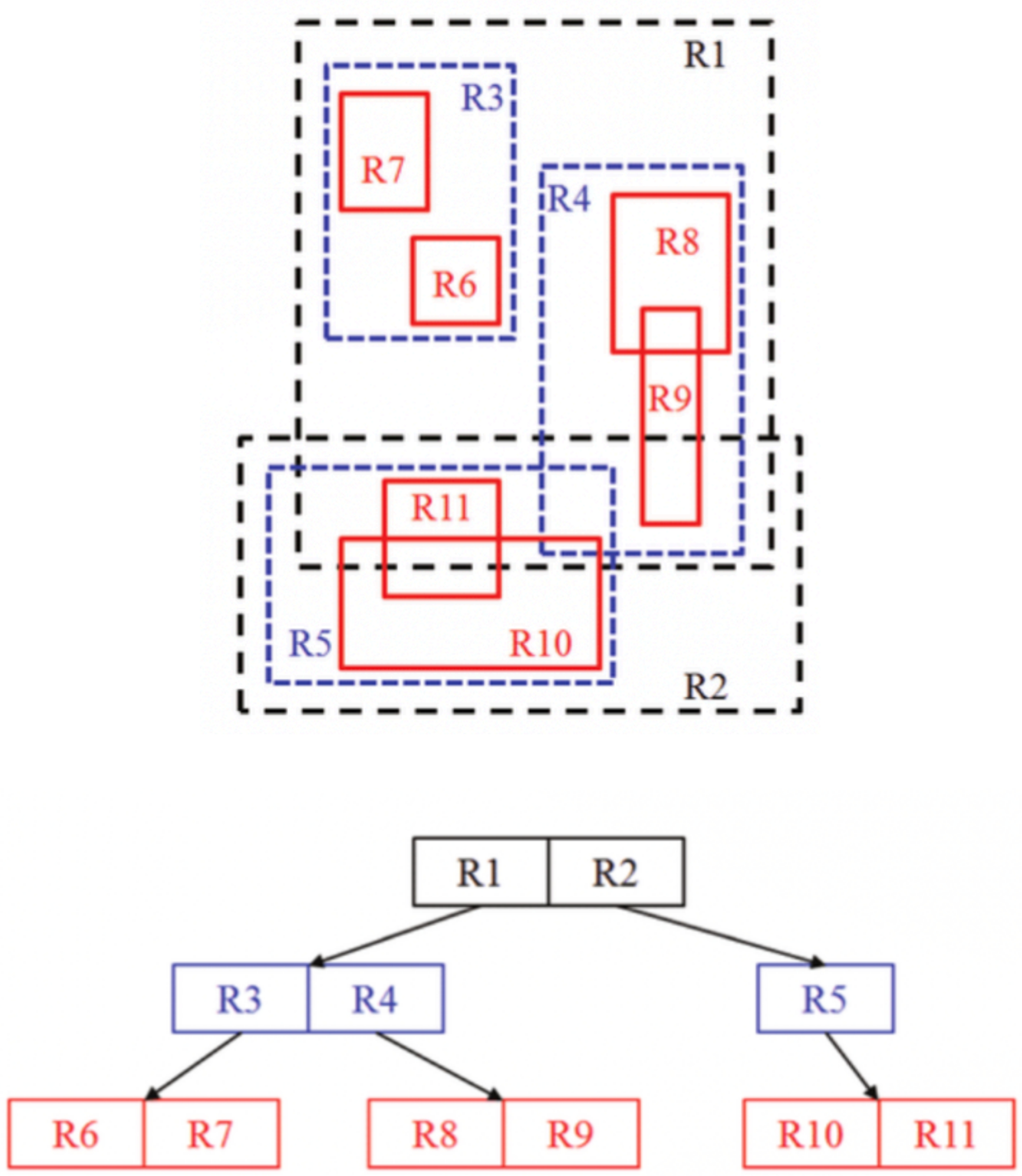
R-Trees

NDBI007: Practical class 6

R-Tree

- Height-*balanced* tree
  - Extension of B+-tree for *spatial data*
- Nodes correspond to disk pages
- Inner node contains n-dimensional bounding box I
  - MBRs (*minimum bounding rectangle*)
  - MBR of a node in MBR of all children
- Leaf level contains pointers to the spatial objects



Splitting in R-Tree: Guttman

- First, we *identify a pair of elements* which would result in the *largest dead space*
  - I.e., we apply method *PickSeeds*
- Next, remaining elements are added one by one
  - If remaining entries need to be assigned into node in order to have the *minimum number of entries*, then assign them
  - Otherwise, Pick the one that would make the *biggest difference in area enlargement* when put to one of two groups (method *PickNext*)
    - Add in to the one with the least difference

SplitNode(P, PP, E)
Input: node P, new node PP, m original entries, new entry E
Output: modified P, PP

PickSeeds(); {chooses first E_i and E_j for P and PP }
WHILE not assigned entry exists DO
  IF remaining entries need to be assigned to P or PP in order to have the minimum number of entries m
    THEN assign them;
  ELSE
    $E_i \longleftarrow$ **PickNext()** {choose where to assign next entry}
    Add E_i into group that will have to be enlarged least to accommodate it. Resolve ties by adding the entry to the group with smaller area, then to the one with fewer entries;

PickSeeds()
FOREACH E_i, E_j $(i \neq j)$ DO
  $d_{ij} \longleftarrow$ area(J) - area(E_i.) - area (E_J.I);
    { J is the MBR covering E_i and E_j }
  pick E_i and E_j with maximal d_{ij};

PickNext()
FOREACH remaining E_i DO
  $d_1 \longleftarrow$ area increase required for MBR of P and E_i.I;
  $d_2 \longleftarrow$ area increase required for MBR of PP and E_i.I;
pick E_i with maximal $|d_1 - d_2|$;

Example 6.1: Guttman's Split

- Split the following overflown node with Guttman's splitnode method
  - The maximum number of items in a node is $M = 8$
  - The minimum number of items in a node is $m = 3$

D							A		
D						A	A	A	
D		F	F	F			A		
					E		I		
		G	G		E		I		
					E				
B			H					C	
B	B		H				C	C	C
B									

Example 6.1: Guttman's Split (Continued)

- We apply Guttman's PickSeeds method to find two elements having the largest dead space if being placed together

Pair	Overall area	Area of the objects	Dead space
AB	9x10=90	5+4=9	90-9=81
AC	4x9=36	5+4=9	36-9=27
AD	9x3=27	5+3=8	27-8=19
…			
CD	9x10=90	4+3=7	90-7=83
…			
HI	5x5=25	2+2=4	25-4=21

