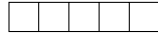
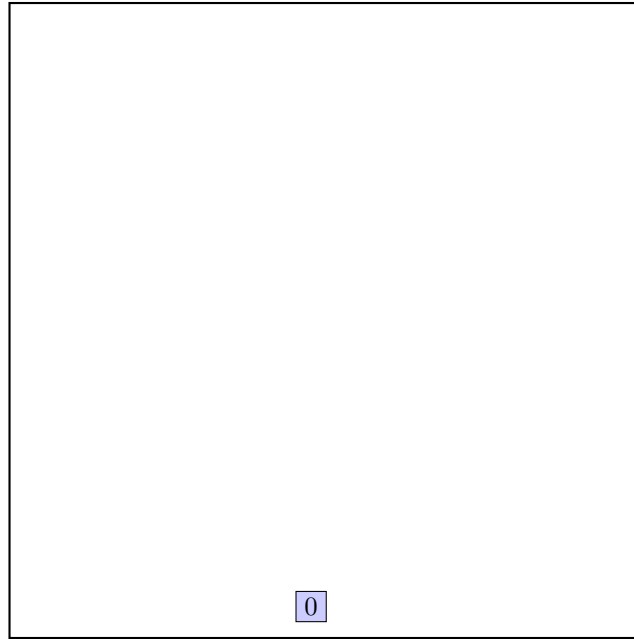


call INSERT, #S(P :X 379/200 :Y 27/250)

structure view:

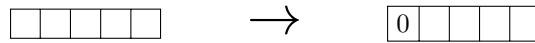


data view:



call CHOOSE-SUBTREE with 0, node *root*, level 0

the node *root* is not full, add the record.



call ADJUST-TREE with *R*, node *root*

we are at the root

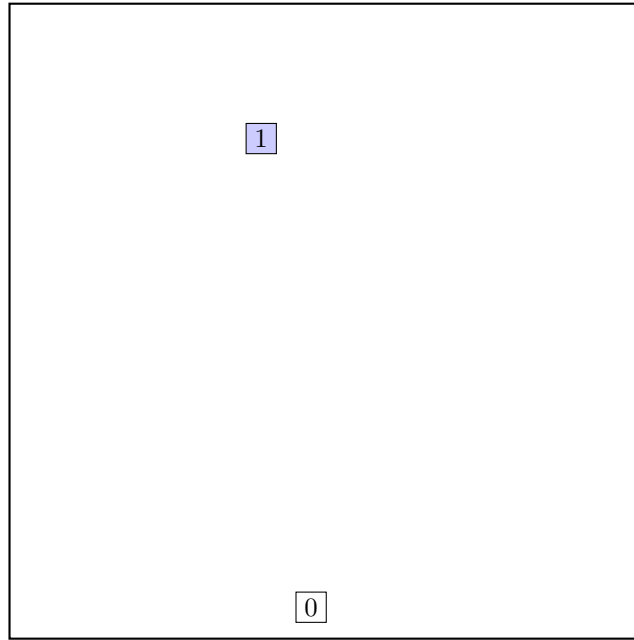
return from ADJUST-TREE

call INSERT, #S(P :X 313/200 :Y 401/125)

structure view:

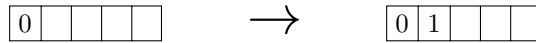


data view:



call CHOOSE-SUBTREE with 1, node *root*, level 0

the node *root* is not full, add the record.



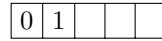
call ADJUST-TREE with *R*, node *root*

we are at the root

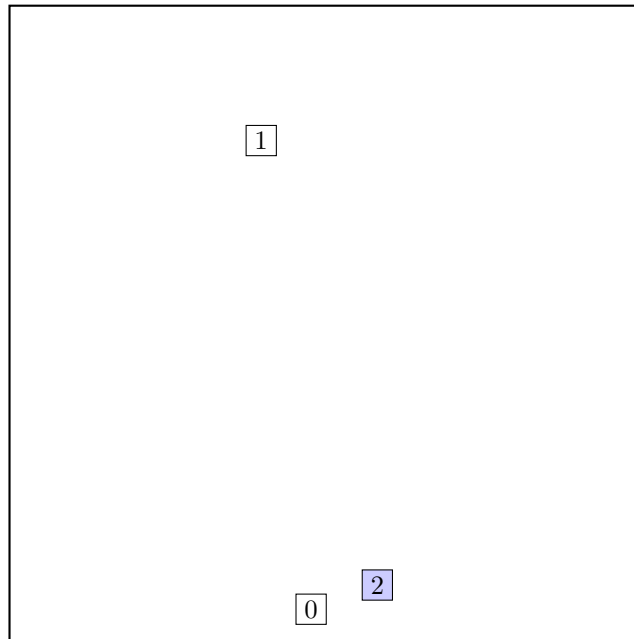
return from ADJUST-TREE

call INSERT, #S(P :X 2333/1000 :Y 67/250)

structure view:

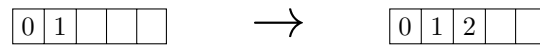


data view:



call CHOOSE-SUBTREE with 2, node *root*, level 0

the node *root* is not full, add the record.



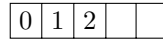
call ADJUST-TREE with *R*, node *root*

we are at the root

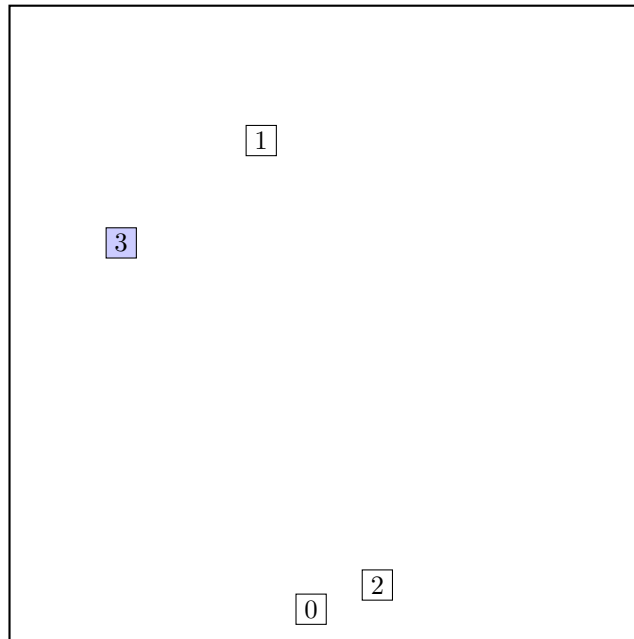
return from ADJUST-TREE

call INSERT, #S(P :X 639/1000 :Y 2531/1000)

structure view:

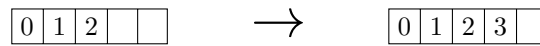


data view:



call CHOOSE-SUBTREE with 3, node *root*, level 0

the node *root* is not full, add the record.



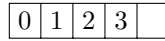
call ADJUST-TREE with *R*, node *root*

we are at the root

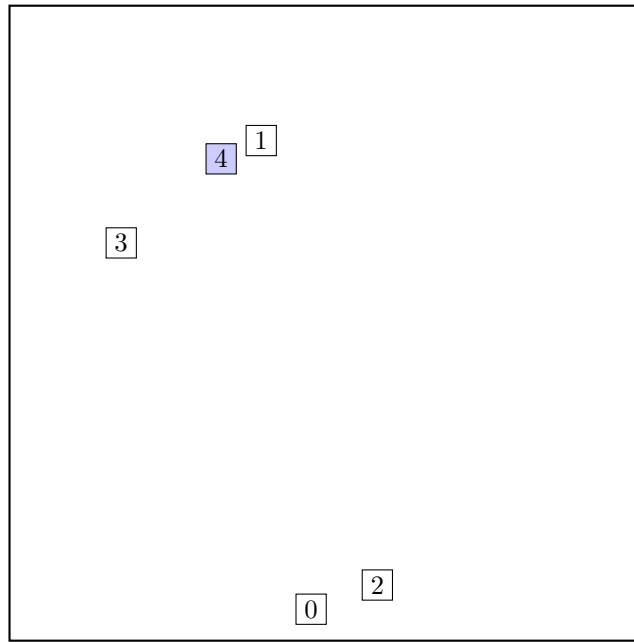
return from ADJUST-TREE

call INSERT, #S(P :X 13/10 :Y 3087/1000)

structure view:

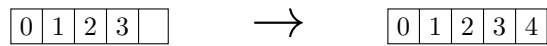


data view:



call CHOOSE-SUBTREE with 4, node *root*, level 0

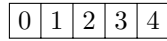
the node *root* is not full, add the record.



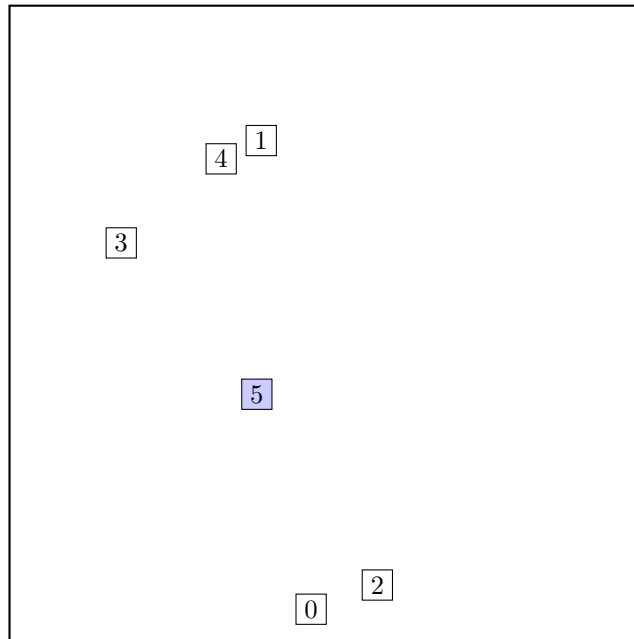
call ADJUST-TREE with *R*, node *root*  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 192/125 :Y 153/100)

structure view:



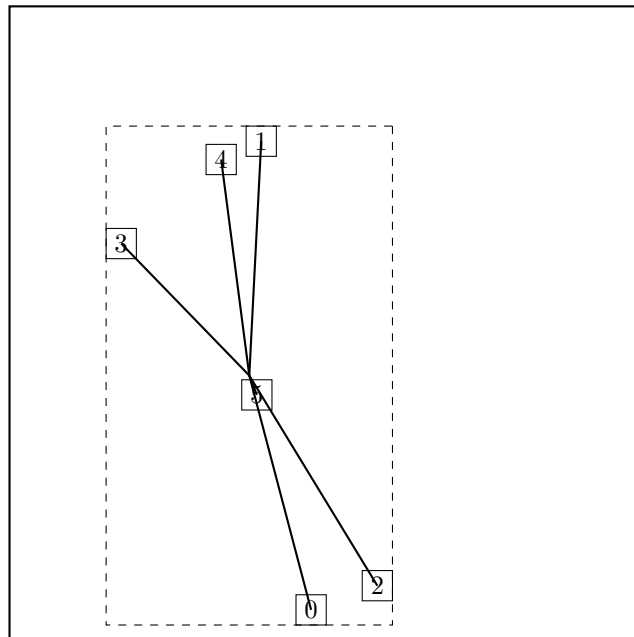
data view:



call CHOOSE-SUBTREE with 5, node *root*, level 0

call OVERFLOW-TREATMENT

call REINSERT



The two most distant nodes (2 0) are removed and reinserted in reversed order.

