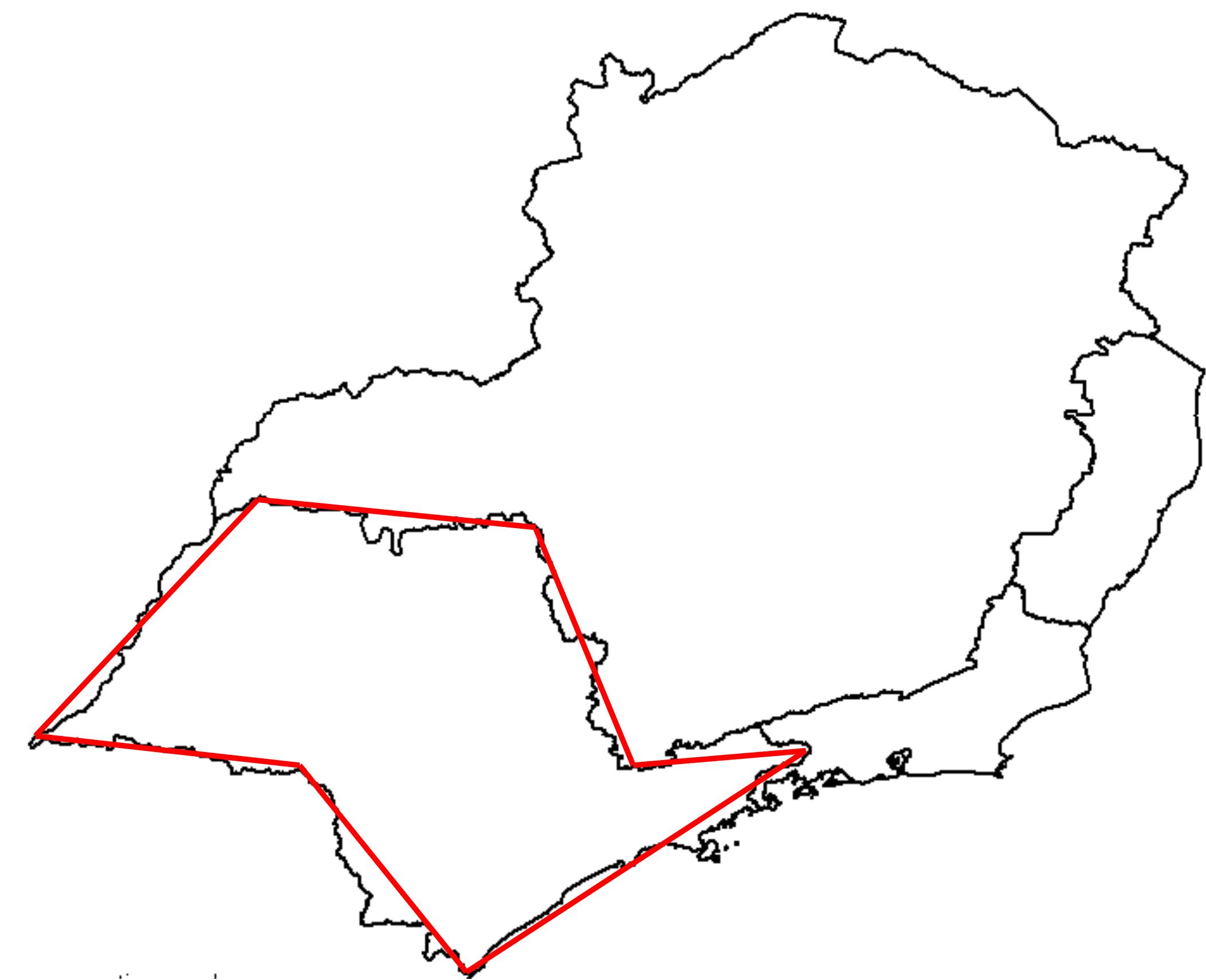
# Tipos de dados espaciais



# Representação dos dados



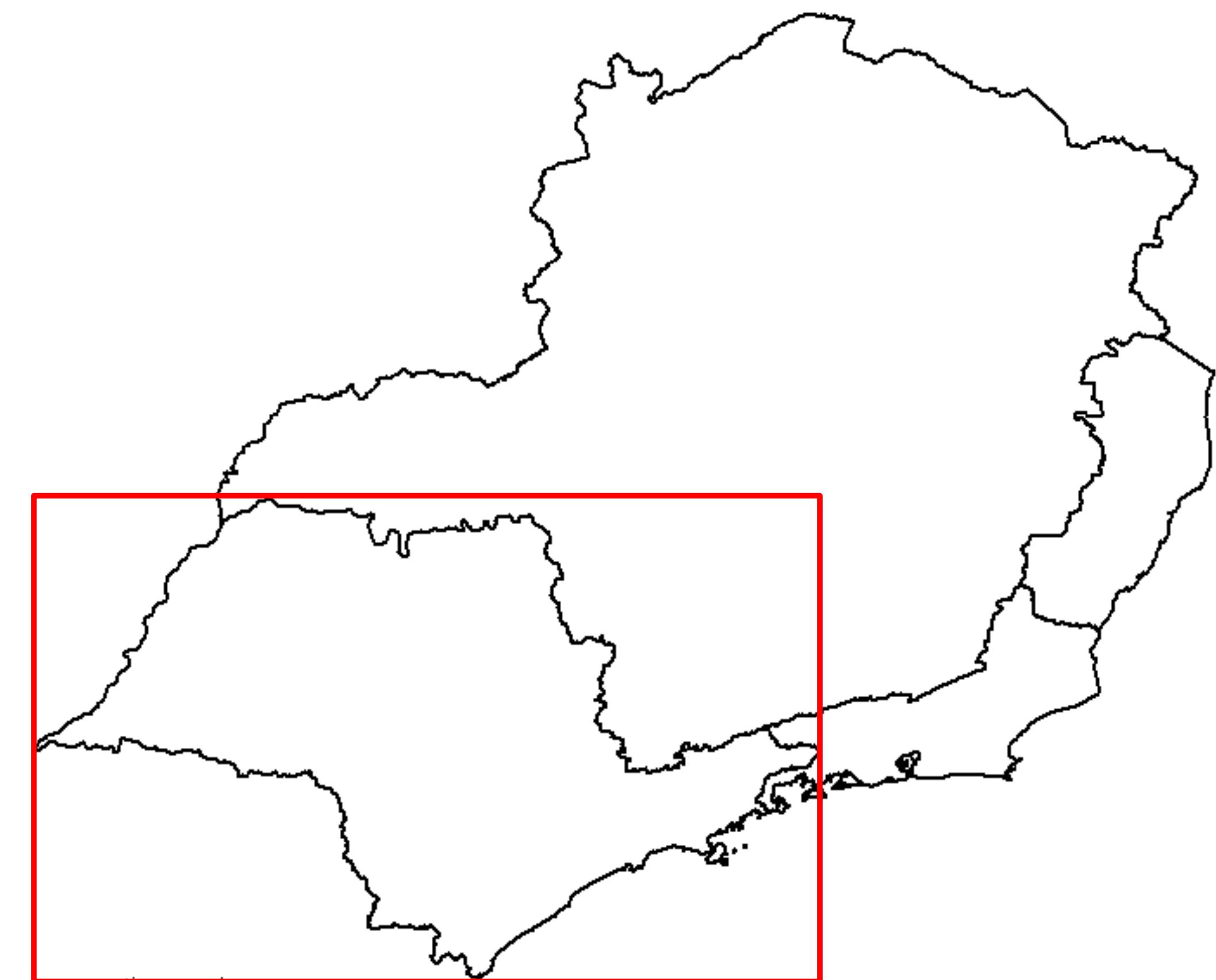
# Representação dos dados



# Representação dos dados



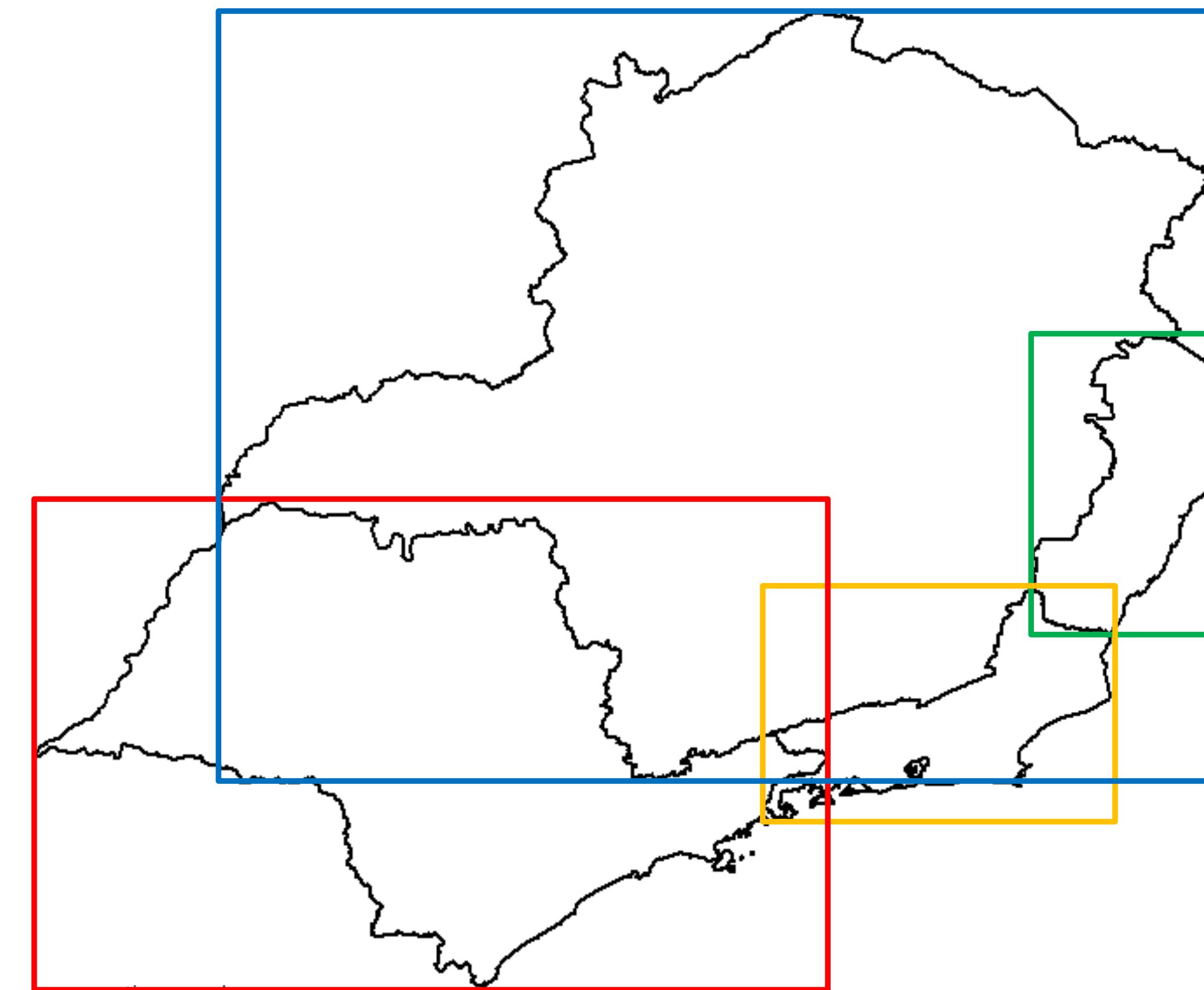
Retângulo delimitador mínimo (*Minimum Bounding Rectangle – MBR*)



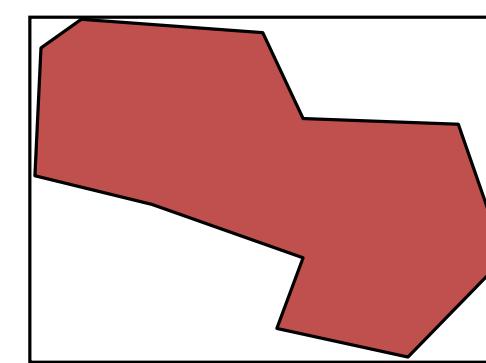
# Representação dos dados



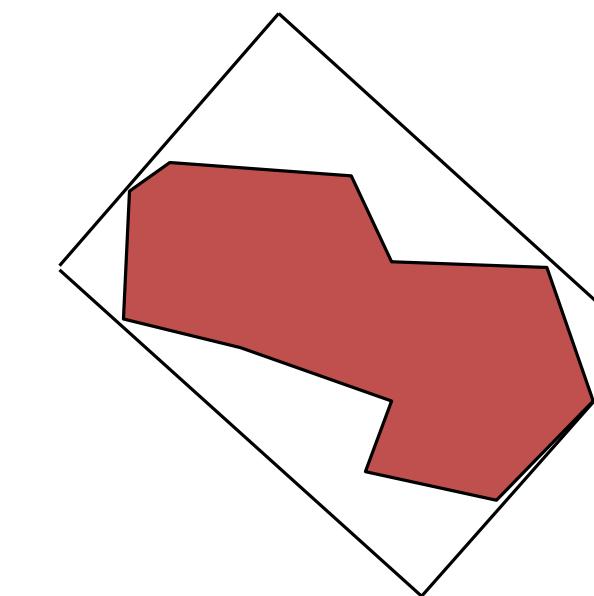
Retângulo delimitador mínimo (*Minimum Bounding Rectangle – MBR*)



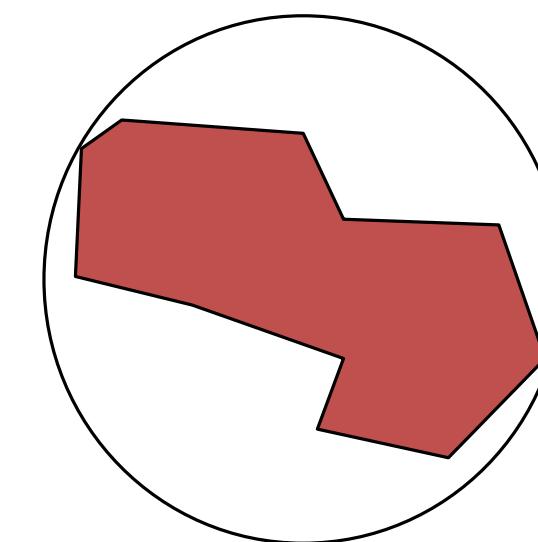
# Representação dos dados



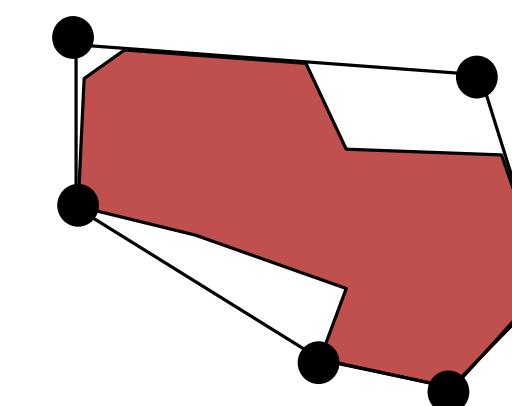
retângulo envolvente mínimo



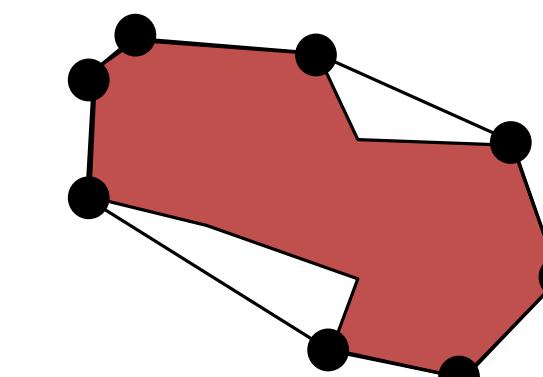
retângulo envolvente mínimo  
rotacionado



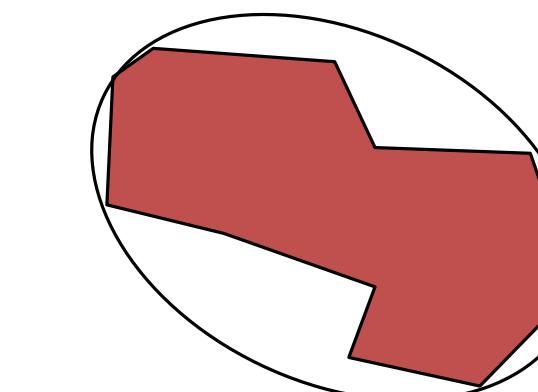
círculo envolvente mínimo



polígono envolvente mínimo  
com 6 vértices



casco convexo



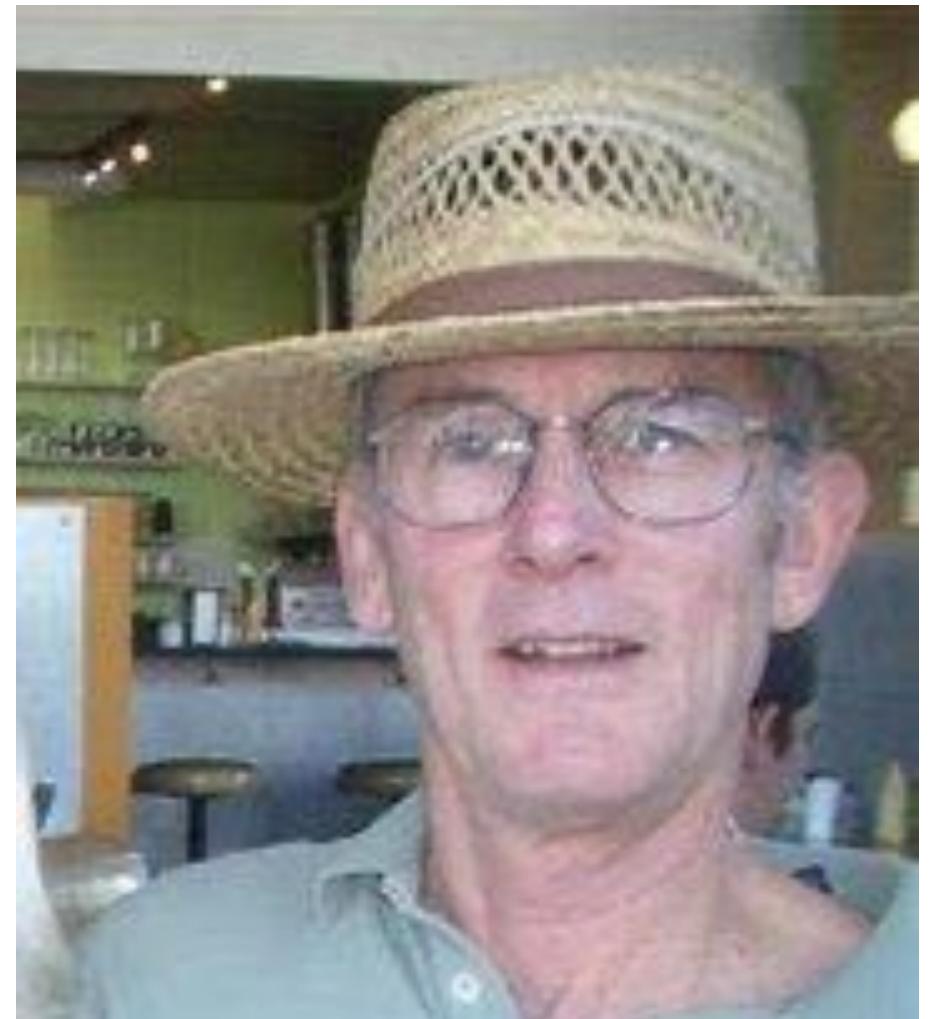
elipse envolvente mínima

# Árvore B



- Dados espaciais são tipos de objetos utilizados com frequência em muitas áreas
  - **Computer-assisted design (CAD), dados geográficos** para análise e mapeamento e **projetos de circuitos (VLSI)** são exemplos de domínios nos quais dados espaciais são criados, pesquisados e excluídos.
- Este tipo de dados requer estruturas de dados especiais para serem processados de forma eficiente.
  - Por exemplo, podemos solicitar que todos os municípios de uma área especificada por coordenadas geográficas sejam impressos ou que todos os edifícios próximos da prefeitura sejam identificados

- Muitas estruturas de dados diferentes foram desenvolvidas para acomodar esse tipo de dados.
  - Um exemplo é uma **Árvore R** (Guttman, 1984)
- Estrutura de dados utilizada principalmente para **indexação espacial**, ou seja, para armazenar dados que existem em **múltiplas dimensões**, como coordenadas em um mapa
- Essa estrutura impõe um **mínimo** e um **máximo** de entradas (chaves) por nó





## ● Aplicações:

- Sistemas de Informações Geográficas (GIS)
- Sistemas CAD
- Arquiteturas VLSI
- Sistemas P2P
- Bioinformática
- Data Streams



- Dinâmica

- Permite novas inserções e remoções

- Hierárquica

- Nós folhas e nós índices

- Armazenamento Secundário

- Nós são páginas de disco de tamanho fixo

- Construção Bottom-Up

- Todos os objetos são inseridos nas folhas

- Balanceada

- Folhas no mesmo nível



- Uma árvore R de ordem  $(m, M)$  é uma estrutura semelhante a uma árvore B contendo:
  - $M$  é o número máximo de entrada por nó
  - $m$  é o número mínimo de entradas em um nó
    - sendo  $m \leq \left\lfloor \frac{M}{2} \right\rfloor$
- Conseqüentemente, uma árvore R não precisa estar pelo menos meio cheia
- Altura máxima da árvore:  $h_{max} = \lceil \log_m N \rceil - 1$ 
  - $N$ : número de objetos inseridos



- O número mínimo de entradas permitido na raiz é 2, a menos que a raiz seja uma folha. Nesse caso, ela pode conter apenas uma ou nenhuma entrada
- Todas as folhas estão no mesmo nível



- Uma folha em uma árvore R contém entradas na forma:

- $(rect, id)$  onde:
  - $rect = ([c_i^1, c_i^2], \dots, [c_n^1, c_n^2])$  é um **retângulo n-dimensional**.  $rect$  é o menor retângulo que contém (delimita) o objeto  $id$ , ou seja, o retângulo delimitador mínimo (*Minimum Bounding Rectangle* – MBR)
  - $c_i^1$  e  $c_i^2$  são coordenadas ao longo do mesmo eixo
  - $id$  é um ponteiro para um registro em um arquivo de dados

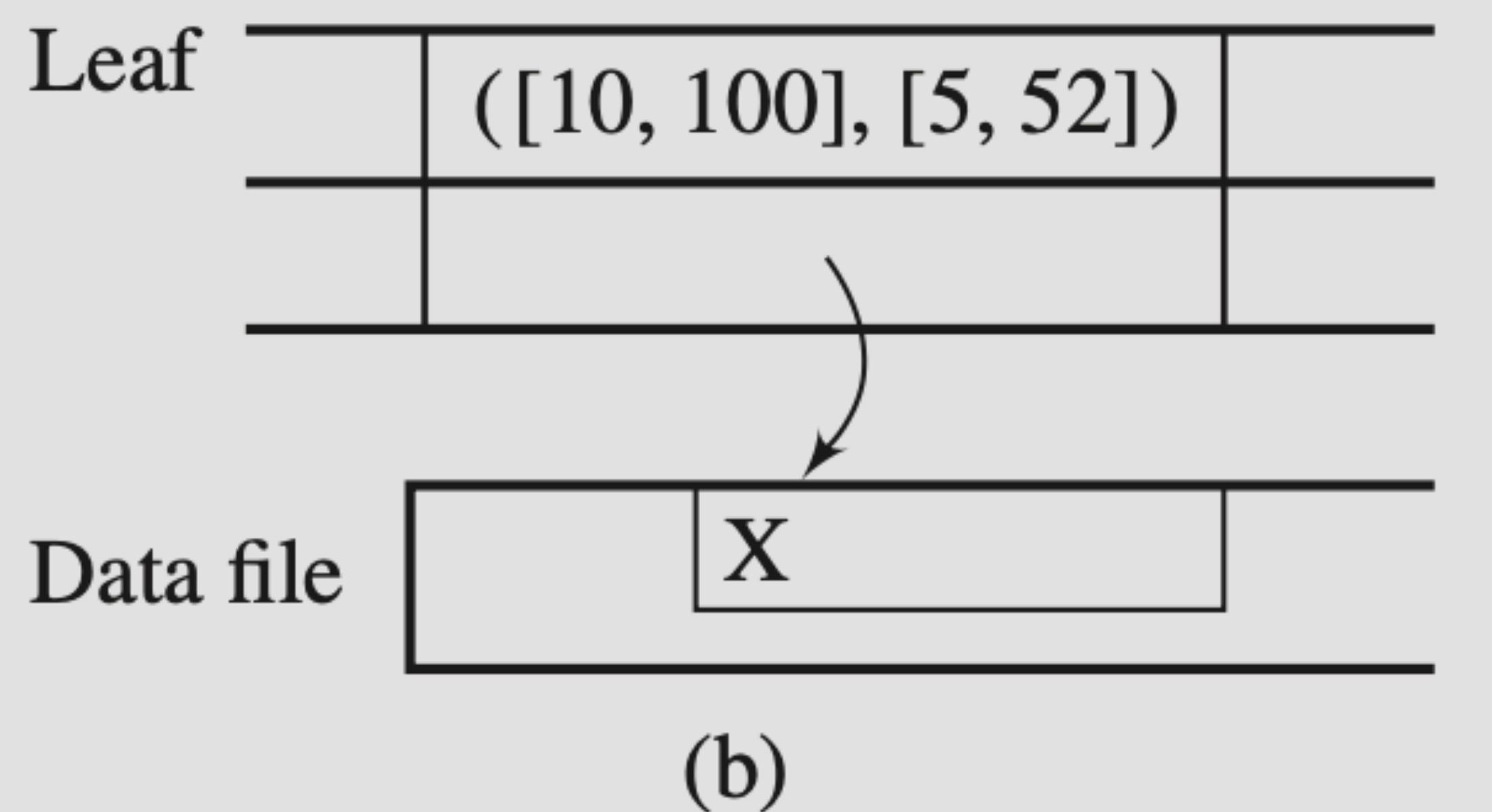
# Estrutura dos nós



- $(rect, id)$  onde  $rect = ([c_i^1, c_i^2], \dots, [c_n^1, c_n^2])$  é um **retângulo n-dimensional**,  $c_i^1$  e  $c_i^2$  são coordenadas ao longo do mesmo eixo e  $id$  é um ponteiro para um registro em um arquivo de dados
  - a entrada em uma folha correspondente a um objeto  $X$  em um plano cartesiano como na Figura é o par  $(([10,100], [5,52]), X)$ .



(a)



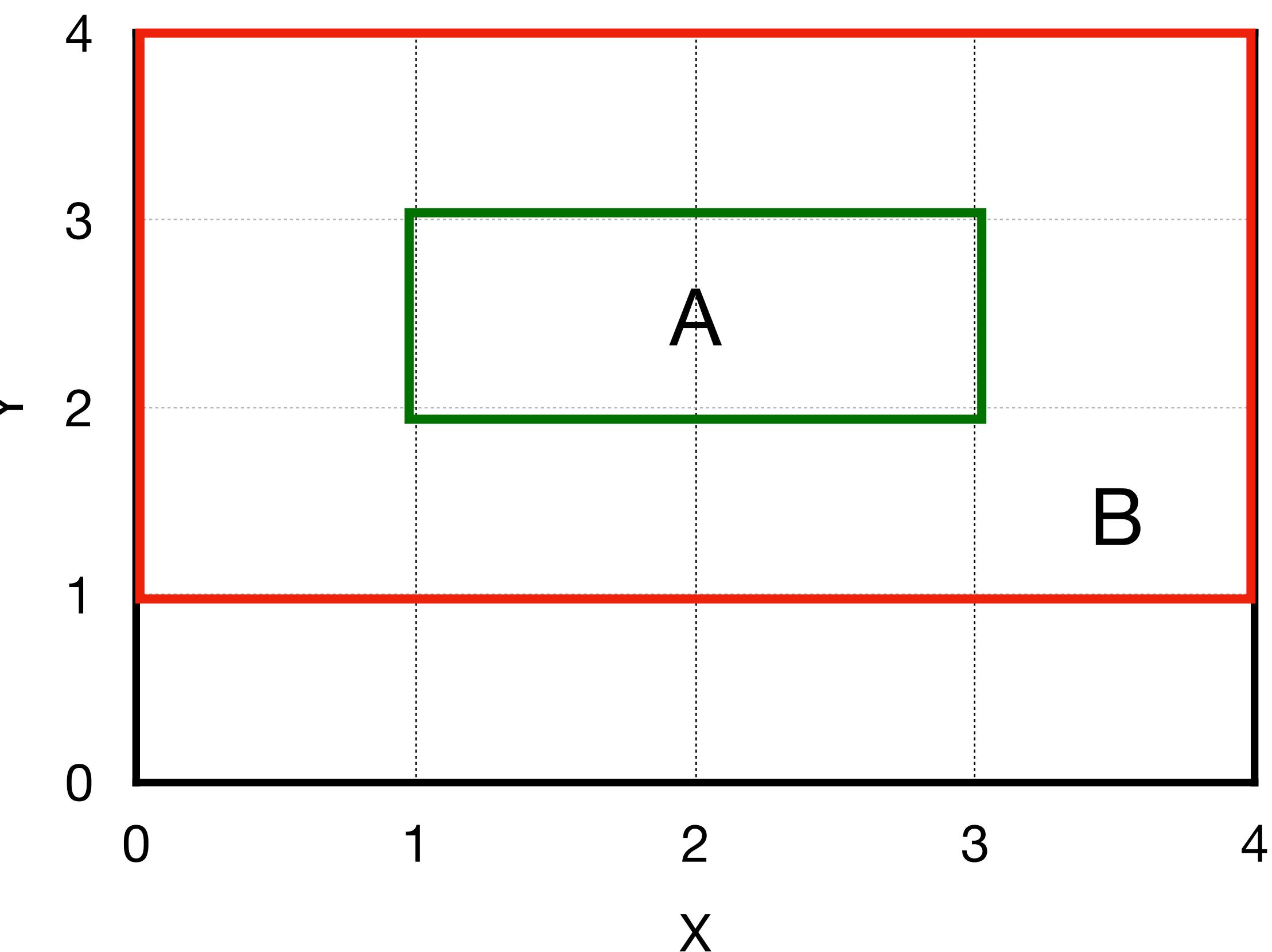
(b)

# Estrutura dos nós



- Um nó não-folha (índice) tem forma:
  - $(rect, prt)$  onde:
    - $prt$ : referência ao nó do nível imediatamente inferior (filho).
    - $rect$  é o menor retângulo que abrange todos os retângulos encontrados em  $prt$ .

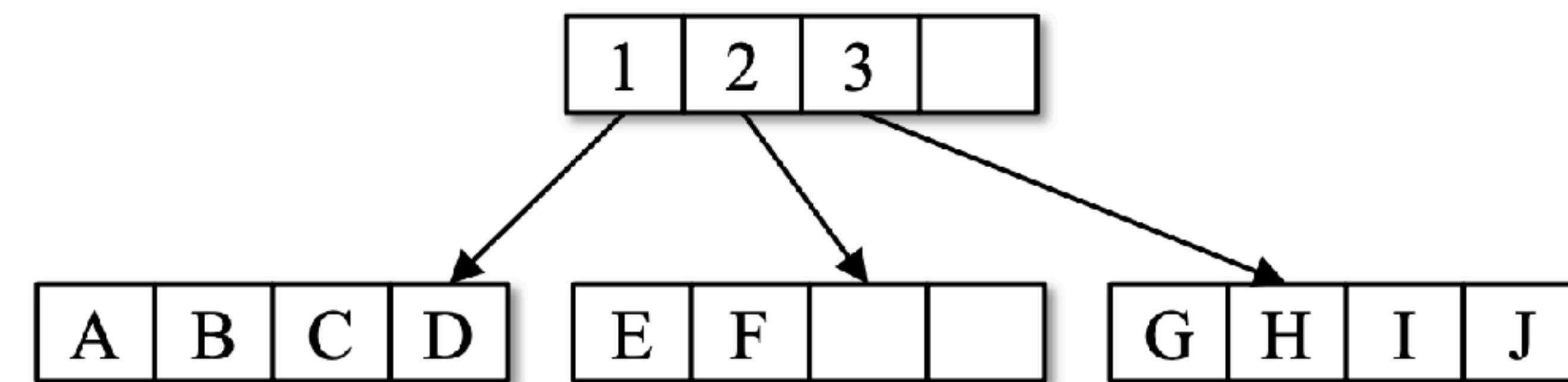
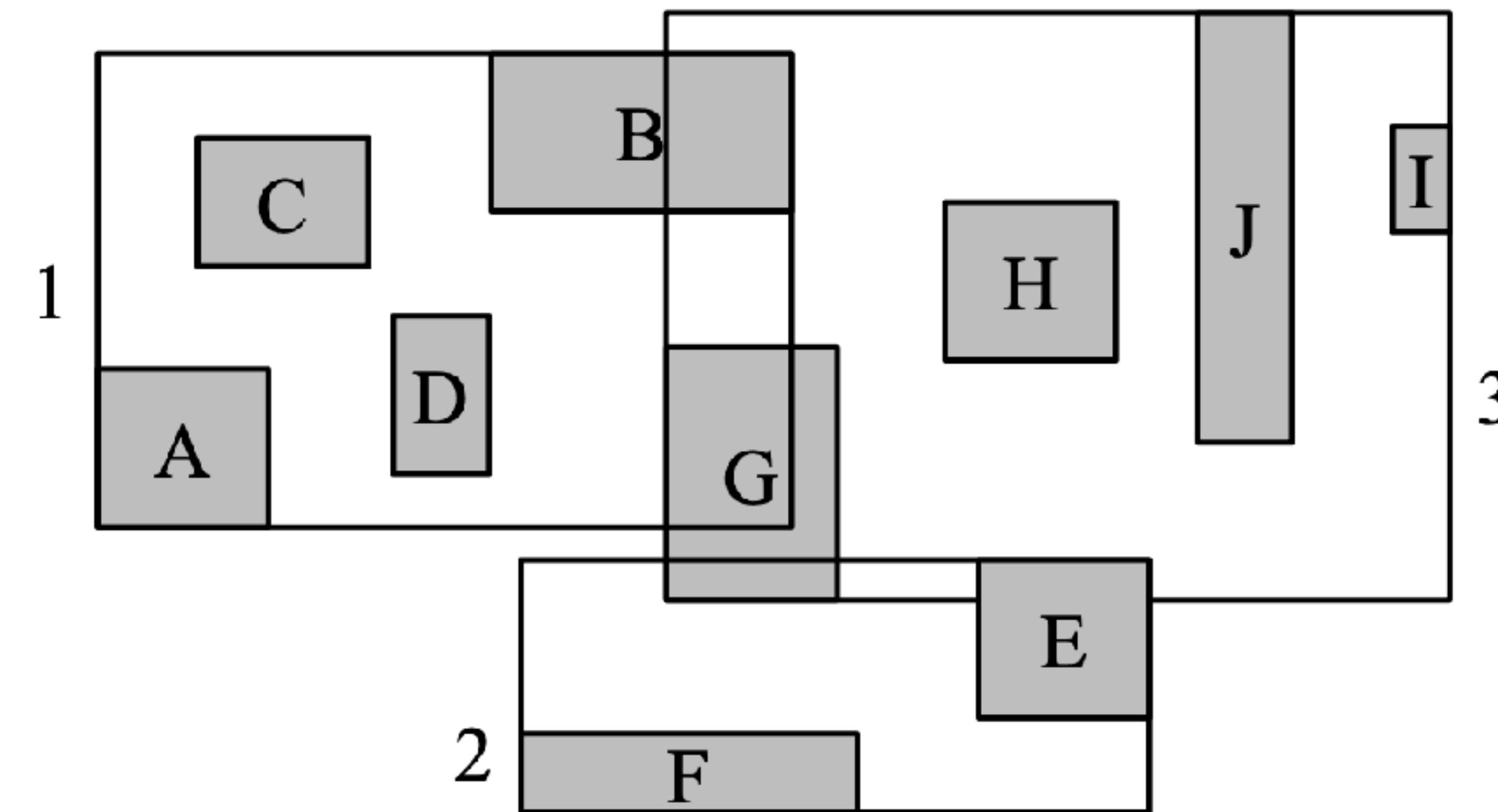
- Verificar se retângulo  $A = ([A_{x_1} A_{x_2}], [A_{y_1} A_{y_2}])$  está completamente contido dentro de  $B = ([B_{x_1} B_{x_2}], [B_{y_1} B_{y_2}])$
- O canto inferior esquerdo de A deve estar à direita e acima do canto inferior esquerdo de B:
  - Se  $A_{x_1} \geq B_{x_1}$  e  $A_{y_1} \geq B_{y_1}$
- O canto superior direito de A deve estar à esquerda e abaixo do canto superior direito de B:
  - Se  $A_{x_2} \leq B_{x_2}$  e  $A_{y_2} \leq B_{y_2}$
- Se ambas as condições forem atendidas, então o retângulo A está completamente contido dentro do retângulo B