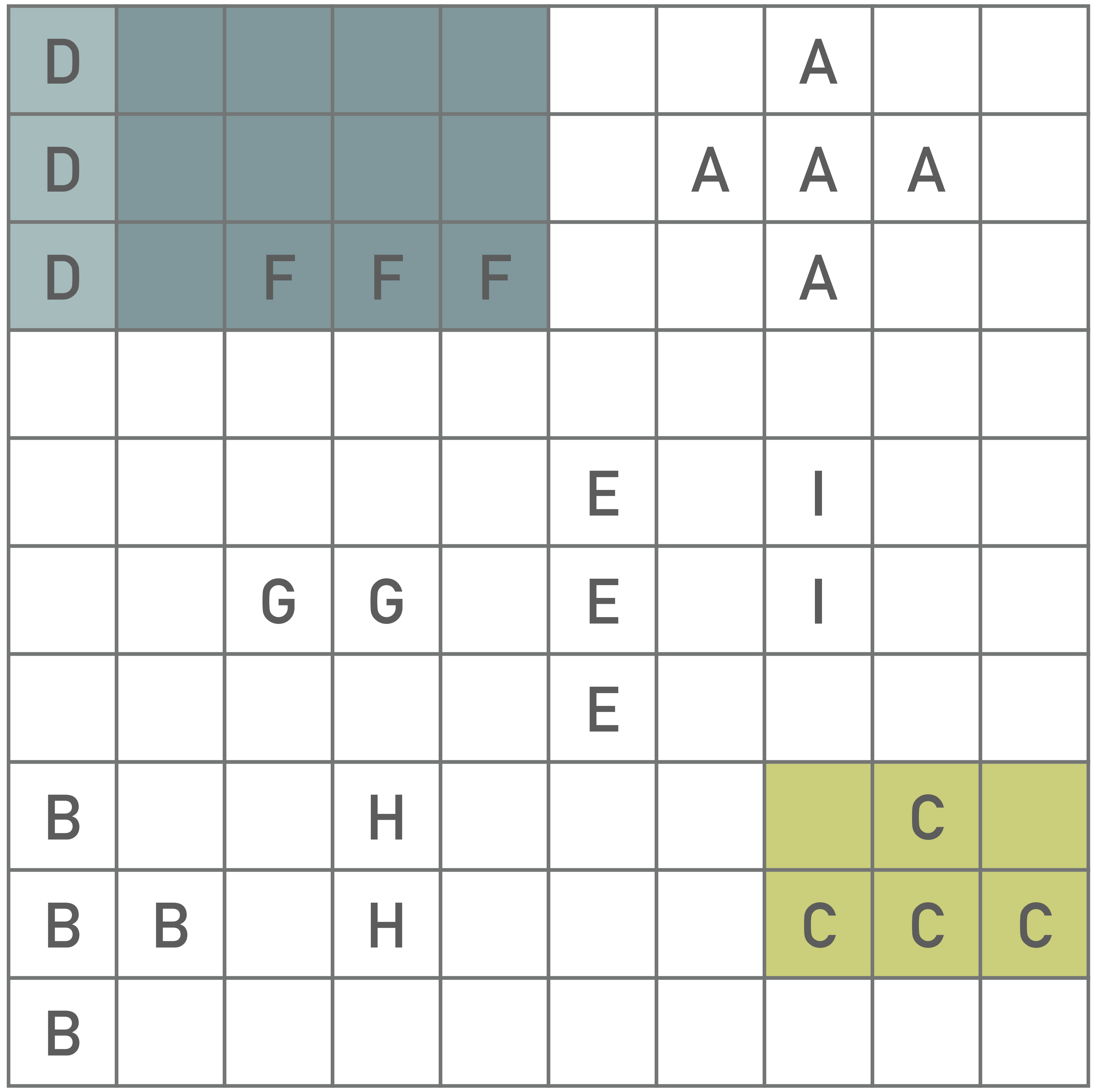
D							A		
D						A	A	A	
D		F	F	F			A		
					E		I		
		G	G		E		I		
					E				
B			H					C	
B	B		H				C	C	C
B									

- The largest dead space has CD thus those will be the seeds of the splitting method

Example 6.1: Guttman's Split (Continued)

- Next, iteratively add such an object into a node which will maximize the difference in the *node area enlargements* if the object was inserted into the first or second node

Object	C	D	Difference
A	4x9-6=30	9x3-3=24	|30-24|=6
B	10x3-6=24	2x10-3=17	|24-17|=7
E	5x5-6=19	6x7-3=39	|19-39|=20
F	8x7-6=50	5x3-3=12	|50-12|=38
G	8x4-6=26	4x6-3=21	|26-21|=5
H	7x2-6=8	4x9-3=33	|8-33|=25
I	3x5-6=9	8x6-3=45	|45-9|=36

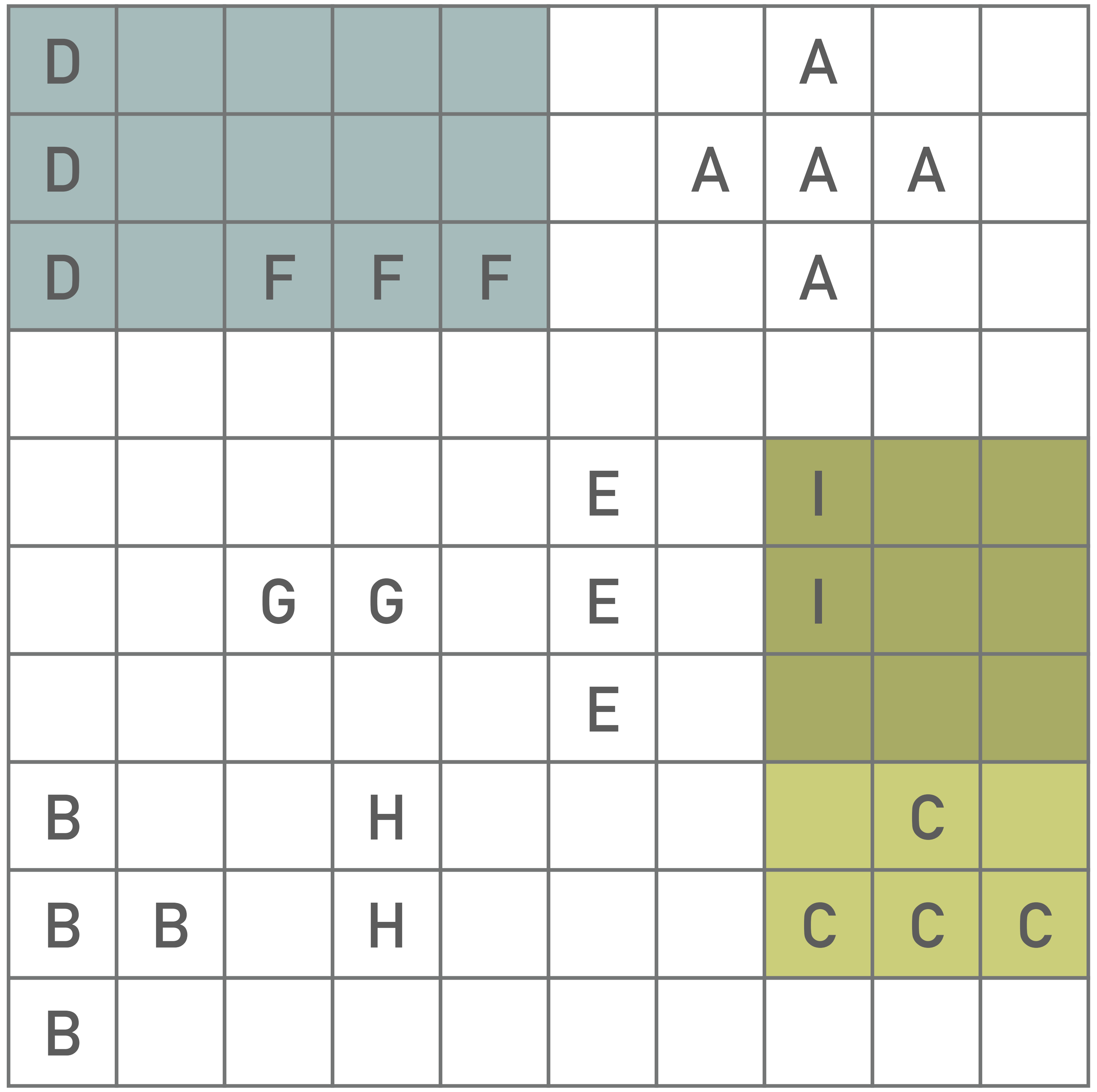


- The biggest difference shows the object F, hence it will be inserted into the node which is closer
- Thus, we have nodes DF and C

Example 6.1: Guttman's Split (Continued)

- Next, iteratively add such an object into a node which will maximize the difference in the node area enlargements if the object was inserted into the first or second node

Object	C	DF	Difference
A	4x9-6=30	9x3-15=12	|30-12|=18
B	10x3-6=24	5x10-15=35	|24-35|=11
E	5x5-6=19	6x7-15=27	|19-27|=8
G	8x4-6=26	5x6-15=15	|26-15|=11
H	7x2-6=8	5x9-15=30	|8-30|=22
I	3x5-6=9	8x6-15=33	|9-33|=24