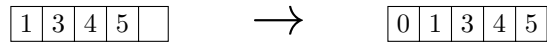
The rest is kept.

Adjust MBR of root.

Reinserting 0.

call CHOOSE-SUBTREE with 0, node *A*, level 0

the node A is not full, add the record.



call ADJUST-TREE with  $R$ , node A

we are at the root

return from ADJUST-TREE

return from REINSERT

Reinserting 2.

call CHOOSE-SUBTREE with 2, node A, level 0

call OVERFLOW-TREATMENT

Second overflow on the same level during one insert — make a split.

call SPLIT-NODE with nodes A and 2

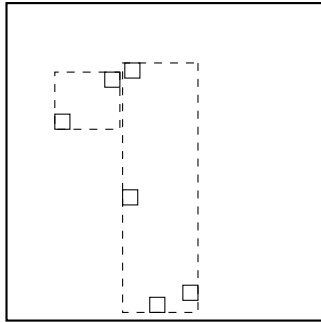
call CHOOSE-SPLIT-AXIS with ...

Axis:  $x$

Entries sorted by low: 3 4 5 1 0 2

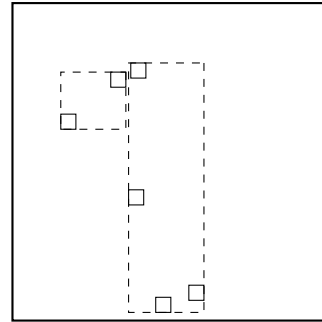
Entries sorted by high: 3 4 5 1 0 2

partition by low: 3 4 — 5 1 0 2



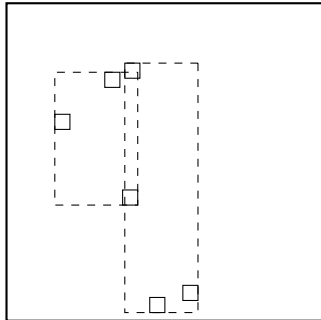
$o = 5.9139996$

partition by high: 3 4 — 5 1 0 2



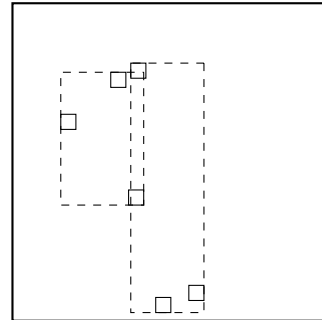
$o = 5.9139996$

partition by low: 3 4 5 — 1 0 2



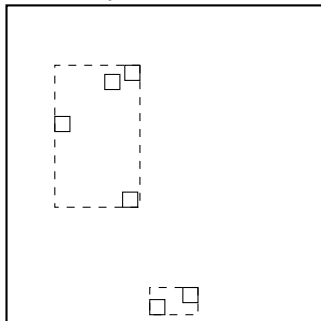
$o = 7.1219998$

partition by high: 3 4 5 — 1 0 2



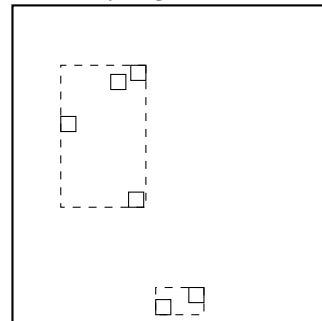
$o = 7.1219998$

partition by low: 3 4 5 1 — 0 2



$o = 4.002$

partition by high: 3 4 5 1 — 0 2



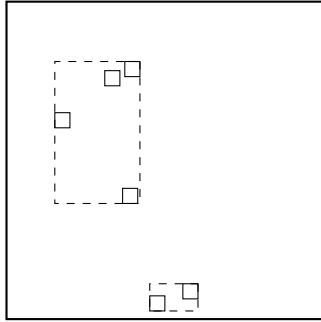
$o = 4.002$

Axis:  $y$

Entries sorted by low: 0 2 5 3 4 1

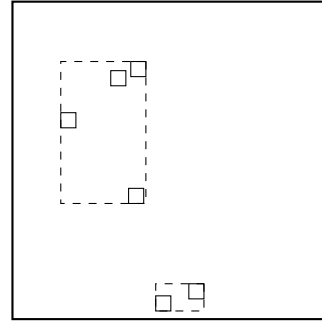
Entries sorted by high: 0 2 5 3 4 1

partition by low: 0 2 — 5 3 4 1



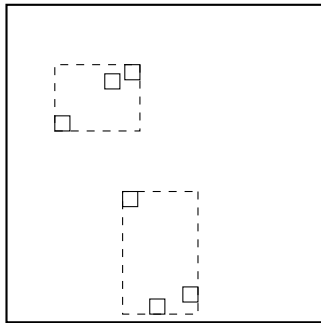
$o = 4.002$

partition by high: 0 2 — 5 3 4 1



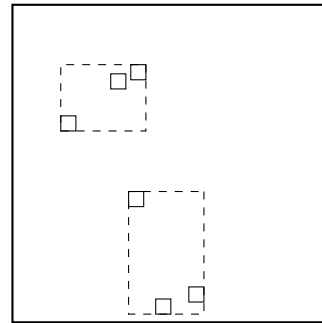
$o = 4.002$

partition by low: 0 2 5 — 3 4 1



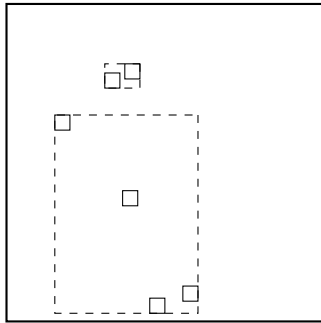
$o = 4.6219998$

partition by high: 0 2 5 — 3 4 1



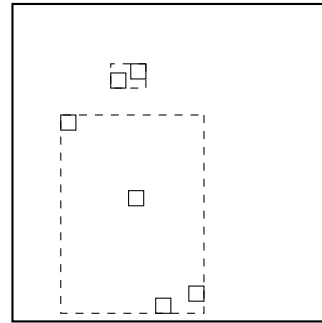
$o = 4.6219998$

partition by low: 0 2 5 3 — 4 1



$o = 5.3029995$

partition by high: 0 2 5 3 — 4 1



$o = 5.3029995$

Minimal margin ( $o = 4.002$ ) was reached for axis  $x$ .  
return from CHOOSE-SPLIT-AXIS

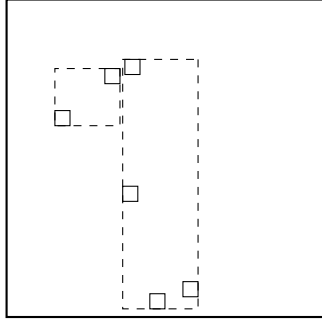
call CHOOSE-SPLIT-INDEX with entries, axis  $x$ ,  $R$

Entries sorted by low at axis  $x$ : 3 4 5 1 0 2

Entries sorted by high at axis  $x$ : 3 4 5 1 0 2

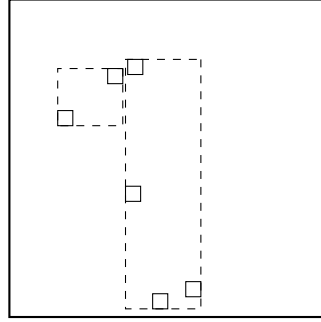


Partition by low: 3 4 — 5 1 0 2



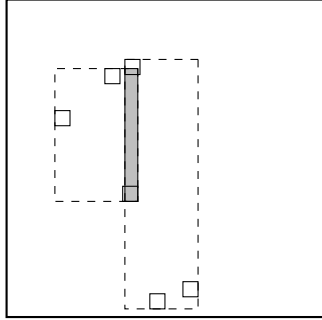
overlap = 0  
 $S = 3.9410153$

Partition by high: 3 4 — 5 1 0 2



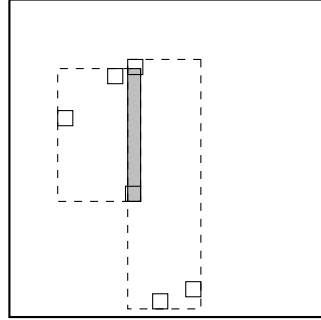
overlap = 0  
 $S = 3.9410153$

Partition by low: 3 4 5 — 1 0 2



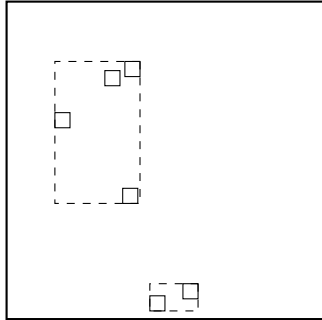
overlap = 0.300447  
 $S = 5.1218286$

Partition by high: 3 4 5 — 1 0 2



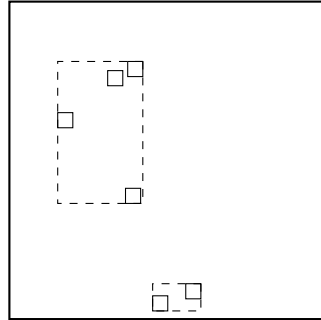
overlap = 0.300447  
 $S = 5.1218286$

Partition by low: 3 4 5 1 — 0 2



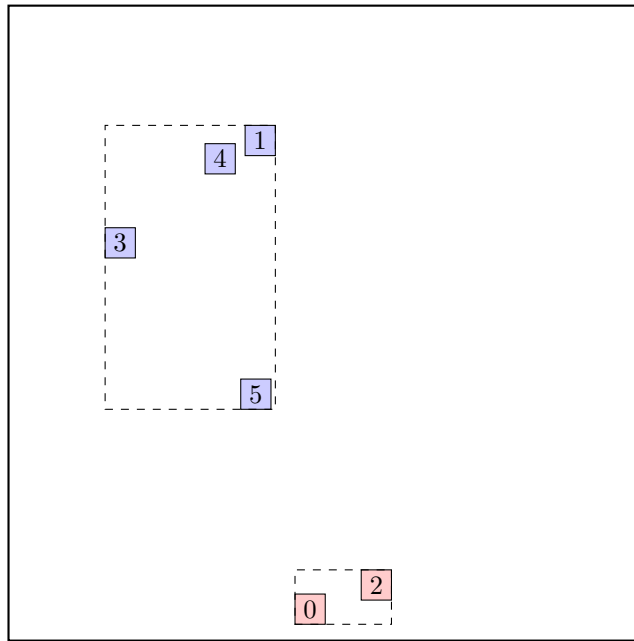
overlap = 0  
 $S = 2.3443084$

Partition by high: 3 4 5 1 — 0 2



overlap = 0  
 $S = 2.3443084$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