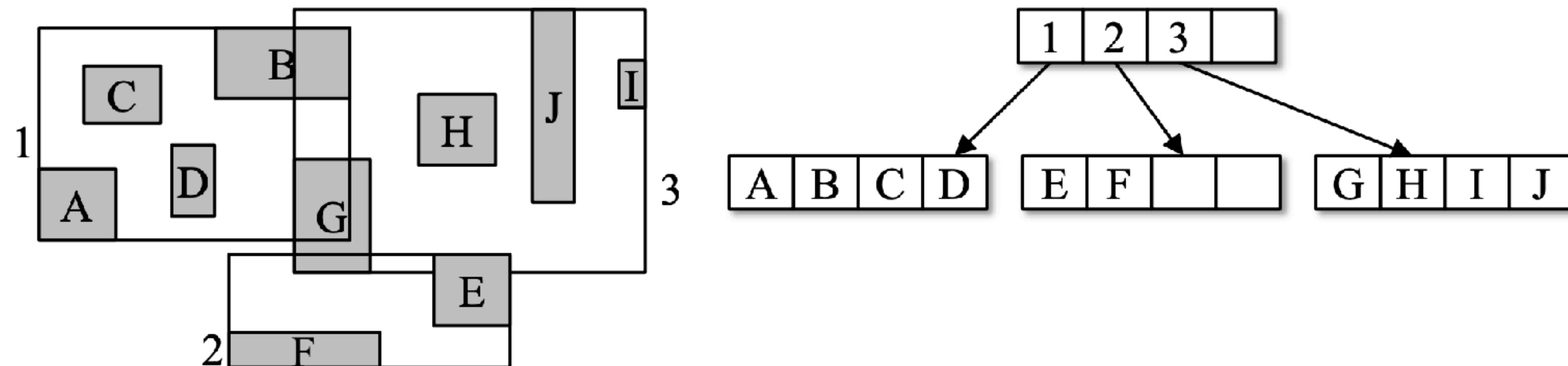
$$A = ([1,3], [2,3])$$

$$B = ([0,4], [1,4])$$

# Exemplo: Árvore R(2, 4)



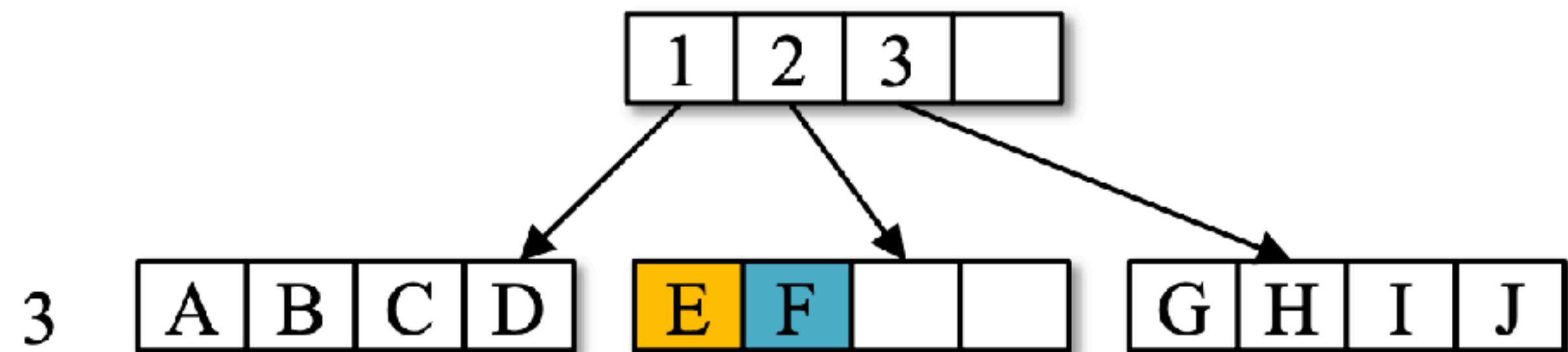
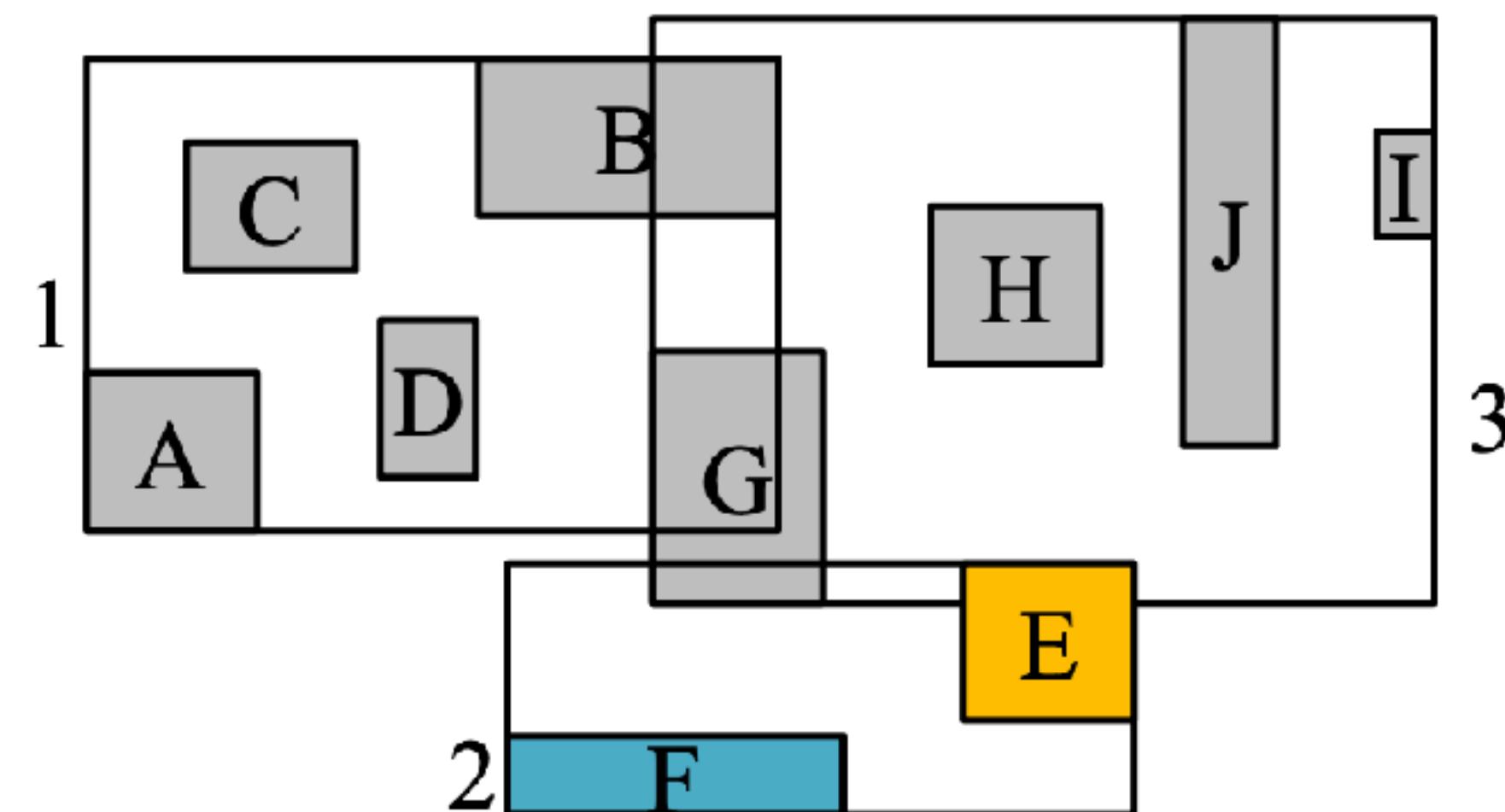
# Representação dos nós



<i>Leaf</i>	$\alpha$	$\beta$	<i>espaço livre</i>	28	0	35	3	30	4	36	7
-------------	----------	---------	---------------------	----	---	----	---	----	---	----	---

<i>Index</i>	b1	b2	b4	<i>esp. livre</i>	32	9	39	49	28	0	36	7	0	15	34	45
--------------	----	----	----	-------------------	----	---	----	----	----	---	----	---	---	----	----	----

# Representação dos nós



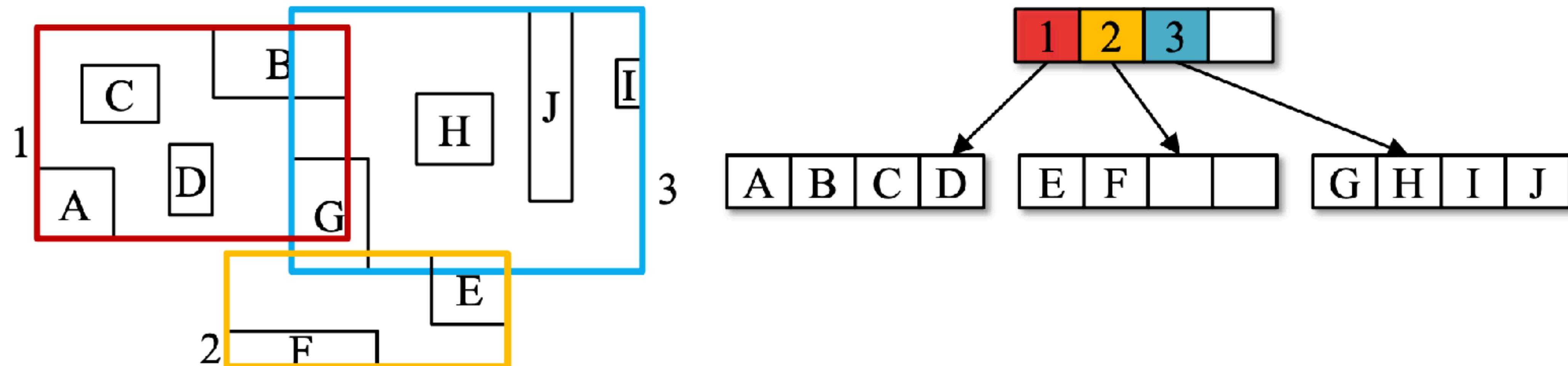
rect

<i>Leaf</i>	α	β	<i>espaço livre</i>	28	0	35	3	30	4	36	7
-------------	---	---	---------------------	----	---	----	---	----	---	----	---

id

<i>Index</i>	b1	b2	b4	<i>esp. livre</i>	32	9	39	49	28	0	36	7	0	15	34	45
--------------	----	----	----	-------------------	----	---	----	----	----	---	----	---	---	----	----	----

# Representação dos nós



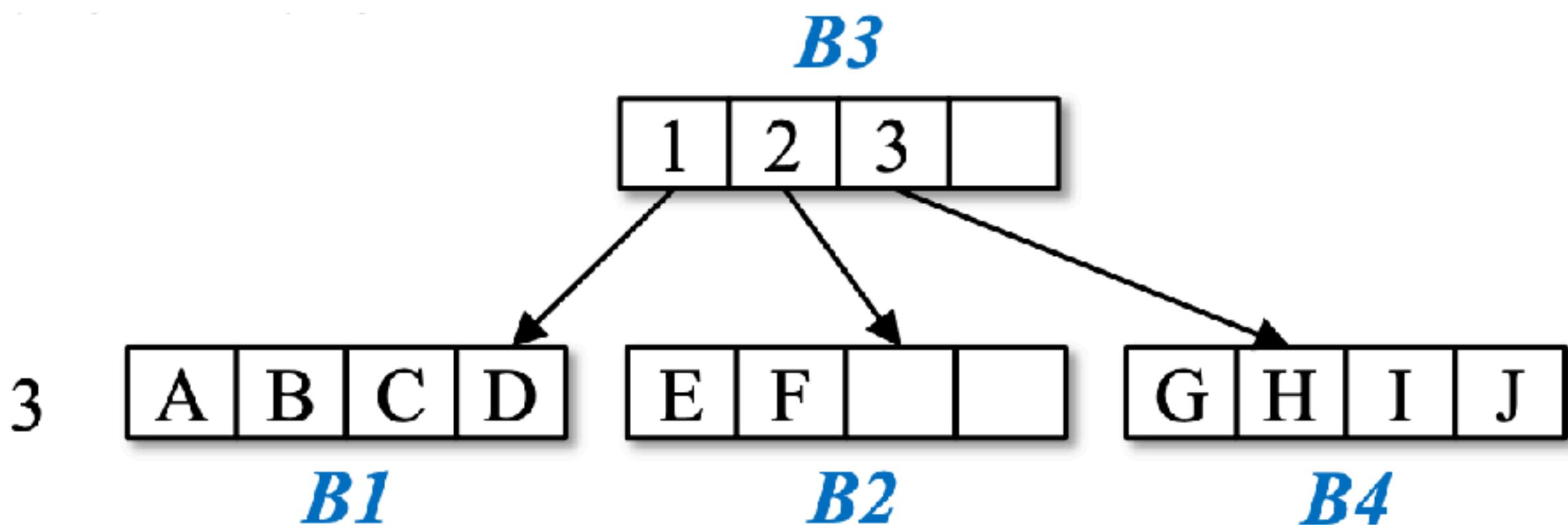
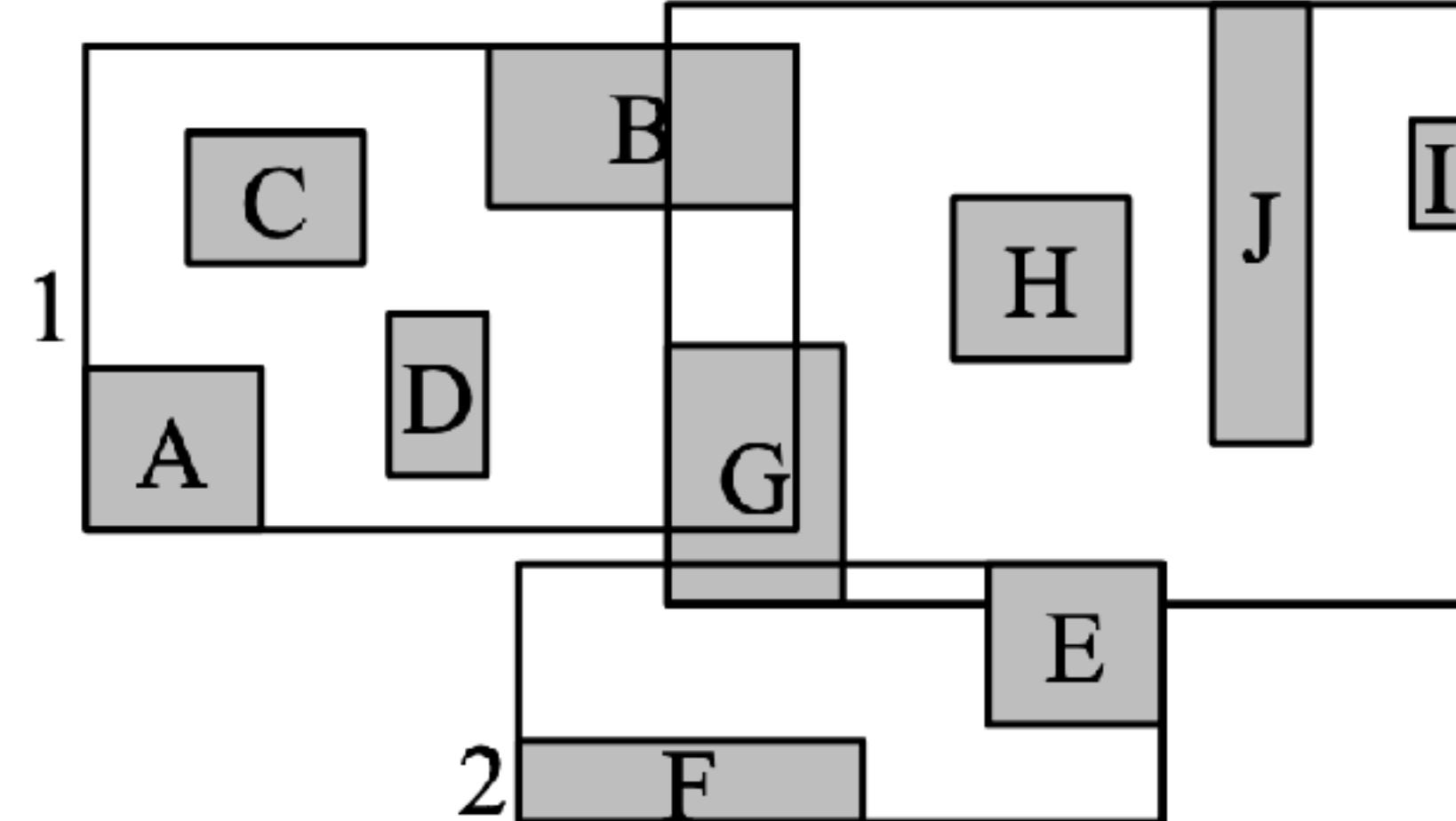
<i>Leaf</i>	a	$\beta$	<i>espaço livre</i>	28	0	35	3	30	4	36	7
-------------	---	---------	---------------------	----	---	----	---	----	---	----	---

PTR

rect

<i>Index</i>	b1	b2	b4	<i>esp. livre</i>	32	9	39	49	28	0	36	7	0	15	34	45
--------------	----	----	----	-------------------	----	---	----	----	----	---	----	---	---	----	----	----

# Representação dos nós



<b>B0</b>	H	b3	512	
-----------	---	----	-----	--

<b>B1</b>	L	$\alpha$	$\beta$	$\gamma$	$\delta$	D	C	B	A
-----------	---	----------	---------	----------	----------	---	---	---	---

<b>B2</b>	L	$\varepsilon$	$\zeta$		F	E
-----------	---	---------------	---------	--	---	---

<b>B3</b>	I	b1	b2	b4		3	2	1
-----------	---	----	----	----	--	---	---	---

<b>B4</b>	L	$\eta$	$\theta$	$\mu$	$\omega$	J	I	H	G
-----------	---	--------	----------	-------	----------	---	---	---	---

# Consulta



Point Query



Window Query



Region Query



Adjacency Query





- **Topológicos:** encontra todos os objetos que interceptam um dado objeto
- **Direcionais:** encontra todos os objetos que, por exemplo, estão ao norte de um dado objeto
- **Distância:** encontra todos os objetos que estão a menos que uma distância  $d$  de um dado objeto (range query) ou os  $k$  objetos mais próximos de um dado objeto ( $k$ -nearest-neighbors query).

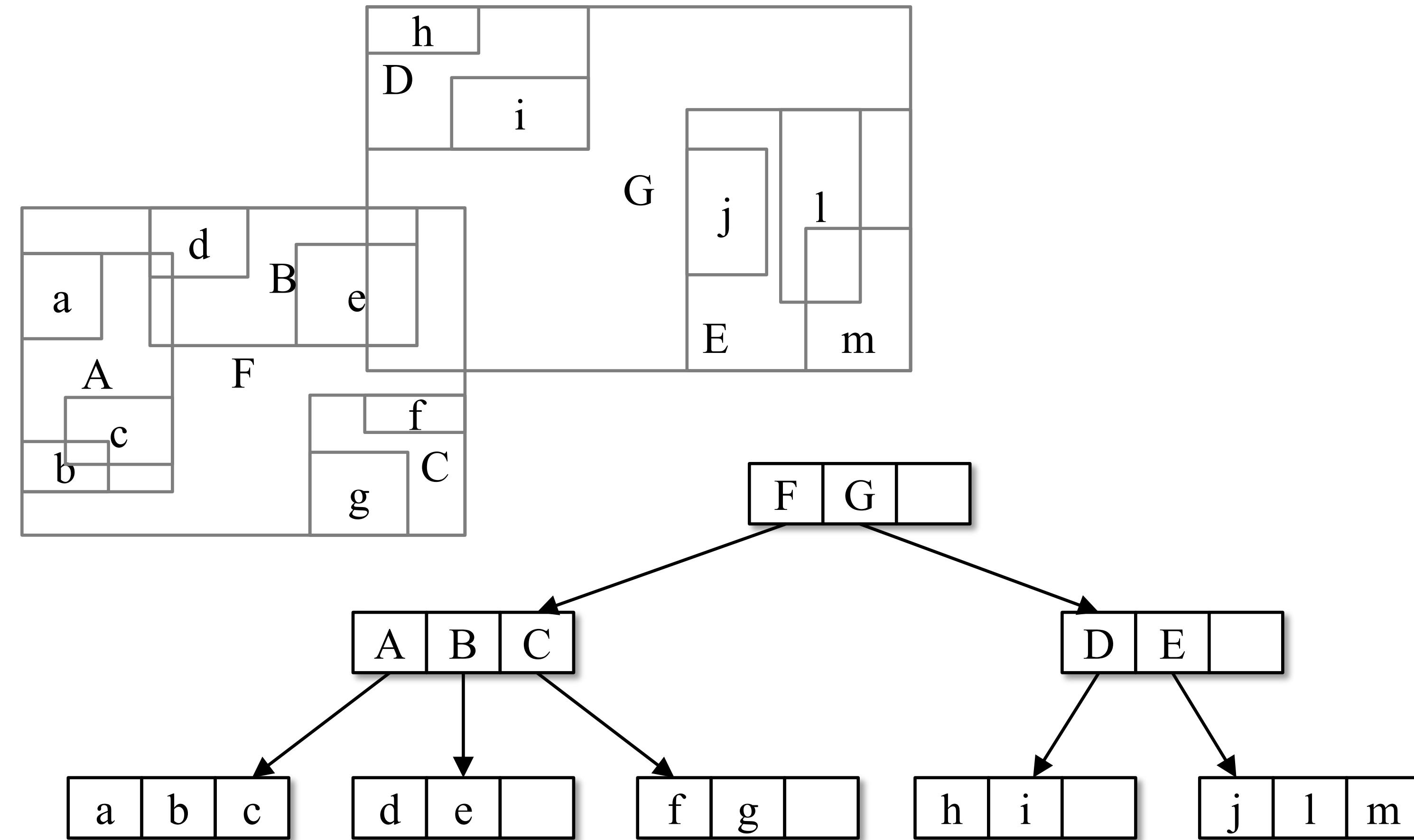
# Consulta - Algoritmo (Range Query)



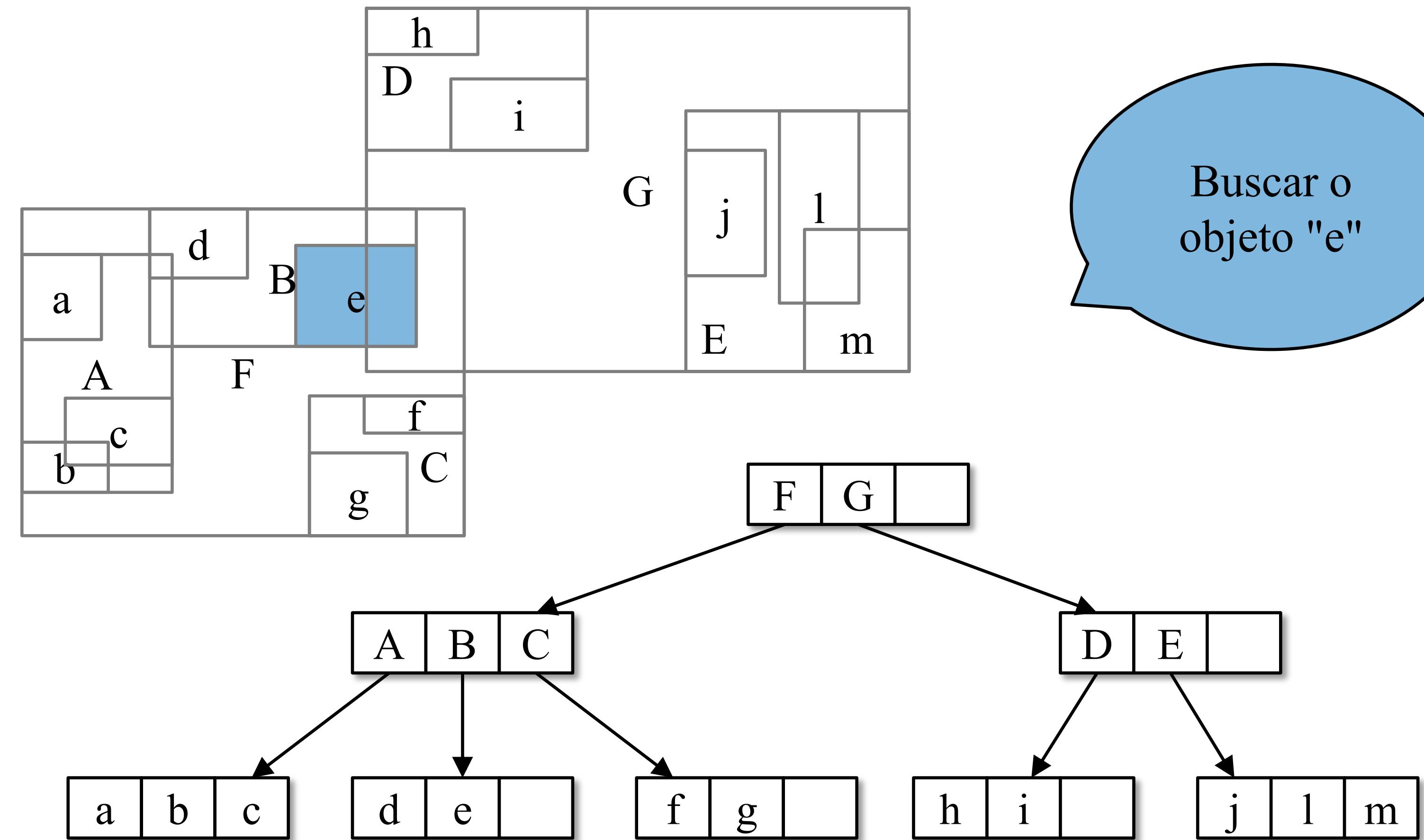
● **Algoritmos:** Encontra todos os objetos interceptados pelo retângulo de busca Q. Devolve um conjunto S de **objetos candidatos**.

- Para todas as entradas de um nó índice, a partir da raiz: Verifica se existe sobreposição
  - Se sim, verifica a respectiva sub-árvore
- Se é um nó folha:
  - Verifica todas as entradas que interceptam Q
    - Adiciona no conjunto resposta S