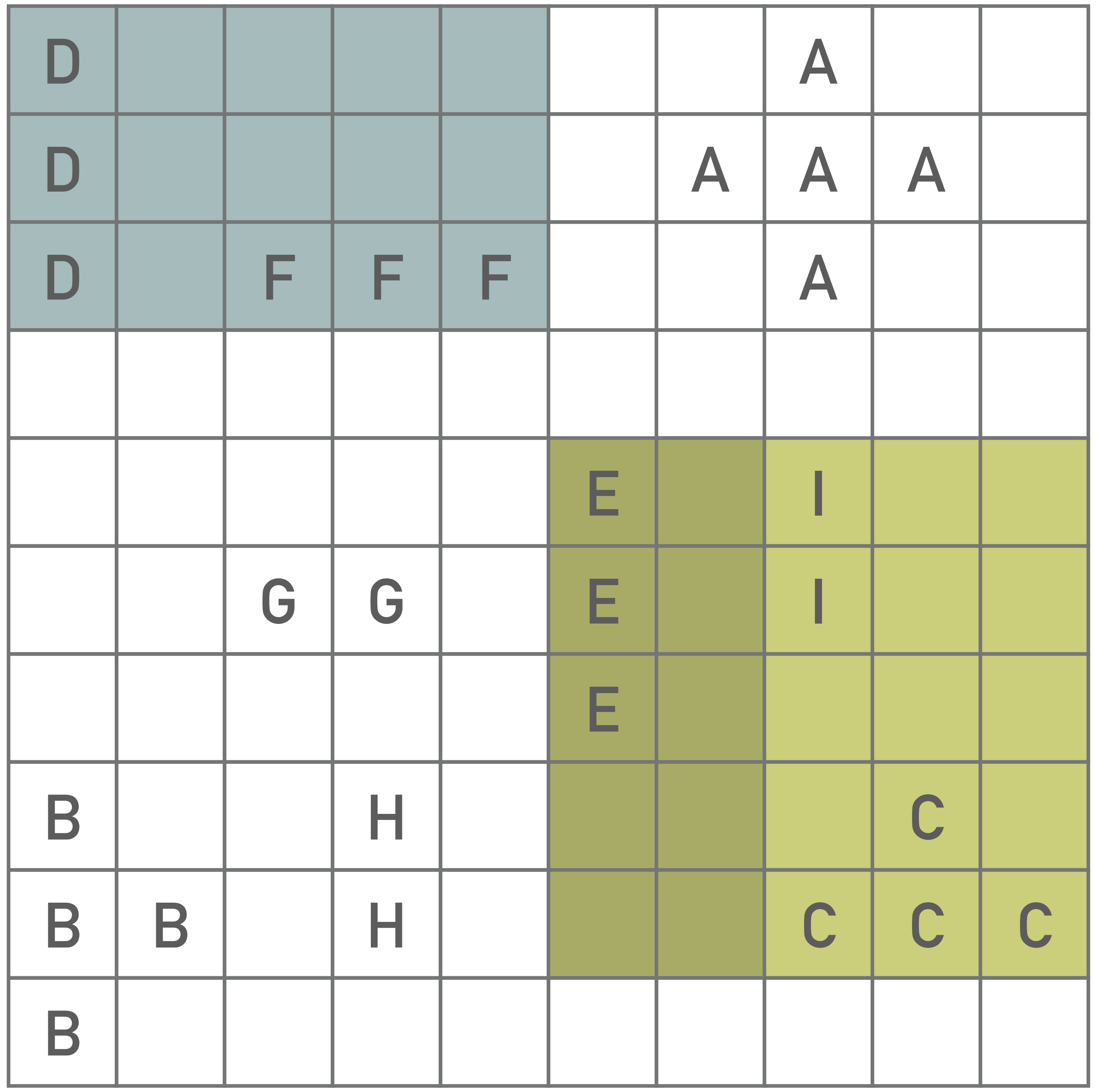


- The biggest difference shows the object I, hence it will be inserted into the node which is closer, i.e., C
- Thus, we have nodes CI and DF

Example 6.1: Guttman's Split (Continued)

- Next, iteratively add such an object into a node which will maximize the difference in the node area enlargements if the object was inserted into the first or second node

Object	CI	DF	Difference
A	4x9-15=21	9x3-15=12	|21-12|=9
B	10x6-15=45	5x10-15=35	|45-35|=10
E	5x5-15=10	6x7-15=27	|10-27|=17
G	8x5-15=25	5x6-15=15	|25-15|=10
H	7x5-15=20	5x9-15=30	|20-30|=10

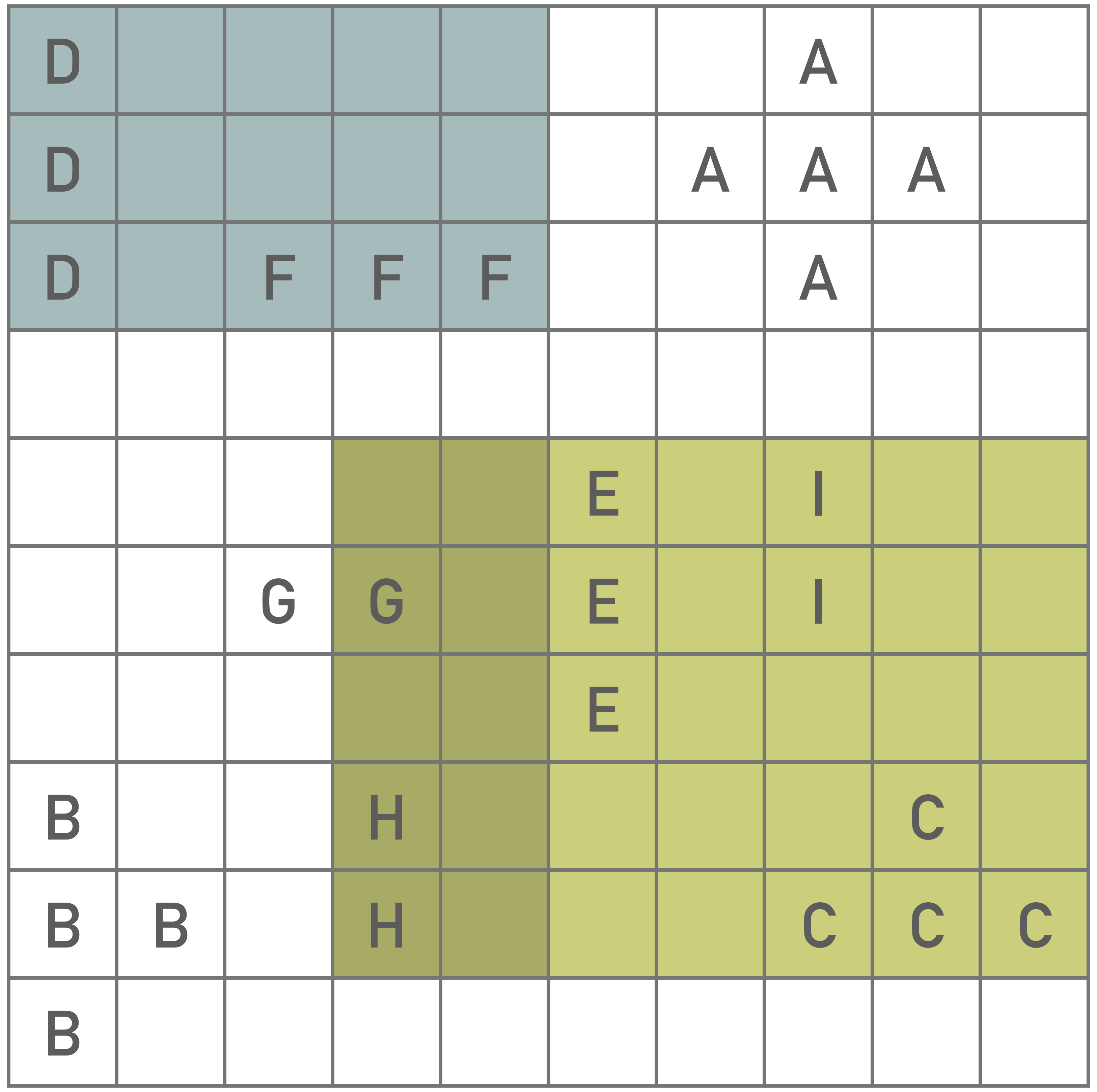


- The biggest difference shows the object E, hence it will be inserted into the node which is closer, i.e., CI
- Thus, we have nodes CIE and DF

Example 6.1: Guttman's Split (Continued)

- Next, iteratively add such an object into a node which will maximize the difference in the node area enlargements if the object was inserted into the first or second node