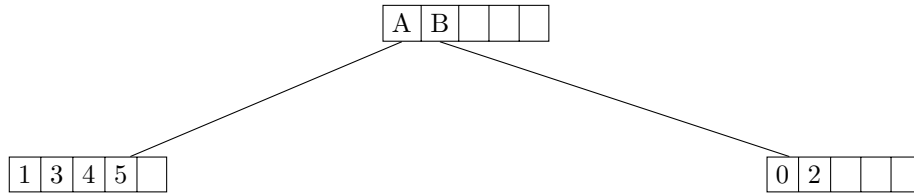
call ADJUST-TREE with  $R$ , node A and the new node  
 we are at the root  
 return from ADJUST-TREE  
 create a new root. return from REINSERT

call ADJUST-TREE with  $R$ , node A  
 update MBR of node A.  
 continue by adjusting the parent node root

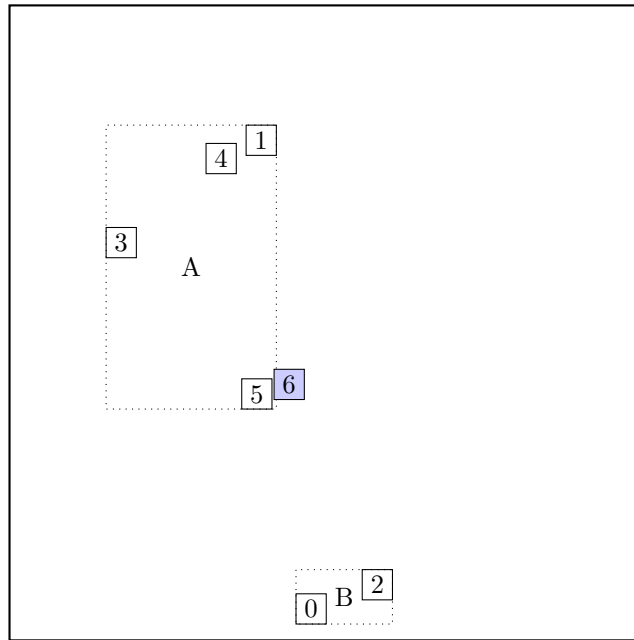
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE

call INSERT, #S(P :X 7/4 :Y 1593/1000)

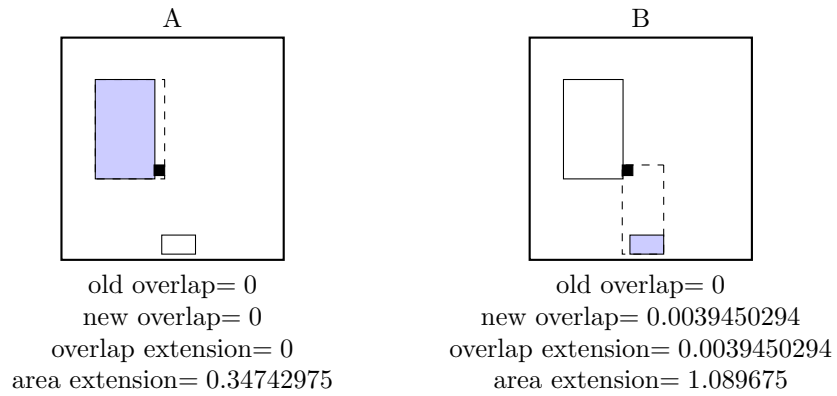
structure view:



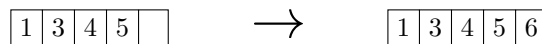
data view:



call CHOOSE-SUBTREE with 6, node *root*, level 1  
Next level are leaf nodes



Node A is chosen  
the node A is not full, add the record.

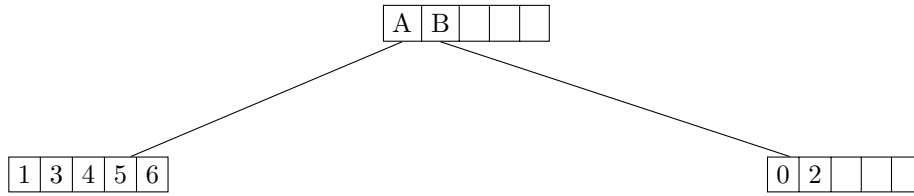


call ADJUST-TREE with *R*, node A  
update MBR of node A.  
continue by adjusting the parent node *root*

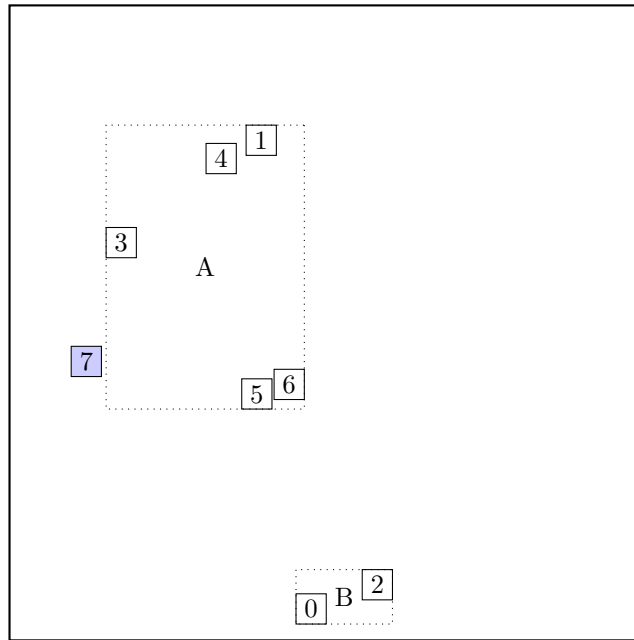
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 51/125 :Y 349/200)

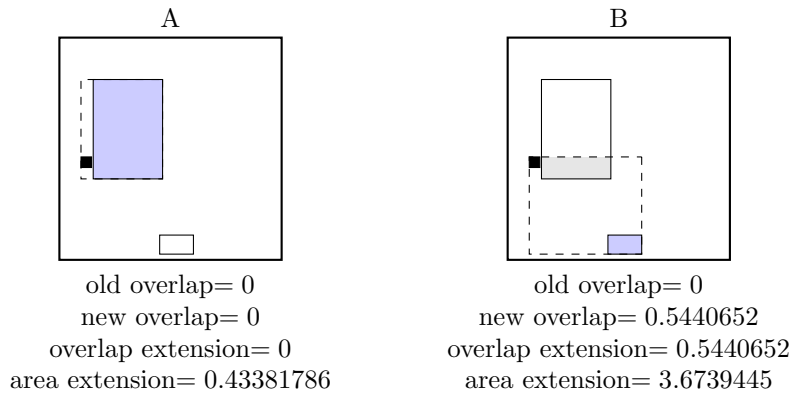
structure view:



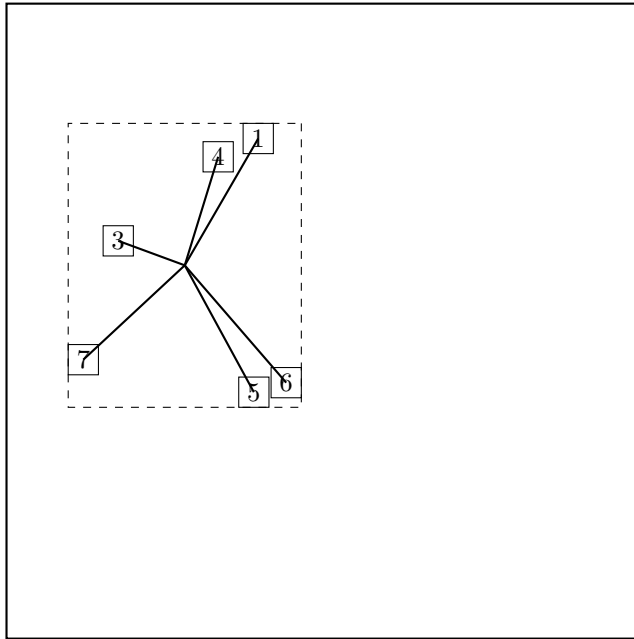
data view:



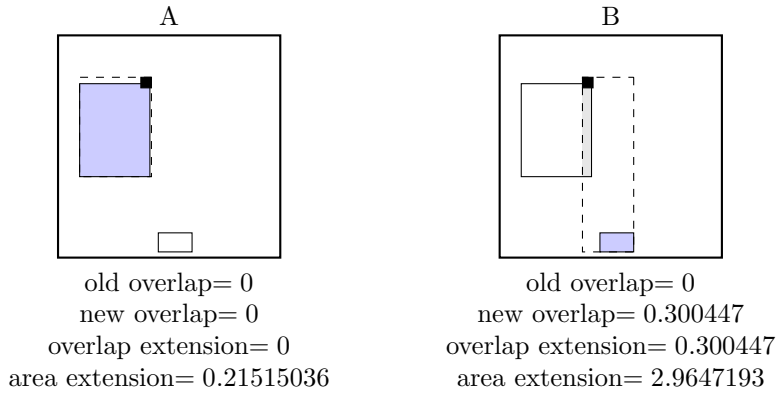
call CHOOSE-SUBTREE with 7, node *root*, level 1  
 Next level are leaf nodes



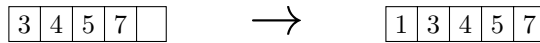
Node A is chosen  
 call OVERFLOW-TREATMENT  
 call REINSERT



The two most distant nodes (6 1) are removed and reinserted in reversed order.  
 The rest is kept.  
 Adjust MBR of A.  
 Reinserting 1.  
 call CHOOSE-SUBTREE with 1, node *root*, level 1  
 Next level are leaf nodes

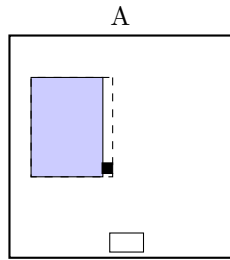


Node A is chosen  
 the node A is not full, add the record.