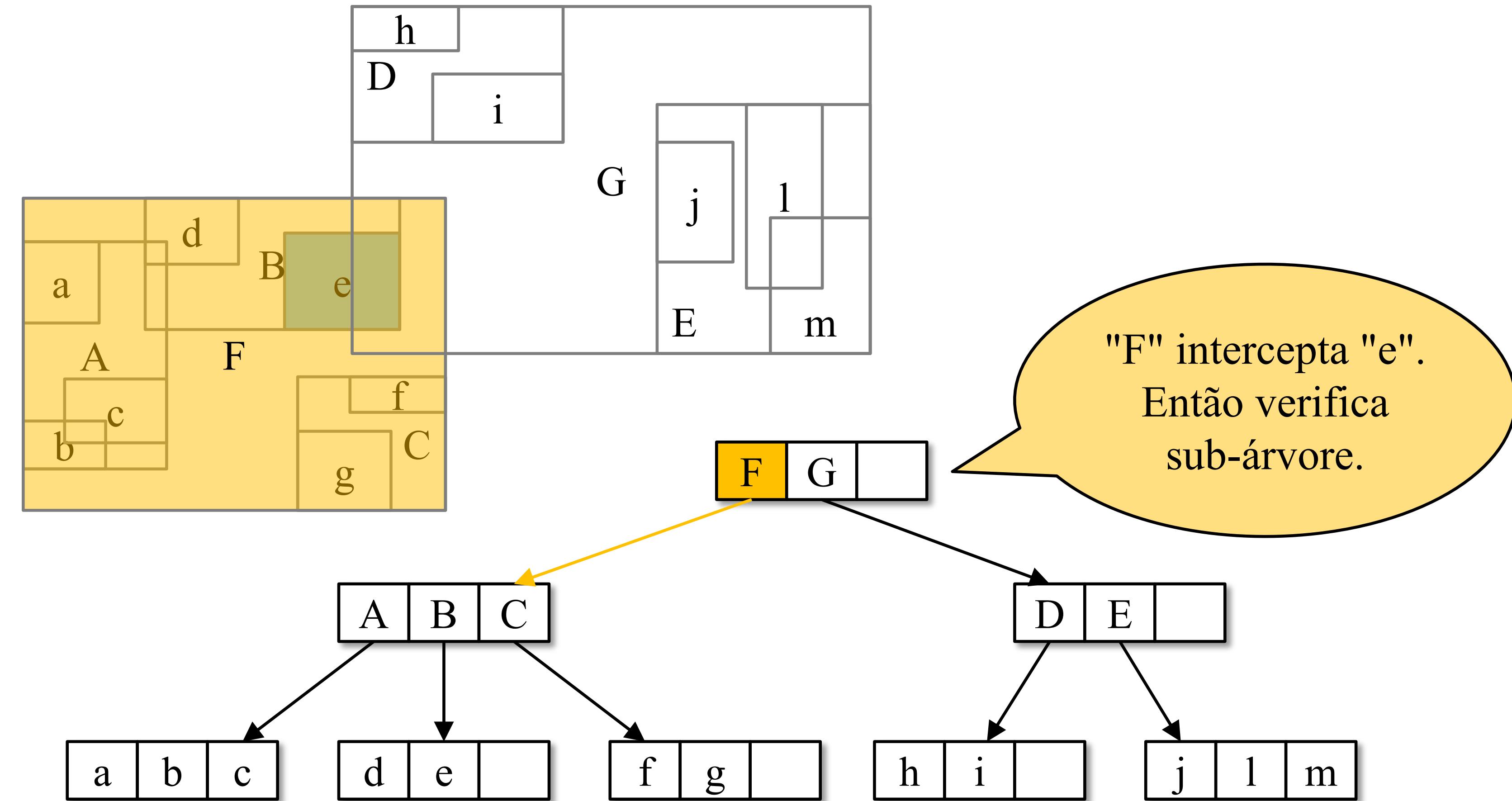
# Exemplo



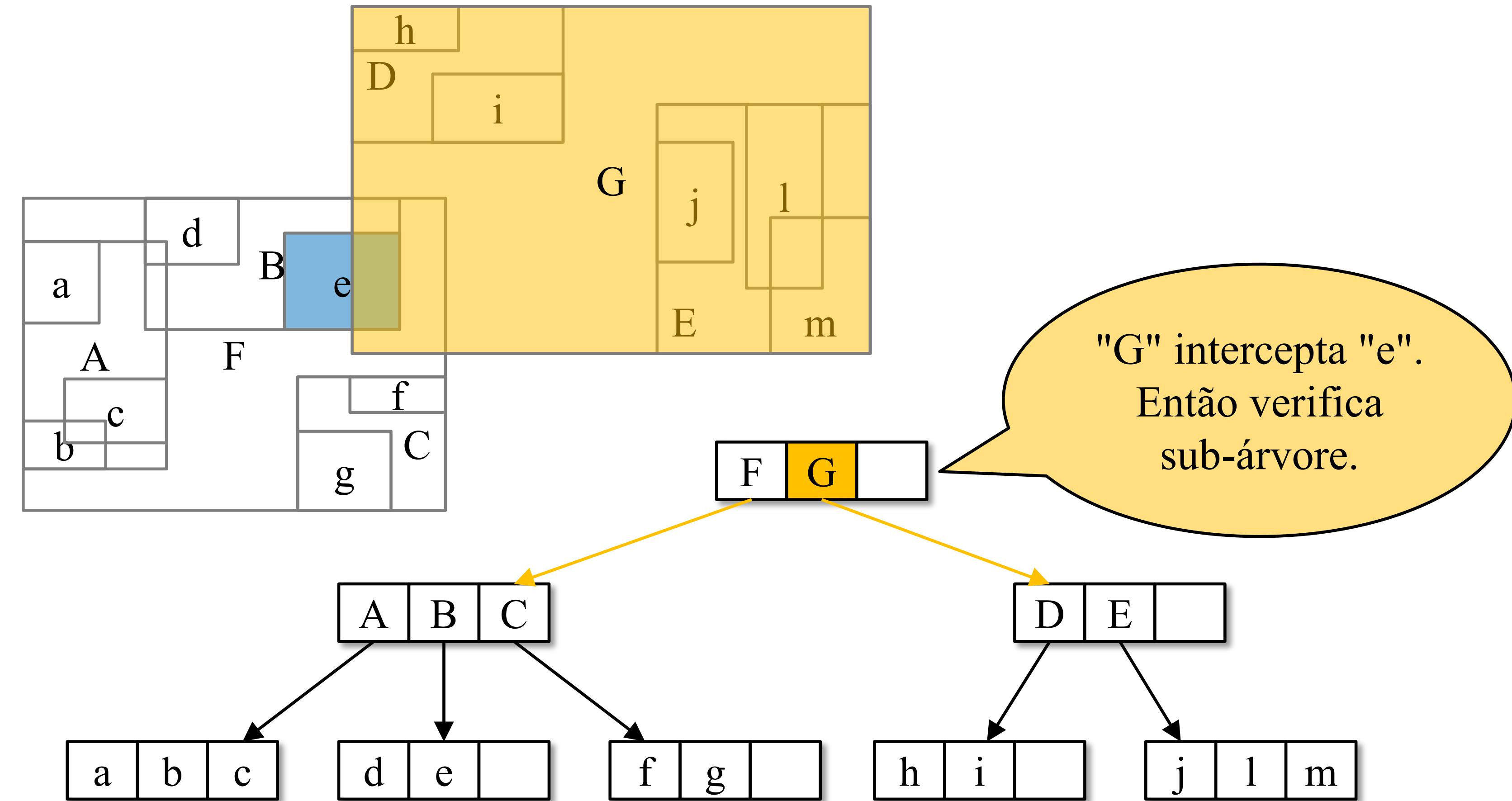
# Exemplo



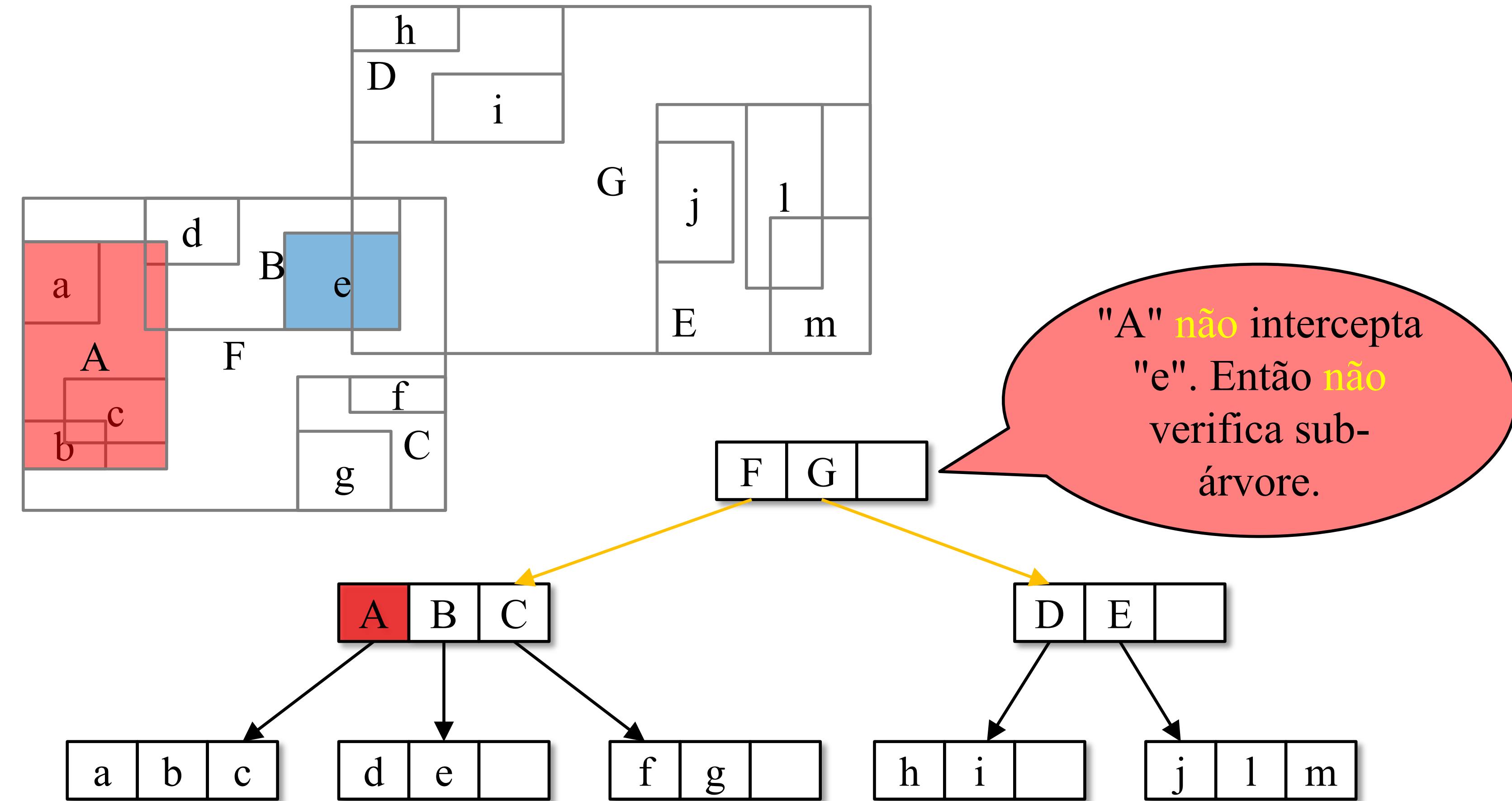
# Exemplo



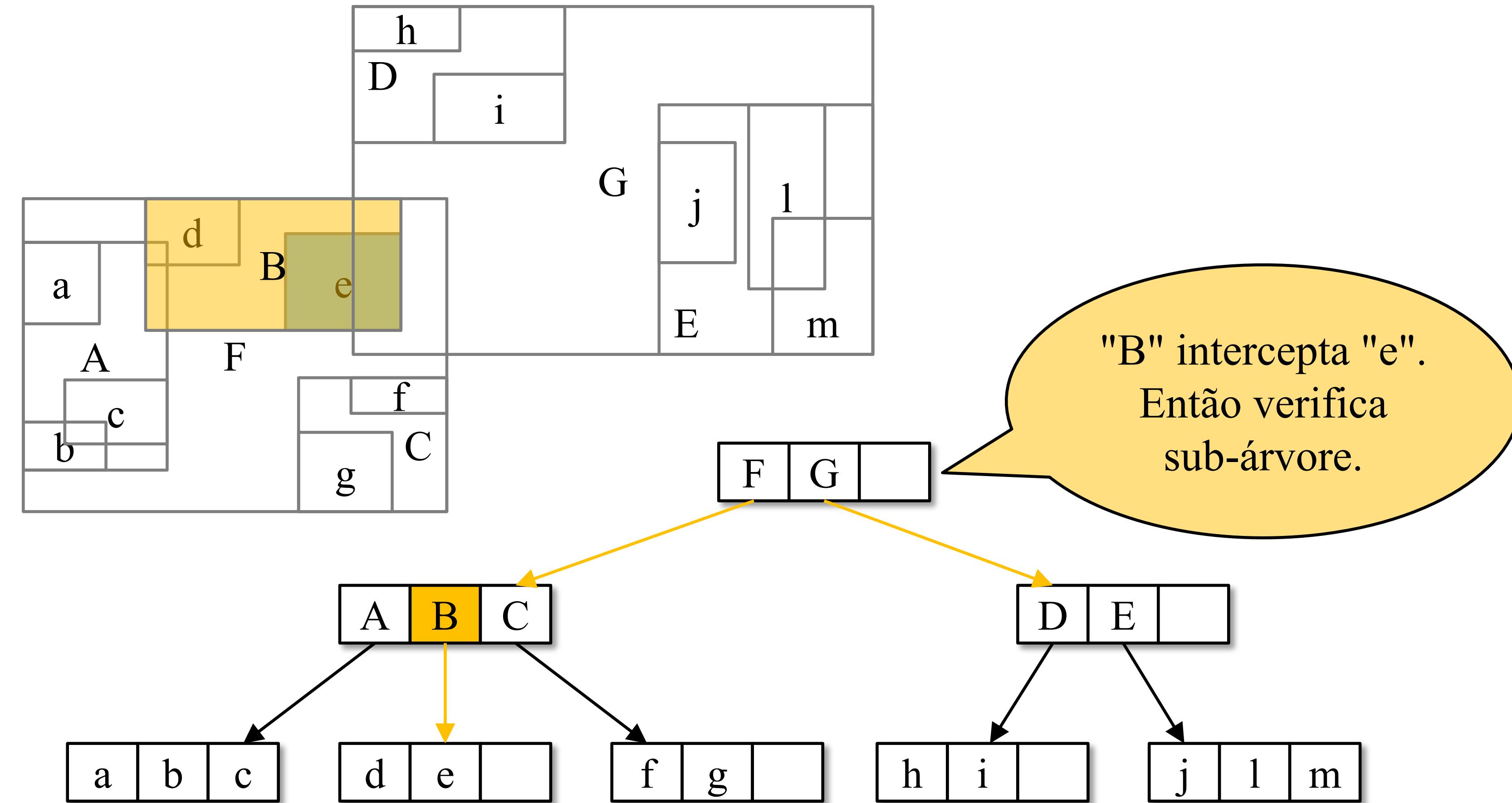
# Exemplo



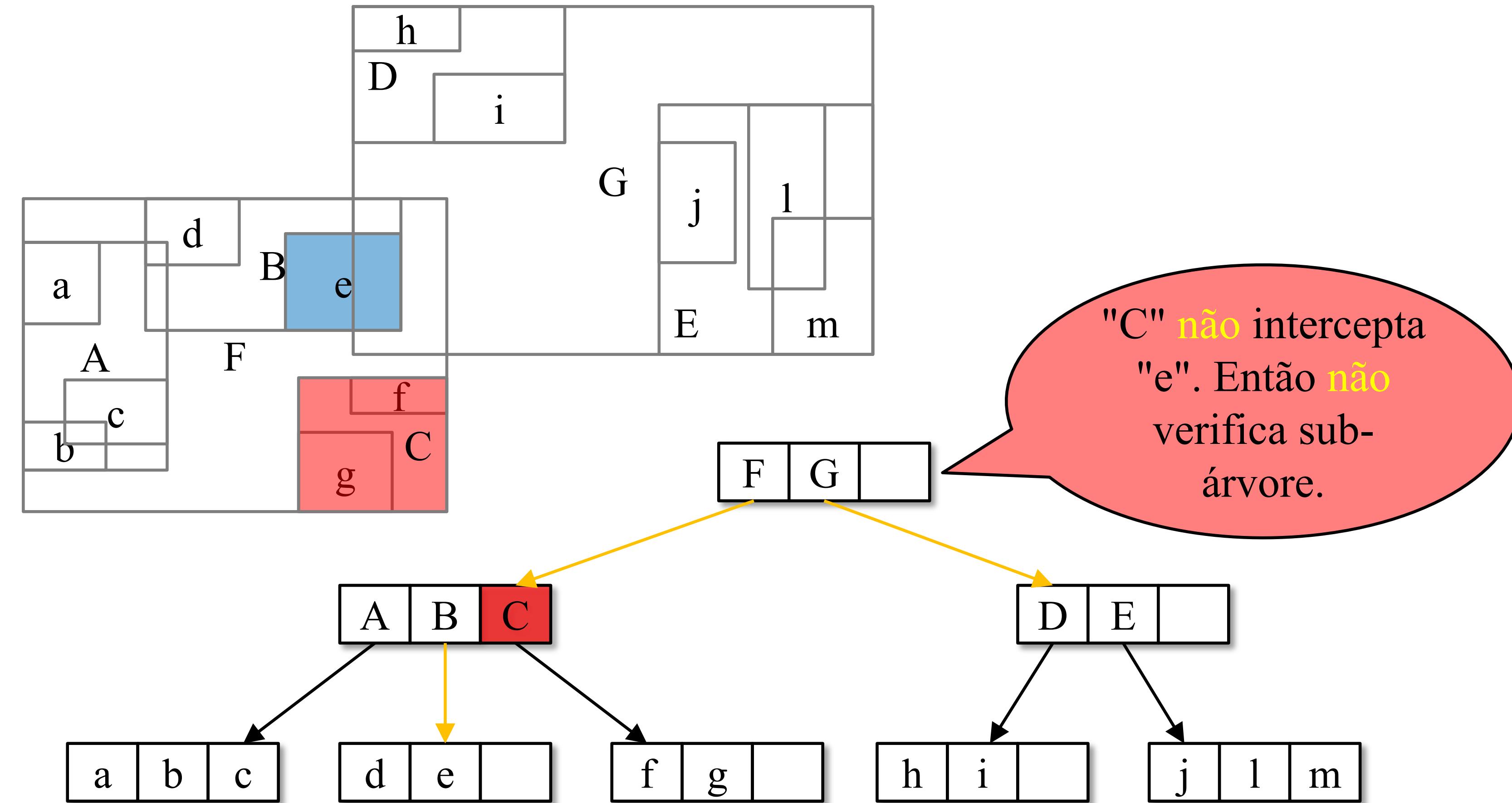
# Exemplo



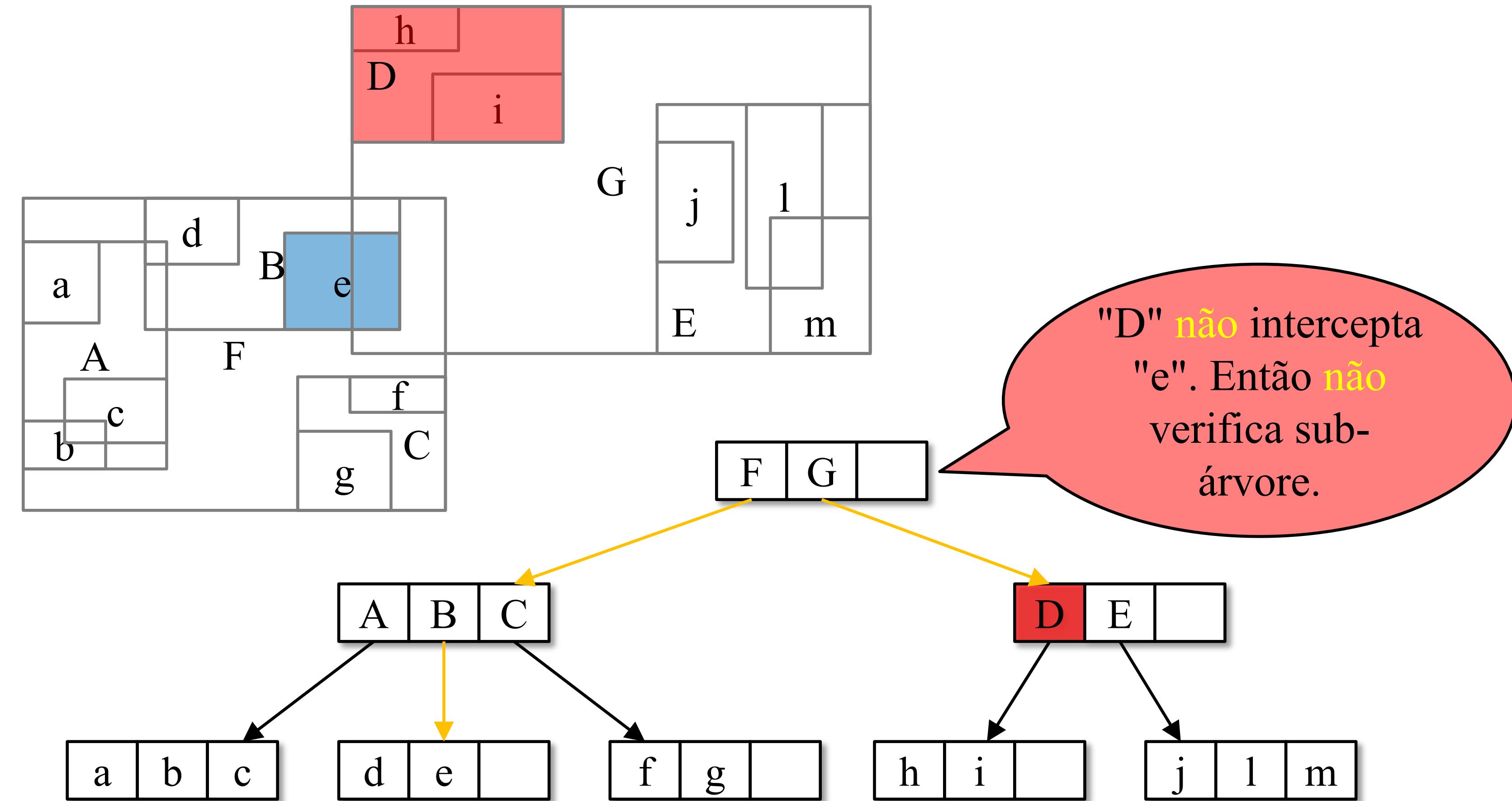
# Exemplo



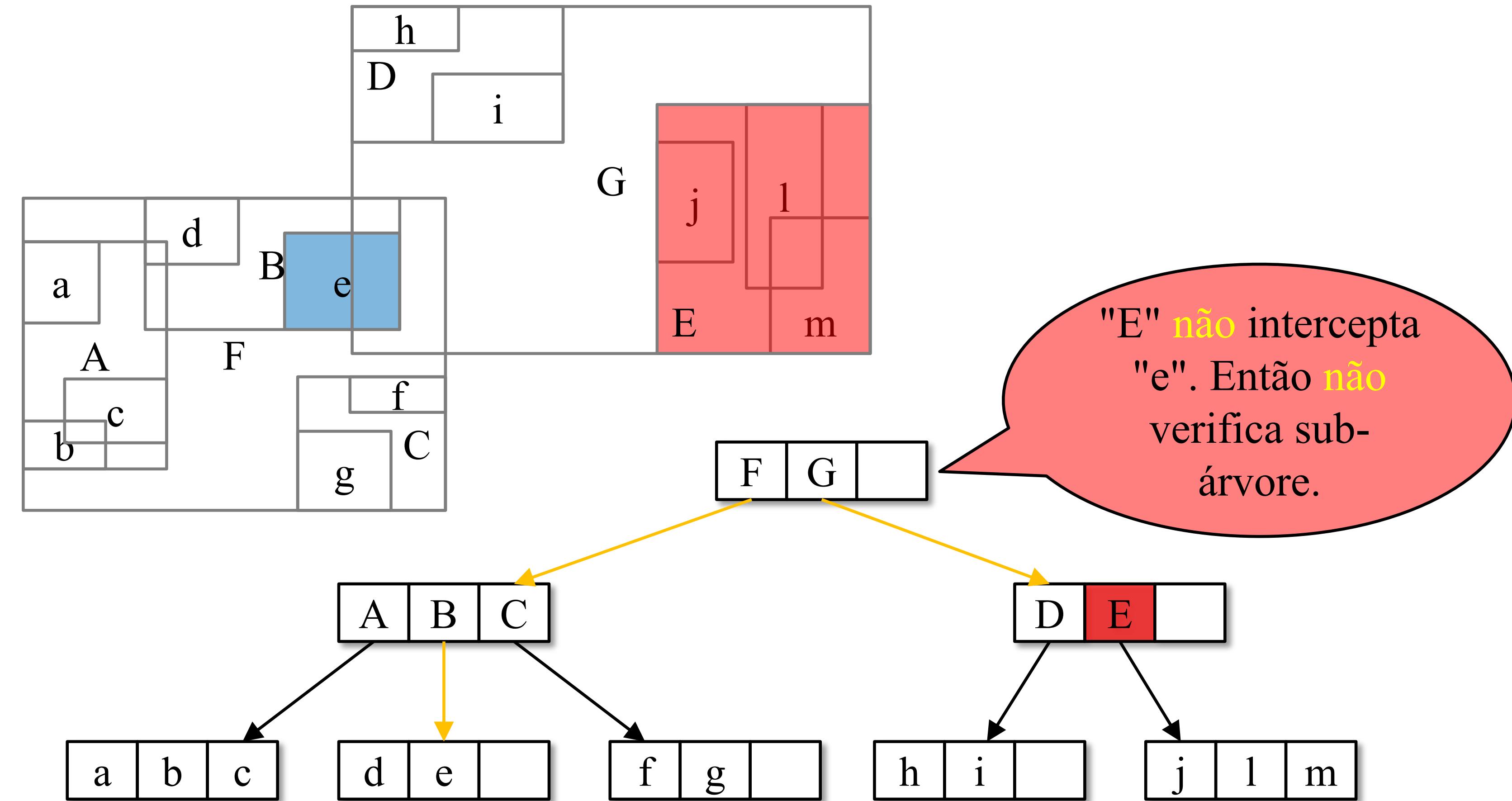
# Exemplo



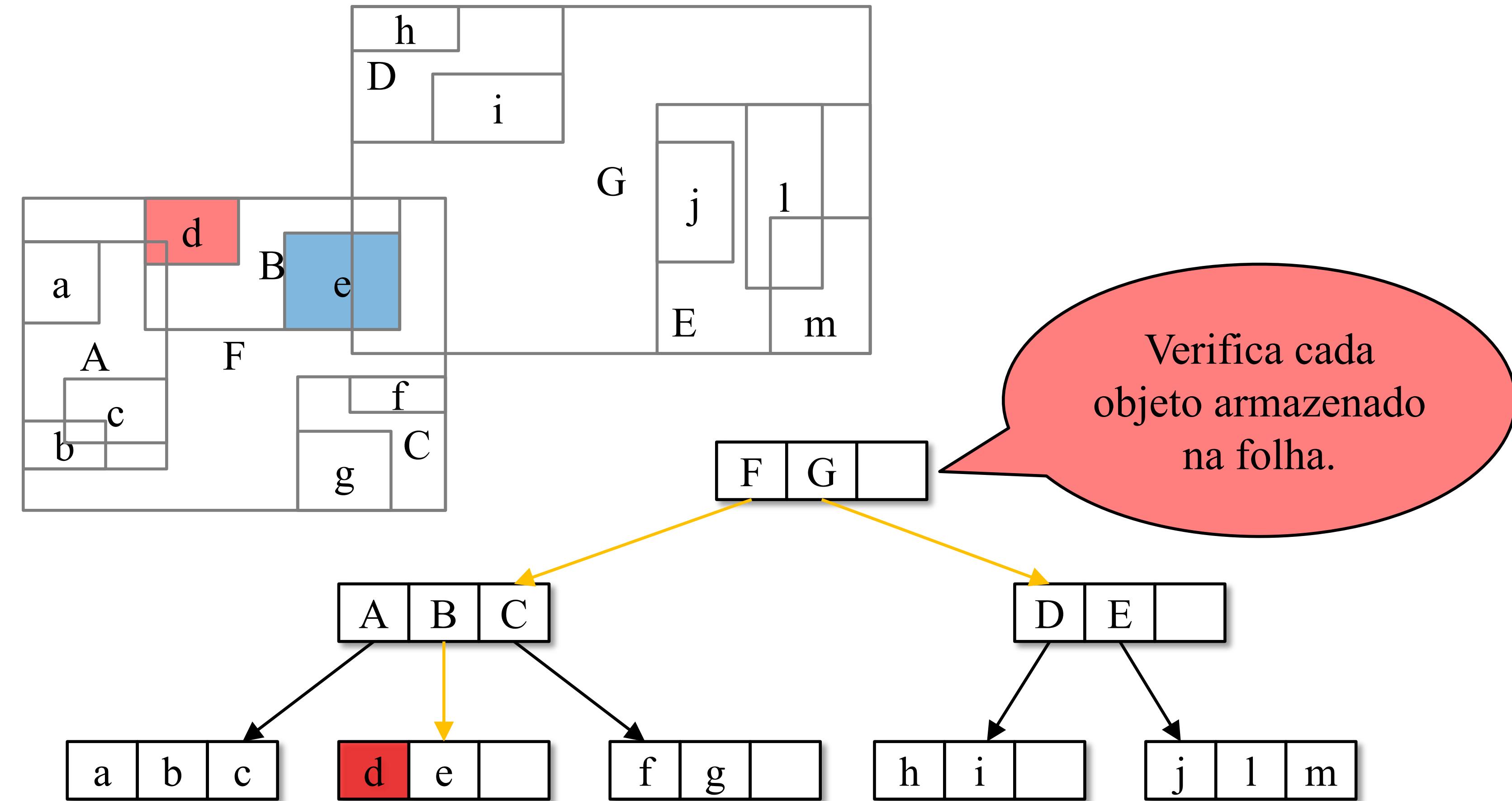
# Exemplo



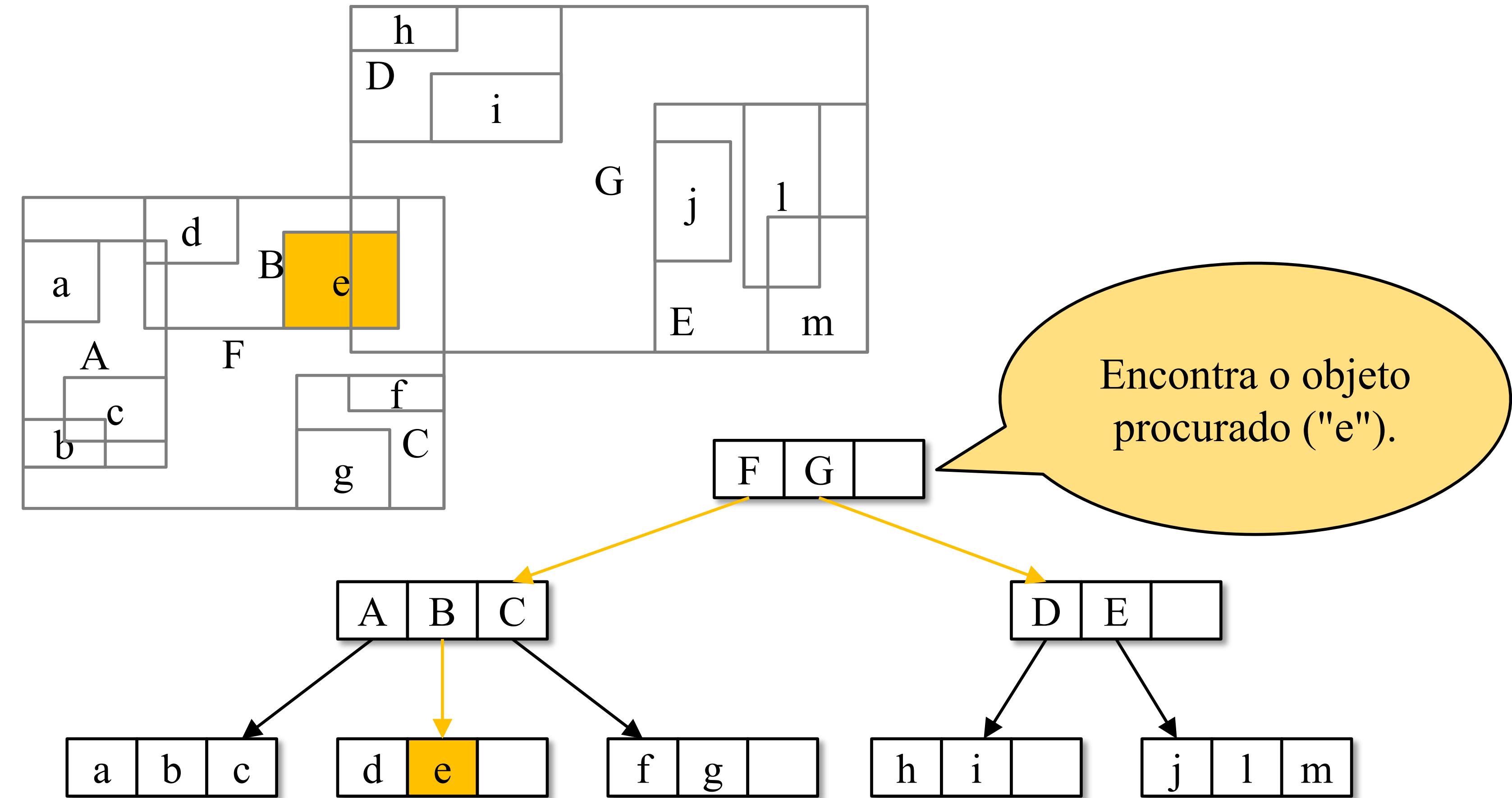
# Exemplo



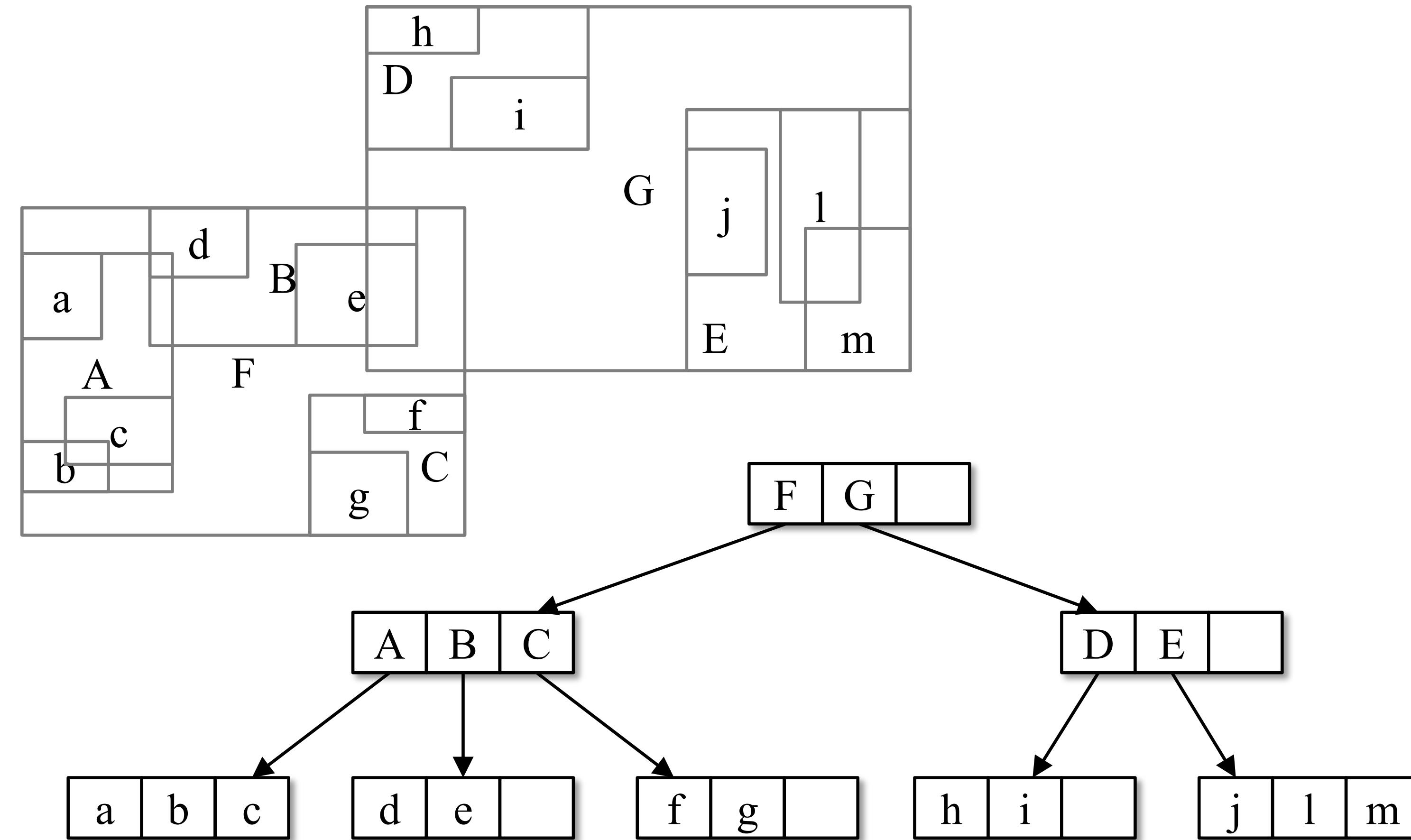
# Exemplo



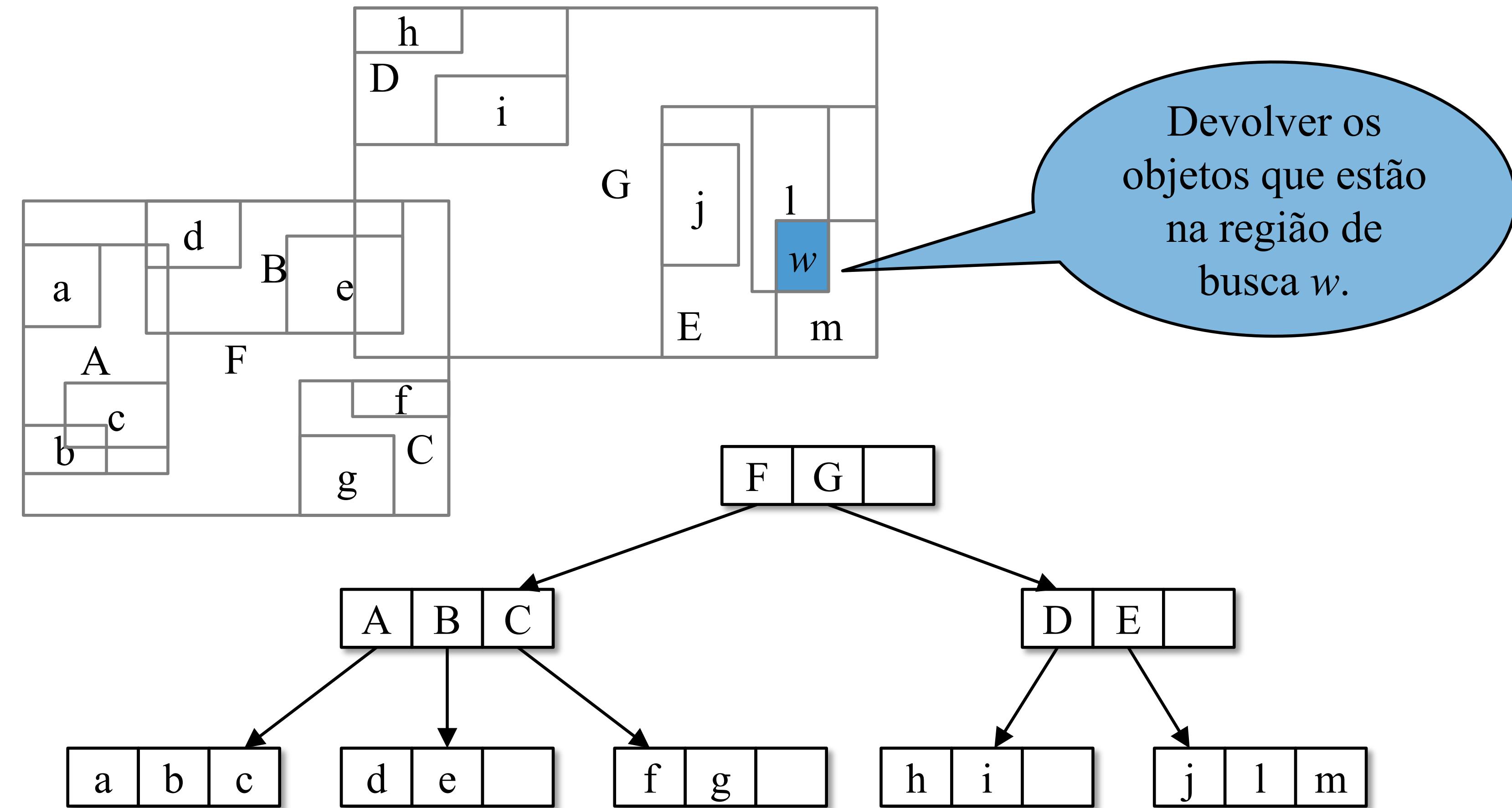
# Exemplo



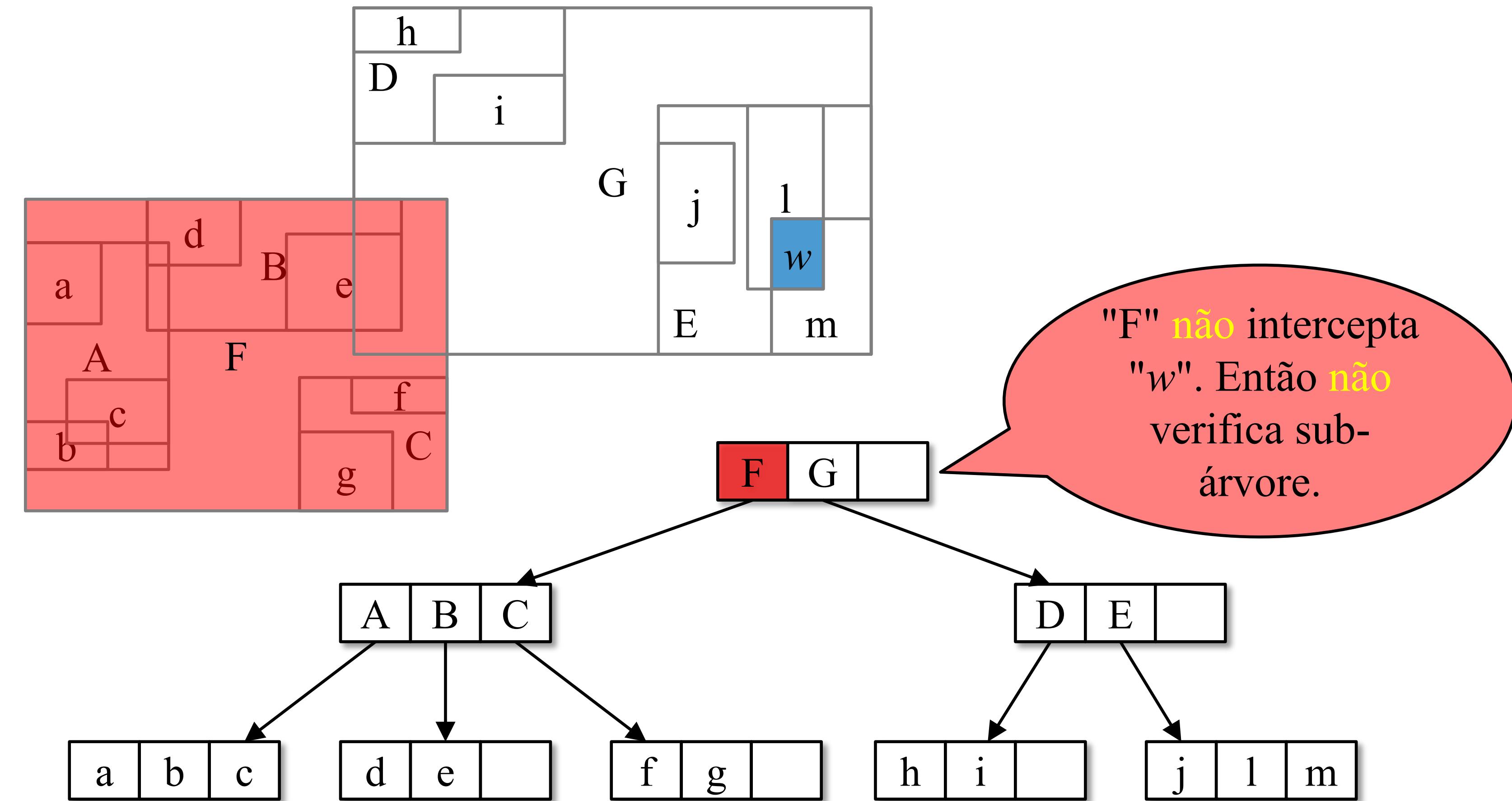
# Exemplo 2



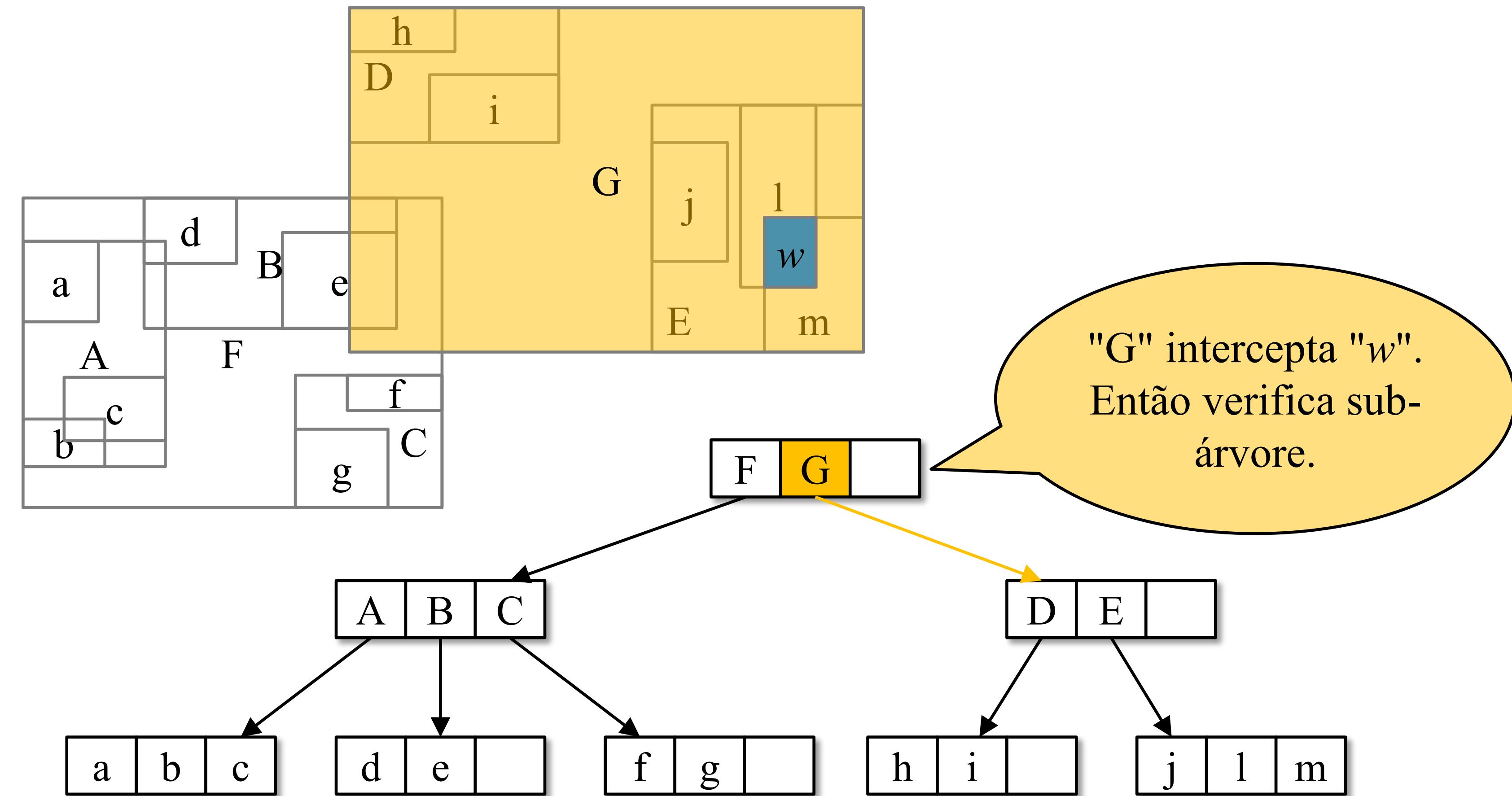
# Exemplo 2



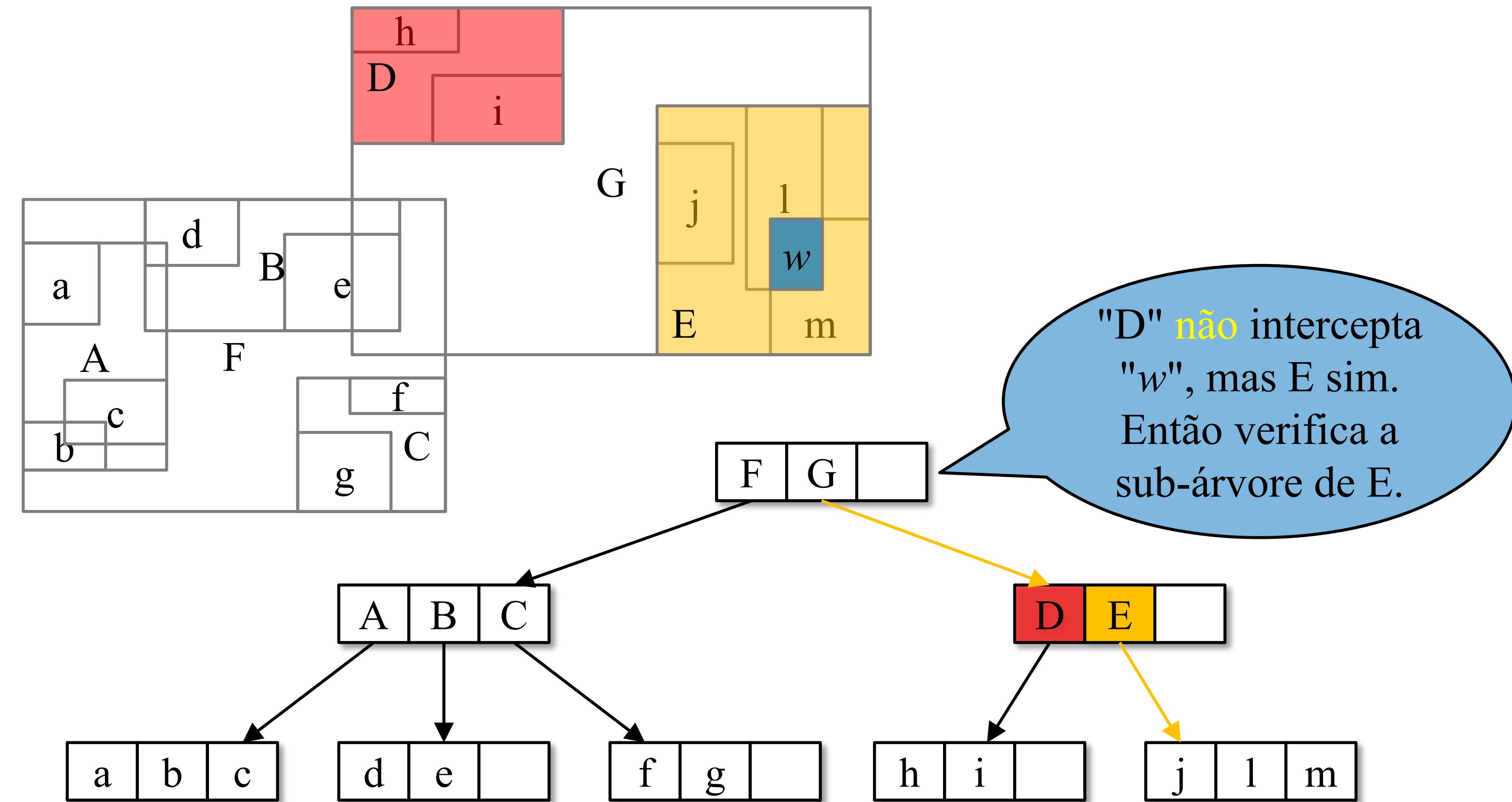
# Exemplo 2



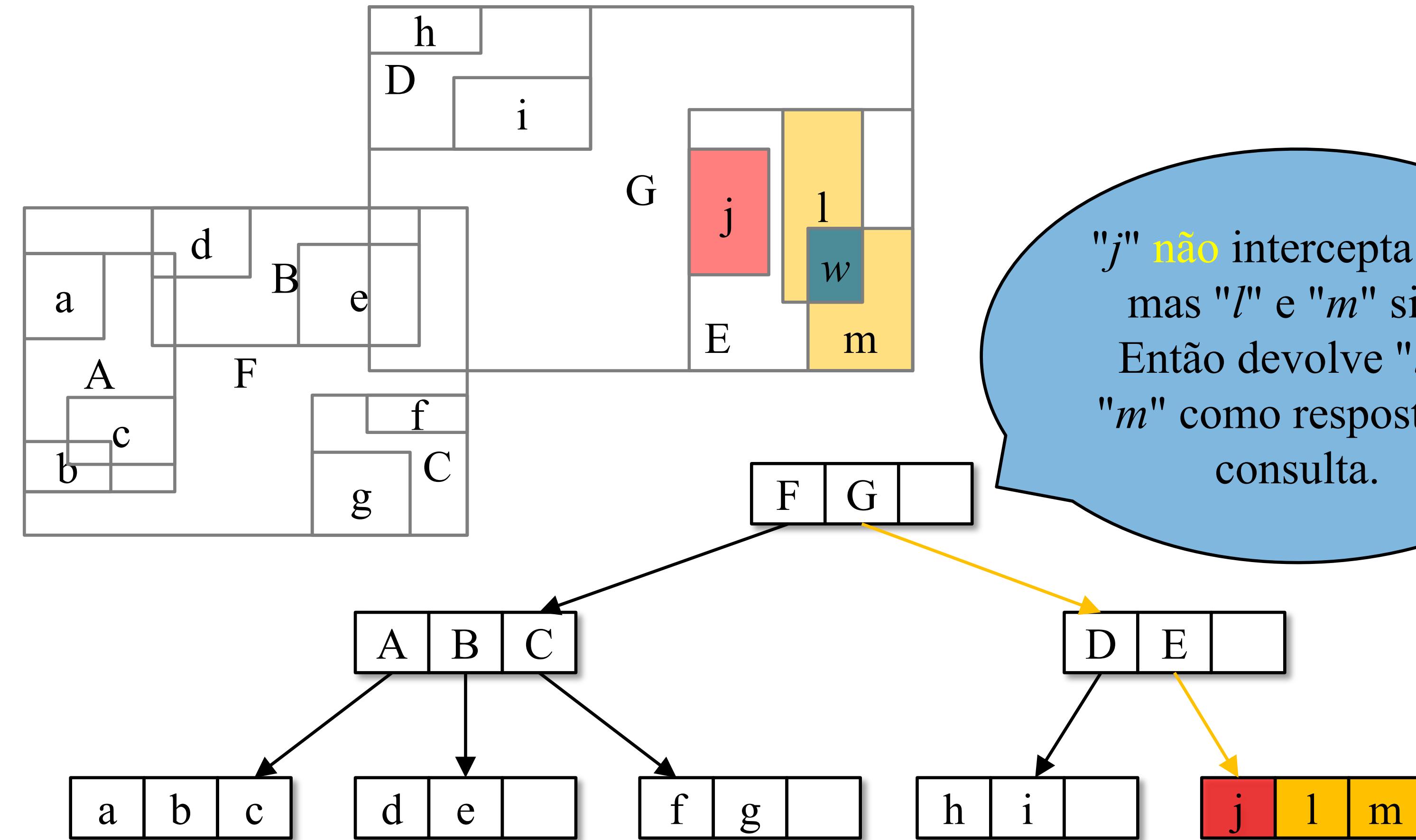
# Exemplo 2



# Exemplo 2



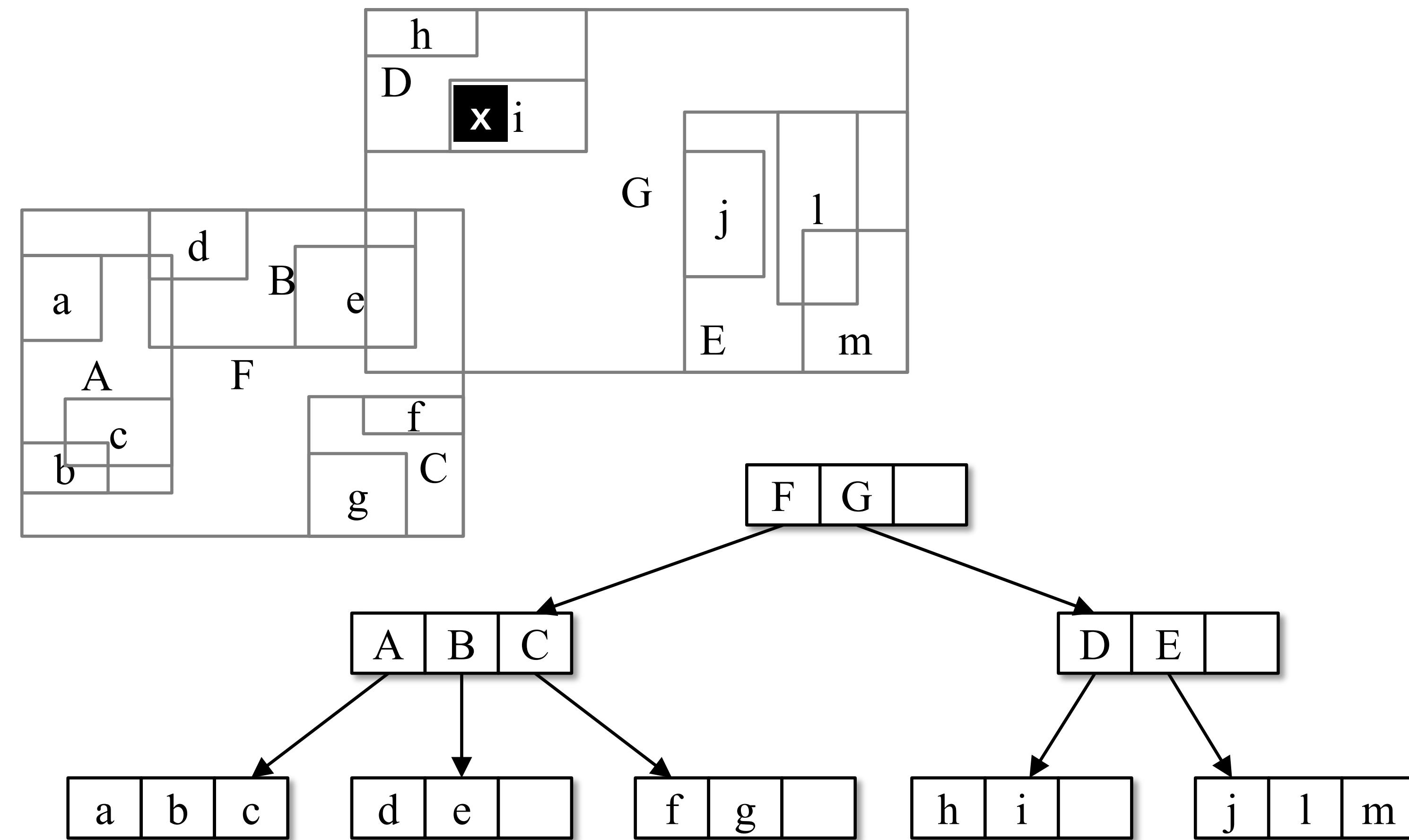
# Exemplo 2



# Exercício



Faça passo a passo a busca do objeto x

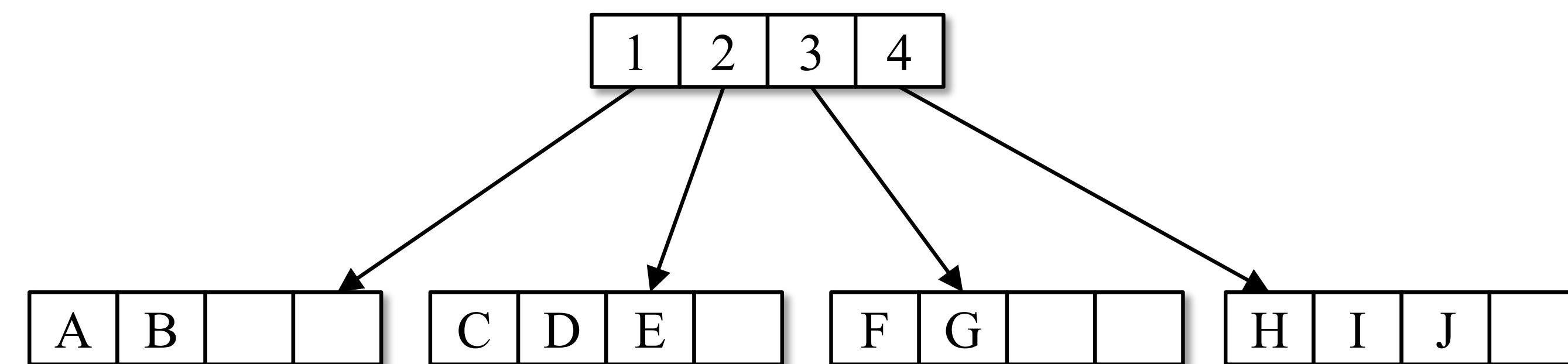
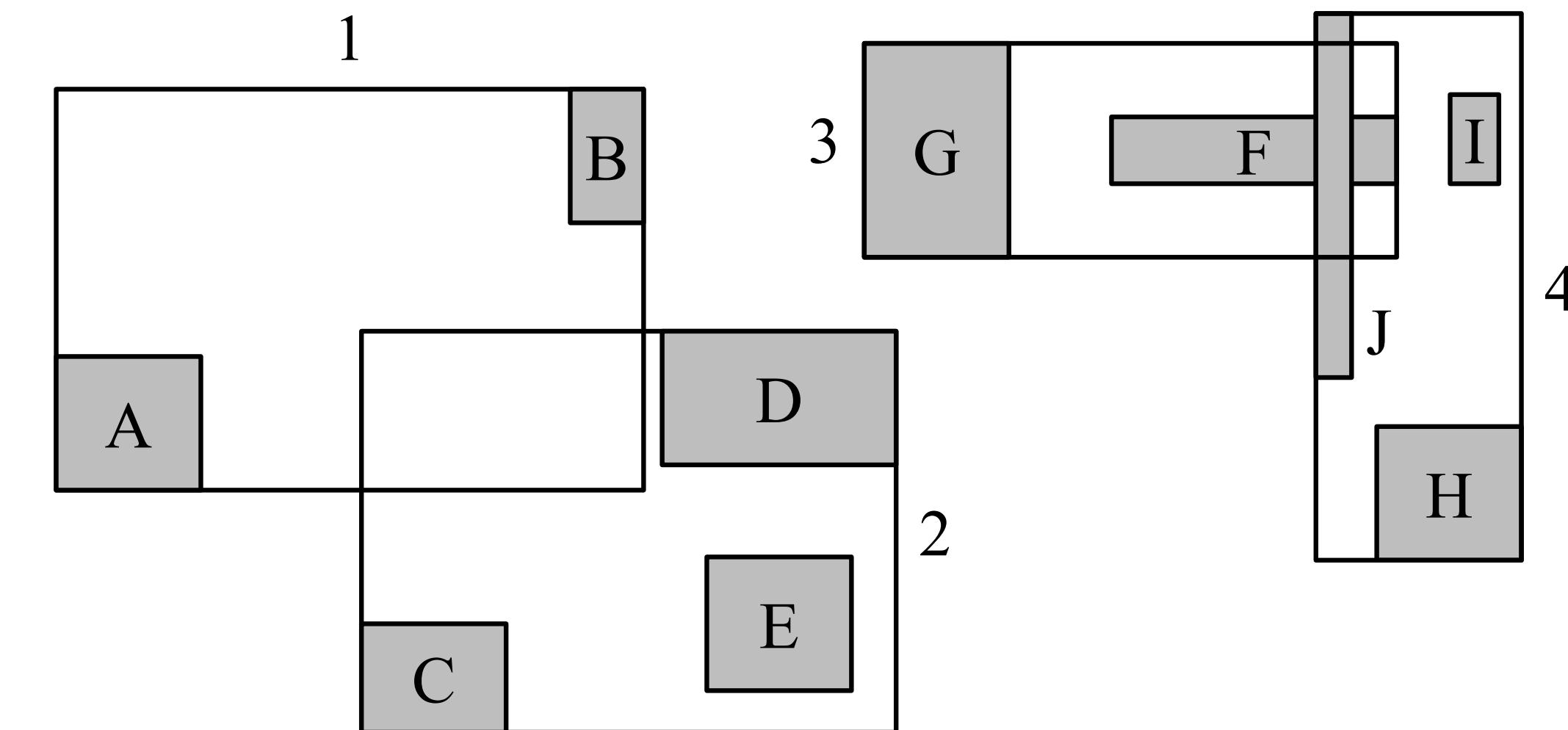


# Inserção

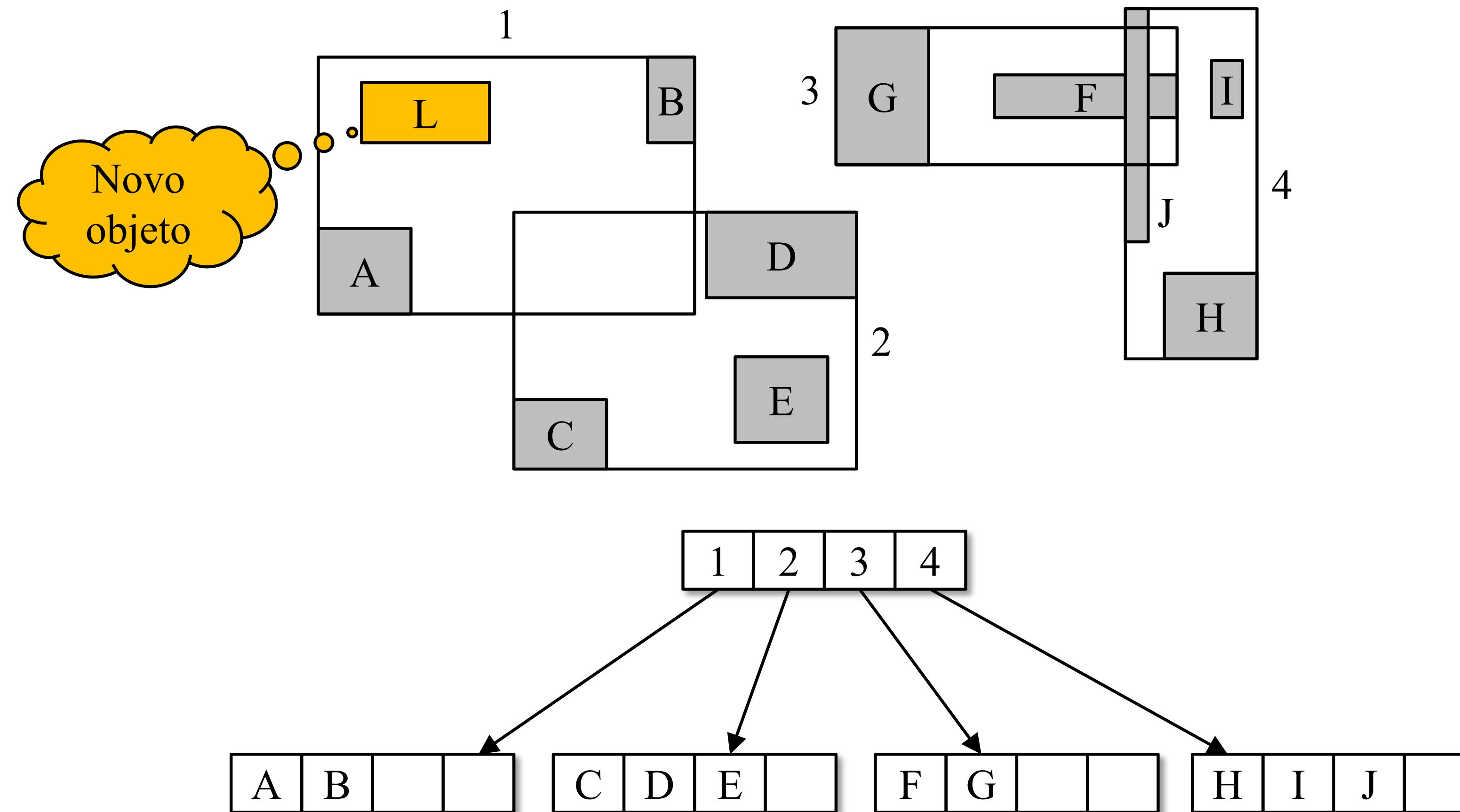