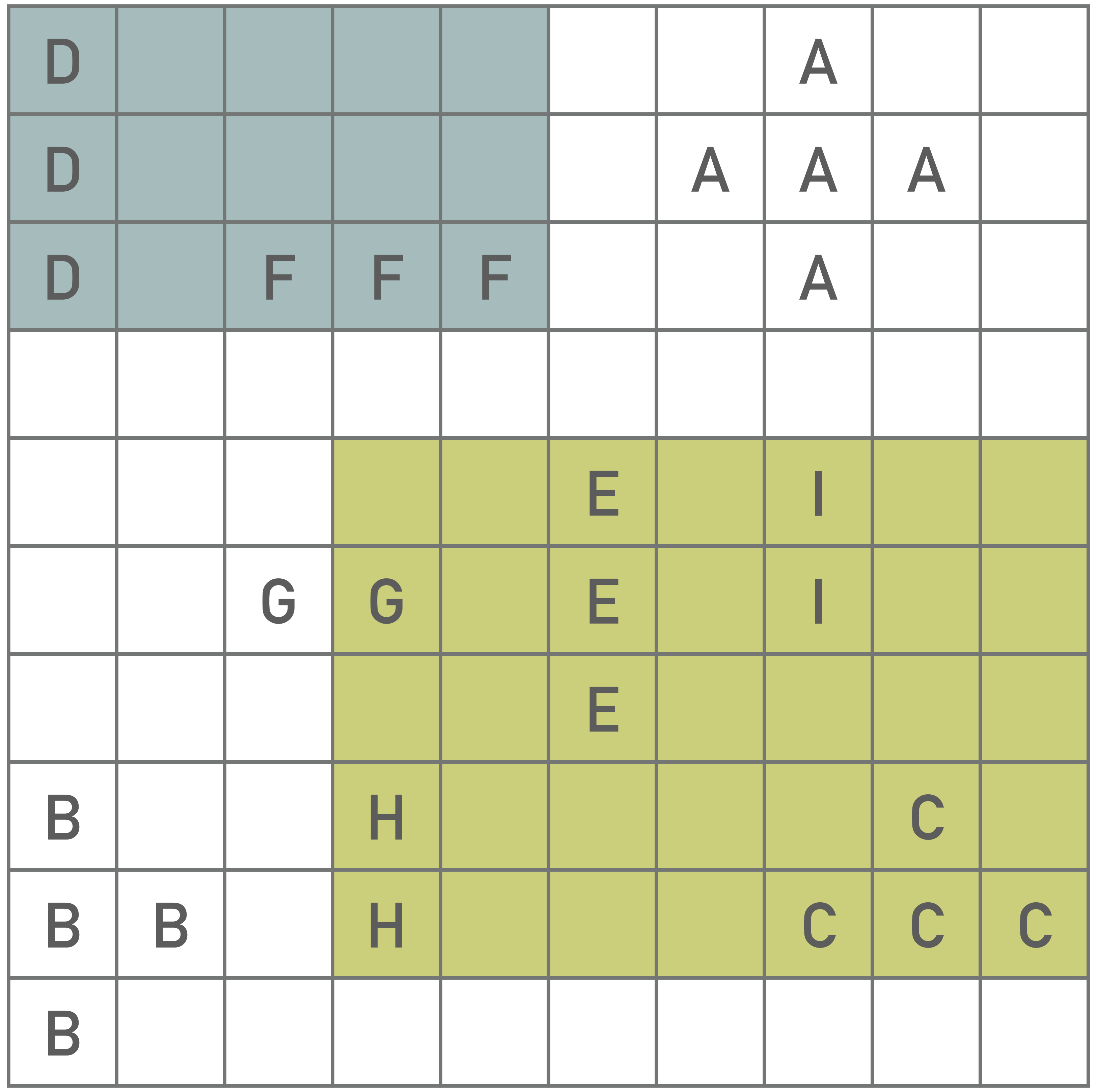
Object	CIE	DF	Difference
A	5x9-25=20	9x3-15=12	|20-12|=8
B	10x6-25=35	5x10-15=35	|35-35|=0
G	8x5-25=15	5x6-15=15	|15-15|=0
H	7x5-25=10	5x9-15=30	|10-30|=20



- The biggest difference shows the object H, hence it will be inserted into the node which is closer, i.e., CIE
- Thus, we have nodes CIEH and DF

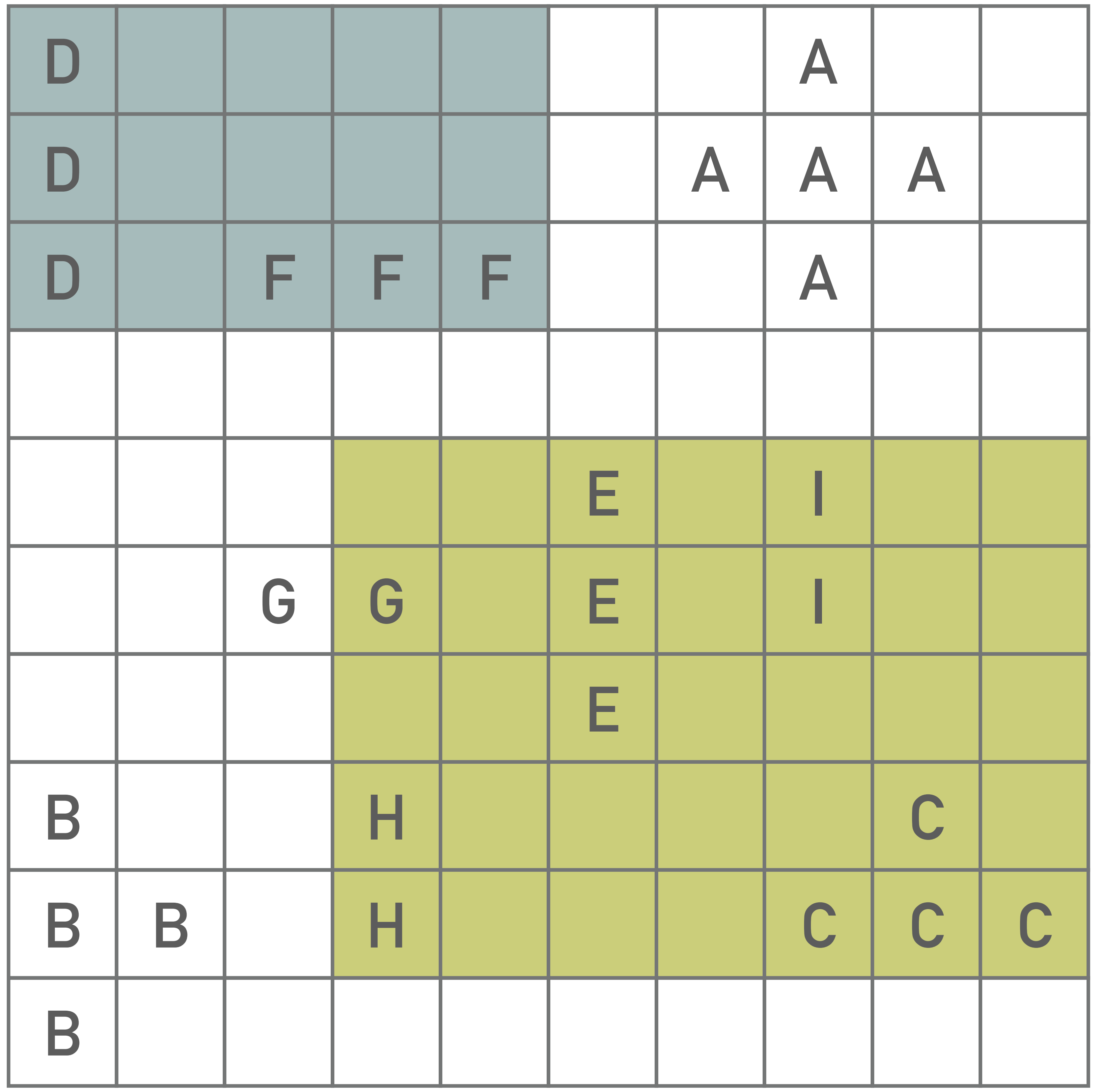
Exercise 6.2

- Finish splitting of the overflown node
  - Continue with Guttman's method
  - The maximum number of items in a node is $M = 8$
  - The minimum number of items in a node is $m = 3$
- If there are more options to choose, explain the reason of yours choice



Exercise 6.3

- Finish splitting of the overflown node
  - Continue with Guttman's method
  - The maximum number of items in a node is $M = 8$
  - This time, the minimum number of items in a node is $m = 4$, i.e., $m = M/2$
- If there are more options to choose, explain the reason of yours choice
- Compare and comment the results of exercises 6.2 and 6.3



Splitting in R-Tree: Greene

- Modification of the split algorithm in original R-Tree (Guttman)
- Splitting is based on a hyperplane which defines in which node the objects will fall
  - I.e., it splits objects into two groups
- We choose an Axis
  - *PickSeeds* work identically to the Guttman's
  - We compute the *normalized distances of each axis* and select the axis having the highest value
- Next, we *order the objects* based on the selected axis
- Finally, we *redistribute* the objects

SplitNode(P, PP, E)
ChooseAxis();
Distribute();

ChooseAxis()
PickSeeds; { from Guttman's version - returns seeds E_i and E_j }
For every axis compute the distance between MBRs E_i, E_j;
Normalize the distance by the respective edge length of the bounding rectangle of the original node;
Pick the axis with greatest normalized separation;