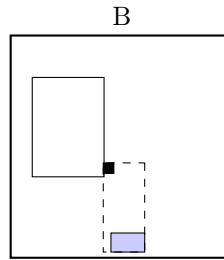


call ADJUST-TREE with *R*, node A  
 update MBR of node A.  
 continue by adjusting the parent node *root*

call ADJUST-TREE with *R*, node *root*  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT  
 Reinserting 6.  
 call CHOOSE-SUBTREE with 6, node *root*, level 1  
 Next level are leaf nodes



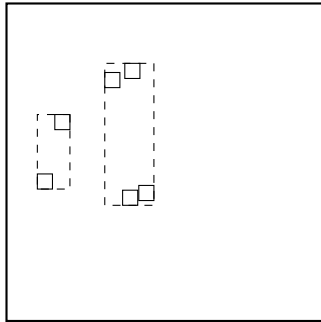
old overlap= 0  
 new overlap= 0  
 overlap extension= 0  
 area extension= 0.34742975



old overlap= 0  
 new overlap= 0.0039450294  
 overlap extension= 0.0039450294  
 area extension= 1.089675

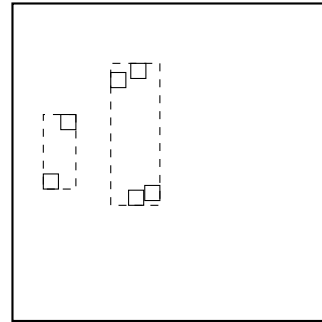
Node A is chosen  
 call OVERFLOW-TREATMENT  
 Second overflow on the same level during one insert — make a split.  
 call SPLIT-NODE with nodes A and 6  
 call CHOOSE-SPLIT-AXIS with ...  
 Axis:  $x$   
 Entries sorted by low: 7 3 4 5 1 6  
 Entries sorted by high: 7 3 4 5 1 6

partition by low: 7 3 — 4 5 1 6



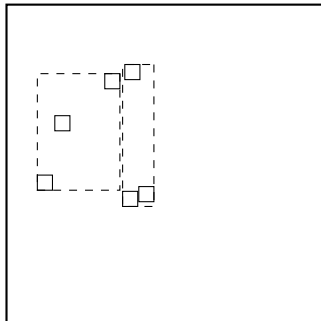
$o = 3.9449997$

partition by high: 7 3 — 4 5 1 6



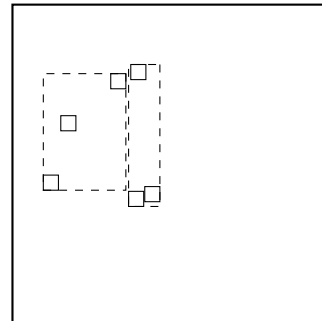
$o = 3.9449997$

partition by low: 7 3 4 — 5 1 6



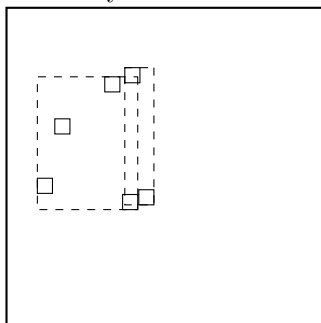
$o = 4.9259997$

partition by high: 7 3 4 — 5 1 6



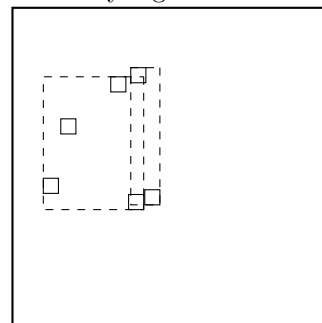
$o = 4.9259997$

partition by low: 7 3 4 5 — 1 6



$o = 5.285$

partition by high: 7 3 4 5 — 1 6



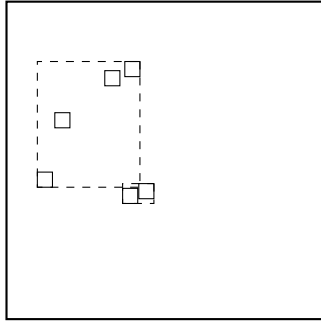
$o = 5.285$

Axis:  $y$

Entries sorted by low: 5 6 7 3 4 1

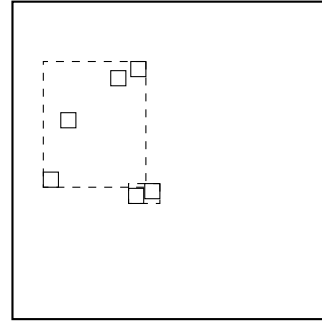
Entries sorted by high: 5 6 7 3 4 1

partition by low: 5 6 — 7 3 4 1



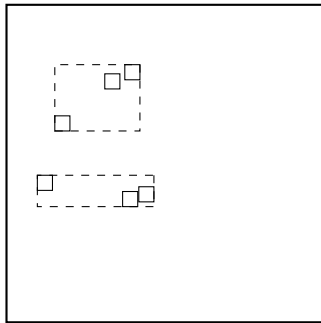
$o = 3.697$

partition by high: 5 6 — 7 3 4 1



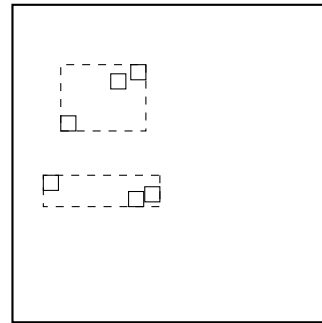
$o = 3.697$

partition by low: 5 6 7 — 3 4 1



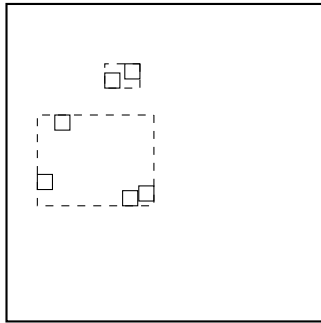
$o = 3.96$

partition by high: 5 6 7 — 3 4 1



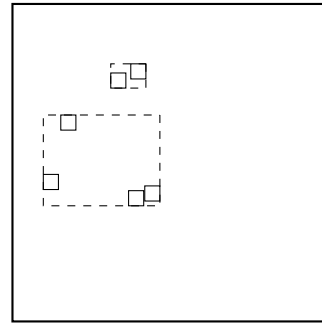
$o = 3.96$

partition by low: 5 6 7 3 — 4 1



$o = 3.529$

partition by high: 5 6 7 3 — 4 1



$o = 3.529$

Minimal margin ( $o = 3.529$ ) was reached for axis  $y$ .  
return from CHOOSE-SPLIT-AXIS

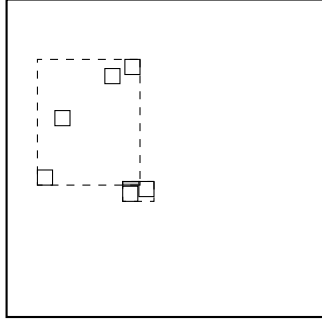
call CHOOSE-SPLIT-INDEX with entries, axis  $y$ ,  $R$