- Percorrer a árvore, a partir do nó raiz, até o nó folha F **mais apropriado**
- A cada nível, escolher a entrada cujo **rect** necessita do **menor aumento de área**. Resolver empates selecionando o de menor área.
- Se o nó folha F contém espaço suficiente, inserir a nova entrada em F e **parar** o processo de inserção. **Caso contrário**, dividir a folha F em F1 e F2.
  - Ajustar a entrada de F no seu nó pai P de modo que seu **rect** cubra apenas F1
  - Adicionar uma entrada em P para F2. Este passo pode fazer o nó P pode **splitar recursivamente**
- Propagar as alterações para os níveis superiores.

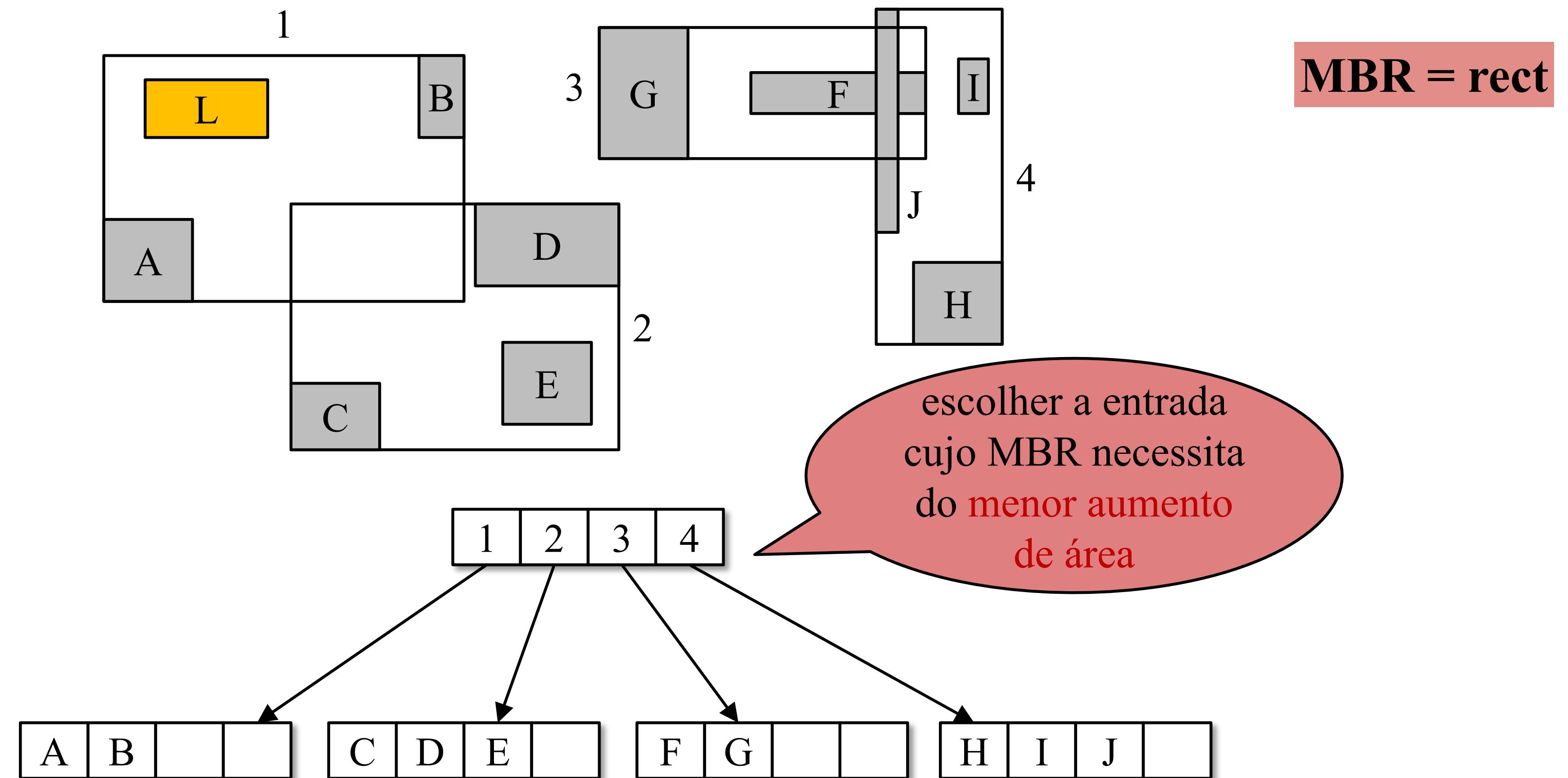
# Inserção - Exemplo: R(2, 4)



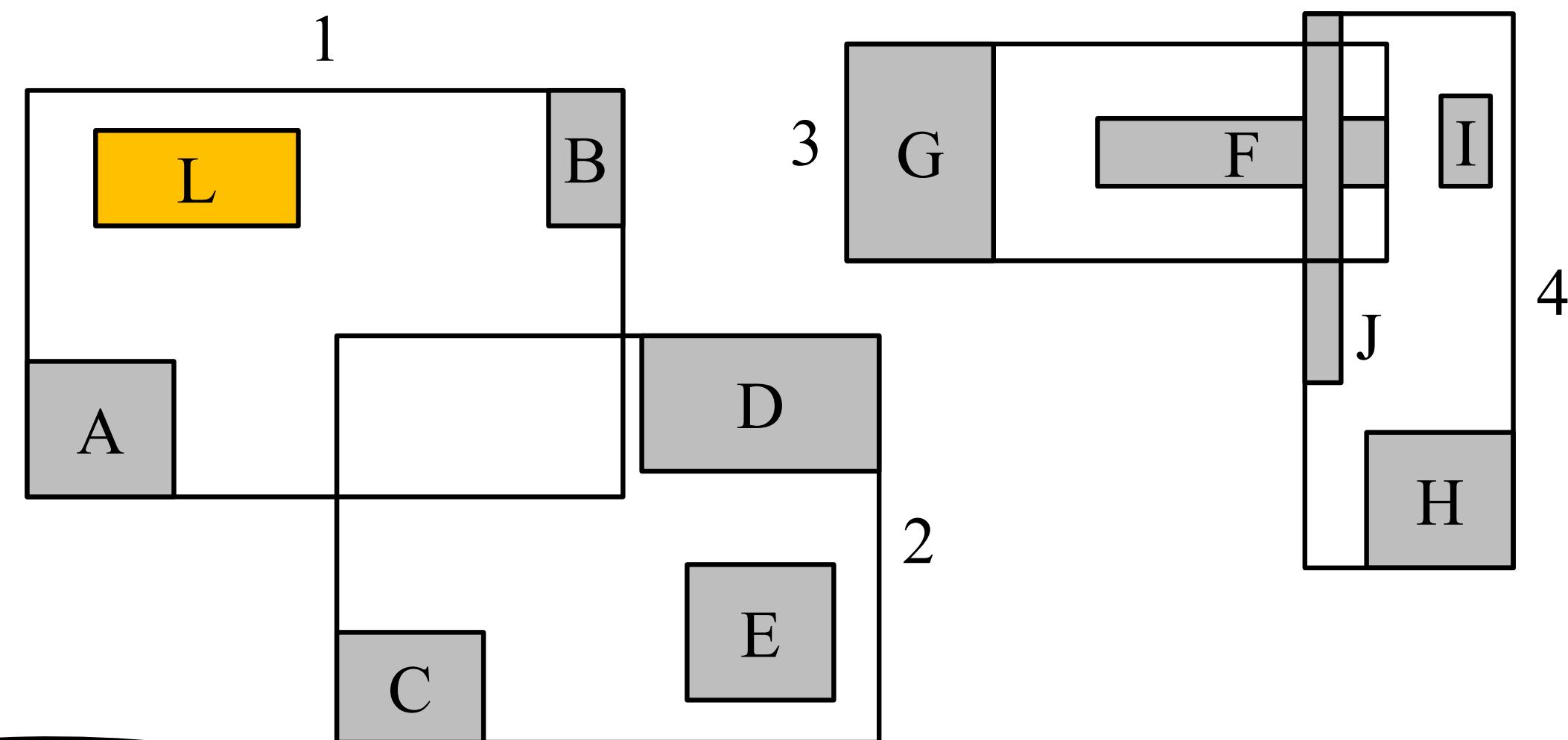
# Inserção - Exemplo: R(2, 4)



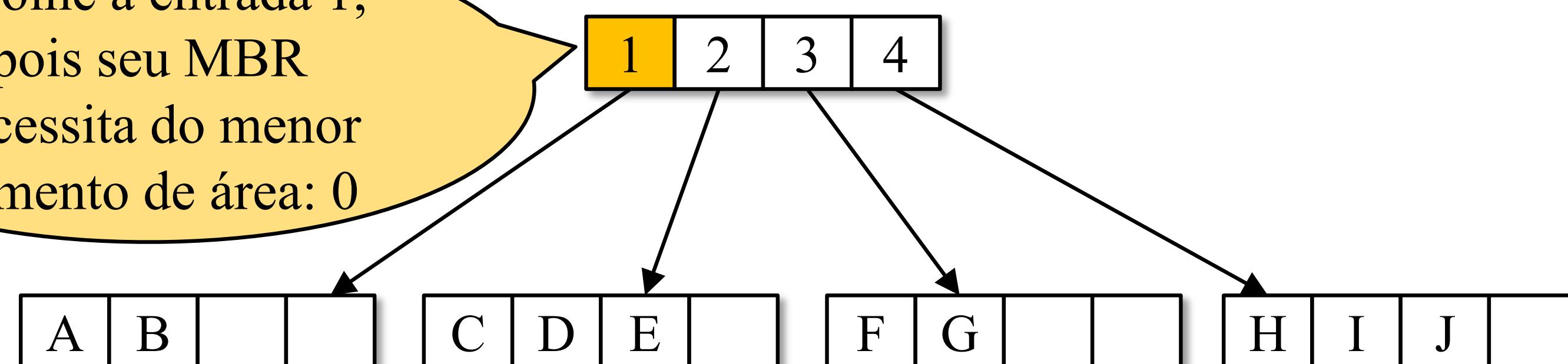
# Inserção - Exemplo: R(2, 4)



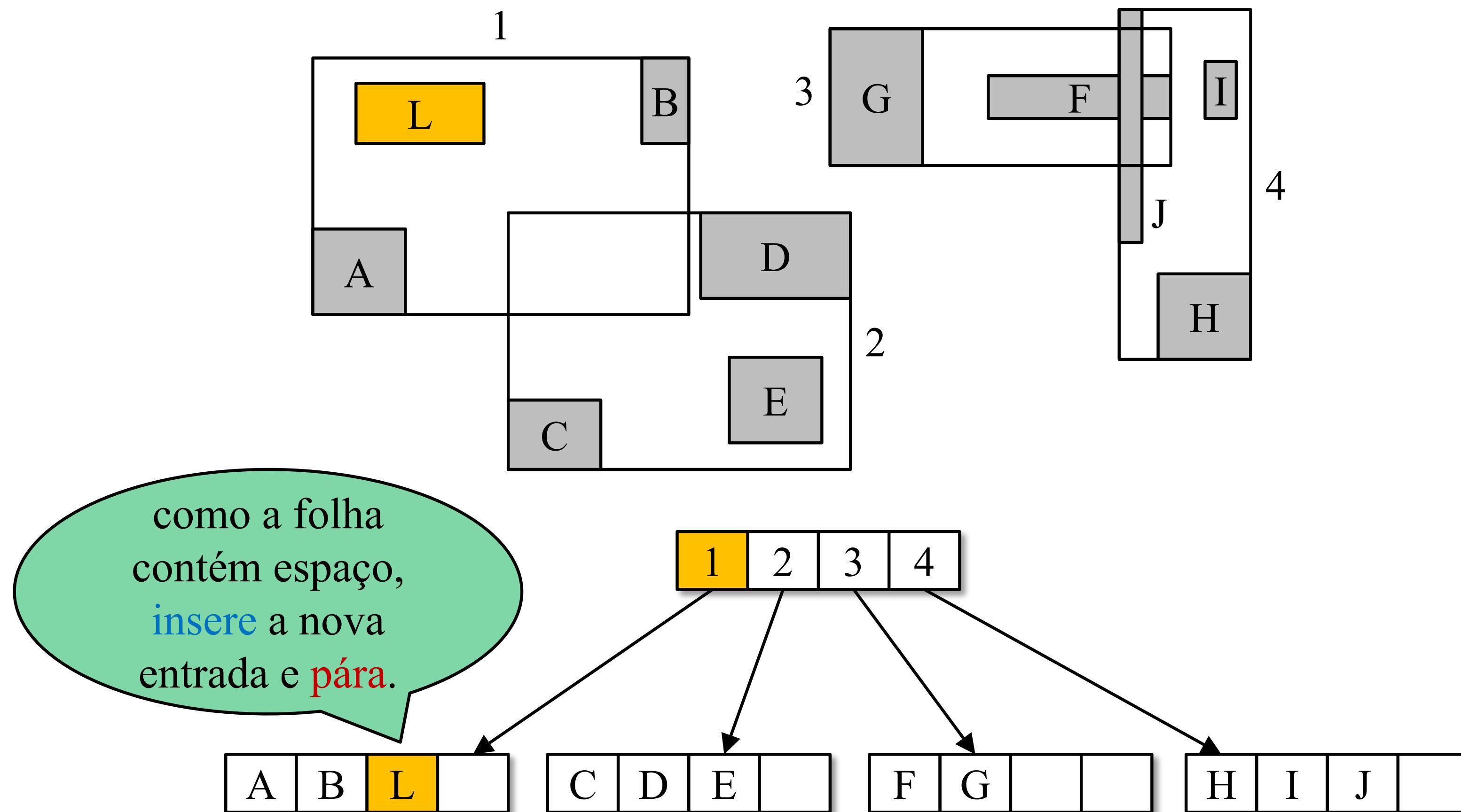
# Inserção - Exemplo: R(2, 4)



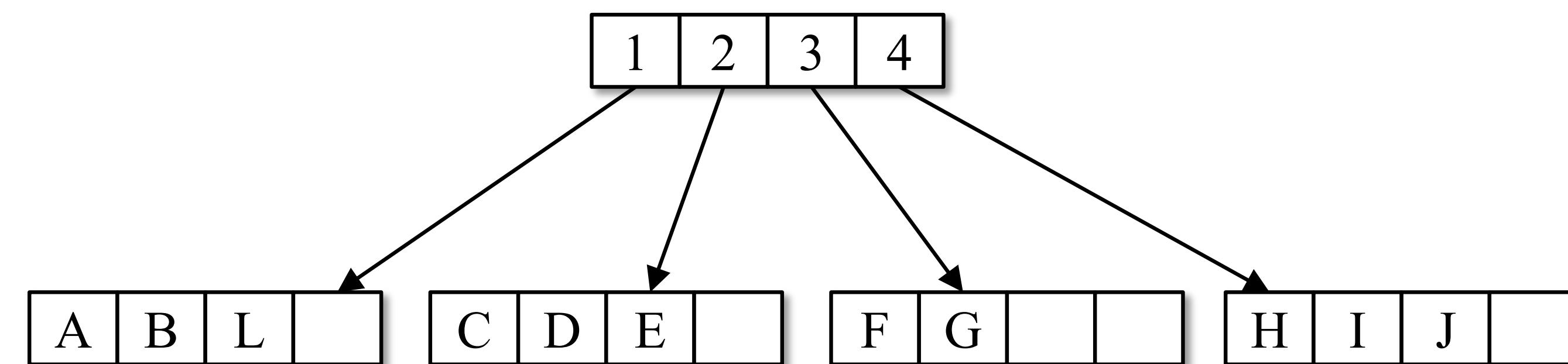
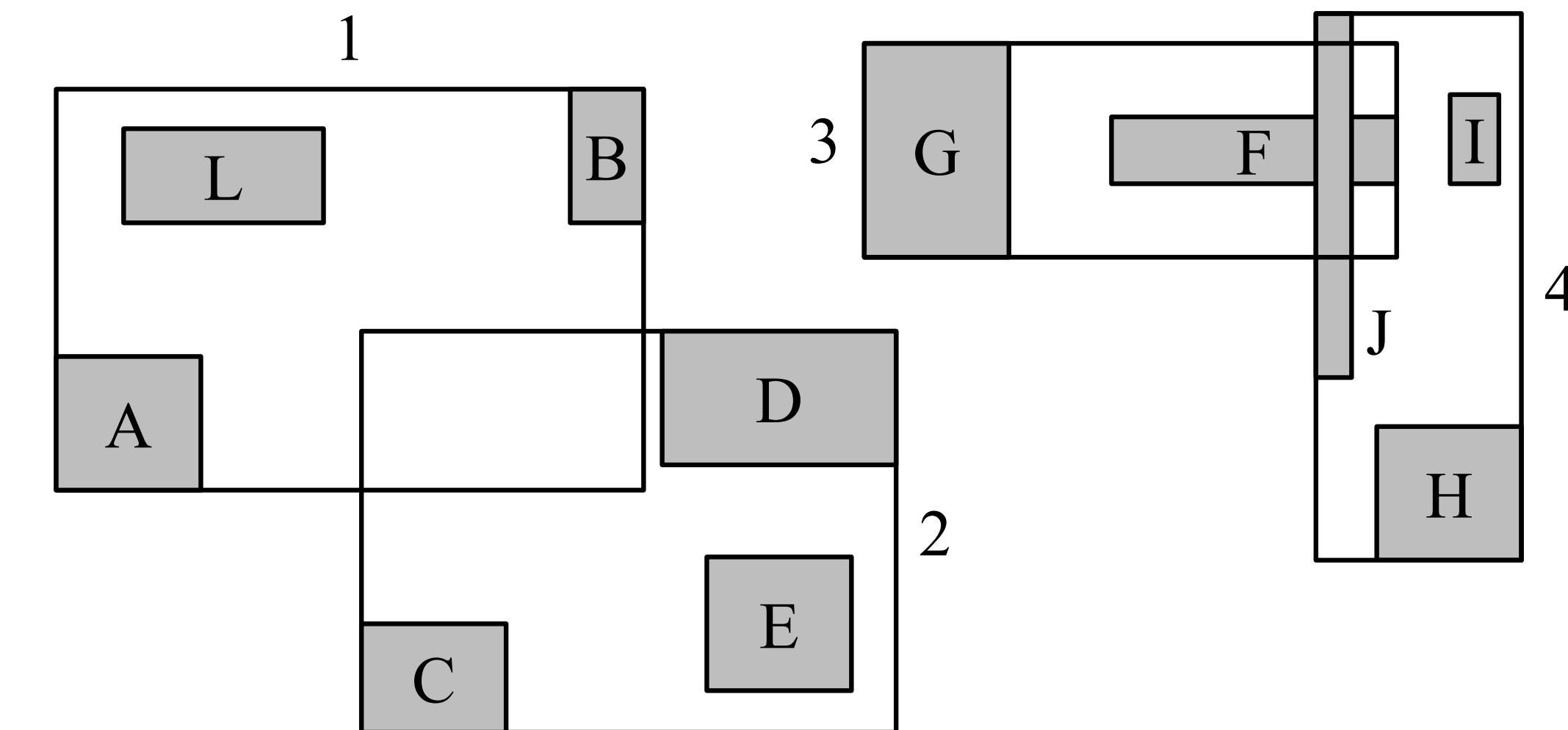
escolhe a entrada 1,  
pois seu MBR  
necessita do menor  
aumento de área: 0



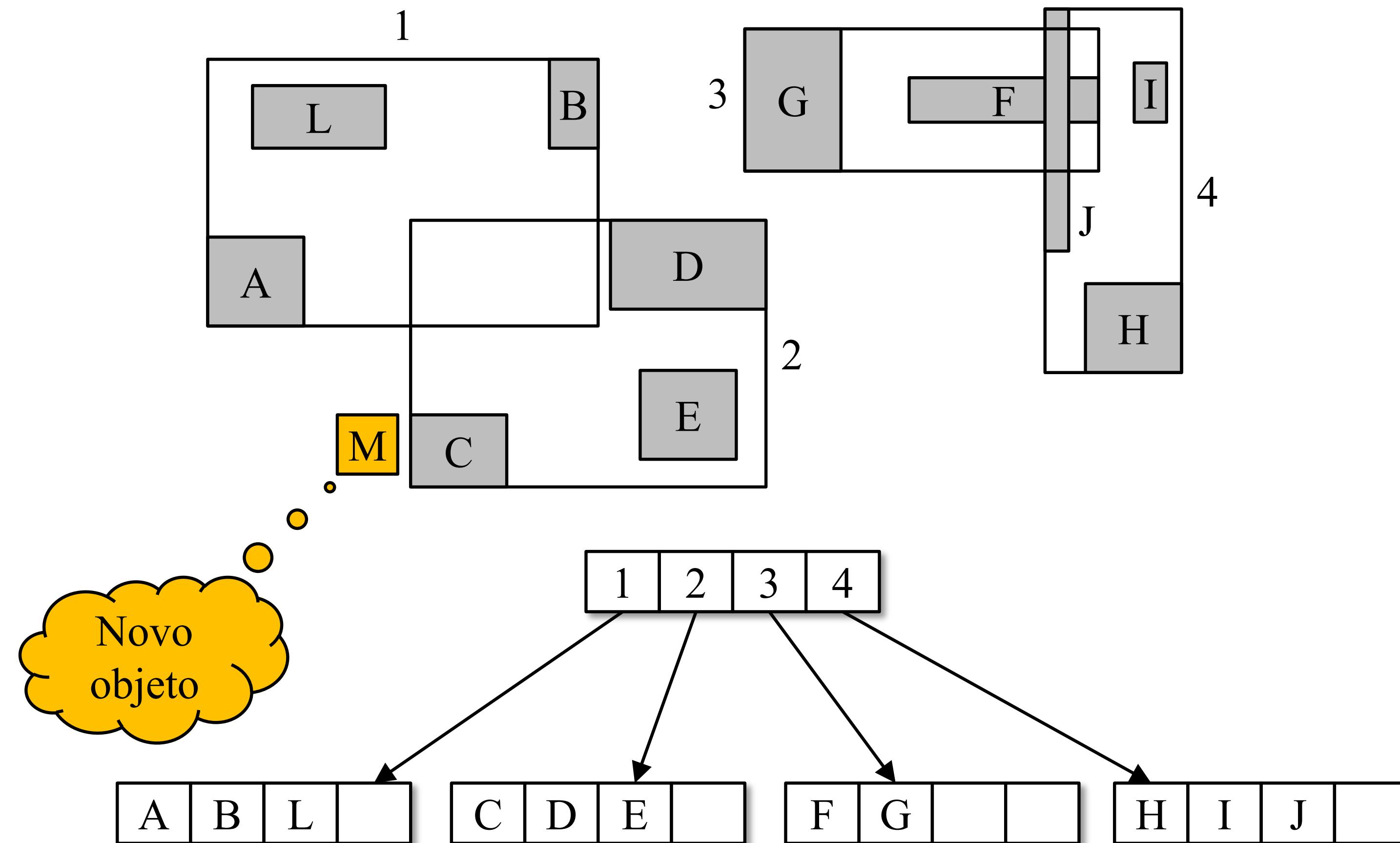
# Inserção - Exemplo: R(2, 4)