Distribute()
Sort E_is in the choosen axis j based on the j-th coordinate;
Add first $\lceil (M+1)/2 \rceil$ records into P and rest of them into PP;

Example 6.4: Greene's Split

- Split the following overflown node with Greene's split method
  - The maximum number of items in a node is $M = 8$
  - The minimum number of items in a node is $m = 3$

- I.e., execute the following methods:
  - PickSeeds (Guttman's)
  - ChooseAxis
  - Distribute (ordering and placement)

D							A		
D						A	A	A	
D		F	F	F			A		
					E		I		
		G	G		E		I		
					E				
B			H					C	
B	B		H				C	C	C
B									

Example 6.4: Greene's Split (Continued)

- We apply Guttman's PickSeeds method to find two elements having the largest dead space if being placed together

Pair	Overall area	Area of the objects	Dead space
AB	9x10=90	5+4=9	90-9=81
AC	4x9=36	5+4=9	36-9=27
AD	9x3=27	5+3=8	27-8=19
...			
CD	9x10=90	4+3=7	90-7=83
...			
HI	5x5=25	2+2=4	25-4=21

D							A		
D						A	A	A	
D		F	F	F			A		
					E		I		
		G	G		E		I		
					E				
B			H					C	
B	B		H				C	C	C
B									

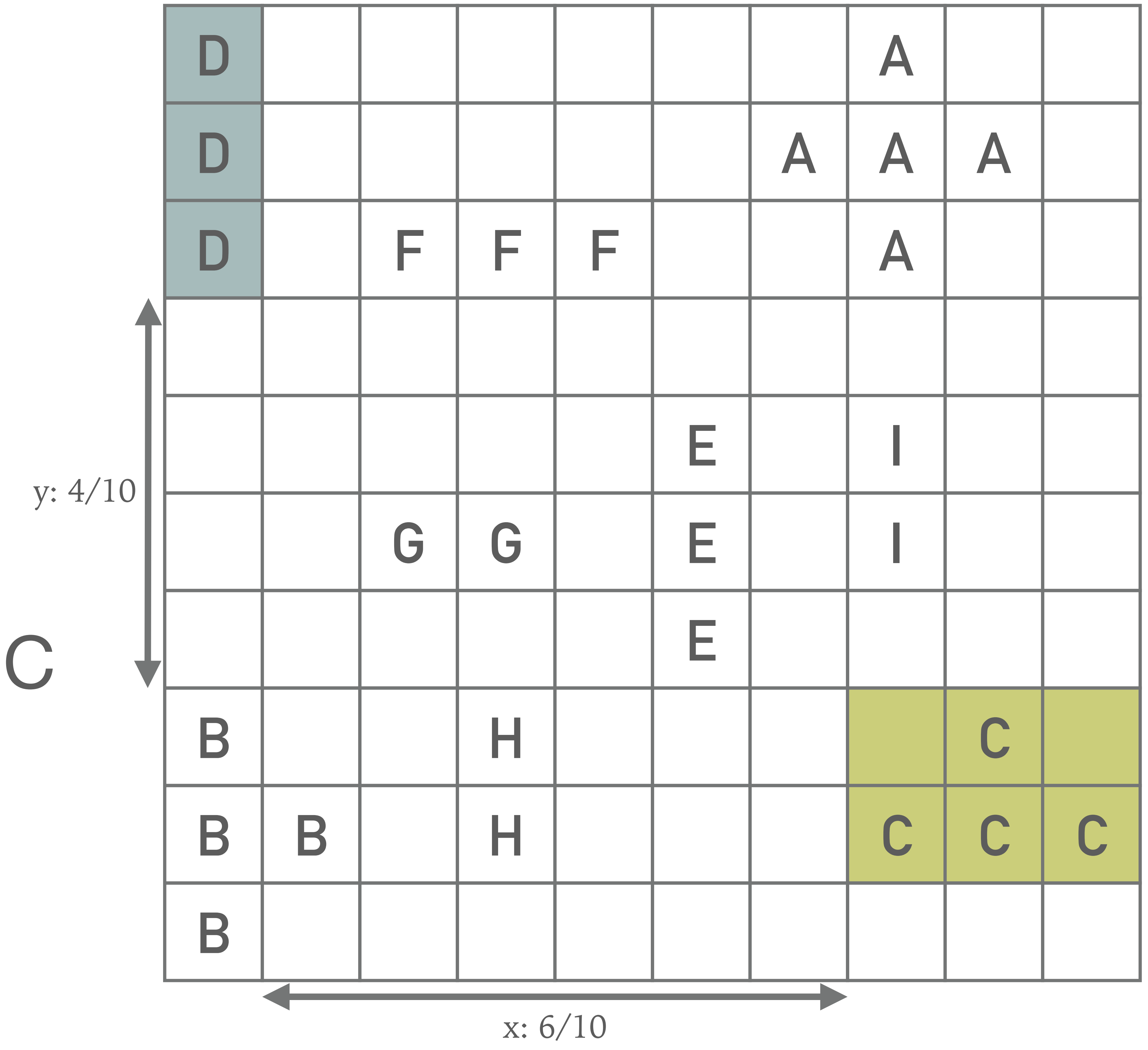
- The largest dead space has CD thus those will be the seeds of the splitting method

Example 6.4: Greene's Split (Continued)

- Having selected seeds, we compute the normalized distances of A and I along each of the axis and pick the axes with higher distance (better separation)

- x: 6/10 = 0.6
- y: 4/10 = 0.4

- In this particular case, the axis x is better separating C and D

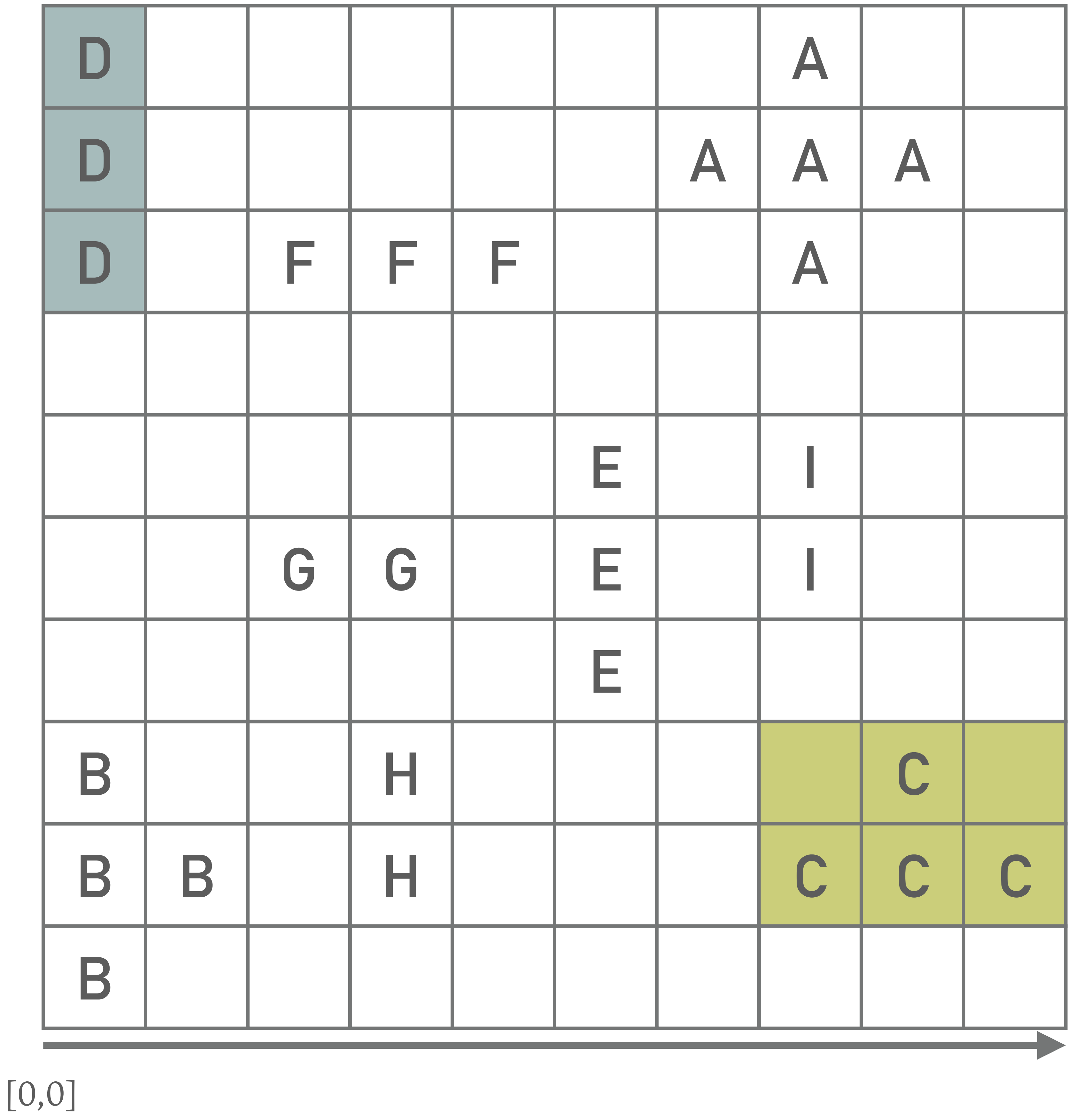


Example 6.4: Greene's Split (Continued)