Entries sorted by low at axis  $y$ : 5 6 7 3 4 1

Entries sorted by high at axis  $y$ : 5 6 7 3 4 1

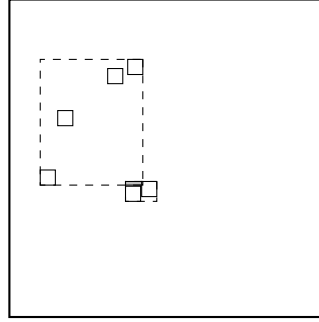


Partition by low: 5 6 — 7 3 4 1



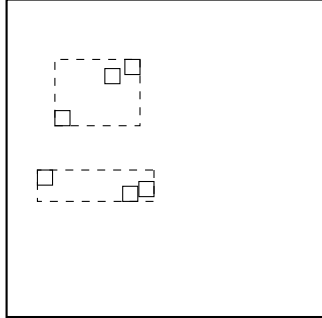
overlap = 0.010992027  
 $S = 2.365573$

Partition by high: 5 6 — 7 3 4 1



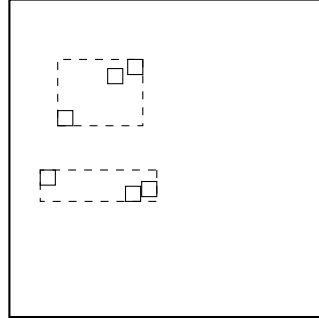
overlap = 0.010992027  
 $S = 2.365573$

Partition by low: 5 6 7 — 3 4 1



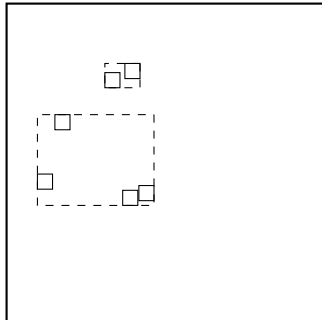
overlap = 0  
 $S = 1.6274321$

Partition by high: 5 6 7 — 3 4 1



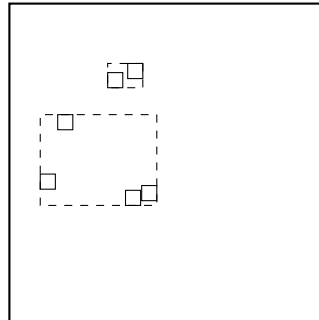
overlap = 0  
 $S = 1.6274321$

Partition by low: 5 6 7 3 — 4 1



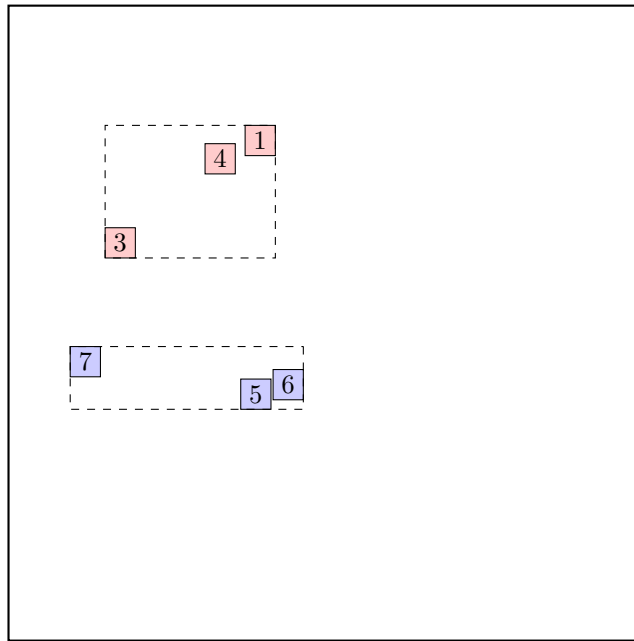
overlap = 0  
 $S = 2.0012069$

Partition by high: 5 6 7 3 — 4 1



overlap = 0  
 $S = 2.0012069$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

call ADJUST-TREE with  $R$ , node A and the new node  
 update MBR of node A.  
 add the new node to the parent node root

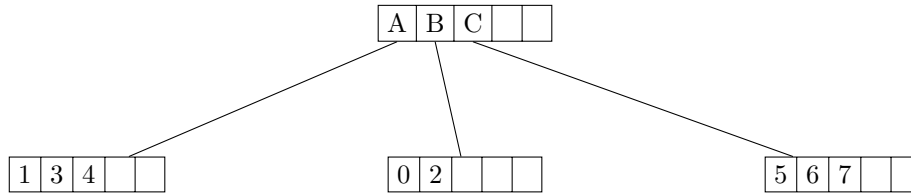
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT

call ADJUST-TREE with  $R$ , node C  
 update MBR of node C.  
 continue by adjusting the parent node root

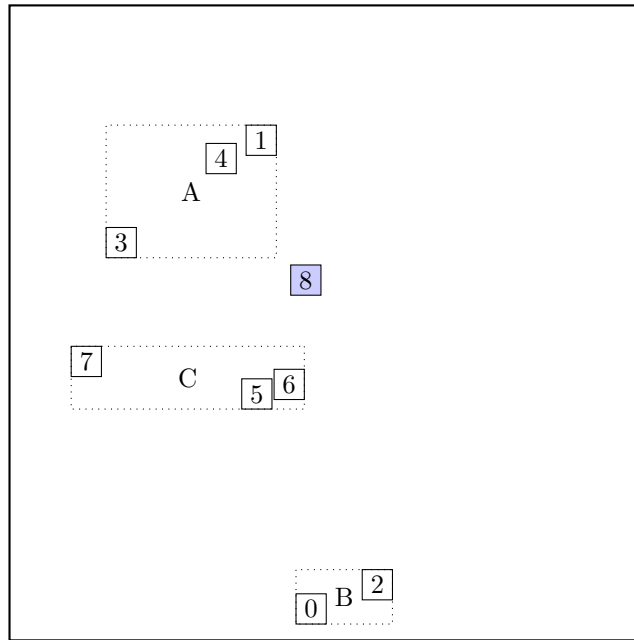
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE

call INSERT, #S(P :X 93/50 :Y 2283/1000)

structure view:



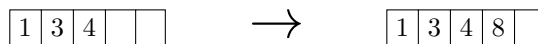
data view:



call CHOOSE-SUBTREE with 8, node *root*, level 1  
Next level are leaf nodes

<p>A</p> <p>old overlap= 0 new overlap= 0 overlap extension= 0 area extension= 0.6111227</p>	<p>B</p> <p>old overlap= 0 new overlap= 0.03735002 overlap extension= 0.03735002 area extension= 1.3686947</p>	<p>C</p> <p>old overlap= 0 new overlap= 0 overlap extension= 0 area extension= 0.93442584</p>
--	--	---

Node A is chosen  
the node A is not full, add the record.

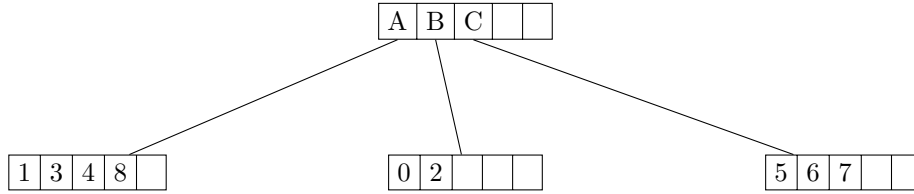


call ADJUST-TREE with *R*, node A  
update MBR of node A.  
continue by adjusting the parent node root

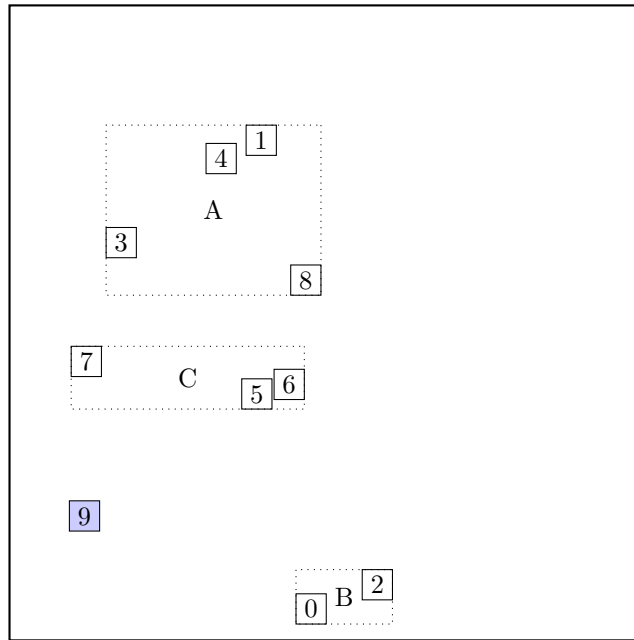
```
call ADJUST-TREE with  $R$ , node root
we are at the root
return from ADJUST-TREE
```

call INSERT, #S(P :X 99/250 :Y 723/1000)

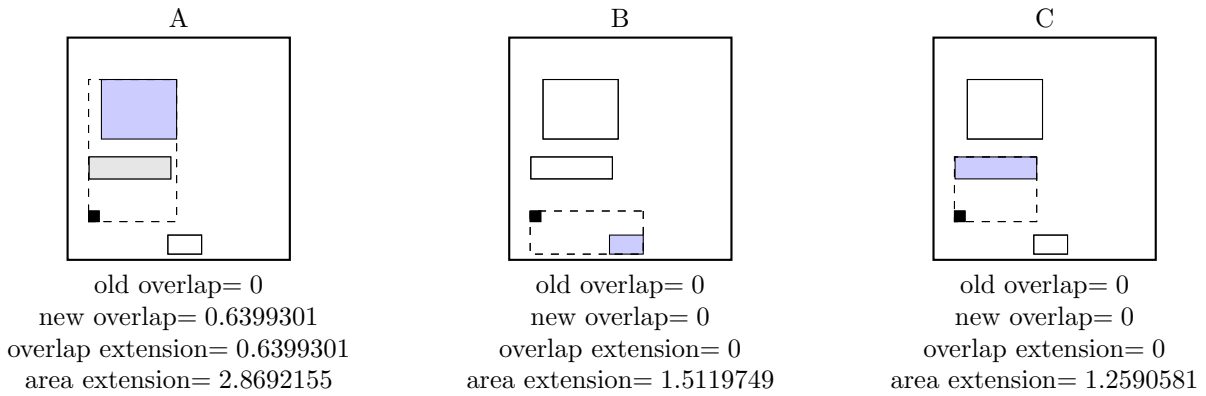
structure view:



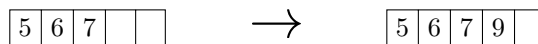
data view:



call CHOOSE-SUBTREE with 9, node *root*, level 1  
Next level are leaf nodes



Node *C* is chosen  
the node *C* is not full, add the record.

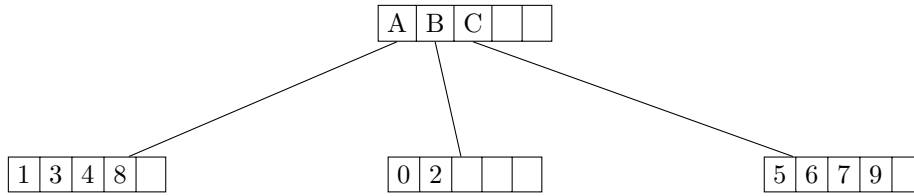


call ADJUST-TREE with *R*, node *C*  
update MBR of node *C*.  
continue by adjusting the parent node *root*

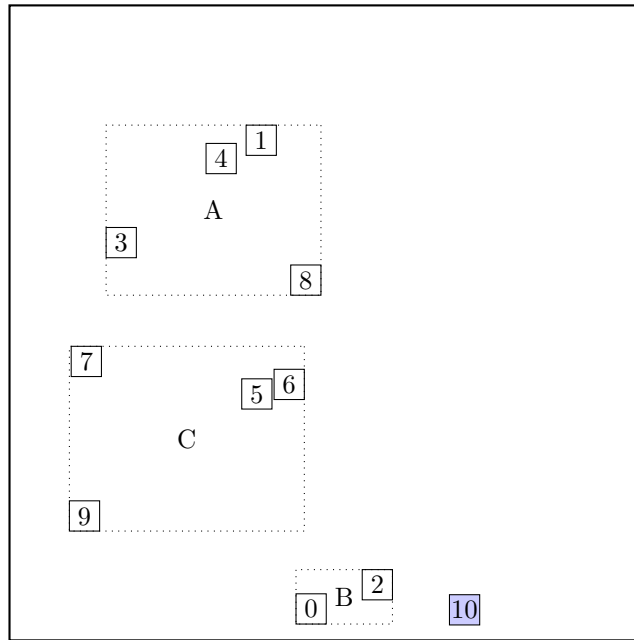
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 291/100 :Y 51/500)