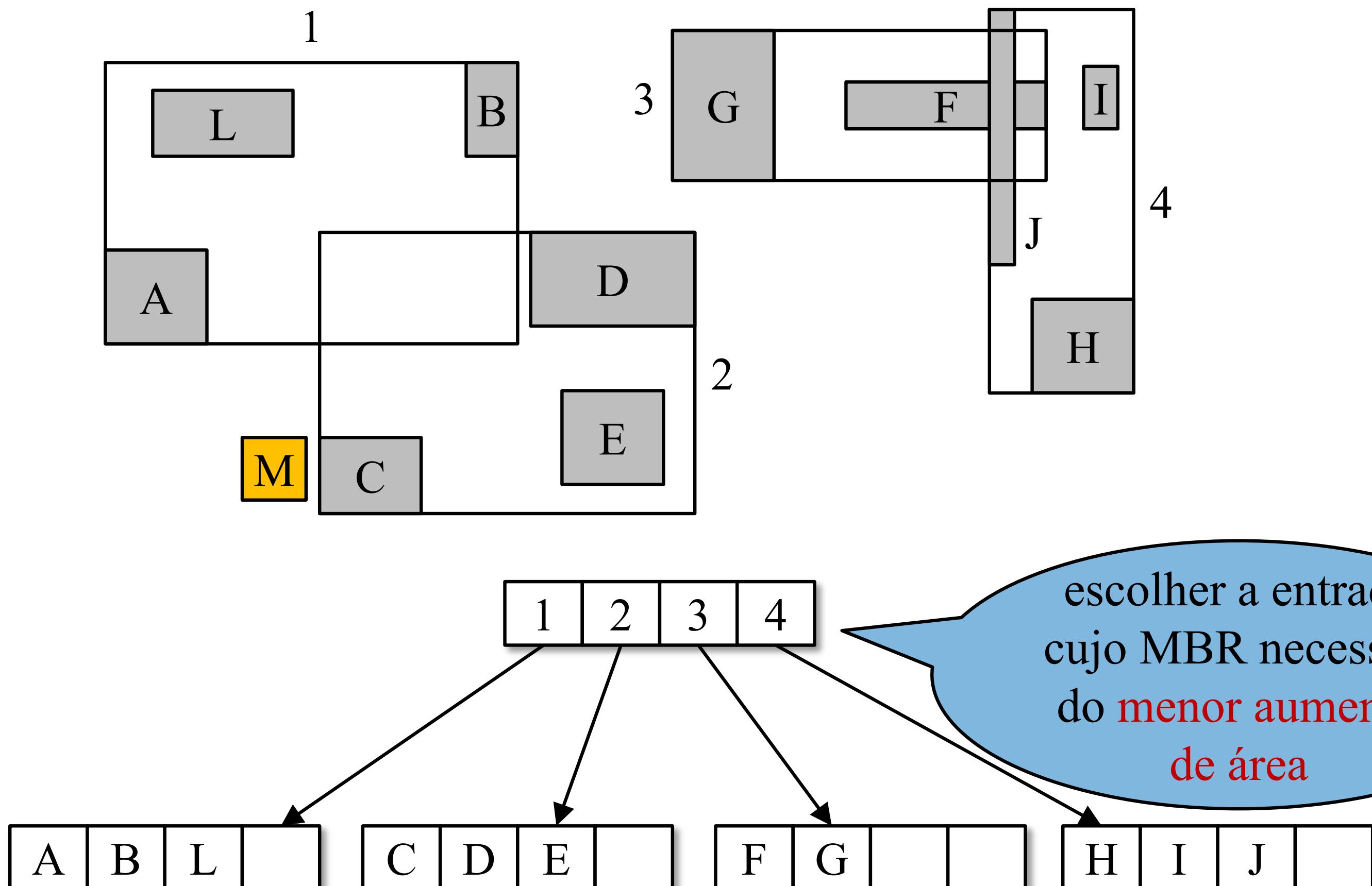
# Inserção - Exemplo: R(2, 4)



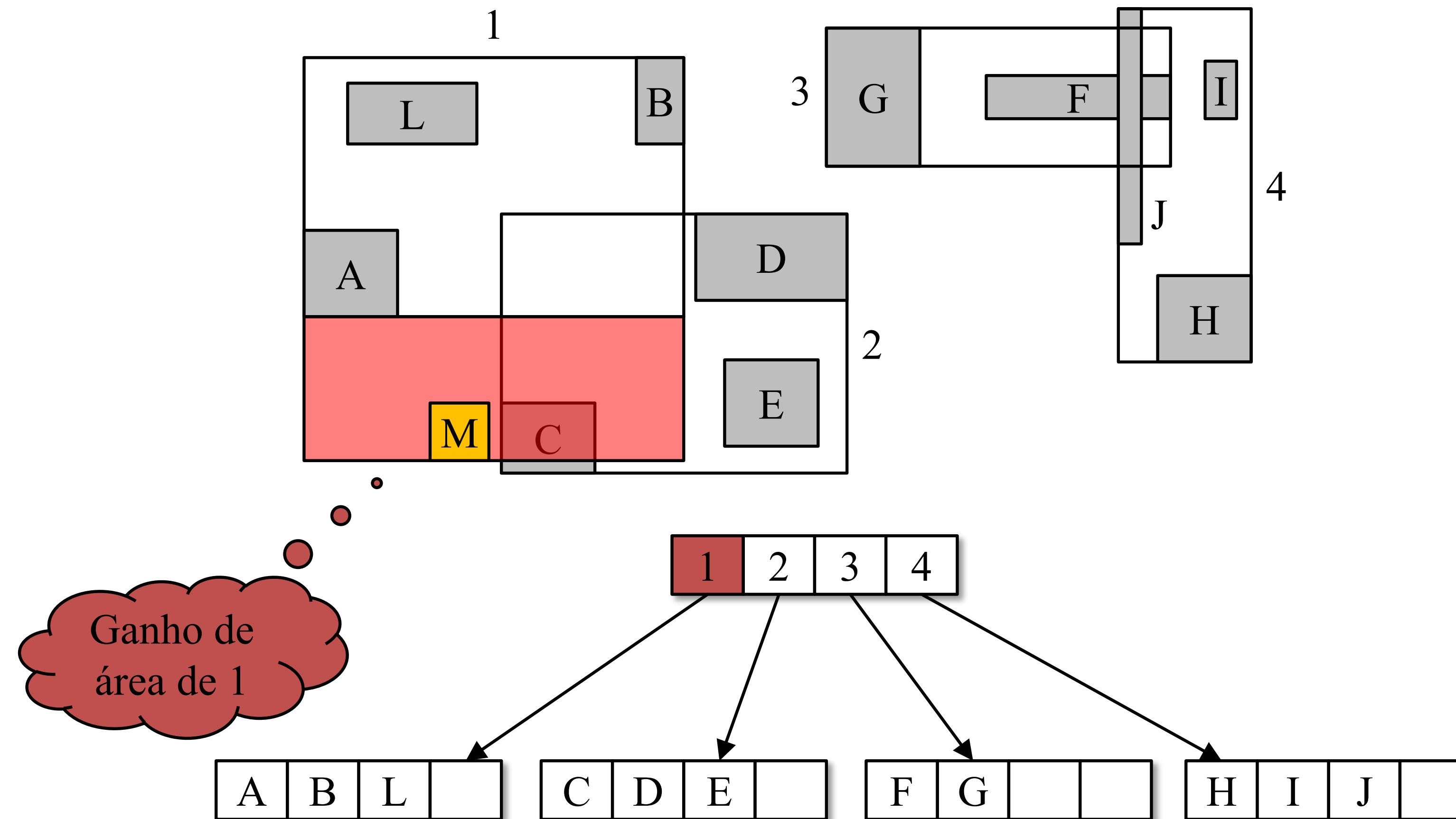
# Inserção - Exemplo: R(2, 4)



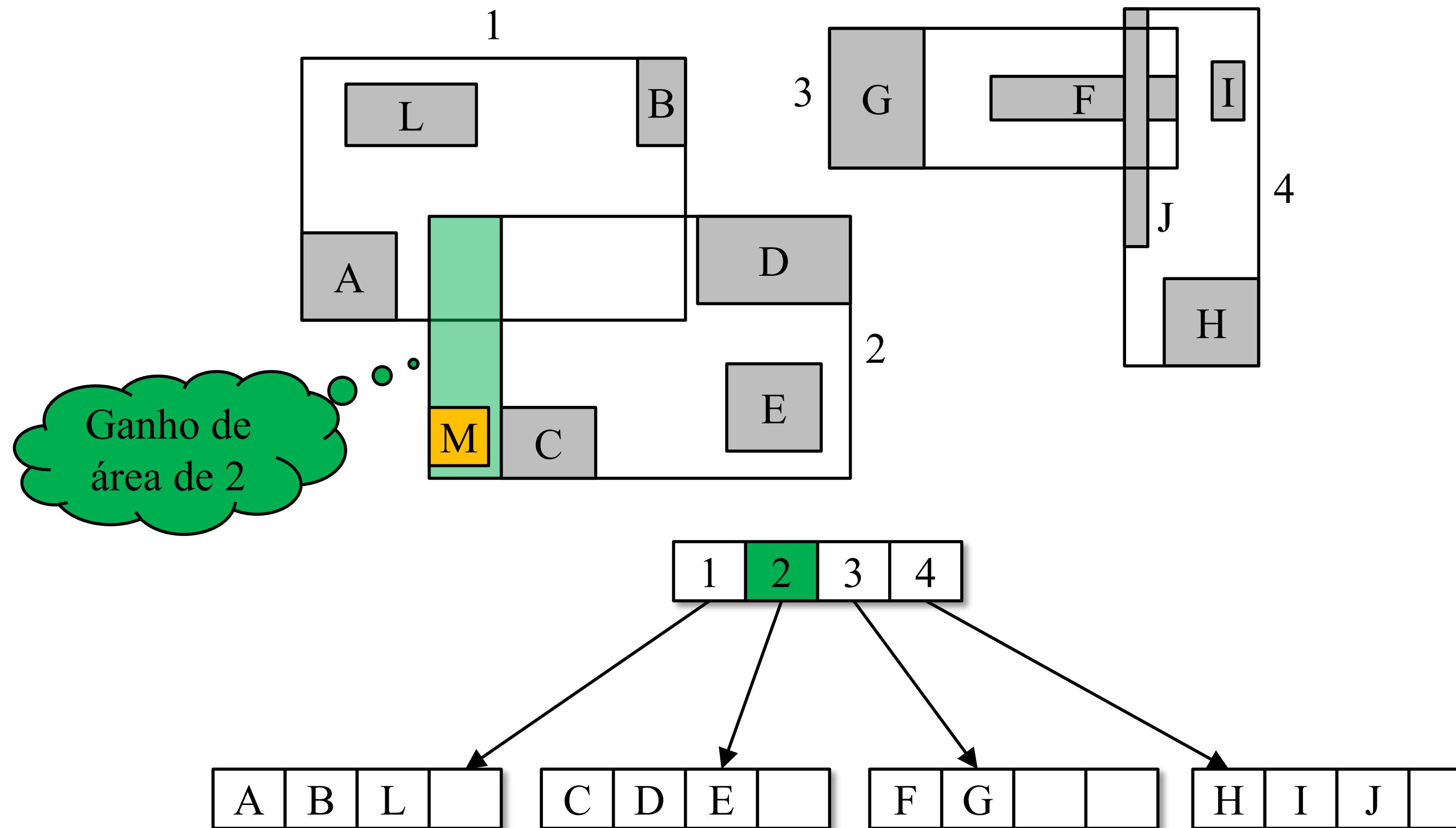
# Inserção - Exemplo: R(2, 4)



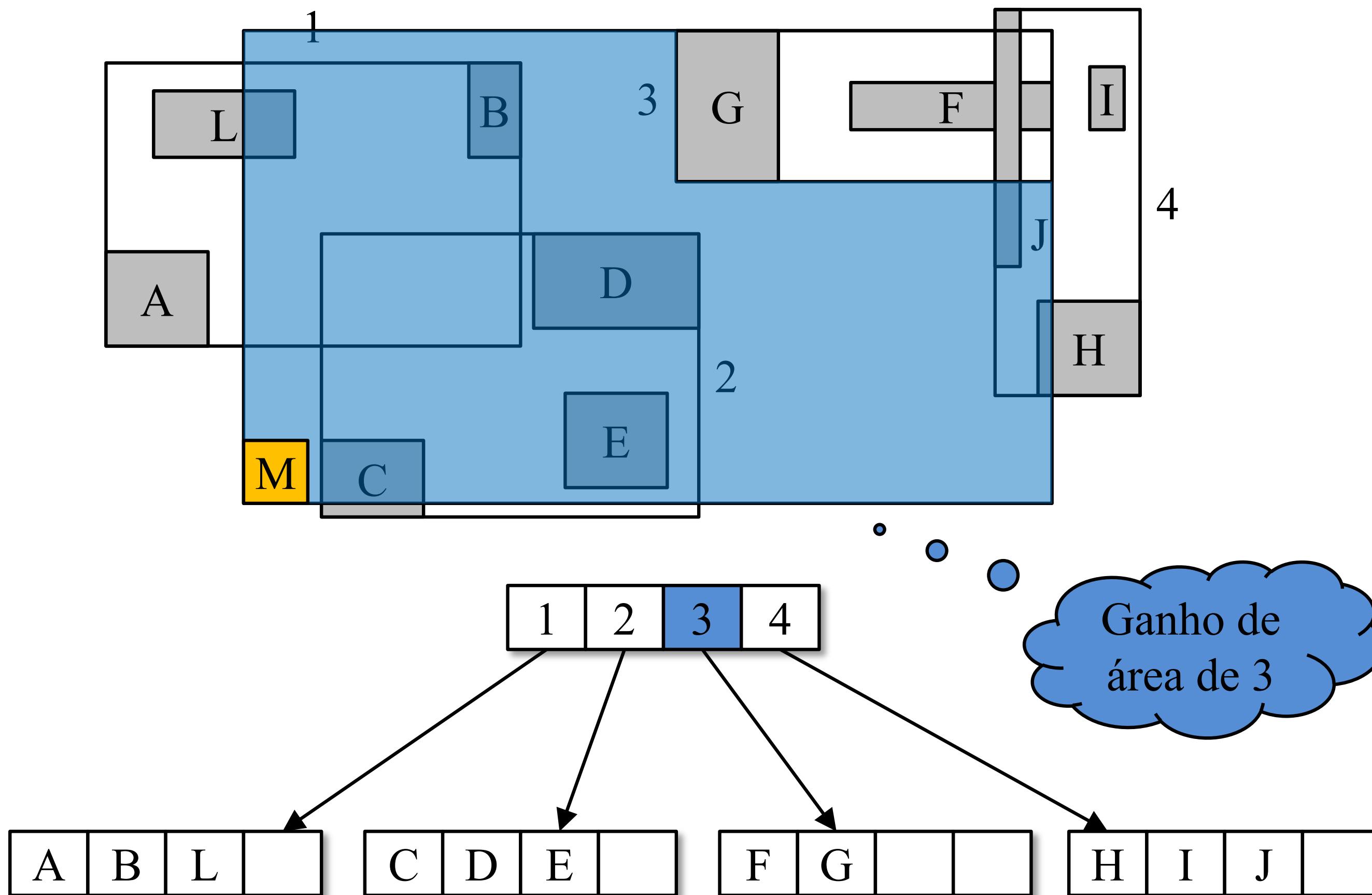
# Inserção - Exemplo: R(2, 4)



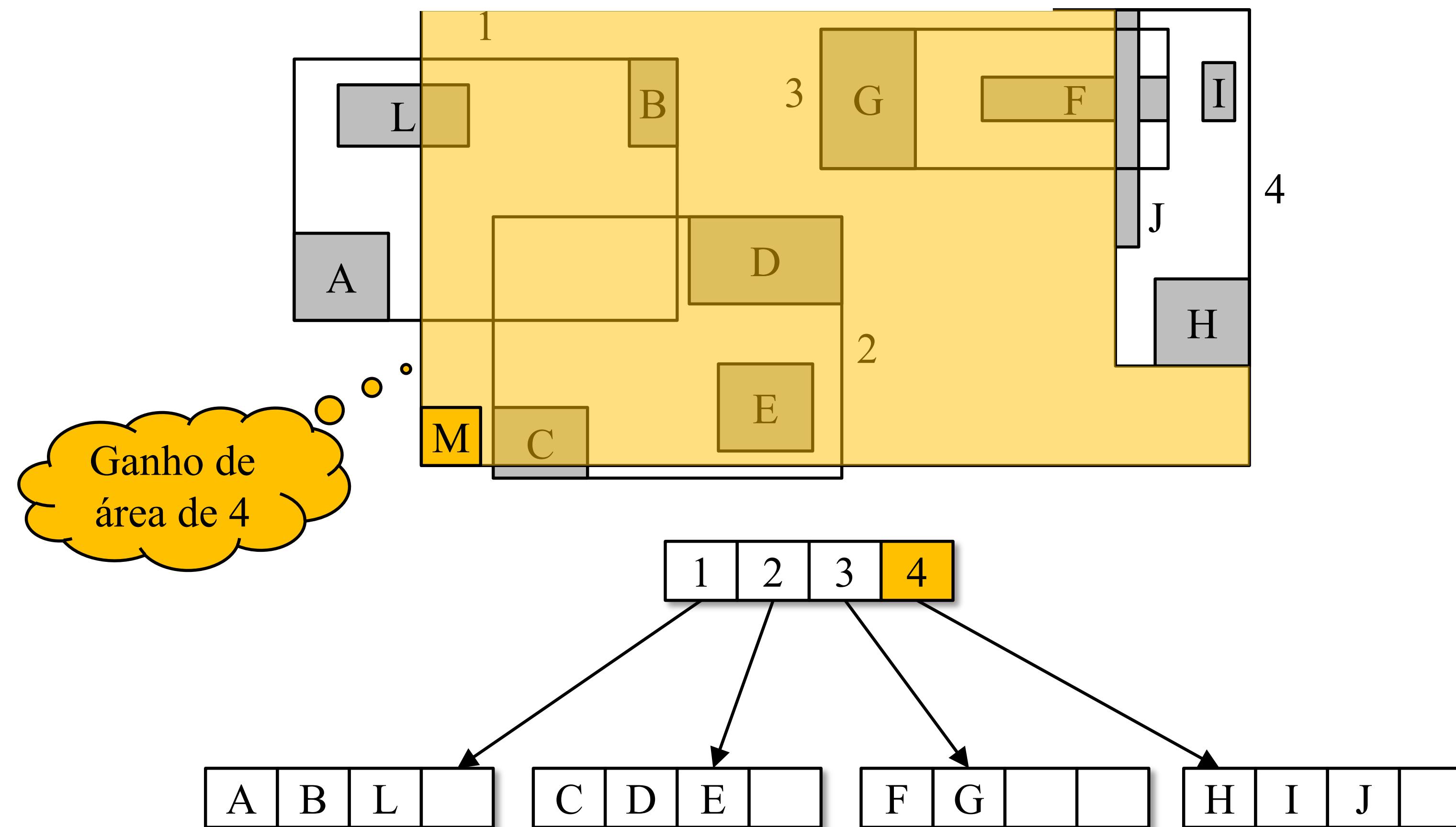
# Inserção - Exemplo: R(2, 4)



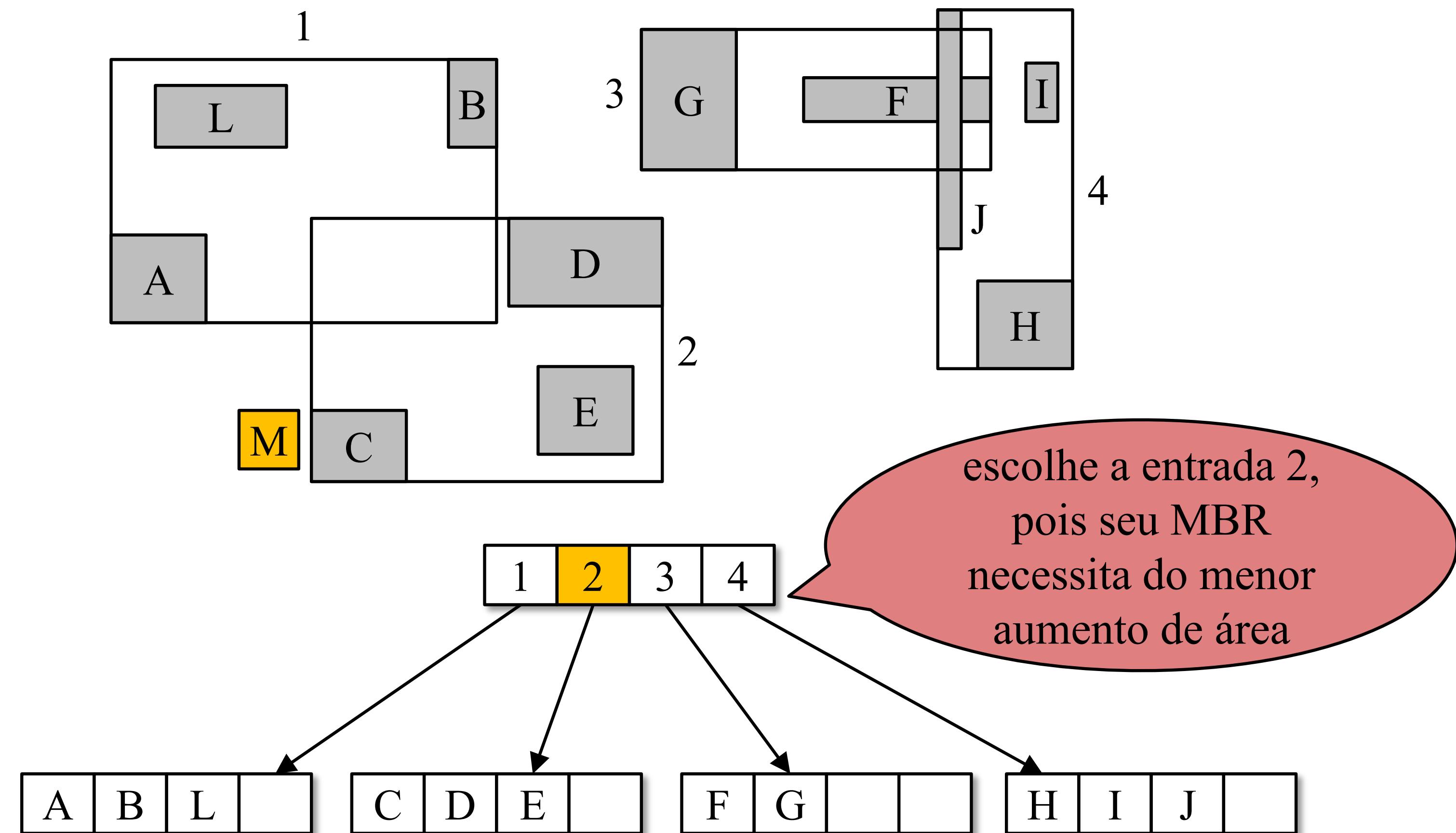
# Inserção - Exemplo: R(2, 4)



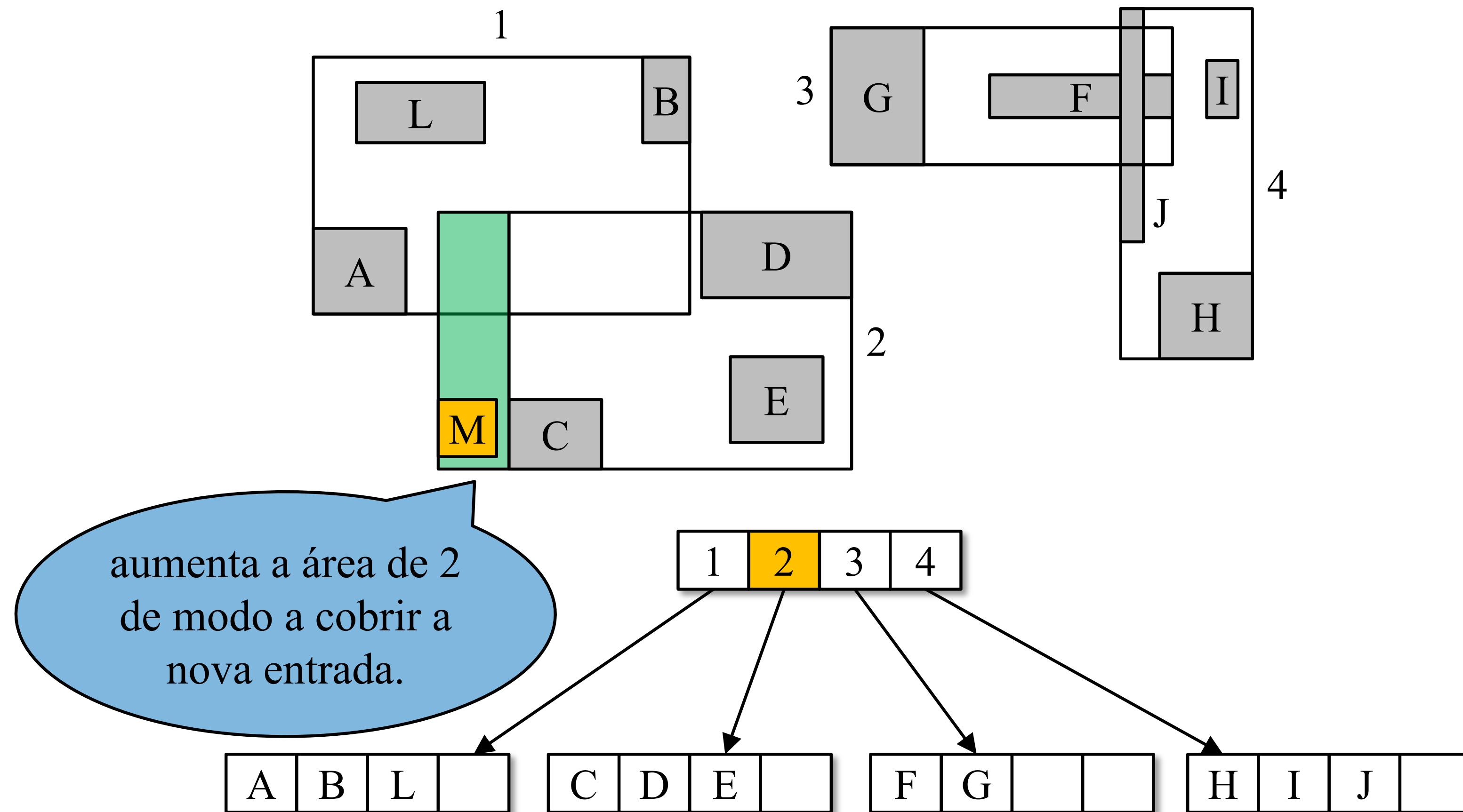
# Inserção - Exemplo: R(2, 4)



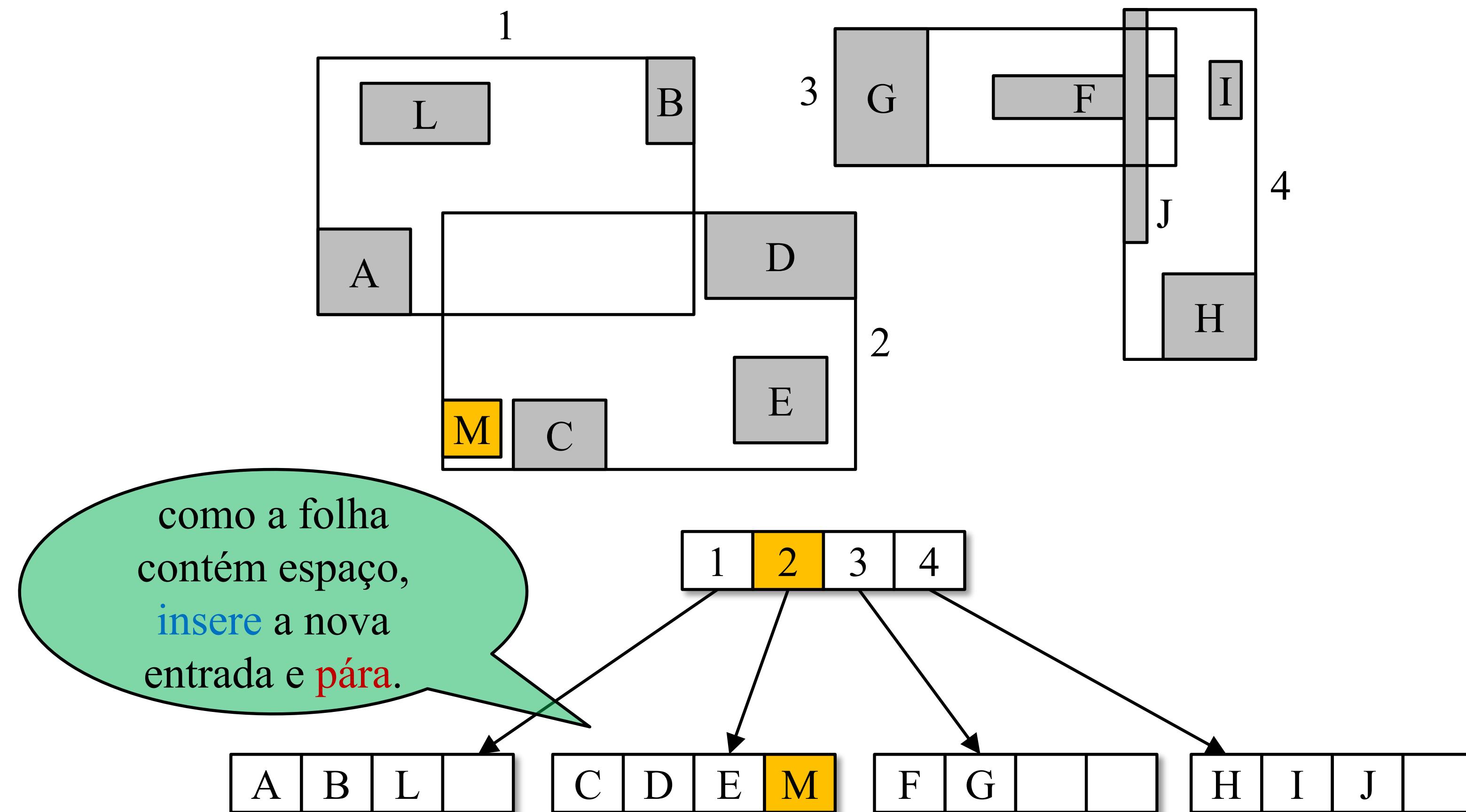
# Inserção - Exemplo: R(2, 4)



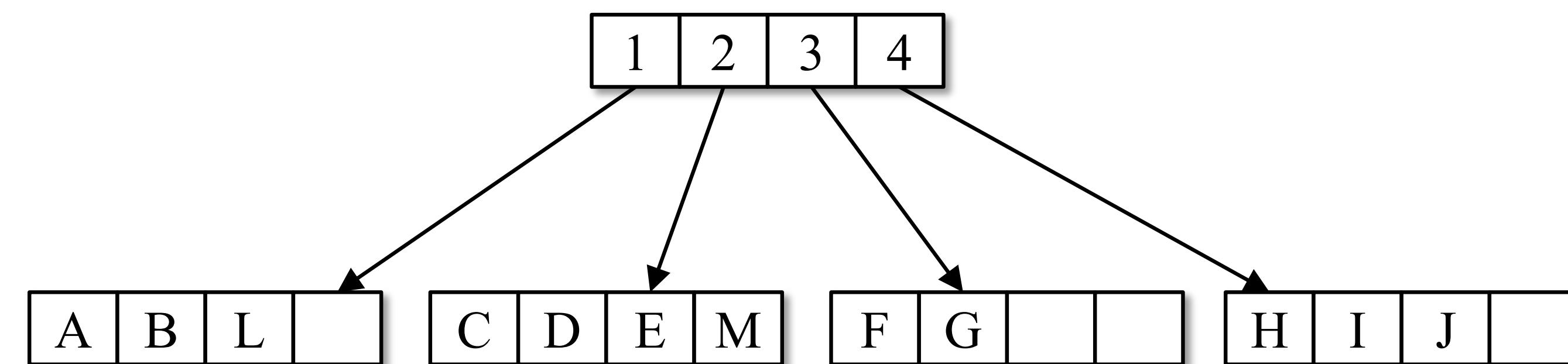
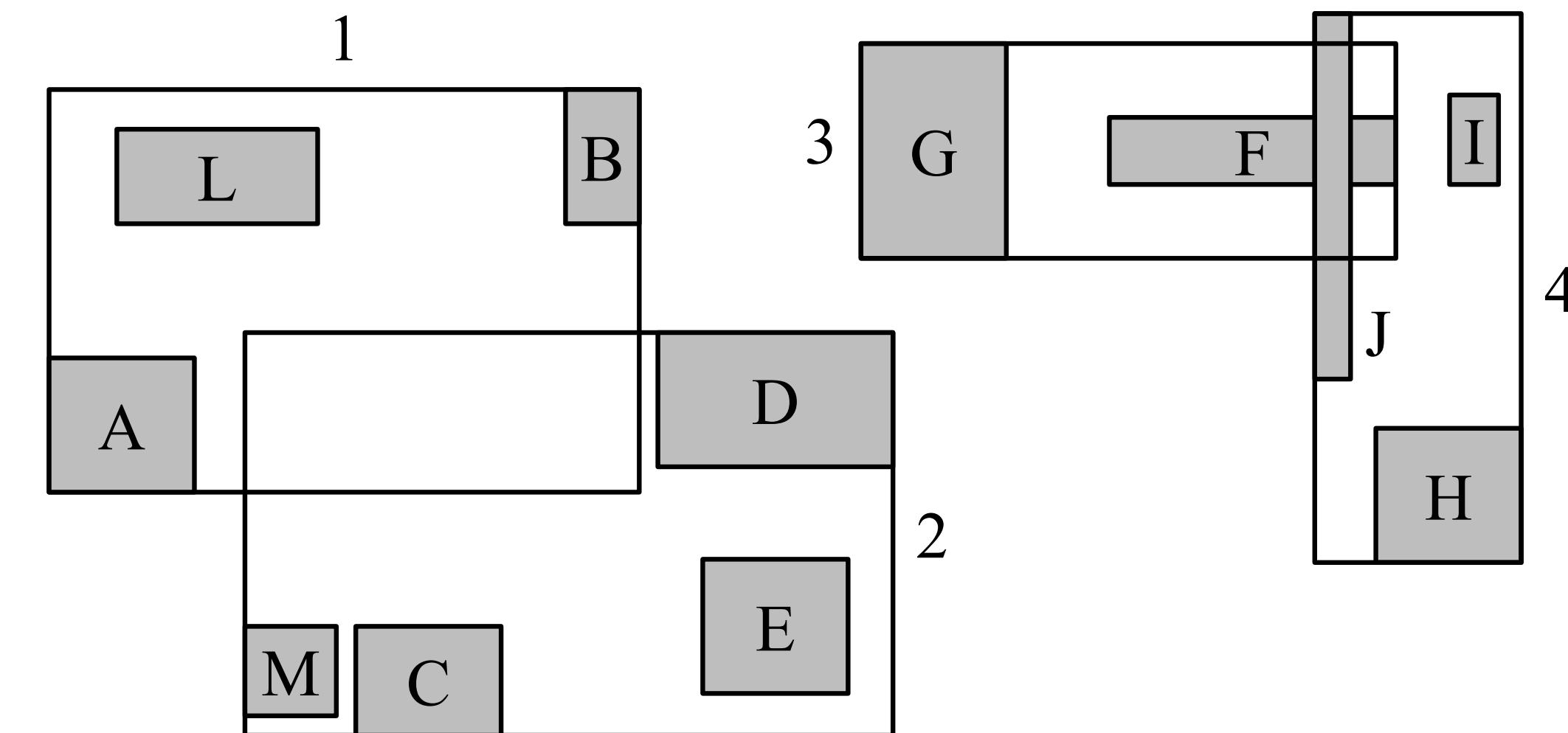
# Inserção - Exemplo: R(2, 4)



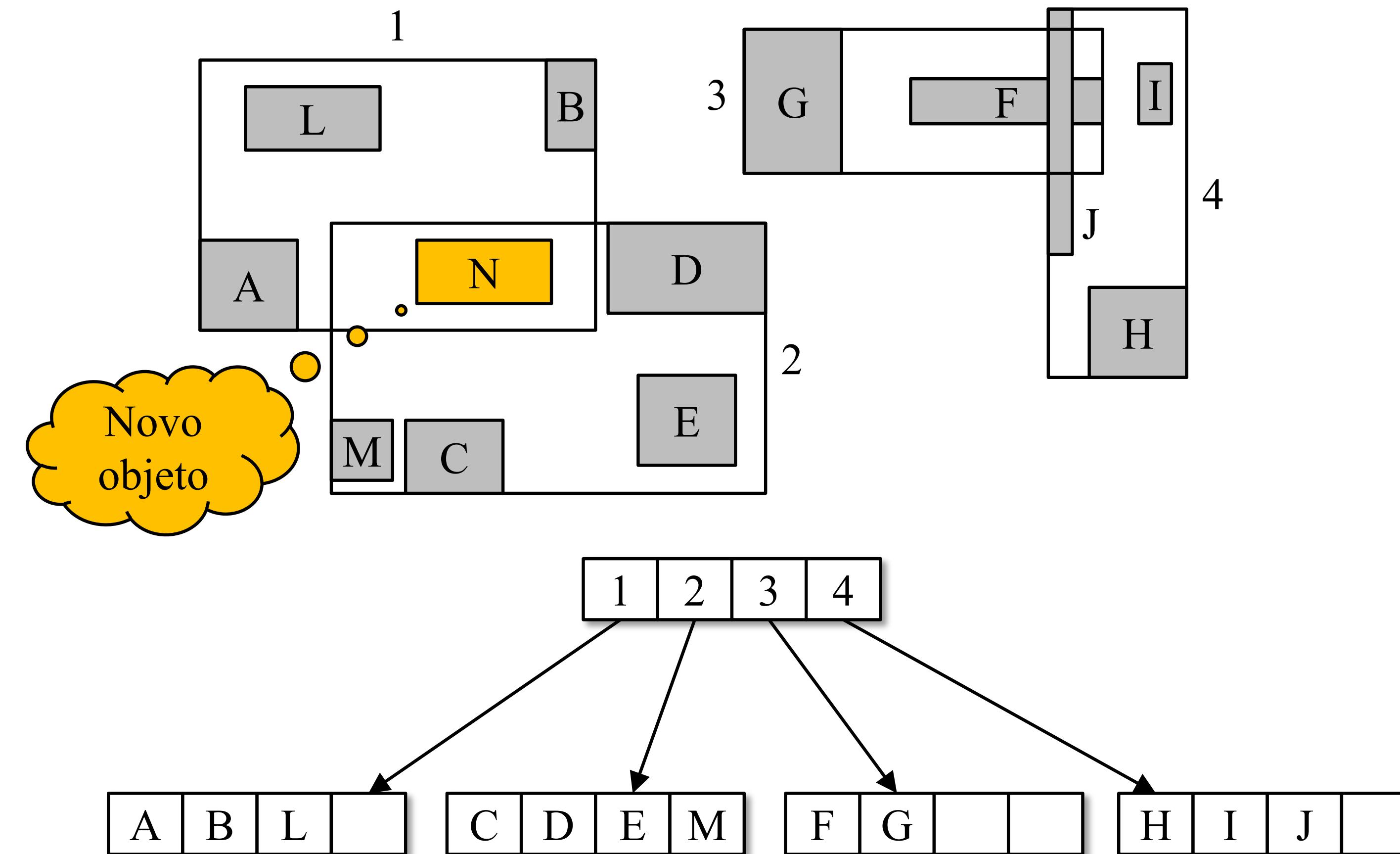
# Inserção - Exemplo: R(2, 4)