- Now we order the objects based on their x-axis
  - We start from the coordination [0,0]

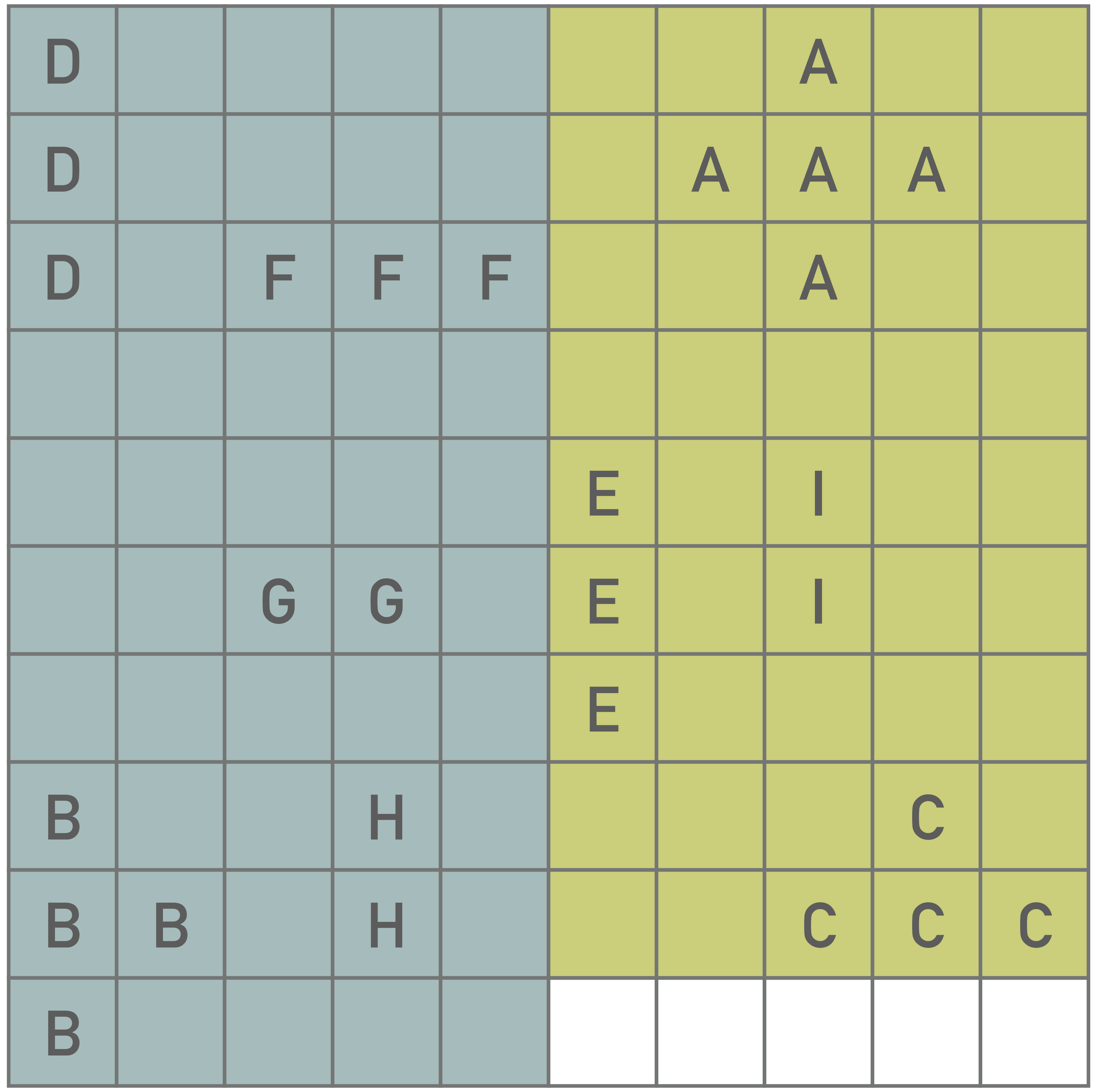
Object	D	B	G	F	H	E	A	I	C
Start	0	0	2	2	3	5	6	7	7
end	0	1	3	4	3	5	8	7	9

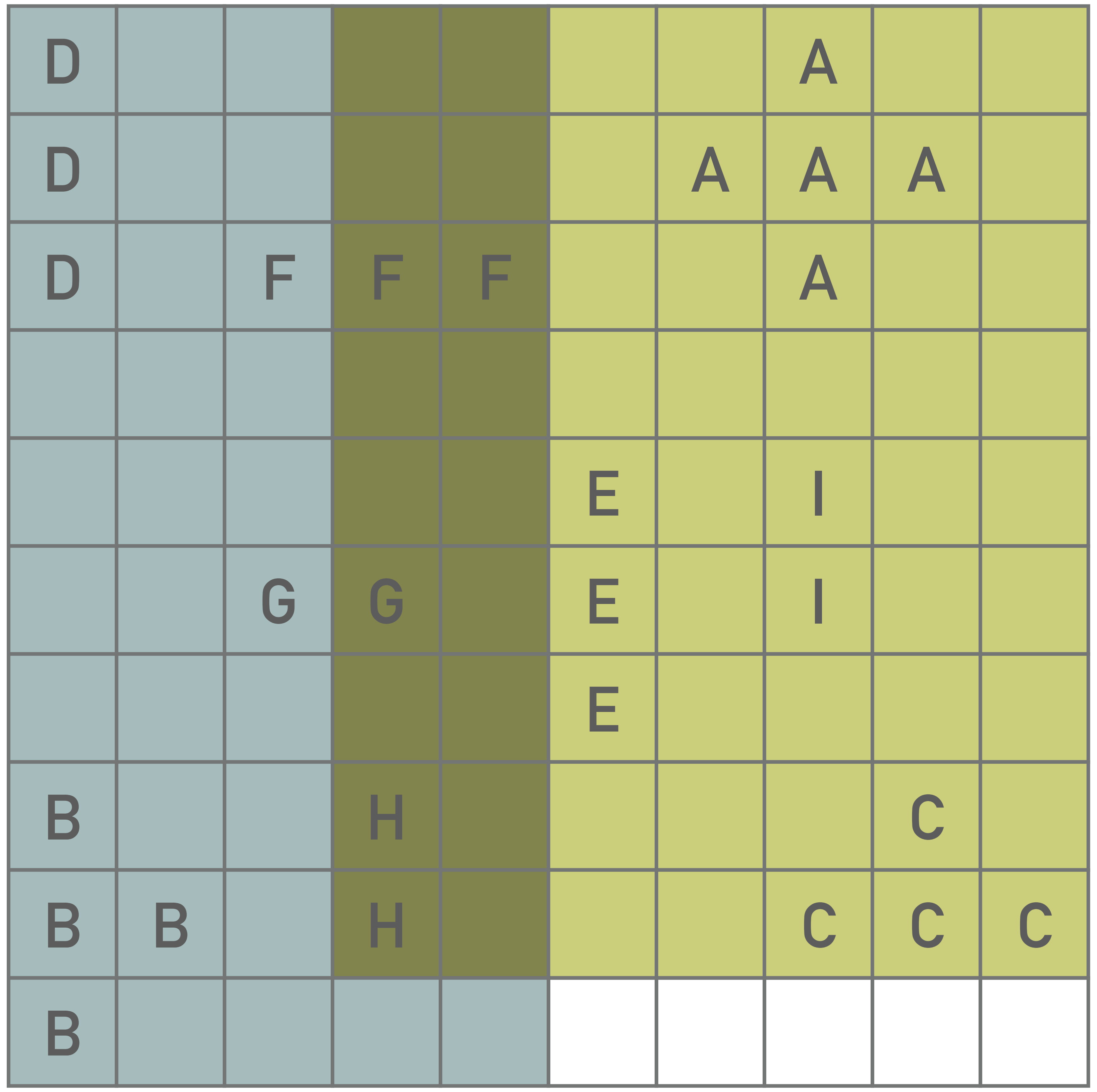
- If two objects start at the same level, we select first the one that ends at lower level
- If two or mode objects starts and ends at the same level, the order is arbitrary