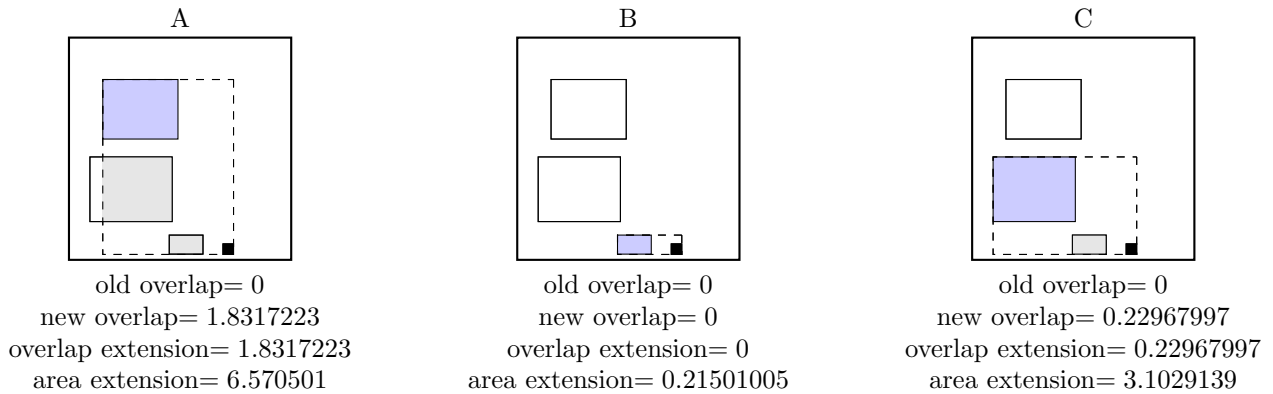
structure view:



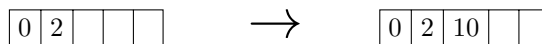
data view:



call CHOOSE-SUBTREE with 10, node *root*, level 1  
Next level are leaf nodes



Node *B* is chosen  
the node *B* is not full, add the record.



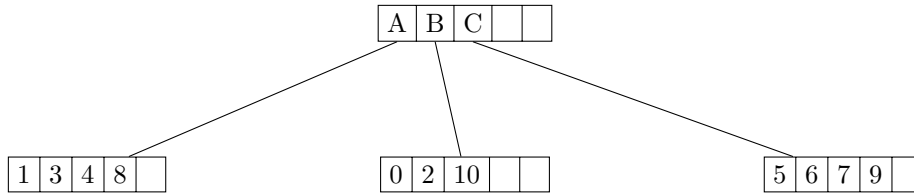
call ADJUST-TREE with *R*, node *B*  
update MBR of node *B*.  
continue by adjusting the parent node *root*

call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

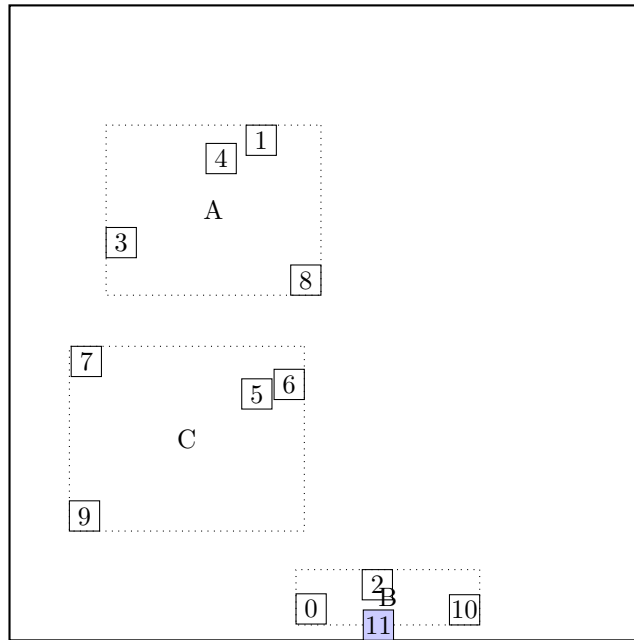


call INSERT, #S(P :X 117/50 :Y 1/1000)

structure view:

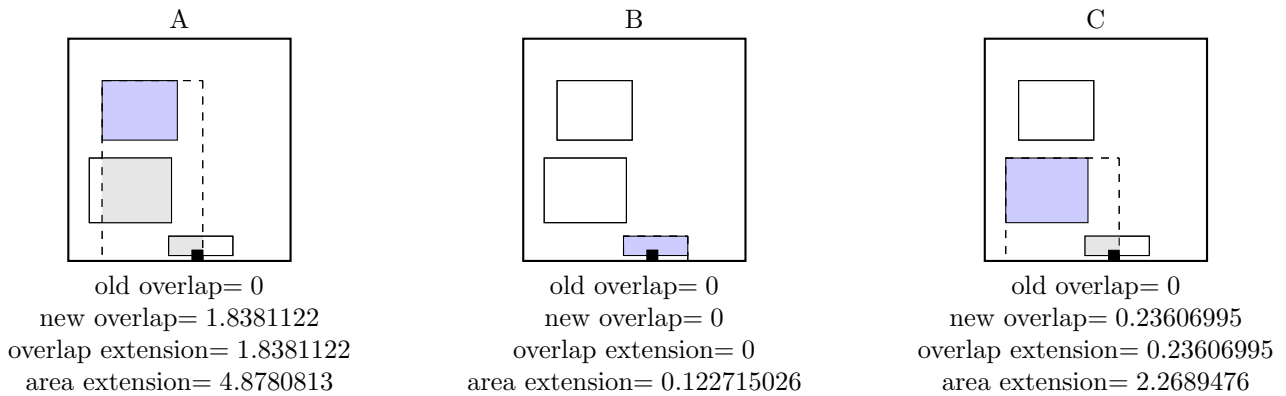


data view:



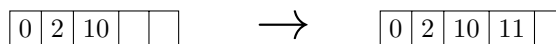
call CHOOSE-SUBTREE with 11, node *root*, level 1

Next level are leaf nodes



Node *B* is chosen

the node *B* is not full, add the record.



call ADJUST-TREE with *R*, node *B*

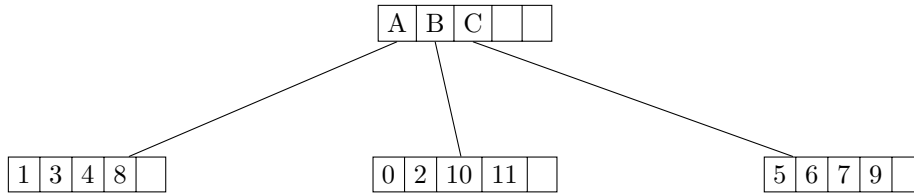
update MBR of node *B*.

continue by adjusting the parent node *root*

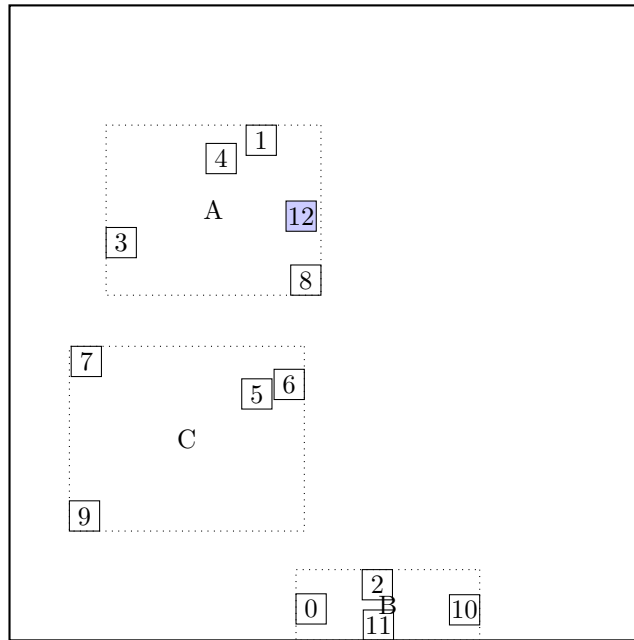
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 1829/1000 :Y 1353/500)

structure view:

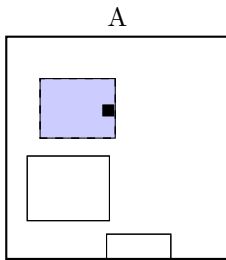


data view:

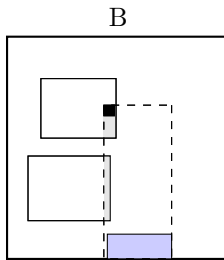


call CHOOSE-SUBTREE with 12, node *root*, level 1

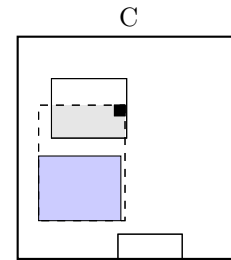
Next level are leaf nodes



old overlap= 0  
new overlap= 0  
overlap extension= 0  
area extension= 0.0



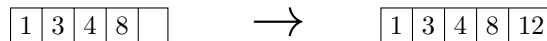
old overlap= 0  
new overlap= 0.2917751  
overlap extension= 0.2917751  
area extension= 3.1539002



old overlap= 0  
new overlap= 0.86596996  
overlap extension= 0.86596996  
area extension= 1.6658509

Node *A* is chosen

the node *A* is not full, add the record.



call ADJUST-TREE with *R*, node *A*

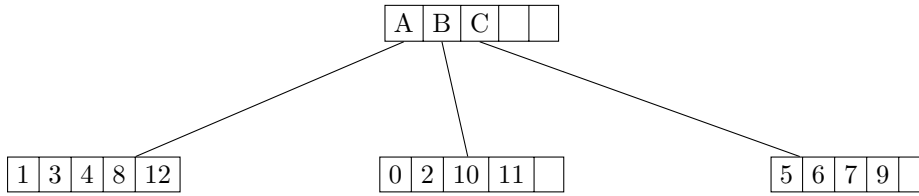
update MBR of node *A*.