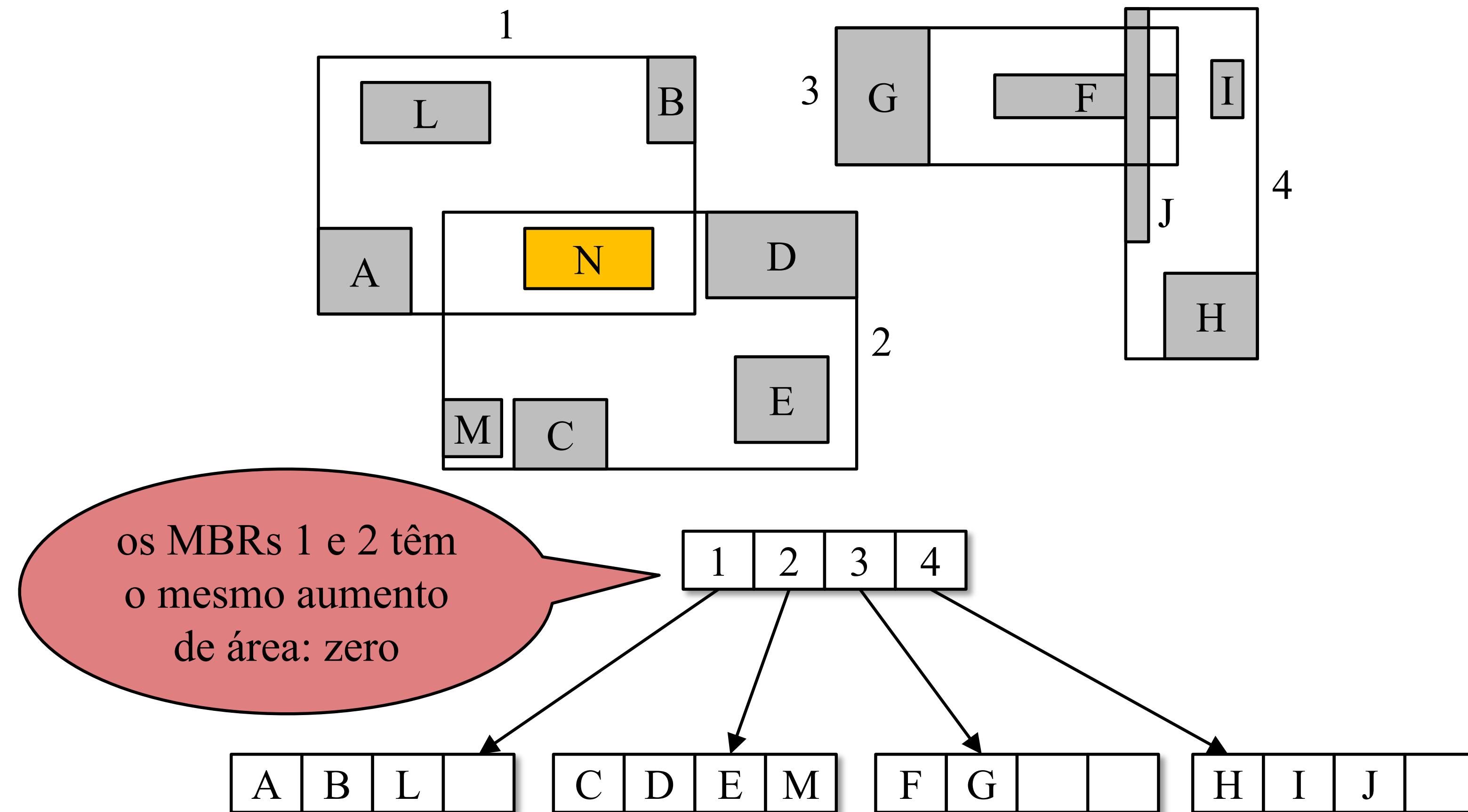
# Inserção - Exemplo: R(2, 4)



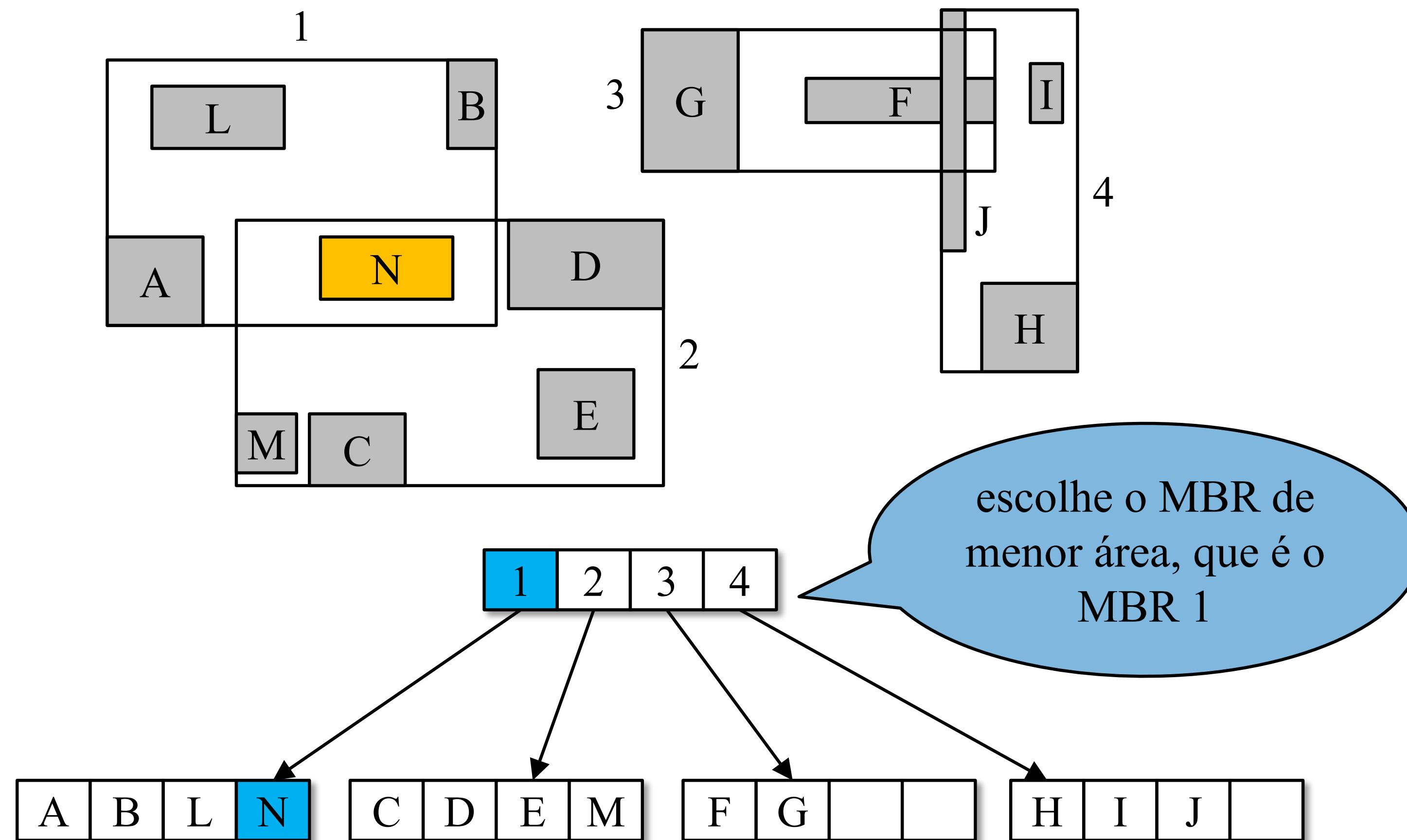
# Inserção - Exemplo: R(2, 4)



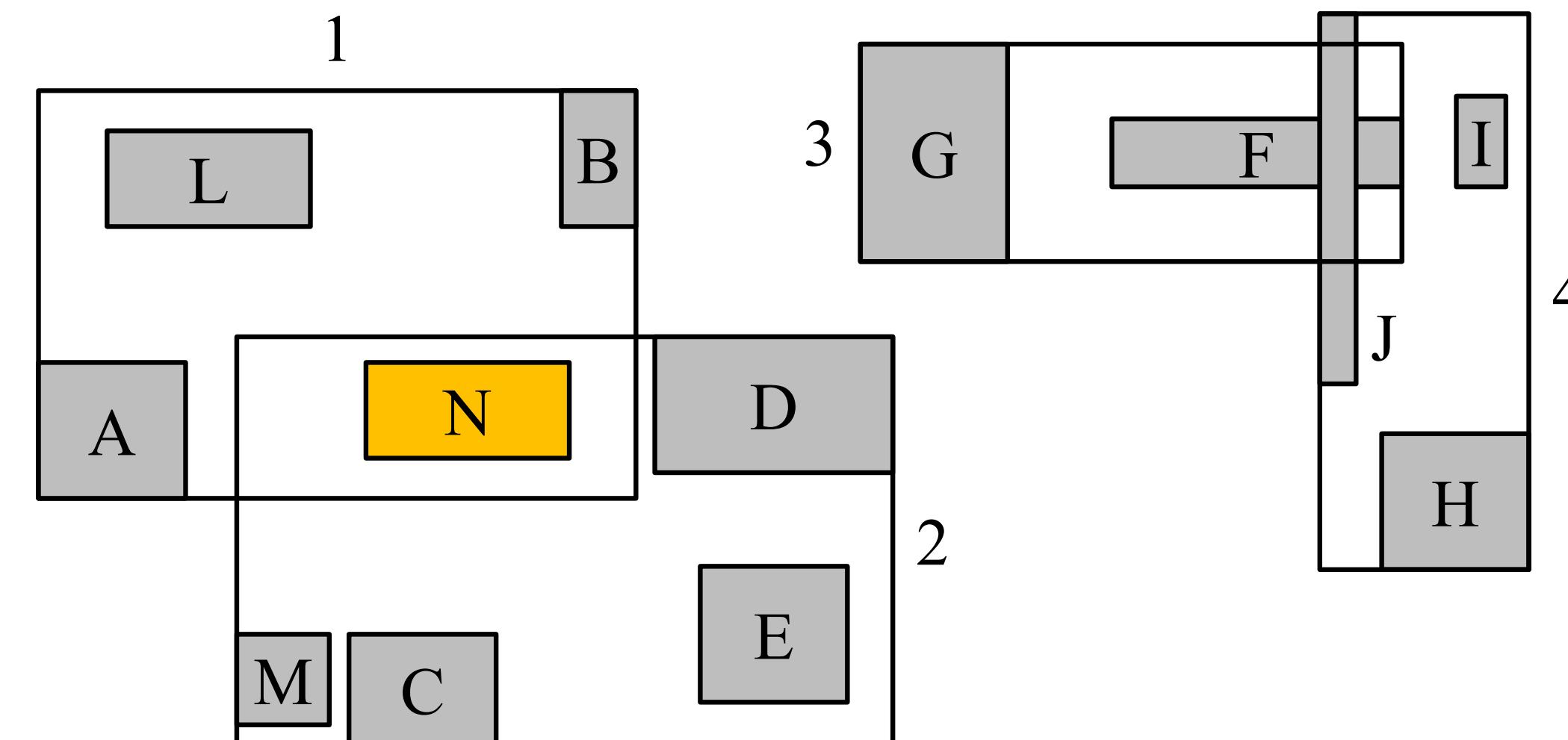
# Inserção - Exemplo: R(2, 4)



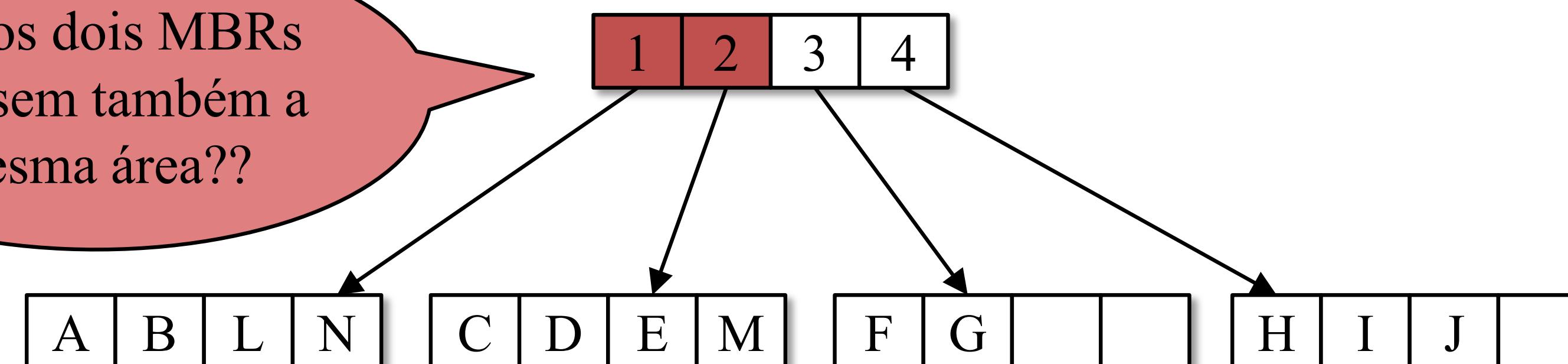
# Inserção - Exemplo: R(2, 4)



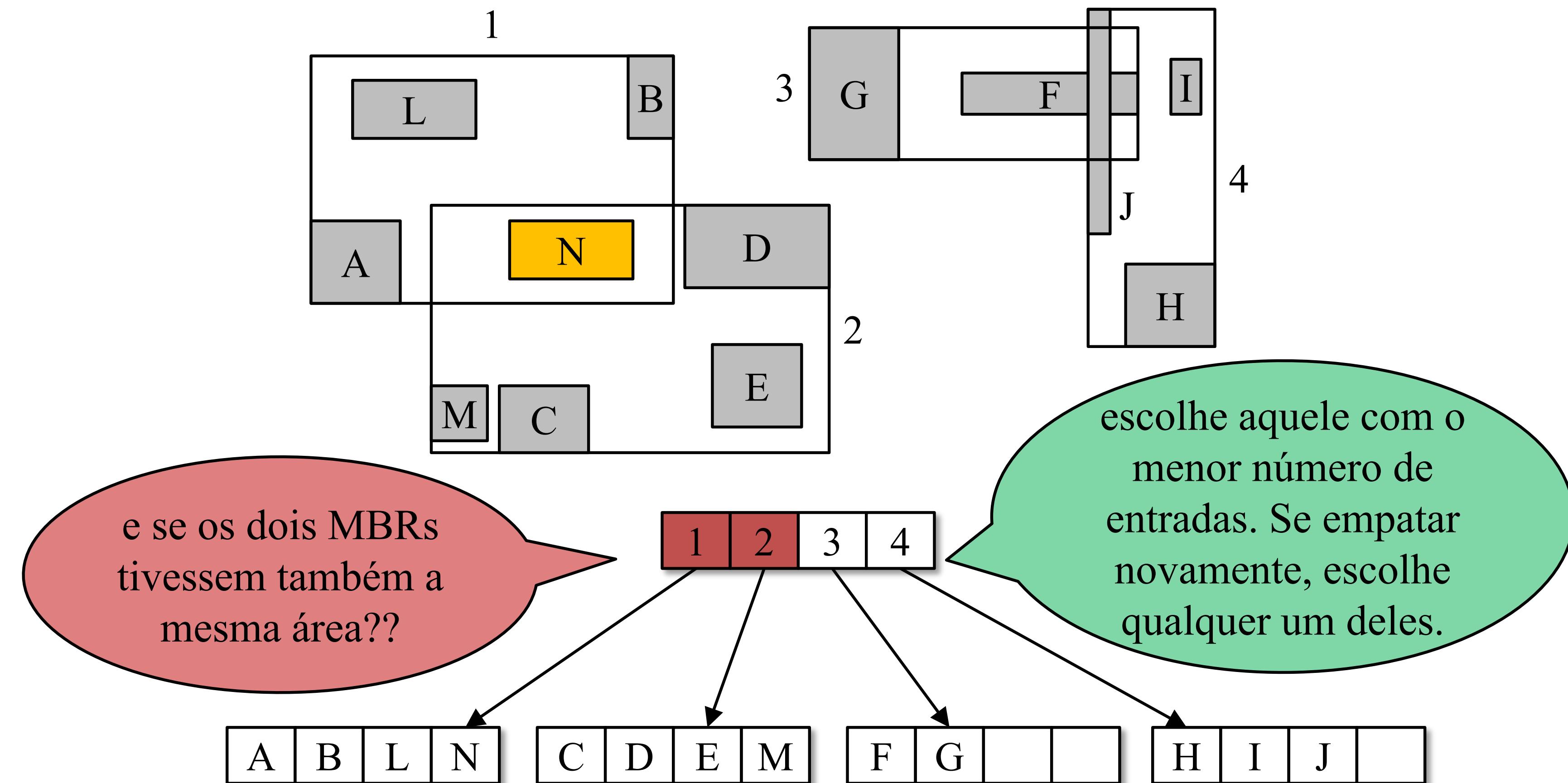
# Inserção - Exemplo: R(2, 4)



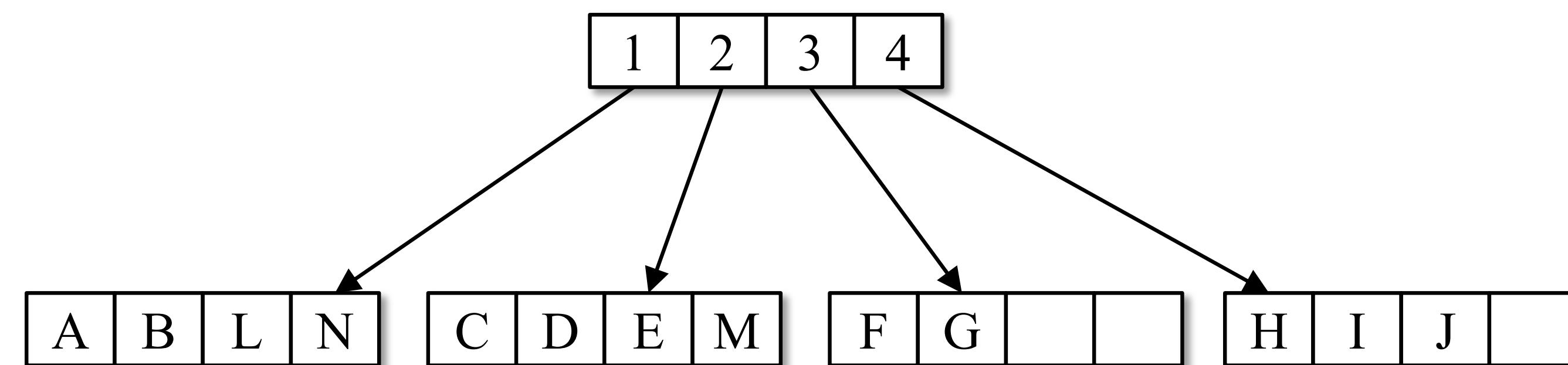
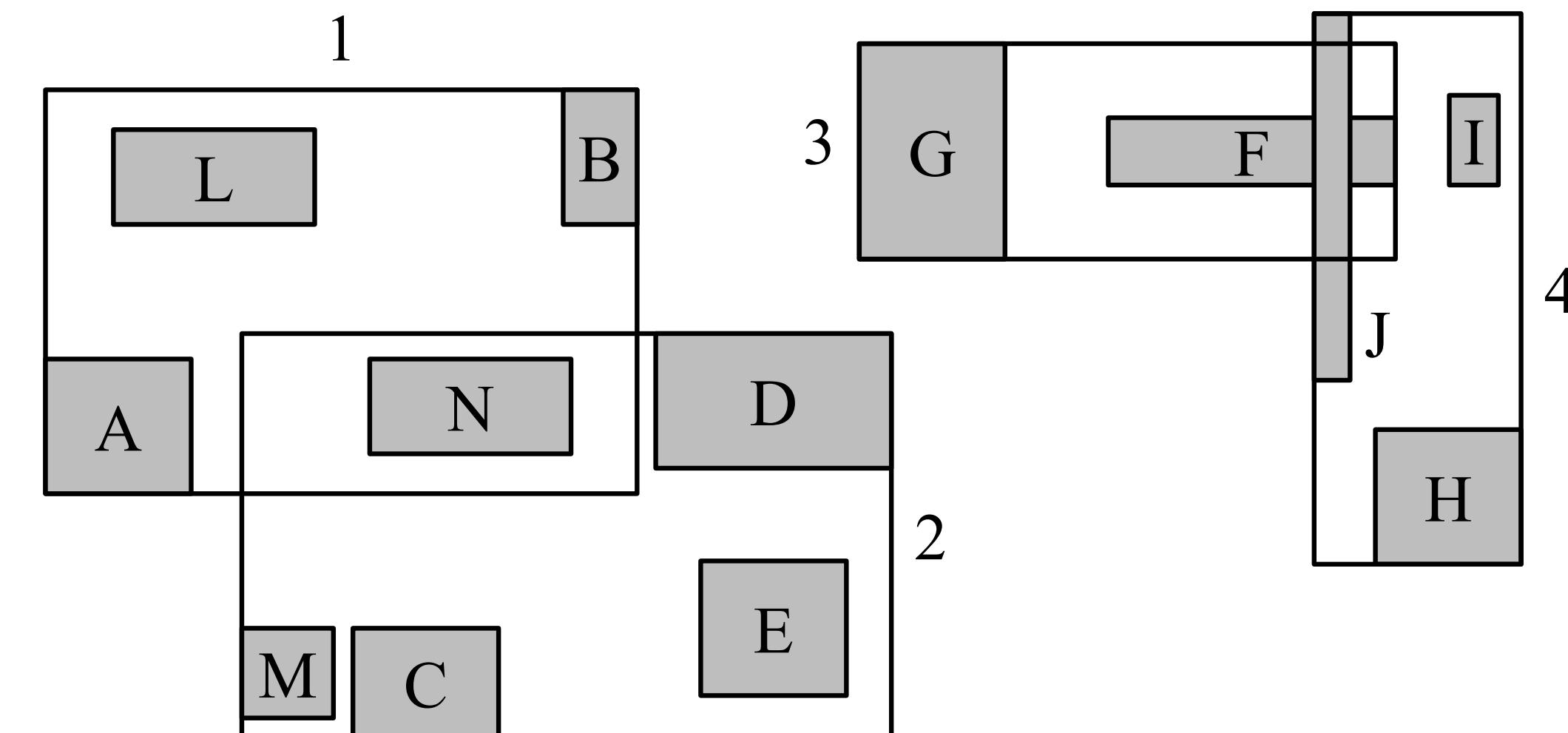
e se os dois MBRs tivessem também a mesma área??



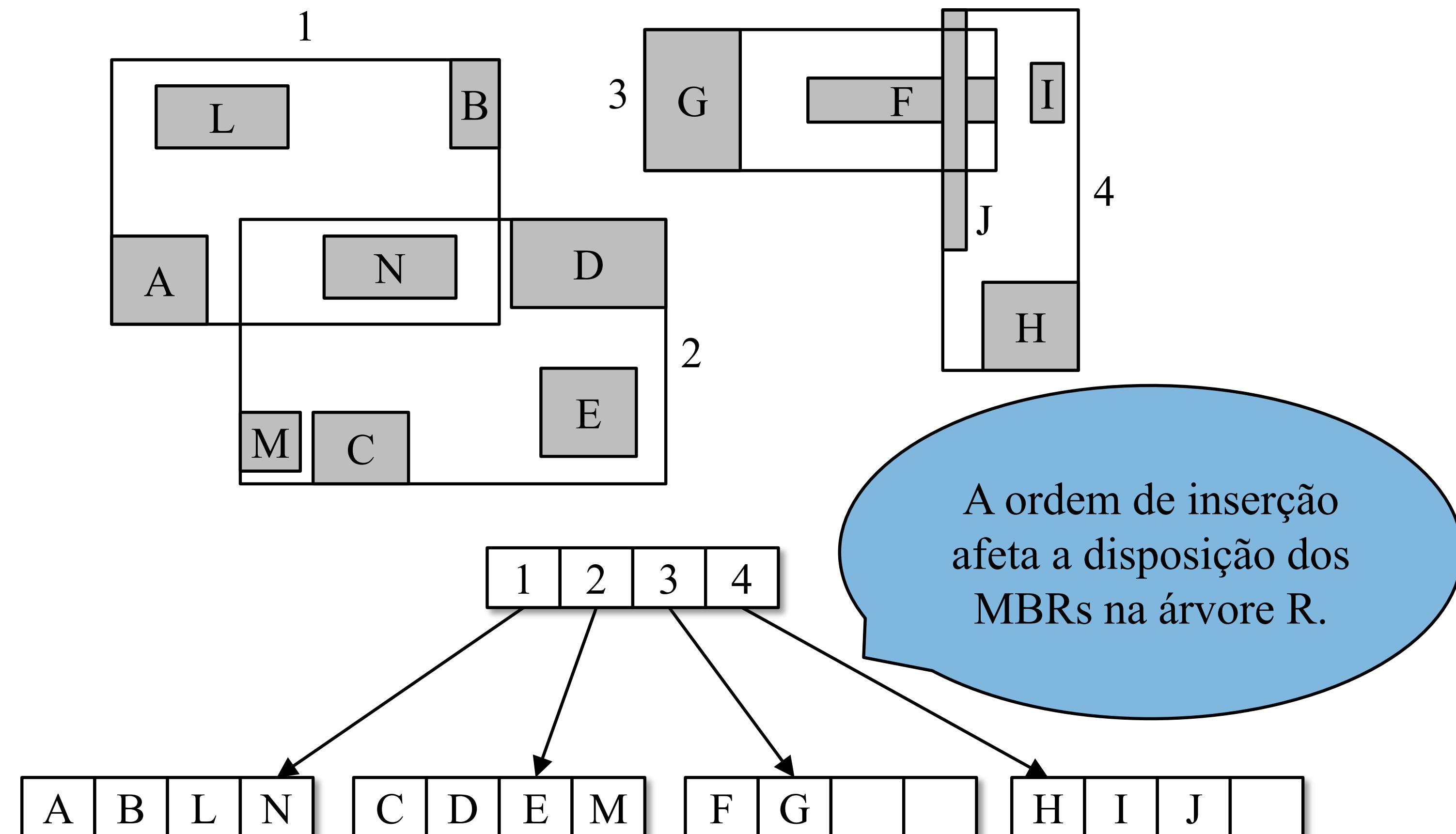
# Inserção - Exemplo: R(2, 4)



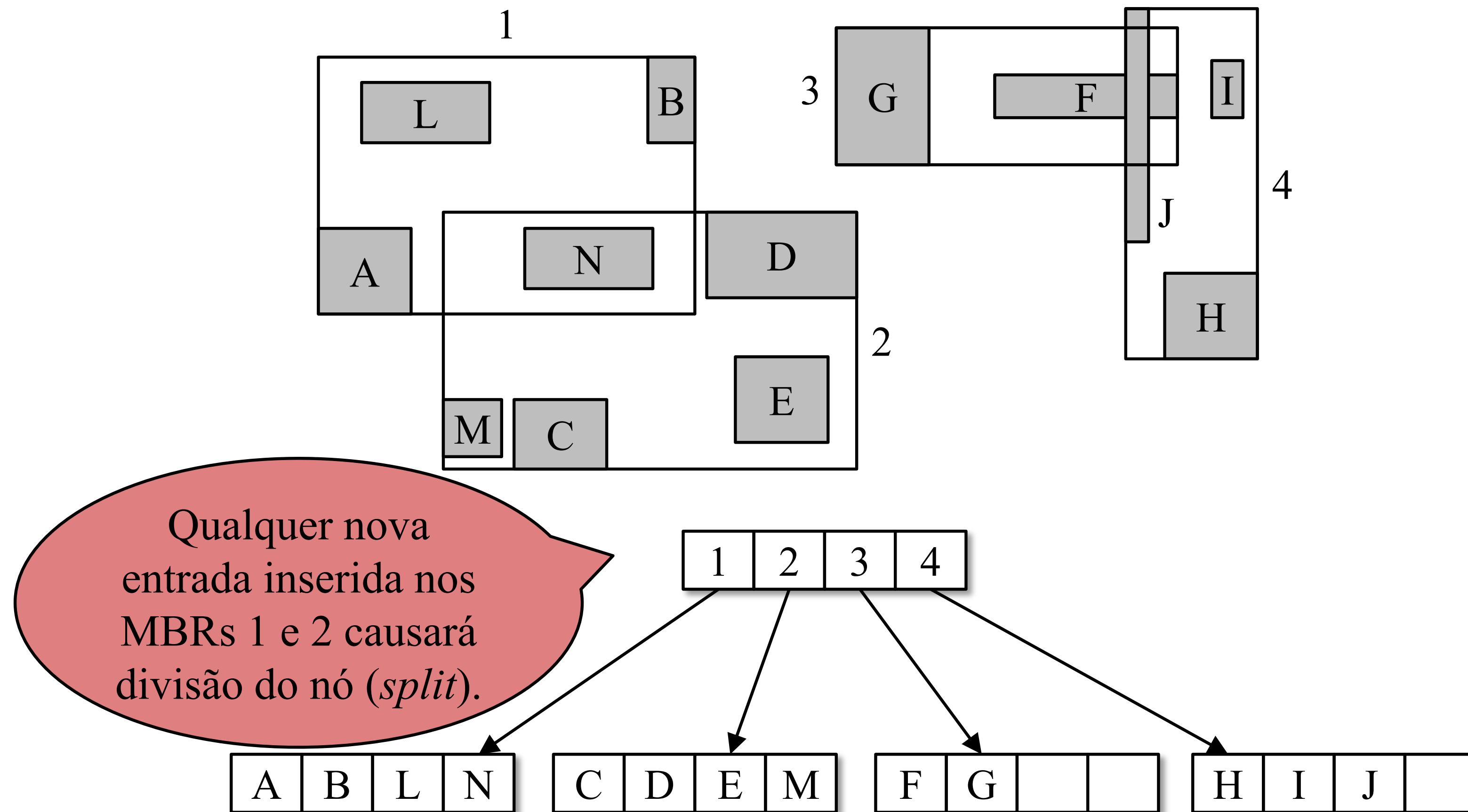
# Inserção - Exemplo: R(2, 4)



# Inserção - Exemplo: R(2, 4)



# Inserção - Exemplo: R(2, 4)



# Split



- Distribui as  $M$  entradas de um nó **mais a nova entrada em dois nós**
- Reduzir a área de cobertura
- Seleção dos primeiros objetos de cada grupo: *seeds*. Na árvore R, **dois** objetos são promovidos ao nó índice.
- Distribuição dos objetos restantes.
- Algoritmos: quadrático, **linear**, exaustivo.



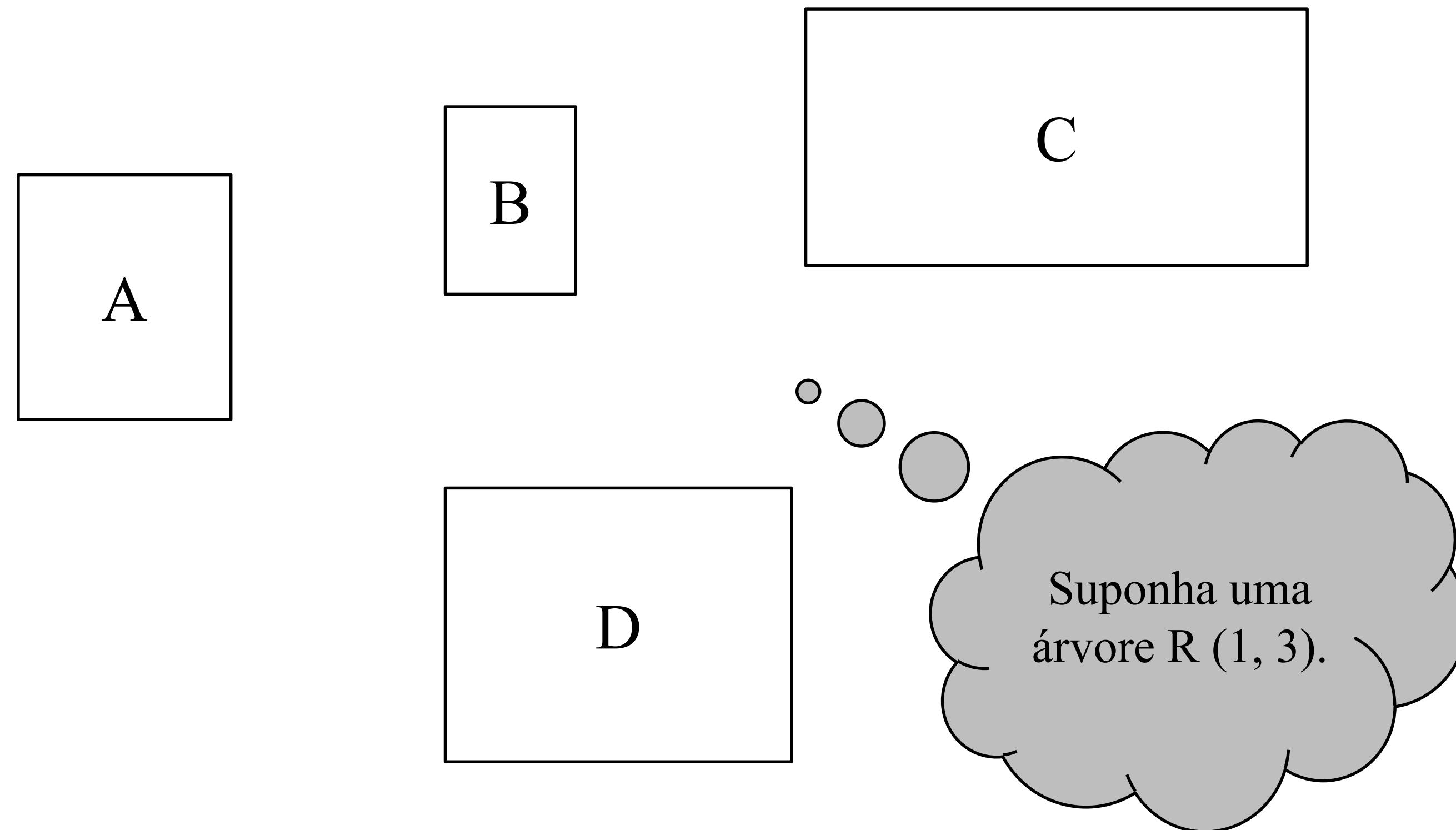
## ● Parte 1: Seleção das seeds

- Ao longo de cada dimensão, encontrar a entrada cujo retângulo tem o lado inferior mais alto, e aquele com o lado mais baixo. Registre a separação. Complexidade de tempo:  $O(M * d)$   $d = \text{no dimensões}$
- Normalize essas separações dividindo a largura de todo o conjunto pela dimensão correspondente -  $O(d)$ .
- Escolha o par com a maior separação normalizada em qualquer dimensão -  $O(d)$ .