Example 6.4: Greene's Split (Continued)

- We place half of the objects in one node and the other half into the second node
  - There are two possible solutions:





Exercise 6.5

- Split the following overflown node with Greene's split method
  - The maximum number of items in a node is $M = 9$
  - The minimum number of items in a node is $m = 3$

G			A			I	I				J
G		A	A	A							
			A					C			
	F							C	C		
	F				H	H		C			
	F										
				E	E	E		B	B	B	
D	D	D							B		

- That is, execute the following methods:
  - PickSeeds
  - ChooseAxis
  - Distribute (ordering and placement)

Splitting in R* Tree

- R* tree tries to minimize coverage (area) and overlap by adding another criterion, i.e., margin
  - It is used only for the level above the leaf level
  - Other levels are split based on Guttman
- For every two group we can compute the following auxiliary values:
  - *margin-value*, i.e., sum of margins (surfaces) of the two groups
  - *overlap-value*, i.e., volume of the overlap of the two groups
  - *area-value*, i.e., sum of volumes of the two groups

Split_RS(P, PP, E)
ChooseSplitAxis();
Distribute();

ChooseSplitAxis()
FOREACH axis DO
Sort the entries along given axis;
$S \longleftarrow$ sum of all margin-values of all different distributions;
Choose the axis with the minimum S as split axis;

Distribute()
Along the split axis, choose the distribution with minimum overlap-value. Resolve ties by choosing the distribution with minimum area-value;

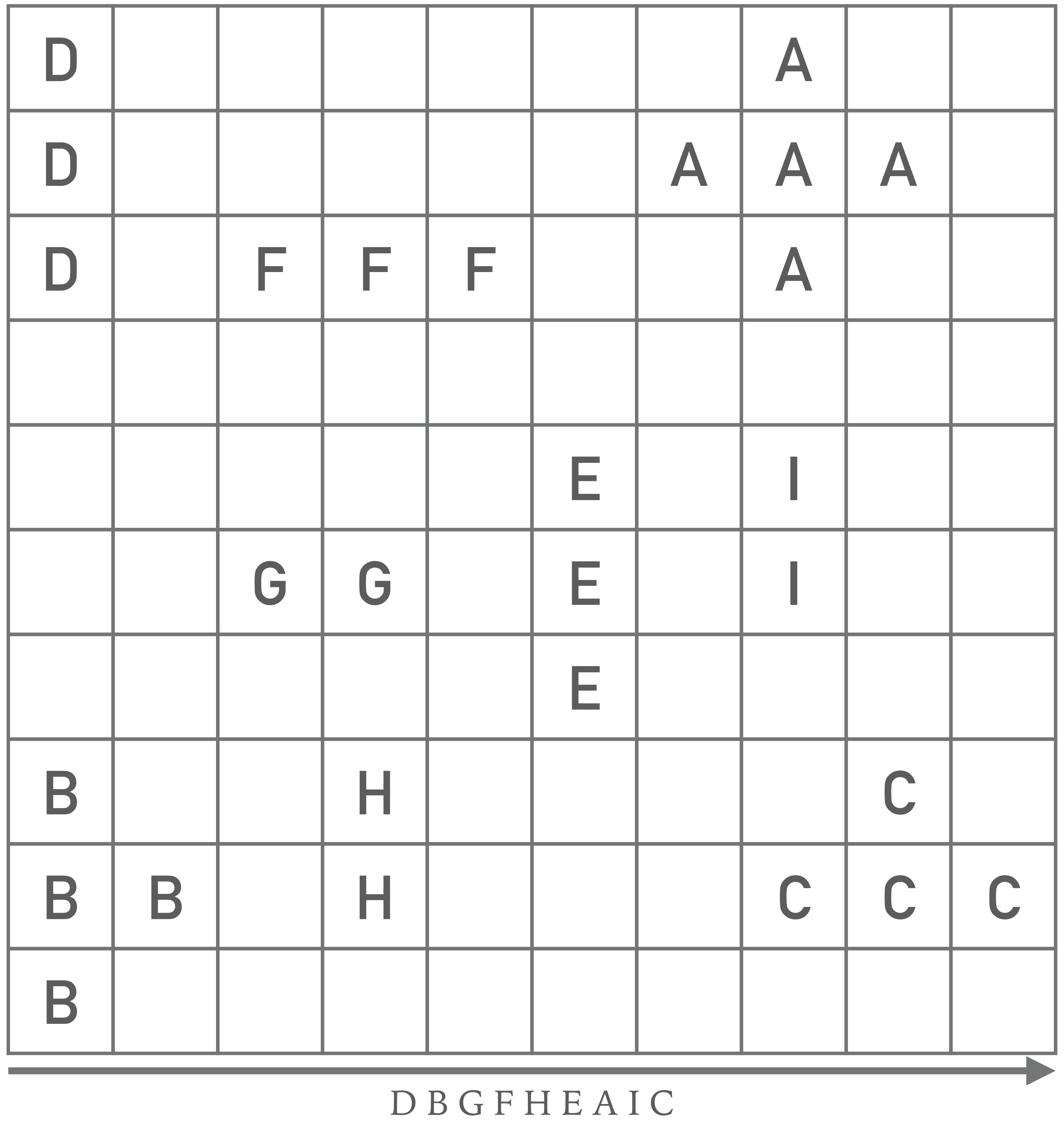
Example 6.6: Splitting in R* Tree

- Split the following overflown node with R* Tree split method
  - The maximum number of items in a node is $M = 8$
  - The minimum number of items in a node is $m = 3$
- That is, execute the following methods:
  - ChooseSplitAxis (i.e., compute margin-value)
  - Distribute (i.e., compute overlap-value and area-value)

D							A		
D						A	A	A	
D		F	F	F			A		
					E		I		
		G	G		E		I		
					E				
B			H					C	
B	B		H				C	C	C
B									

Example 6.6: Splitting in R* Tree (Continued)

- First, we compute the margin-values for every possible distributions of objects with regard to the x and y axis
  - margin-value: $margin(MBR(G_1)) * 2 + margin(MBR(G_2)) * 2$
- These are summed and such an axis is chosen which minimizes the sum