continue by adjusting the parent node *root*

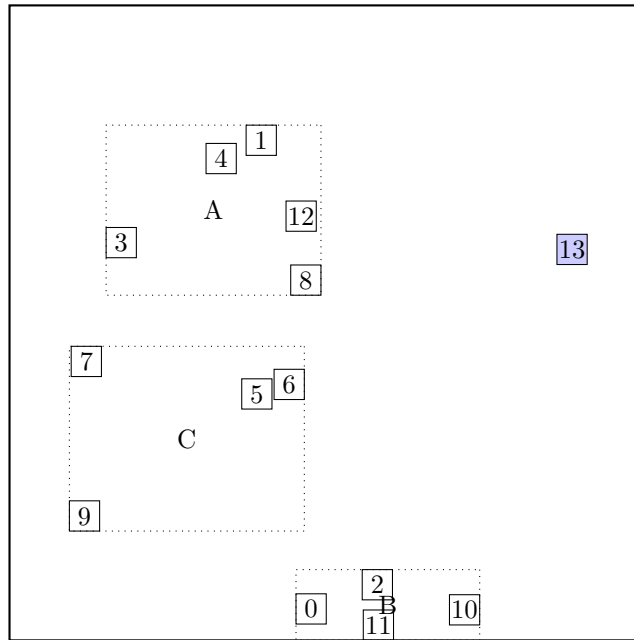
```
call ADJUST-TREE with  $R$ , node root
we are at the root
return from ADJUST-TREE
```

call INSERT, #S(P :X 3621/1000 :Y 1243/500)

structure view:

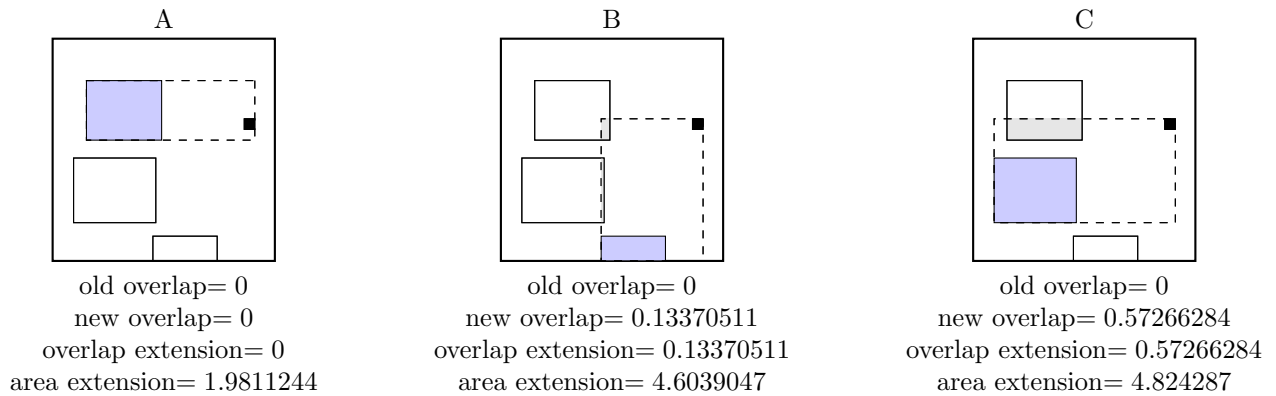


data view:



call CHOOSE-SUBTREE with 13, node *root*, level 1

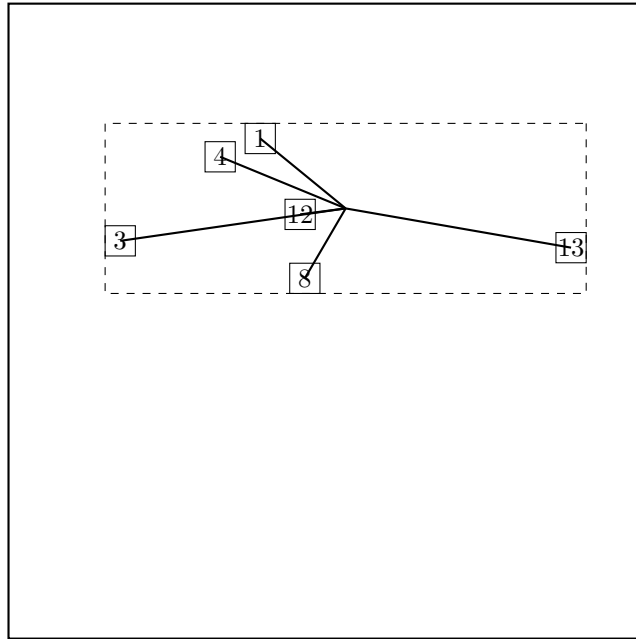
Next level are leaf nodes



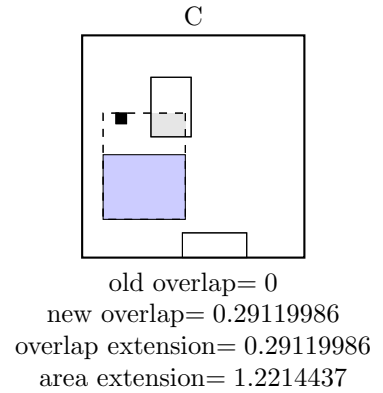
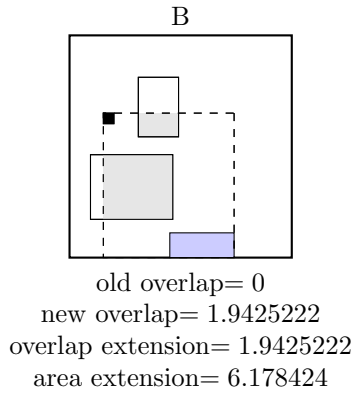
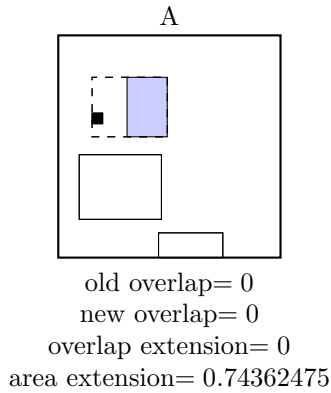
Node A is chosen

call OVERFLOW-TREATMENT

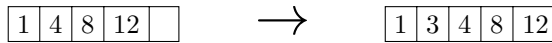
call REINSERT



The two most distant nodes (13 3) are removed and reinserted in reversed order.  
 The rest is kept.  
 Adjust MBR of A.  
 Reinserting 3.  
 call CHOOSE-SUBTREE with 3, node *root*, level 1  
 Next level are leaf nodes

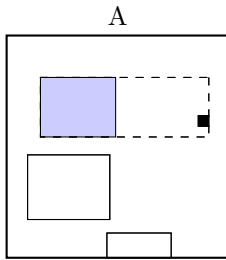


Node A is chosen  
 the node A is not full, add the record.

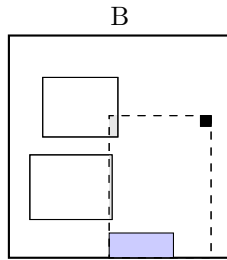


call ADJUST-TREE with *R*, node A  
 update MBR of node A.  
 continue by adjusting the parent node *root*

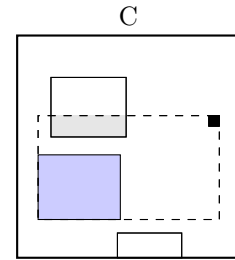
call ADJUST-TREE with *R*, node *root*  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT  
 Reinserting 13.  
 call CHOOSE-SUBTREE with 13, node *root*, level 1  
 Next level are leaf nodes



old overlap= 0  
 new overlap= 0  
 overlap extension= 0  
 area extension= 1.9811244



old overlap= 0  
 new overlap= 0.13370511  
 overlap extension= 0.13370511  
 area extension= 4.6039047



old overlap= 0  
 new overlap= 0.57266284  
 overlap extension= 0.57266284  
 area extension= 4.824287

Node A is chosen

call OVERFLOW-TREATMENT

Second overflow on the same level during one insert — make a split.

call SPLIT-NODE with nodes A and 13

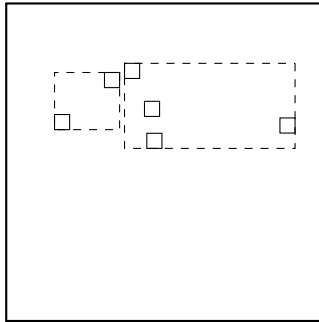
call CHOOSE-SPLIT-AXIS with ...