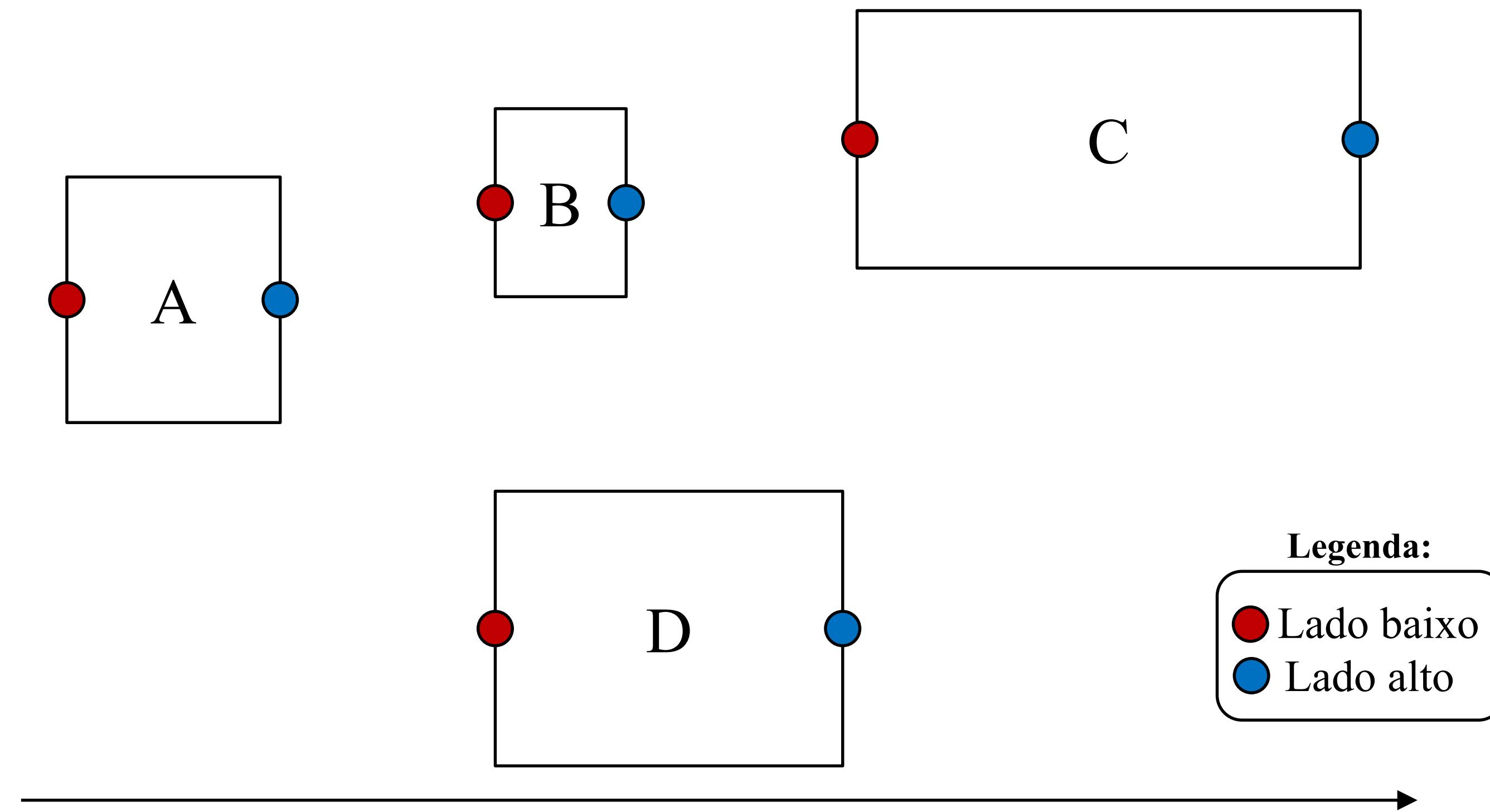
## ● Parte 2: Redistribuição das $M - 1$ entradas

- Até que não reste mais entradas ( $O(1)$ ), selecionar uma entrada E e inserir no nó que requer o menor aumento de seu MBR ( $O(M)$ ). Complexidade total de tempo:  $O(M * d)$