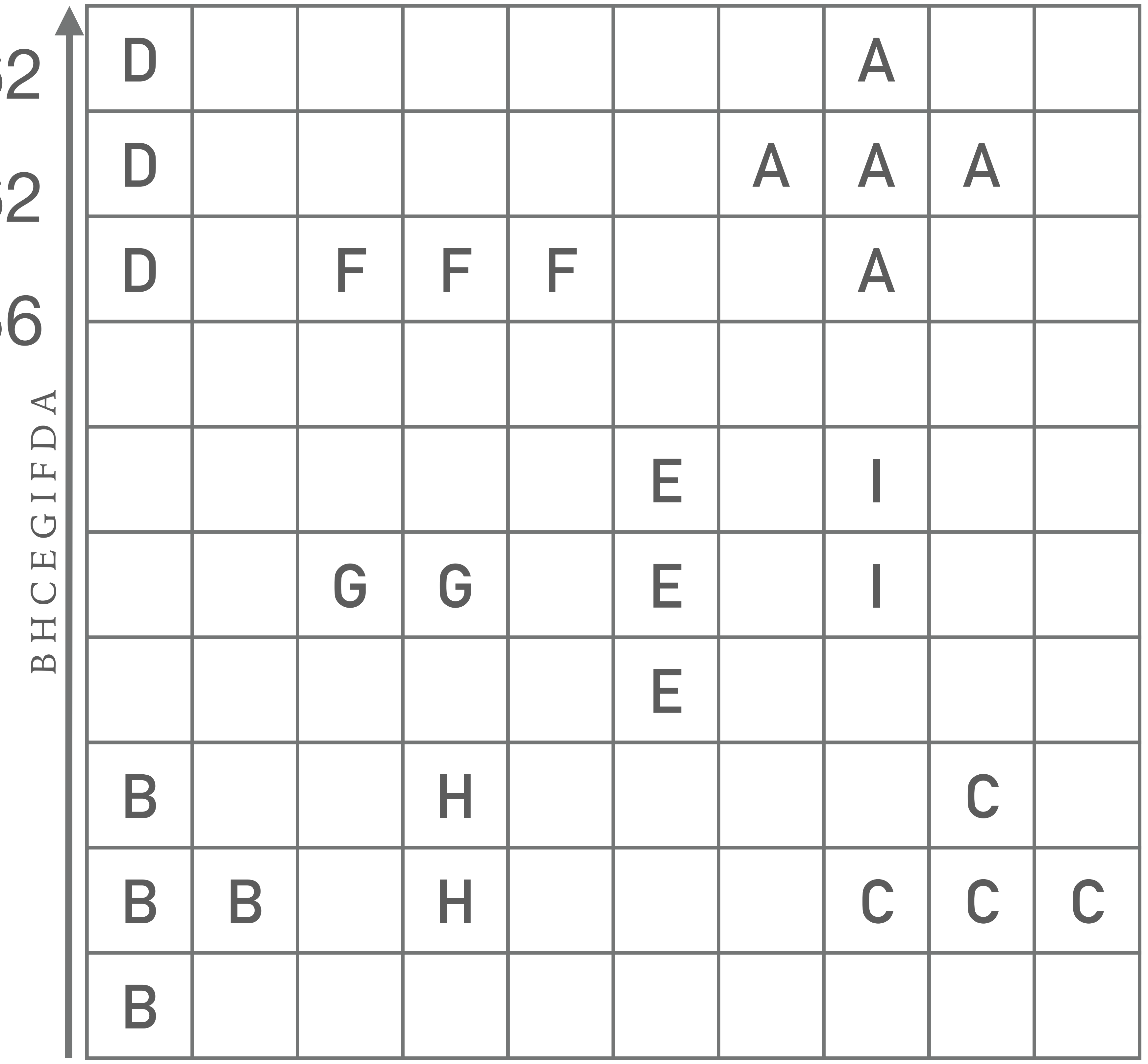


- Ordering* based on the x-axis: D B G F H E A I C
  - margin-value (DBG || FHEAIC) = (4 + 10) x 2 + (8 + 9) x 2 = 62
  - margin-value (DBGF || HEAIC) = (5 + 10) x 2 + (7 + 9) x 2 = 62
  - margin-value (DBGFH || EAIC) = (5 + 10) x 2 + (5 + 9) x 2 = 58
  - margin-value (DBGFHE || AIC) = (6 + 10) x 2 + (4 + 9) x 2 = 58
  - Sum = 62 + 62 + 58 + 58 = 240

* If two objects start at the same level, we select first the one that ends at lower level. Or if two or mode objects starts and ends at the same level, the order is arbitrary.

Example 6.6: Splitting in R* Tree (Continued)

- Ordering* based on the y-axis: B H C E G I F D A
  - margin-value (BHC || EGIFDA) = (10 + 3) x 2 + (9 + 7) x 2 = 58
  - margin-value (BHCE || GIFDA) = (10 + 6) x 2 + (9 + 6) x 2 = 62
  - margin-value (BHCEG || IFDA) = (10 + 6) x 2 + (9 + 6) x 2 = 62
  - margin-value (BHCEGI || FDA) = (10 + 6) x 2 + (9 + 3) x 2 = 56
  - Sum = 58 + 62 + 62 + 56 = 238

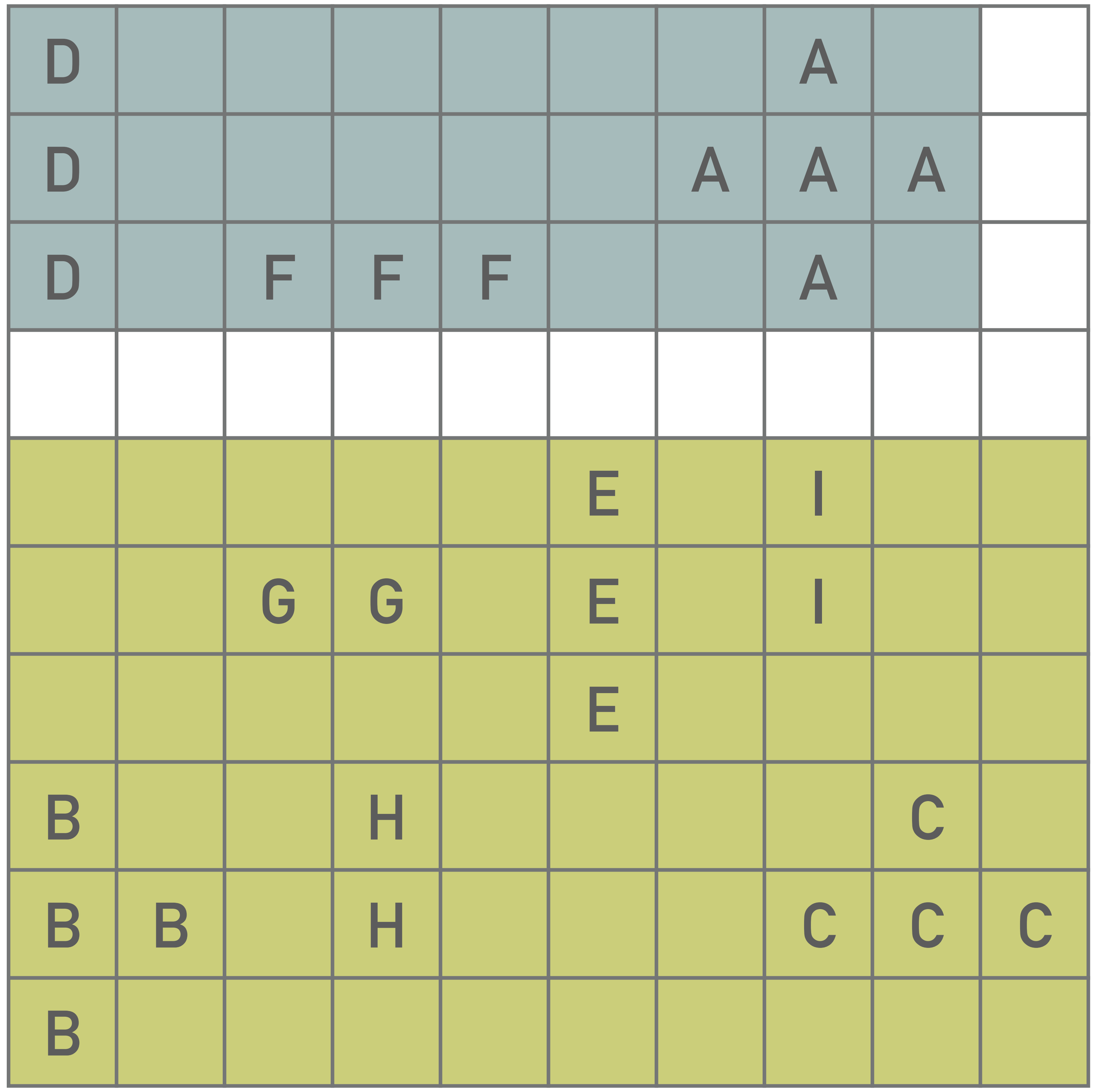


- X-axis: 240
- Y-axis: 238
  - Therefore we chose splitting along the y-axis

* If two objects start at the same level, we select first the one that ends at lower level. Or if two or mode objects starts and ends at the same level, the order is arbitrary.

Example 6.6: Splitting in R* Tree (Continued)

- Now we compute the overlap-values among all the distributions (along y-axis) and pick the distribution that minimizes the overlap
  - overlap-value (BHC || EGIFDA) = 0
  - overlap-value (BHCE || GIFDA) = 14 (rows GEI, EI)
  - overlap-value (BHCEG || IFDA) = 7 (row EI)
  - overlap-value (BHCEGI || FDA) = 0



- If more distributions lead to the minimum overlap, the one is chosen which shows the smallest area-value
  - area-value (BHC || EGIFDA) = (10 x 3) + (9 x 7) = 93
  - area-value (BHCEGI || FDA) = (10 x 6) + (9 x 3) = 87

Exercise 6.7

- Split the following overflown node with R* Tree split method
  - The maximum number of items in a node is $M = 9$
  - The minimum number of items in a node is $m = 3$

- That is, execute the following methods:
  - ChooseSplitAxis
  - Distribute

- Illustrate the result

G			A			I	I				J
G		A	A	A							
			A					C			
	F							C	C		
	F				H	H		C			
	F										
				E	E	E		B	B	B	
D	D	D							B		