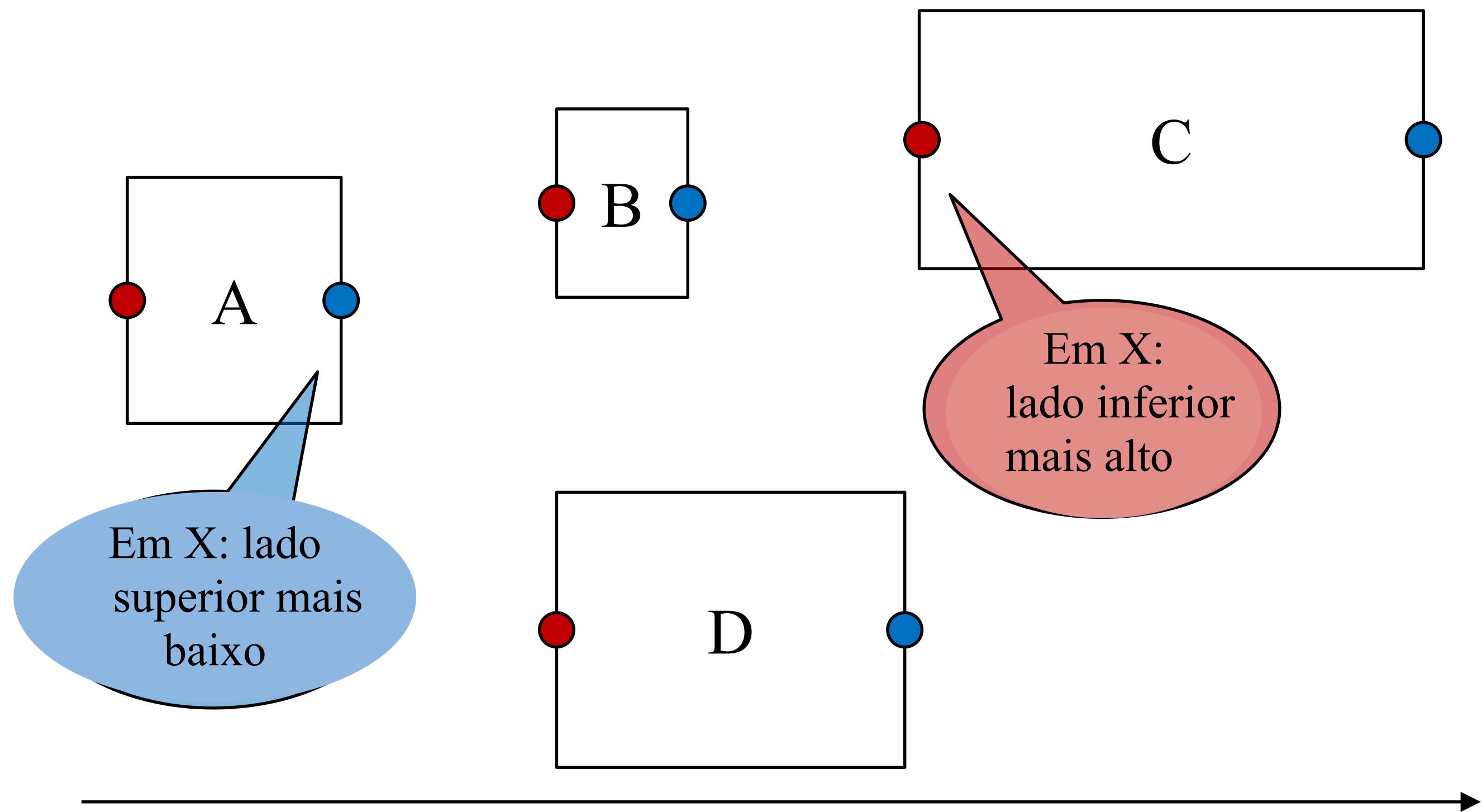
# Split: Algoritmo Linear - Exemplo



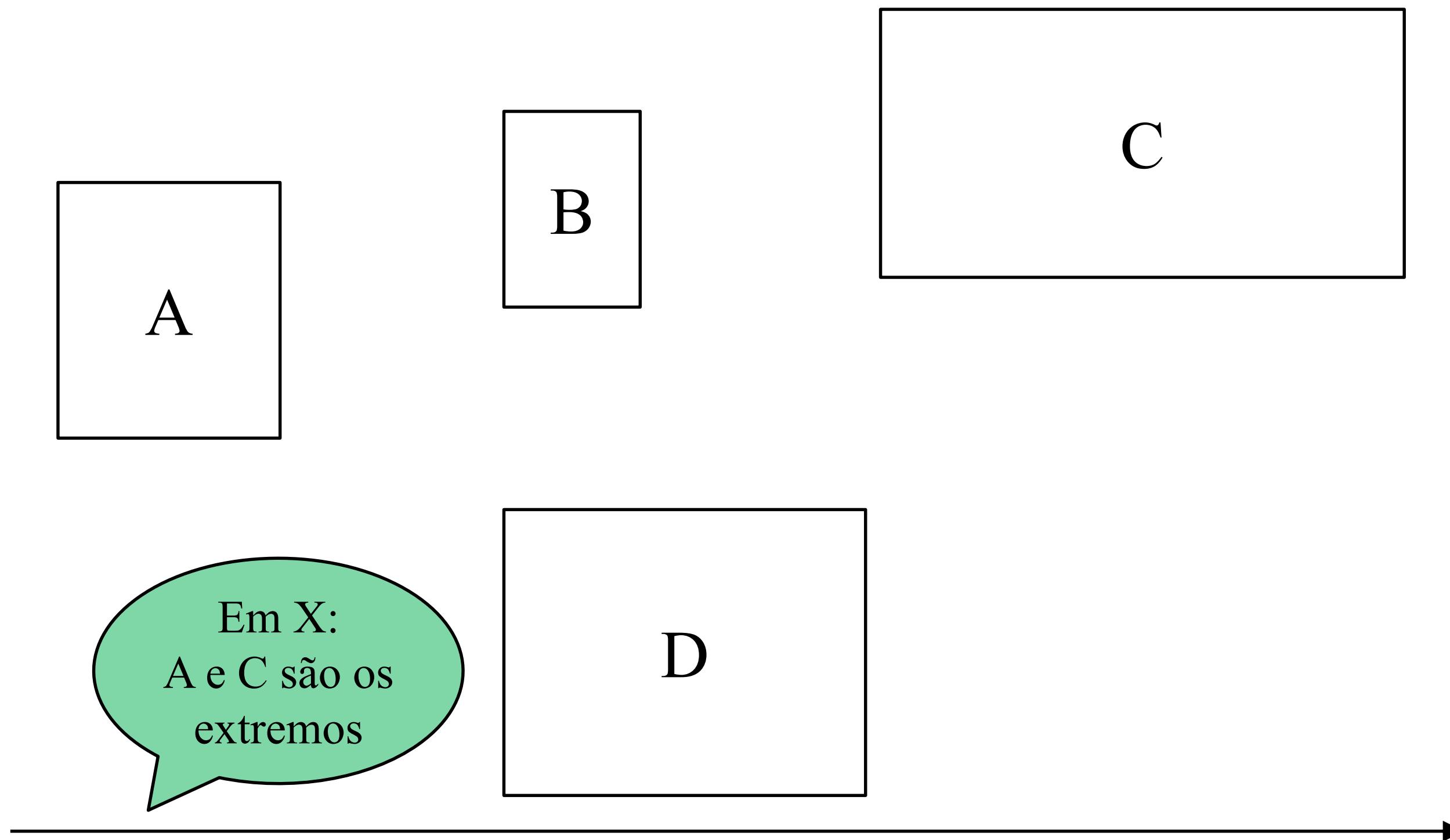
# Split: Algoritmo Linear - Exemplo



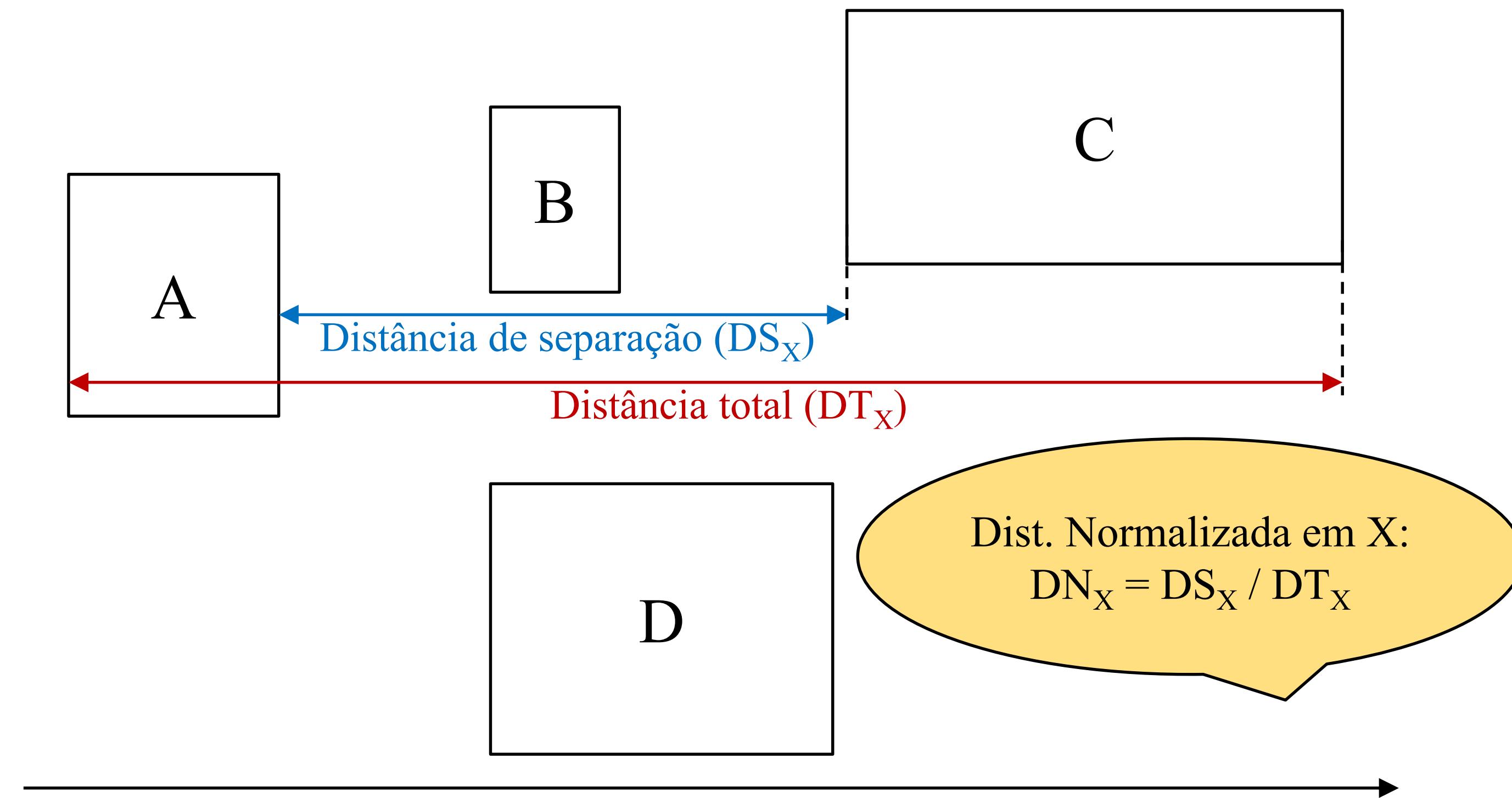
# Split: Algoritmo Linear - Exemplo



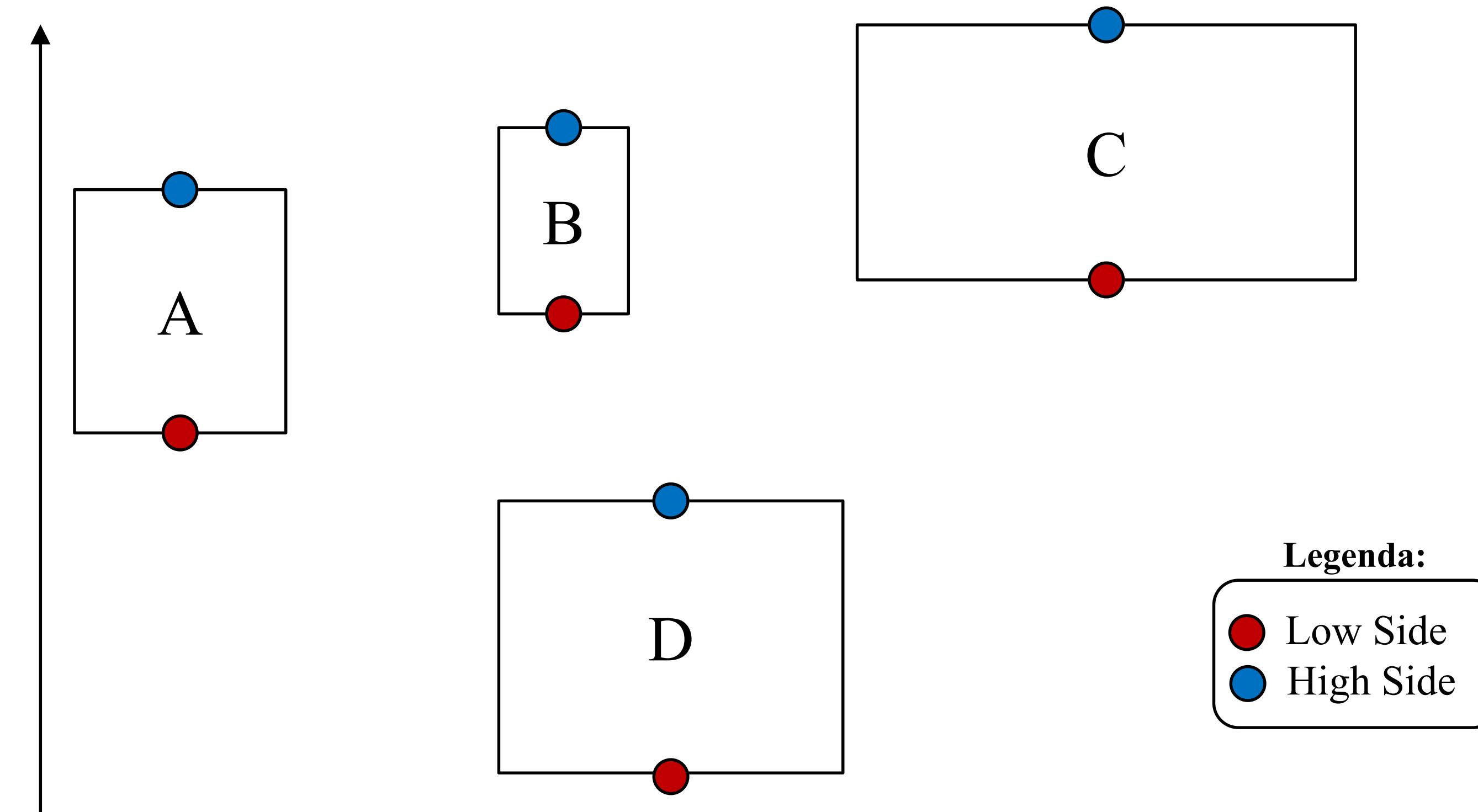
# Split: Algoritmo Linear - Exemplo



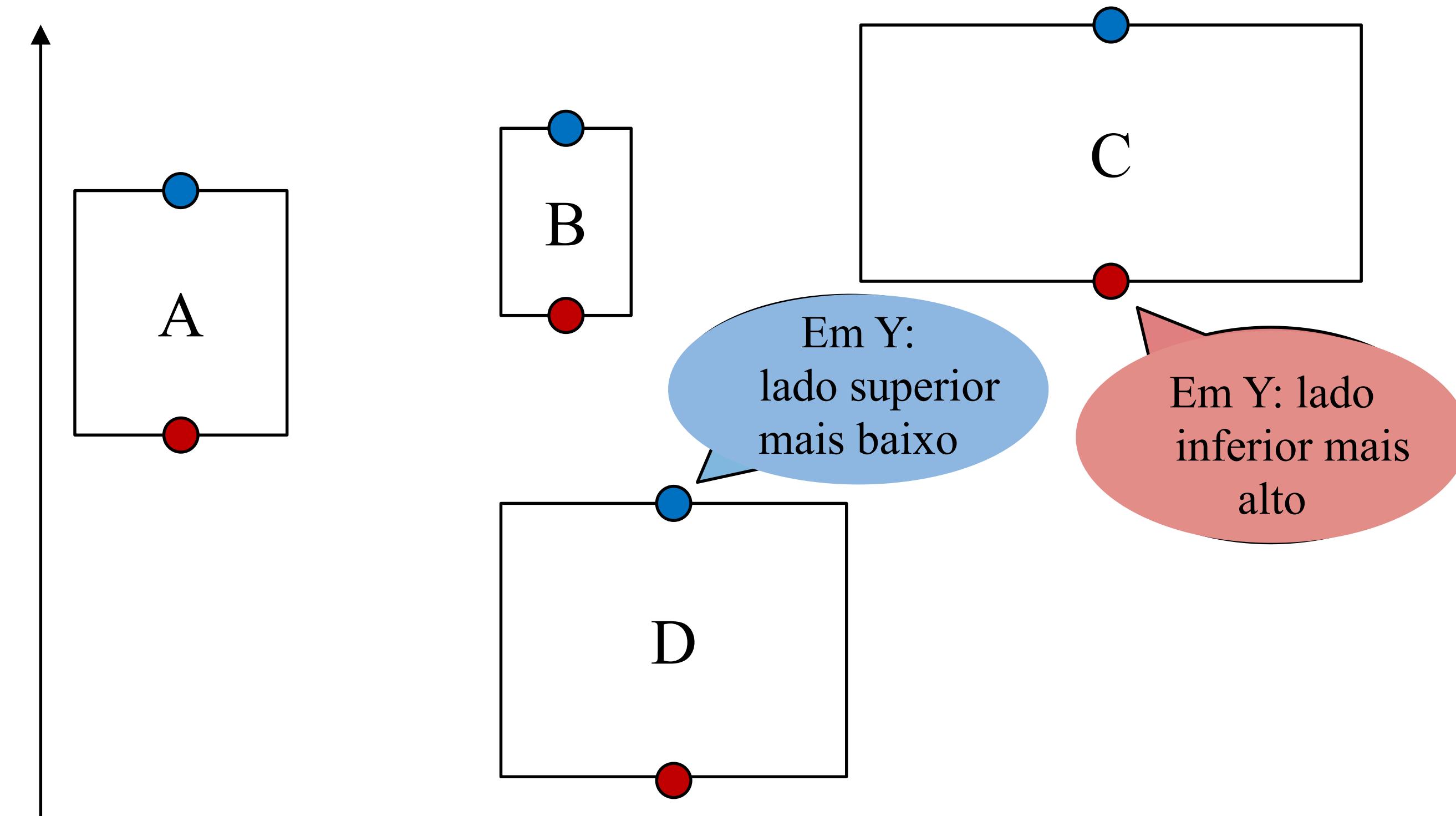
# Split: Algoritmo Linear - Exemplo



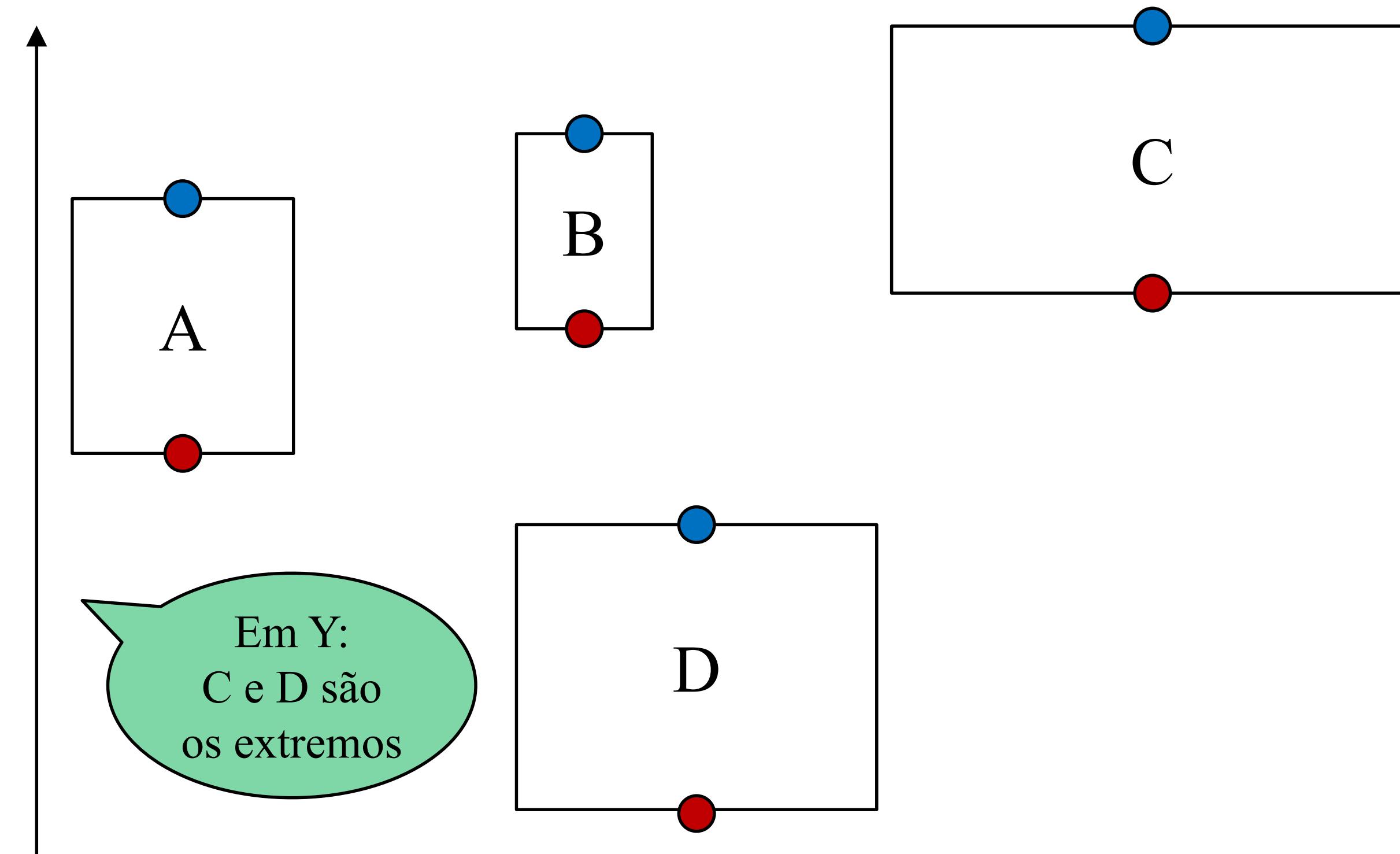
# Split: Algoritmo Linear - Exemplo



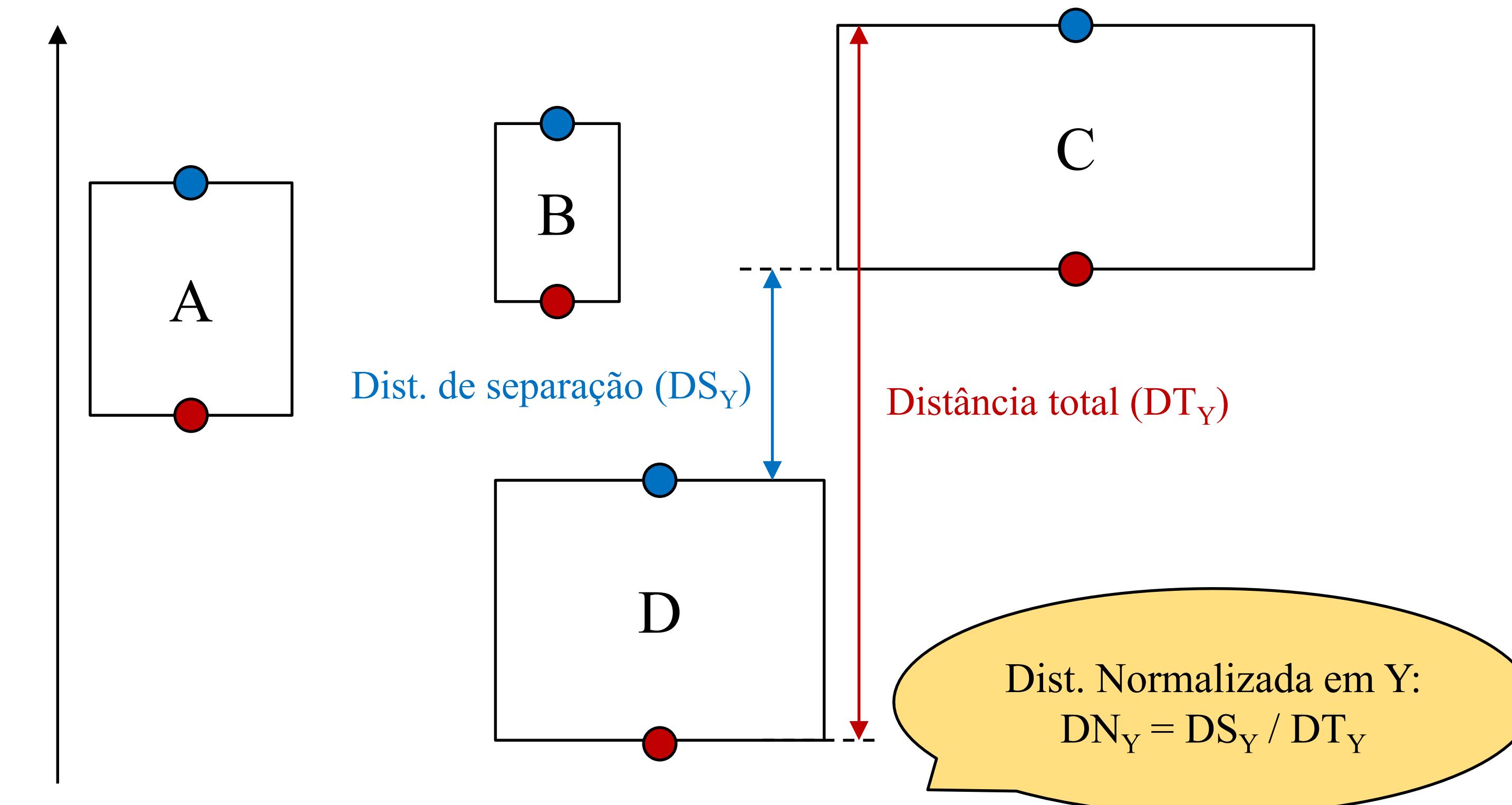
# Split: Algoritmo Linear - Exemplo



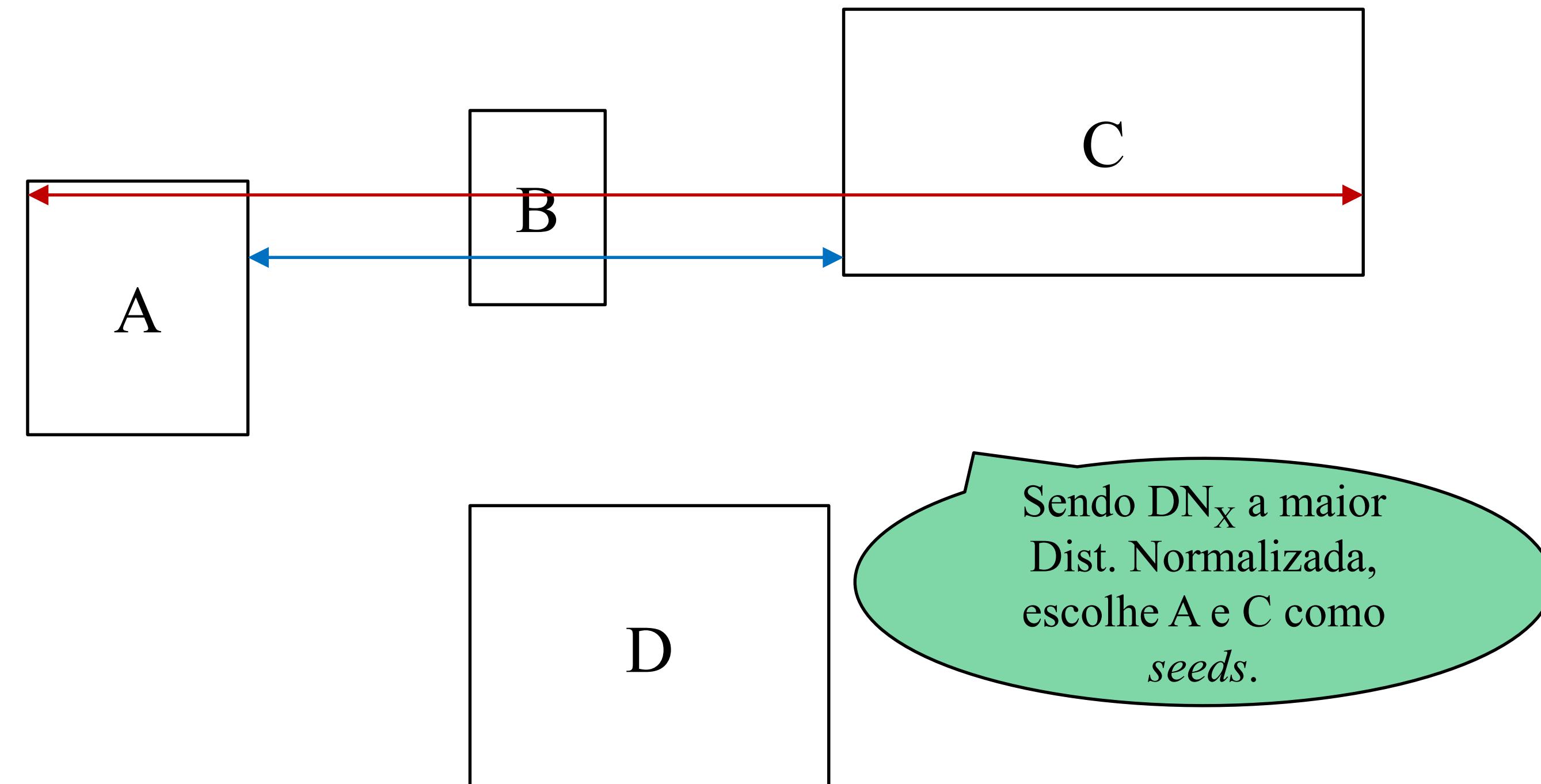
# Split: Algoritmo Linear - Exemplo



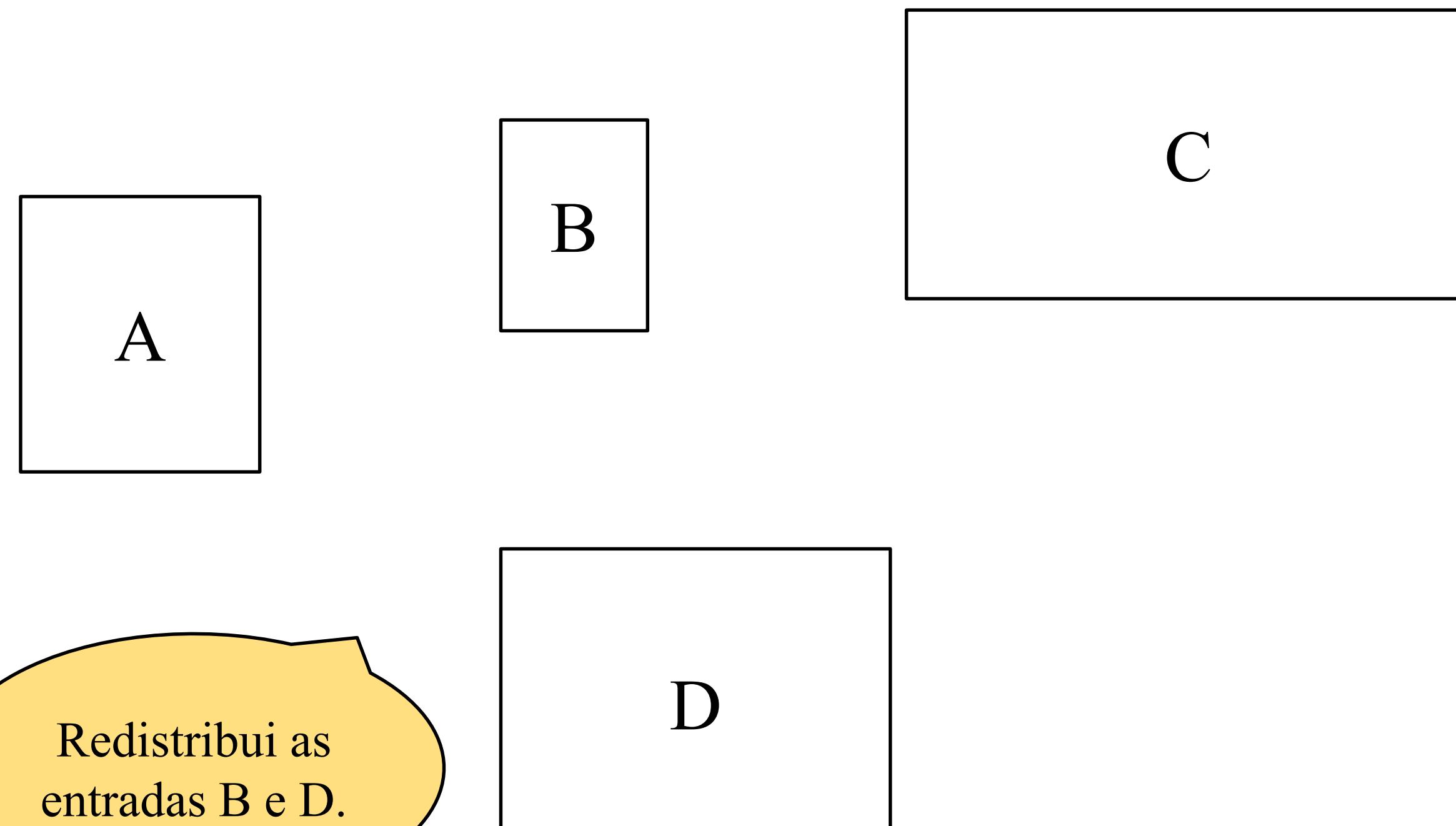
# Split: Algoritmo Linear - Exemplo



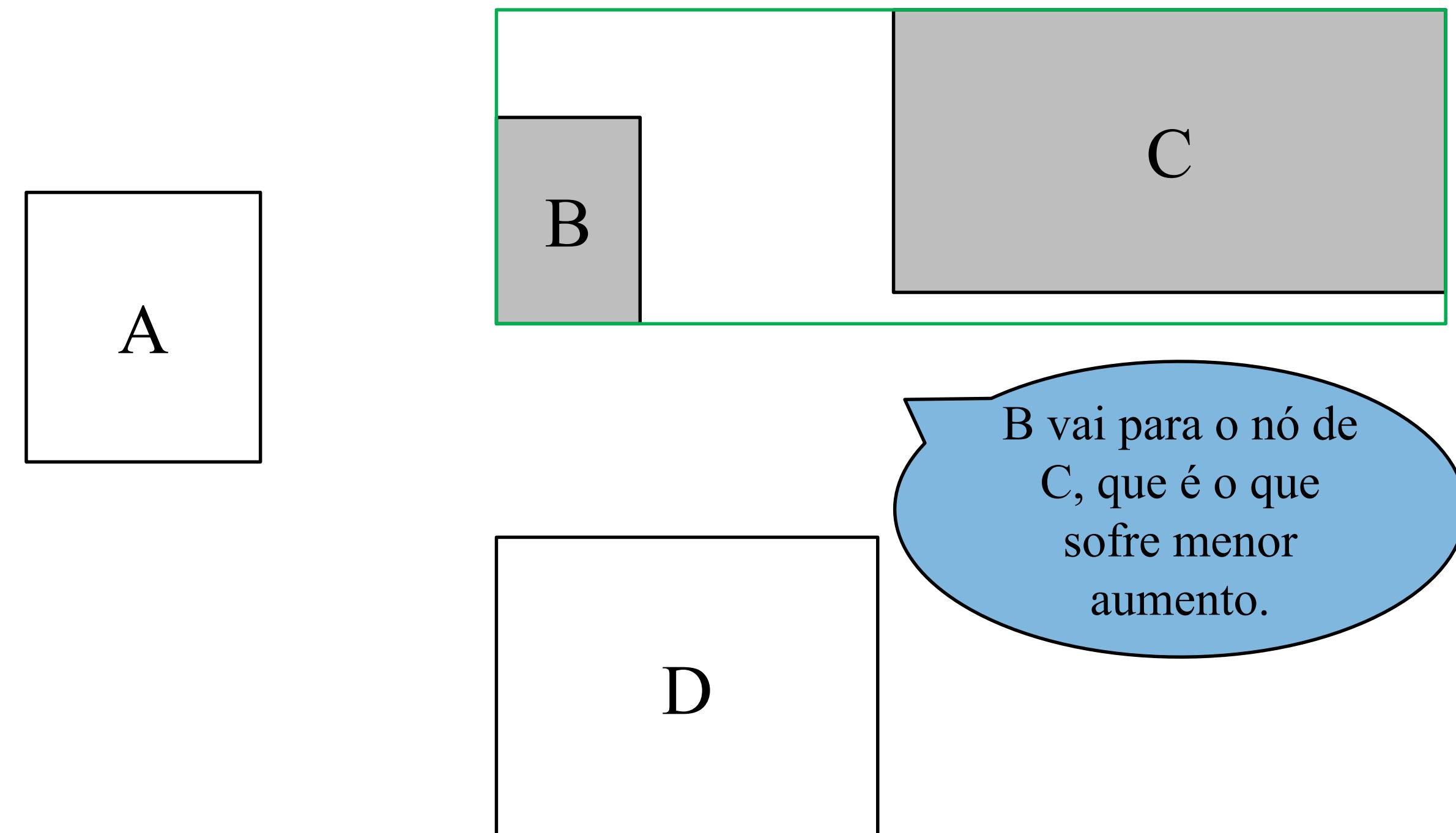
# Split: Algoritmo Linear - Exemplo



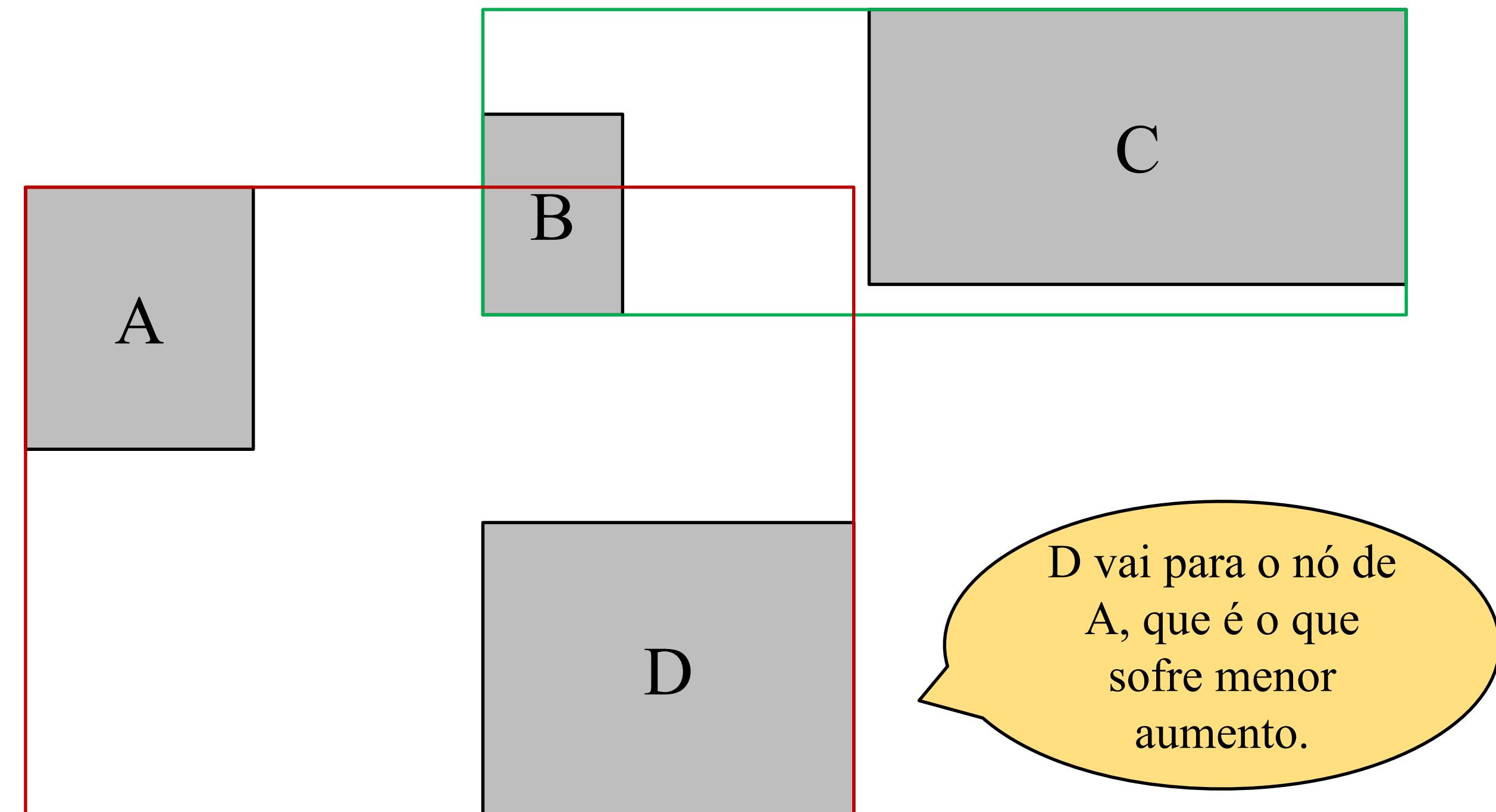
# Split: Algoritmo Linear - Exemplo



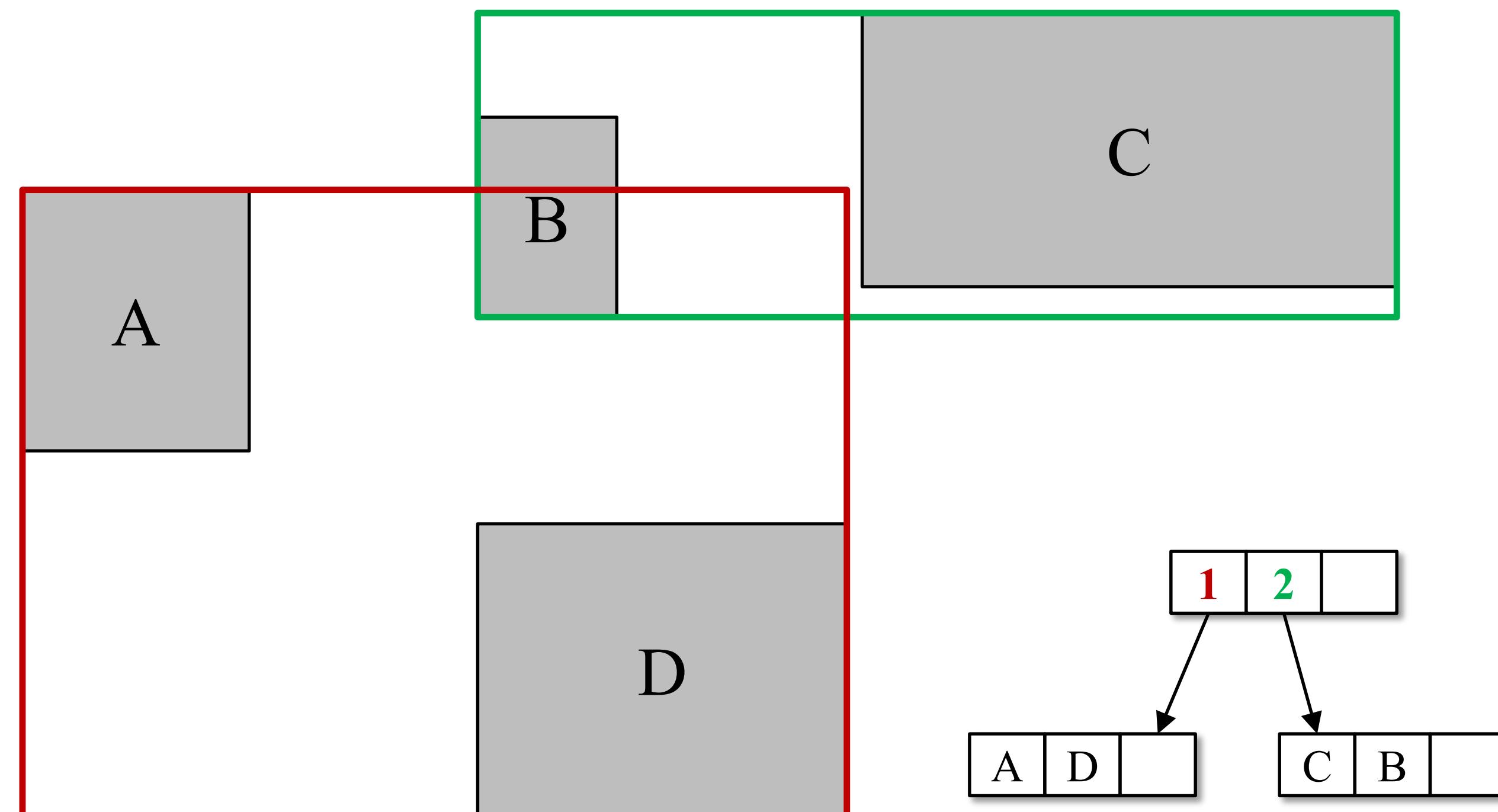
# Split: Algoritmo Linear - Exemplo



# Split: Algoritmo Linear - Exemplo



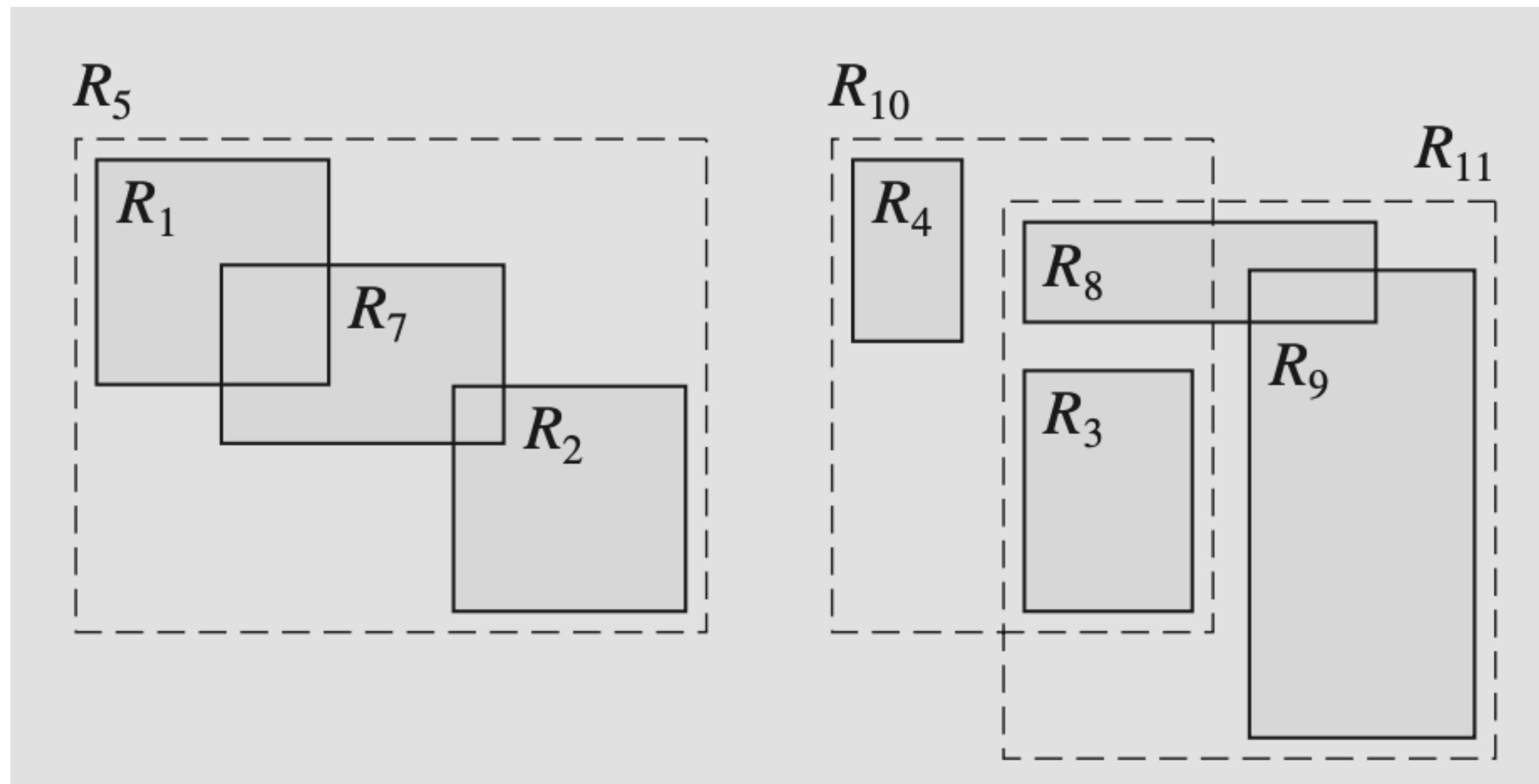
# Split: Algoritmo Linear - Exemplo



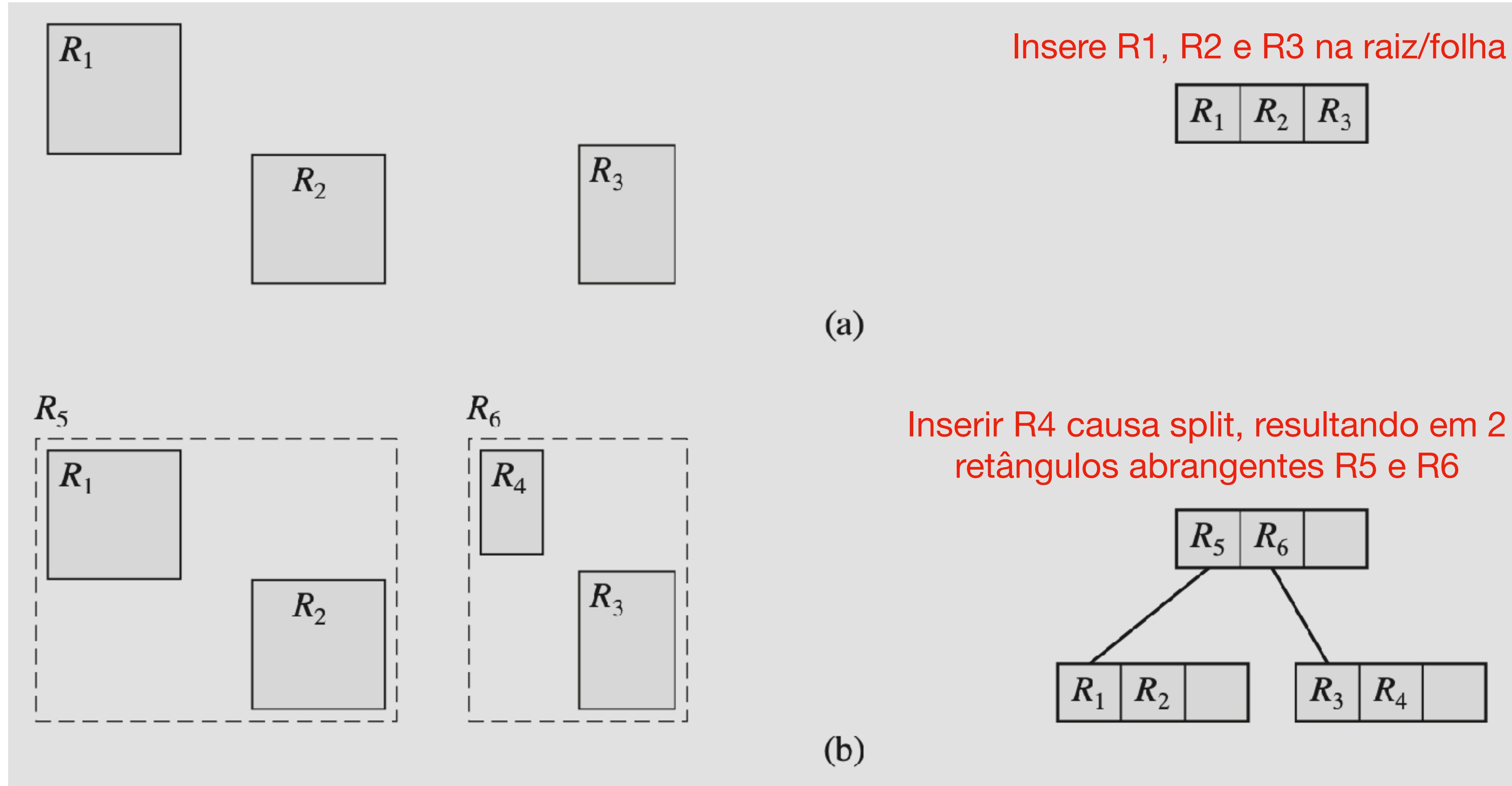
# Exercício



- Considerando o mapa final abaixo e a seguinte ordem de inserção dos objetos: **R1, R2, R3, R4, R7, R8, R9**. Construa a Árvore R(2,3) passo a passo a cada split, explicando e desenhando como fica o mapa a cada passo.



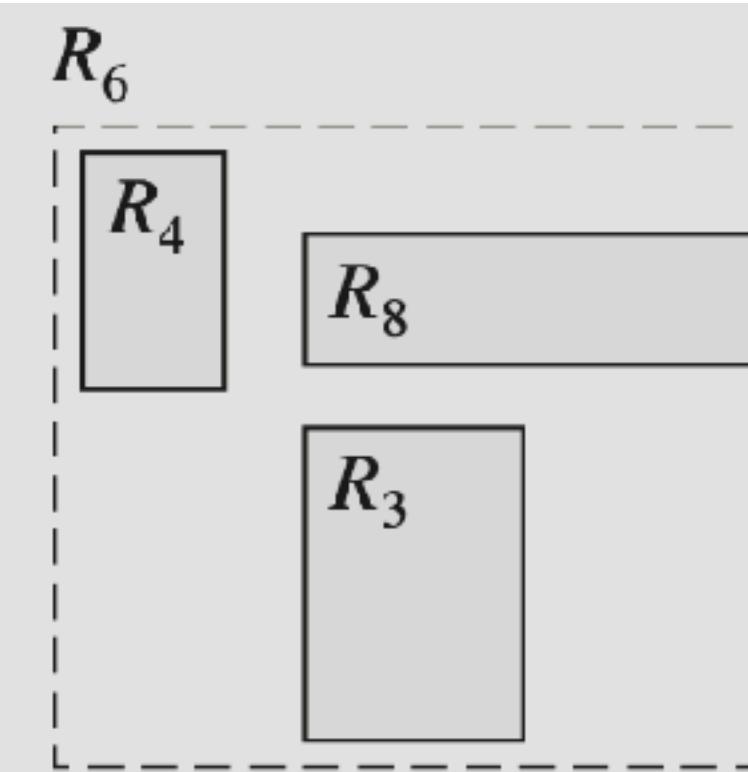
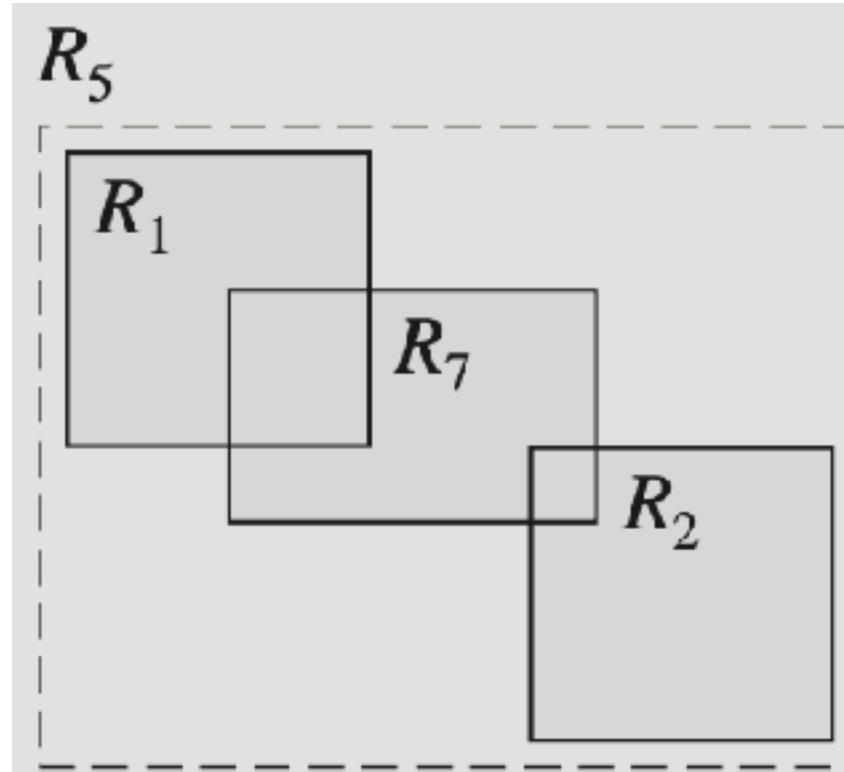
## ● Solução:



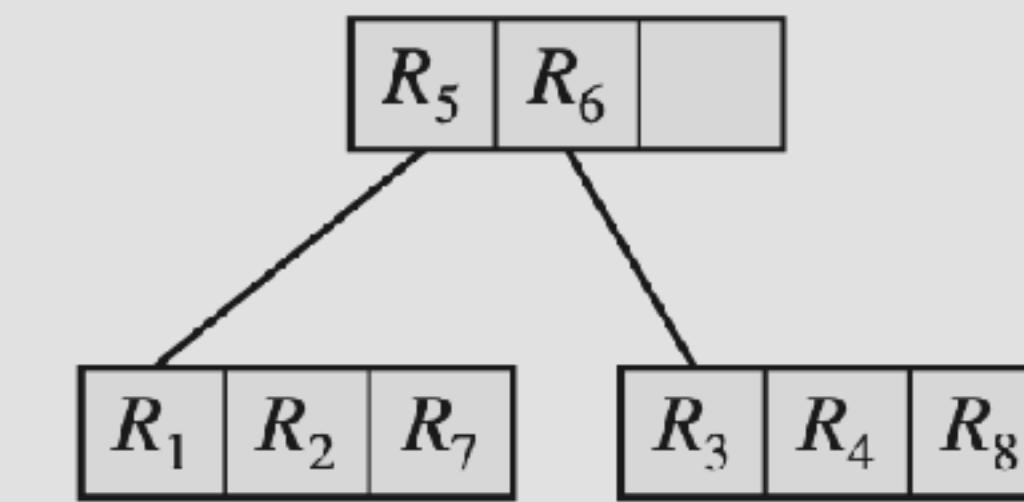
# Exercício



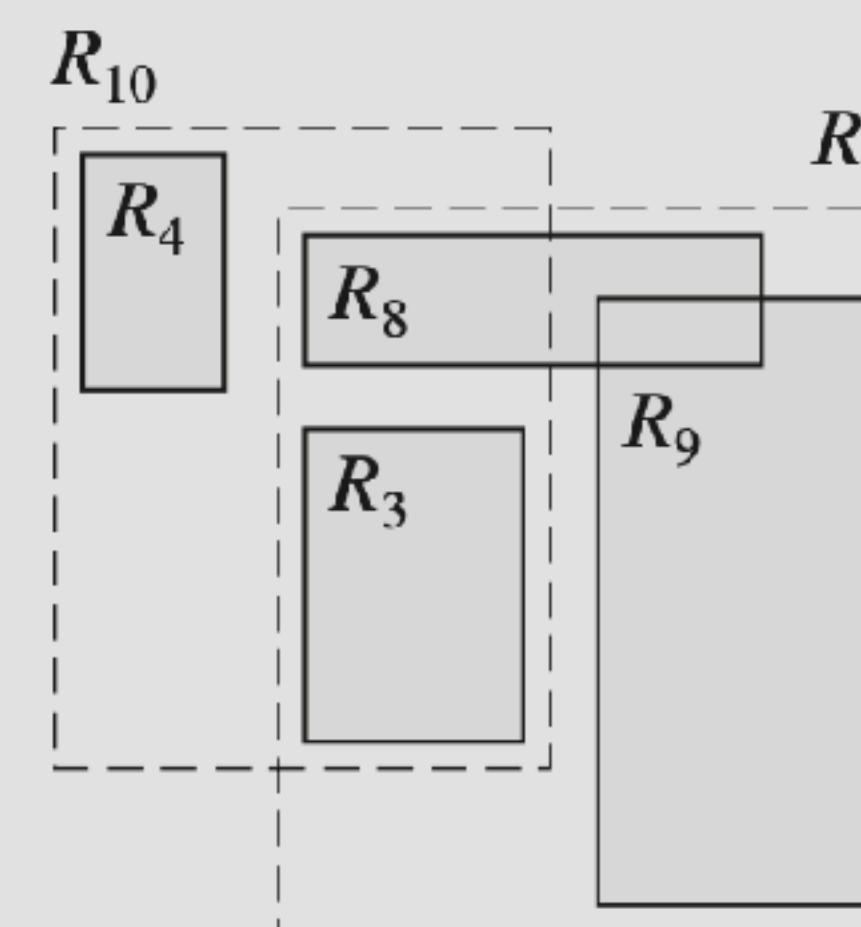
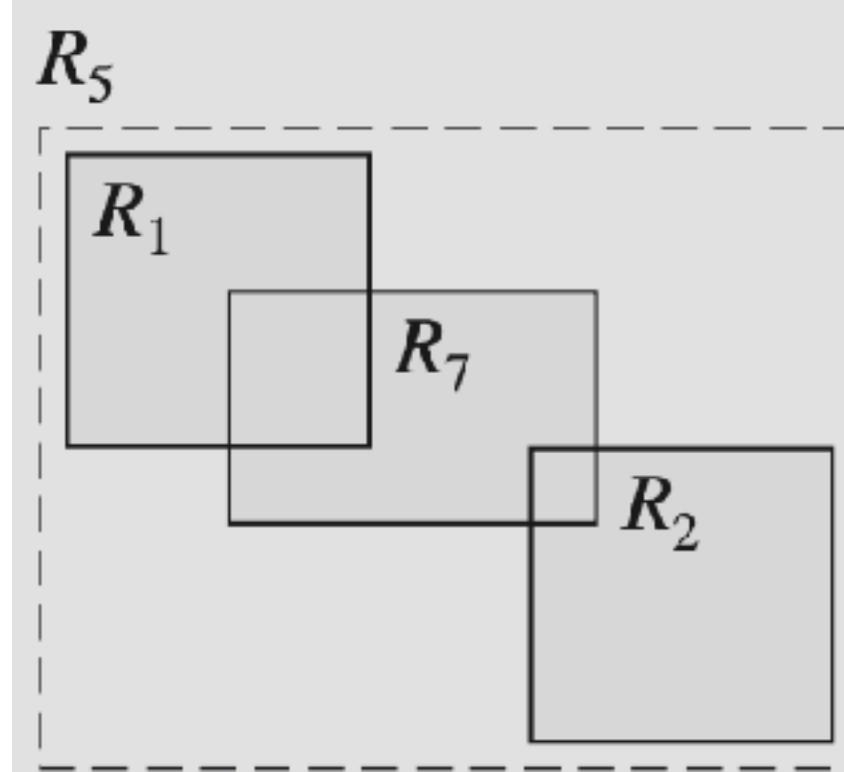
## ● Solução:



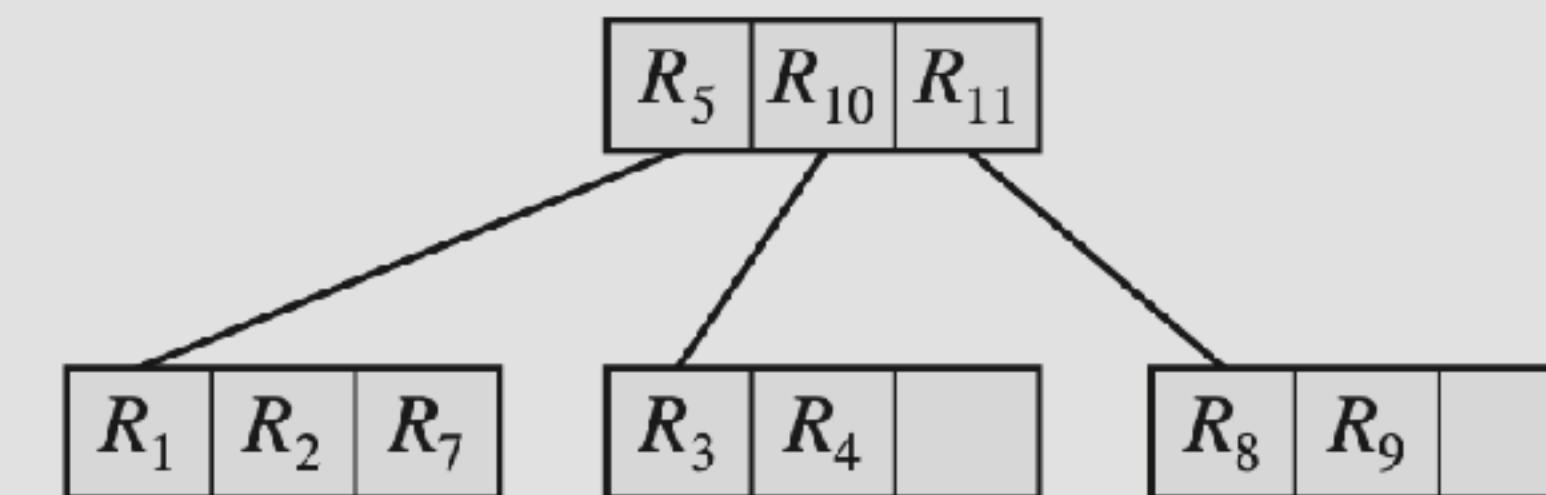
Inserir  $R_7$  não muda nada  
Inserir  $R_8$  faz aumentar  $R_6$  para acomodar  $R_8$



(c)



Inserção de  $R_9$  causa split.  $R_6$  é descartado e  $R_{10}$  e  $R_{11}$  são criados

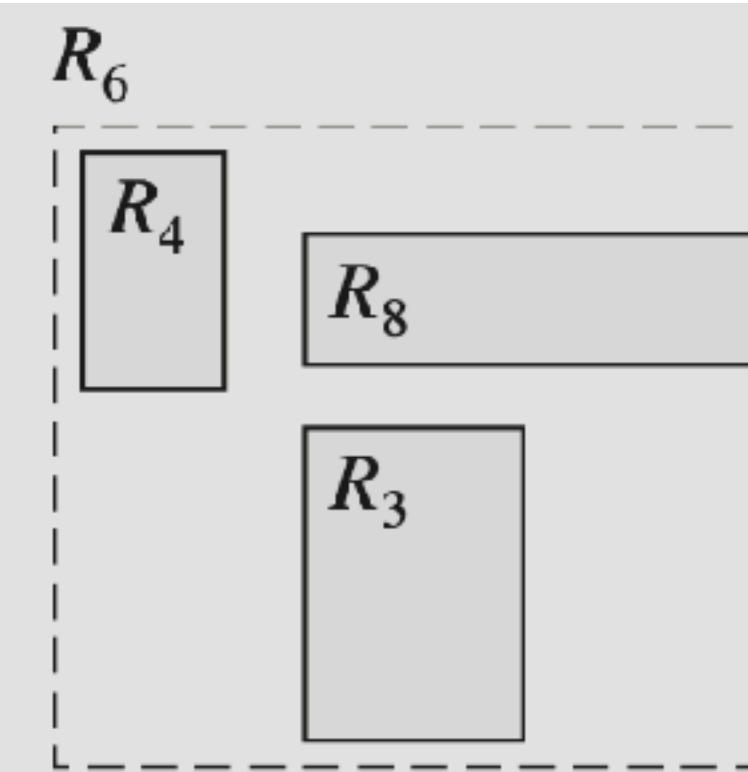
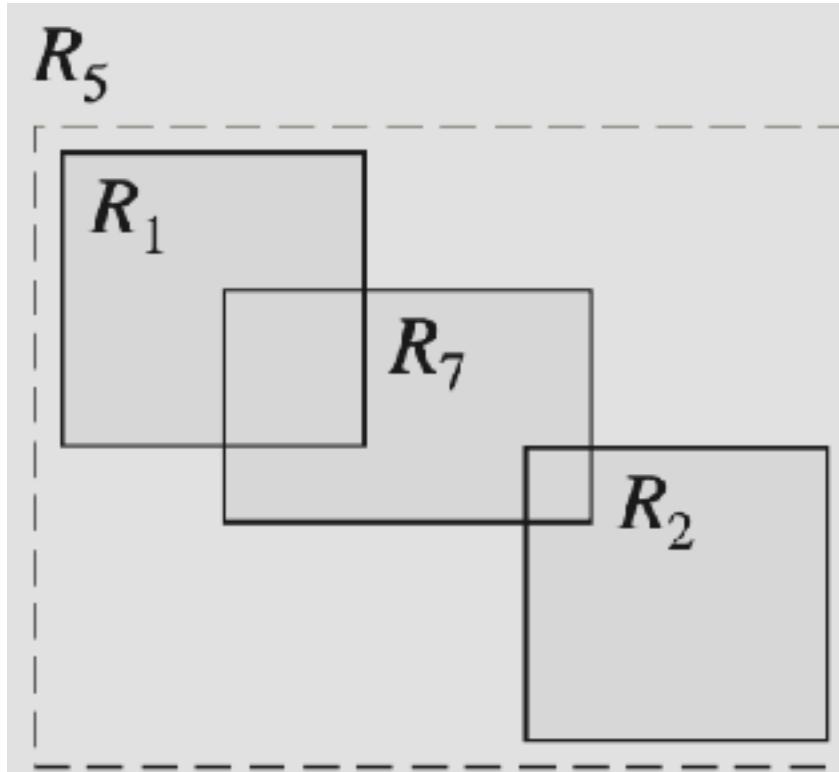


(d)

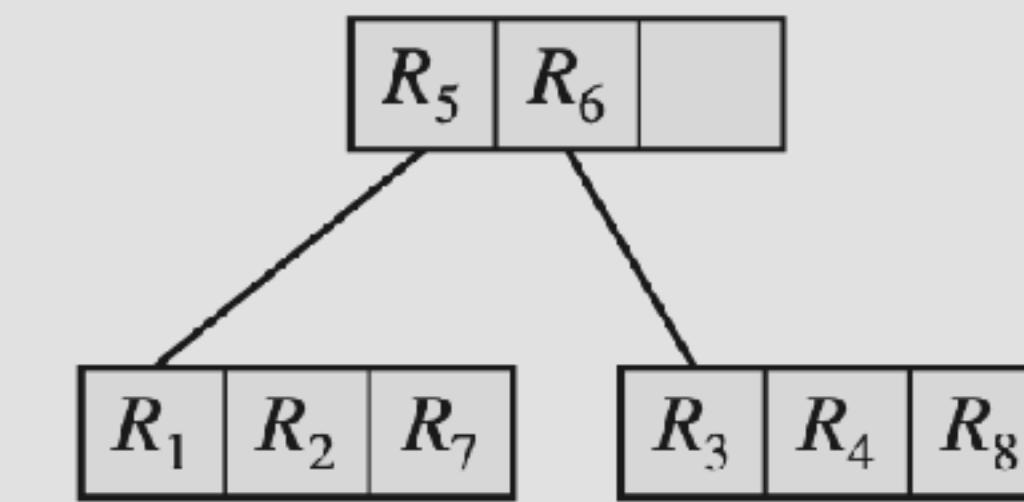
# Exercício



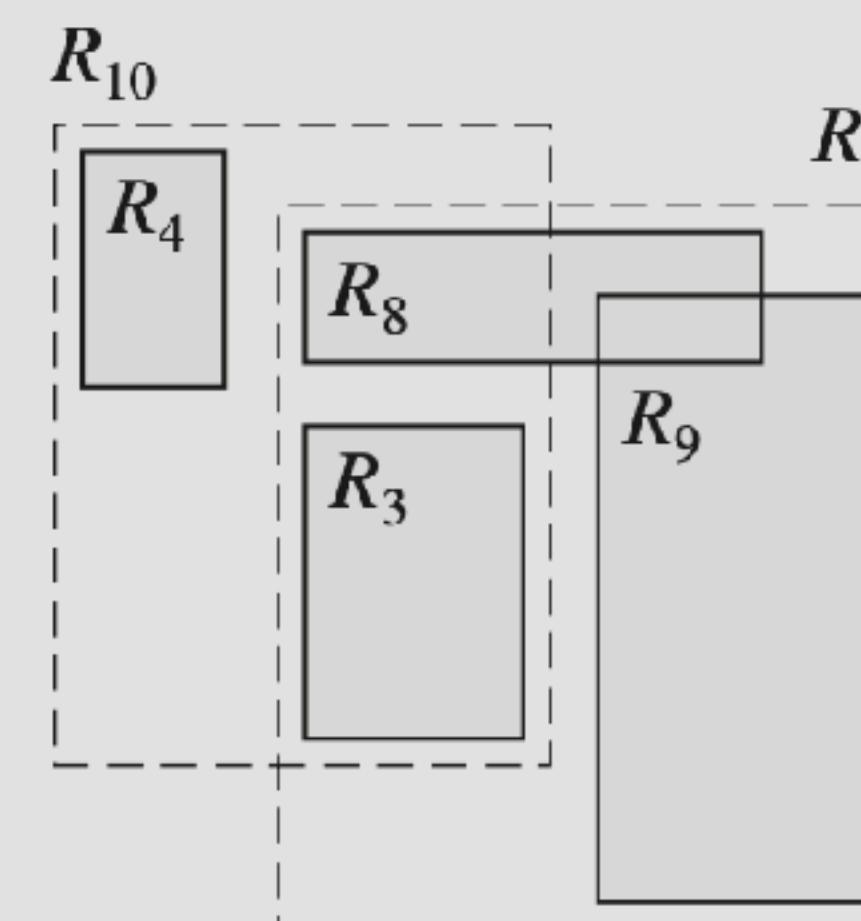
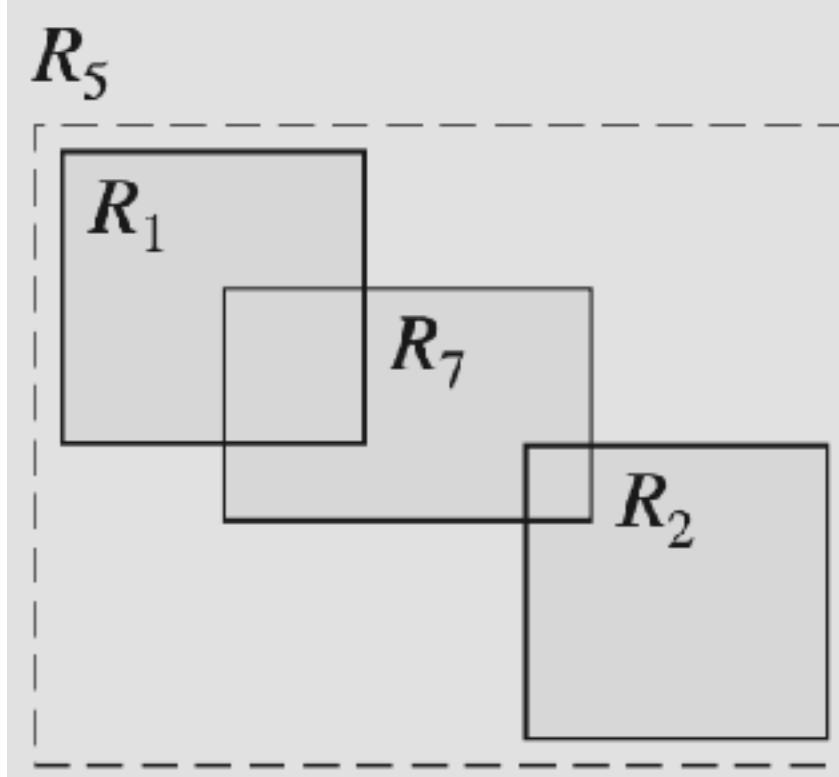
## ● Solução:



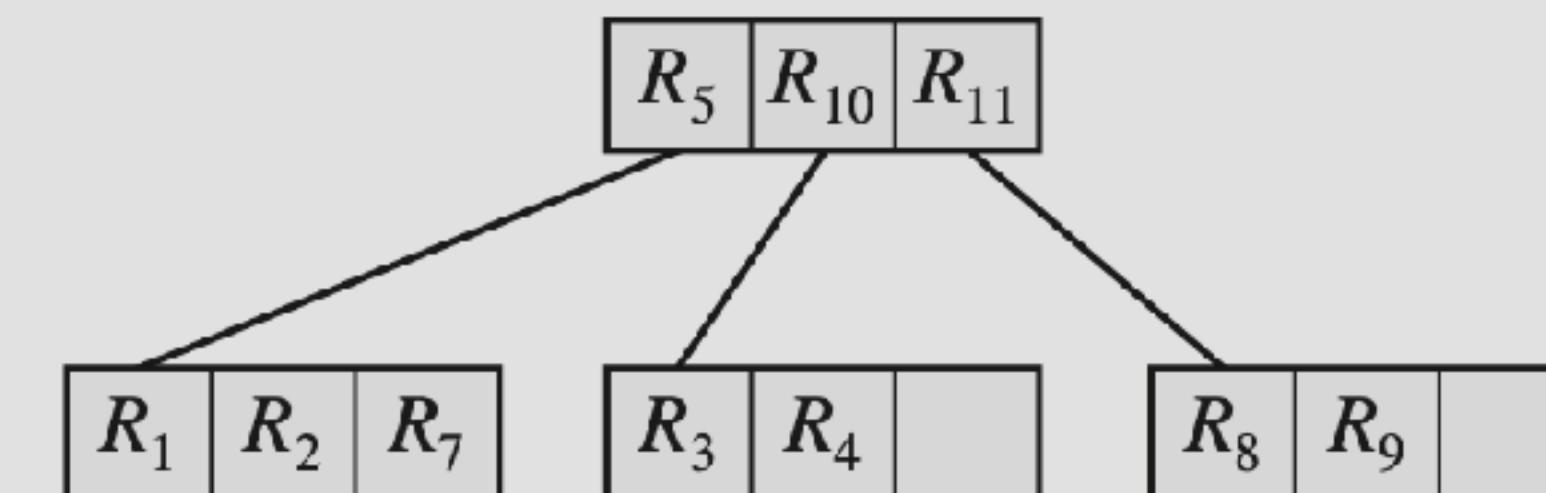
Inserir  $R_7$  não muda nada  
Inserir  $R_8$  faz aumentar  $R_6$  para acomodar  $R_8$



(c)



Inserção de  $R_9$  causa split.  $R_6$  é descartado e  $R_{10}$  e  $R_{11}$  são criados



(d)



- Adam Drozdek, Data Structures and Algorithms in C++. 4 ed. 2012
- Contém conteúdo extraído de notas de aula disciplina SCC 5789 – Base de Dados - Árvore R. Profa. Dra. Cristina Dutra de Aguiar Ciferri