Axis:  $x$

Entries sorted by low: 3 4 1 12 8 13

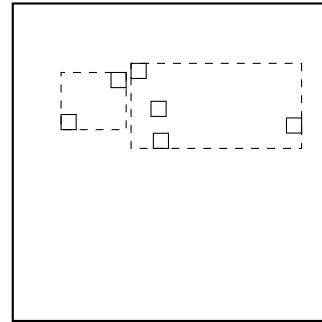
Entries sorted by high: 3 4 1 12 8 13

partition by low: 3 4 — 1 12 8 13



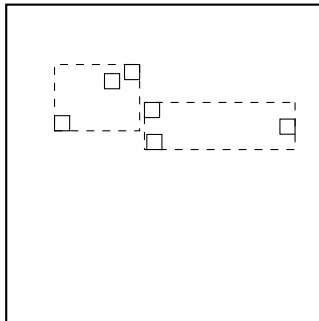
$o = 4.9979997$

partition by high: 3 4 — 1 12 8 13



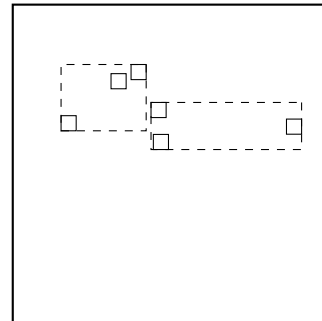
$o = 4.9979997$

partition by low: 3 4 1 — 12 8 13



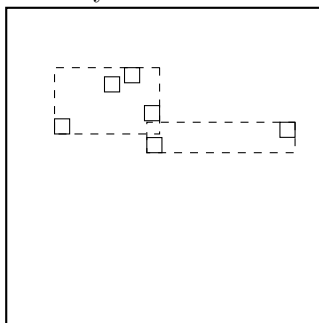
$o = 4.618$

partition by high: 3 4 1 — 12 8 13



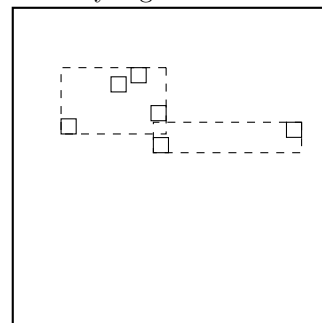
$o = 4.618$

partition by low: 3 4 1 12 — 8 13



$o = 4.6309996$

partition by high: 3 4 1 12 — 8 13



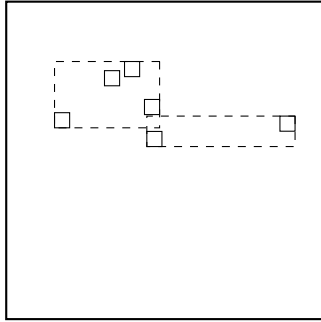
$o = 4.6309996$

Axis:  $y$

Entries sorted by low: 8 13 3 12 4 1

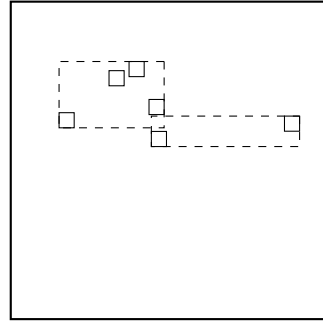
Entries sorted by high: 8 13 3 12 4 1

partition by low: 8 13 — 3 12 4 1



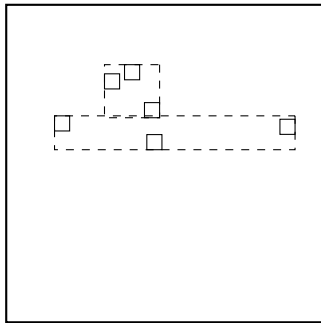
$o = 4.6309996$

partition by high: 8 13 — 3 12 4 1



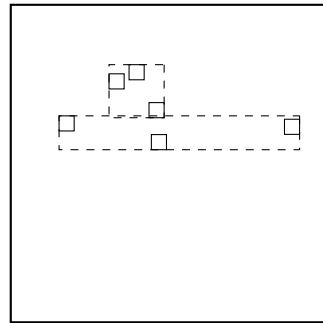
$o = 4.6309996$

partition by low: 8 13 3 — 12 4 1



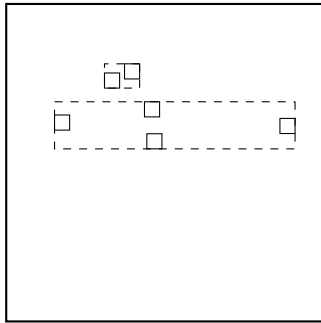
$o = 5.0609994$

partition by high: 8 13 3 — 12 4 1



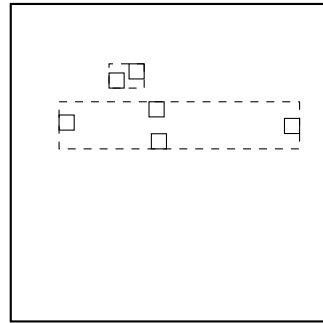
$o = 5.0609994$

partition by low: 8 13 3 12 — 4 1



$o = 4.5909996$

partition by high: 8 13 3 12 — 4 1



$o = 4.5909996$

Minimal margin ( $o = 4.5909996$ ) was reached for axis  $y$ .  
return from CHOOSE-SPLIT-AXIS

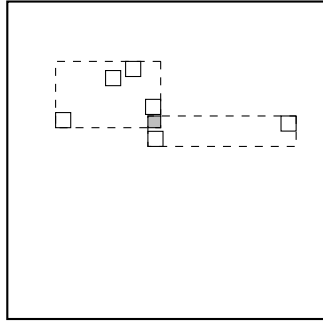
call CHOOSE-SPLIT-INDEX with entries, axis  $y$ ,  $R$

Entries sorted by low at axis  $y$ : 8 13 3 12 4 1

Entries sorted by high at axis  $y$ : 8 13 3 12 4 1

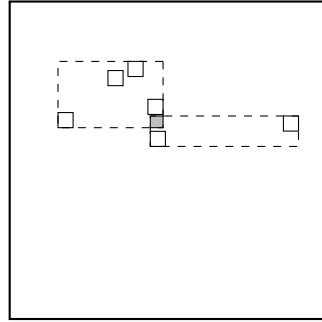


Partition by low: 8 13 — 3 12 4 1



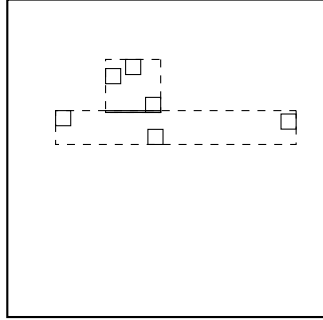
overlap = 0.026194999  
 $S = 2.0093127$

Partition by high: 8 13 — 3 12 4 1



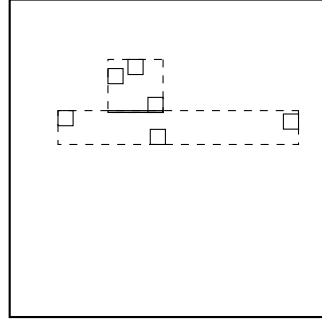
overlap = 0.026194999  
 $S = 2.0093127$

Partition by low: 8 13 3 — 12 4 1



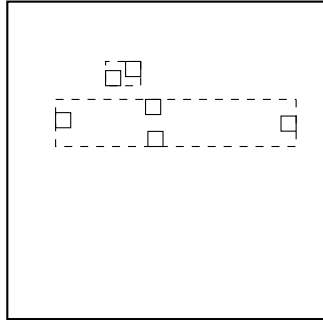
overlap = 0.018224724  
 $S = 1.9372928$

Partition by high: 8 13 3 — 12 4 1



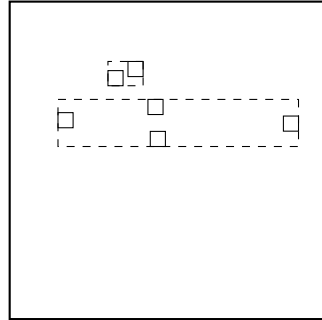
overlap = 0.018224724  
 $S = 1.9372928$

Partition by low: 8 13 3 12 — 4 1



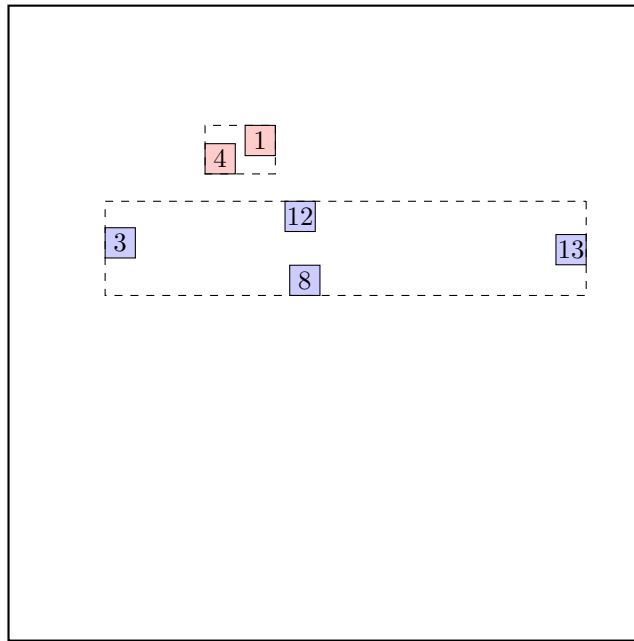
overlap = 0  
 $S = 2.1316507$

Partition by high: 8 13 3 12 — 4 1



overlap = 0  
 $S = 2.1316507$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

call ADJUST-TREE with  $R$ , node A and the new node  
 update MBR of node A.  
 add the new node to the parent node root

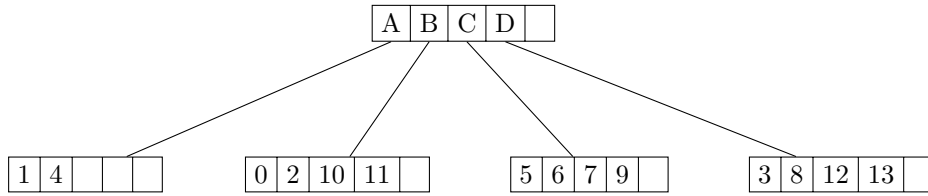
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT

call ADJUST-TREE with  $R$ , node D  
 update MBR of node D.  
 continue by adjusting the parent node root

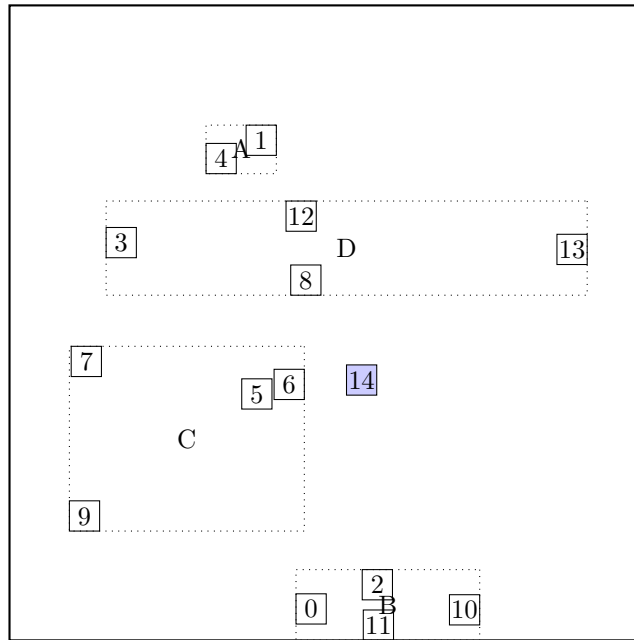
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE

call INSERT, #S(P :X 2229/1000 :Y 811/500)

structure view:

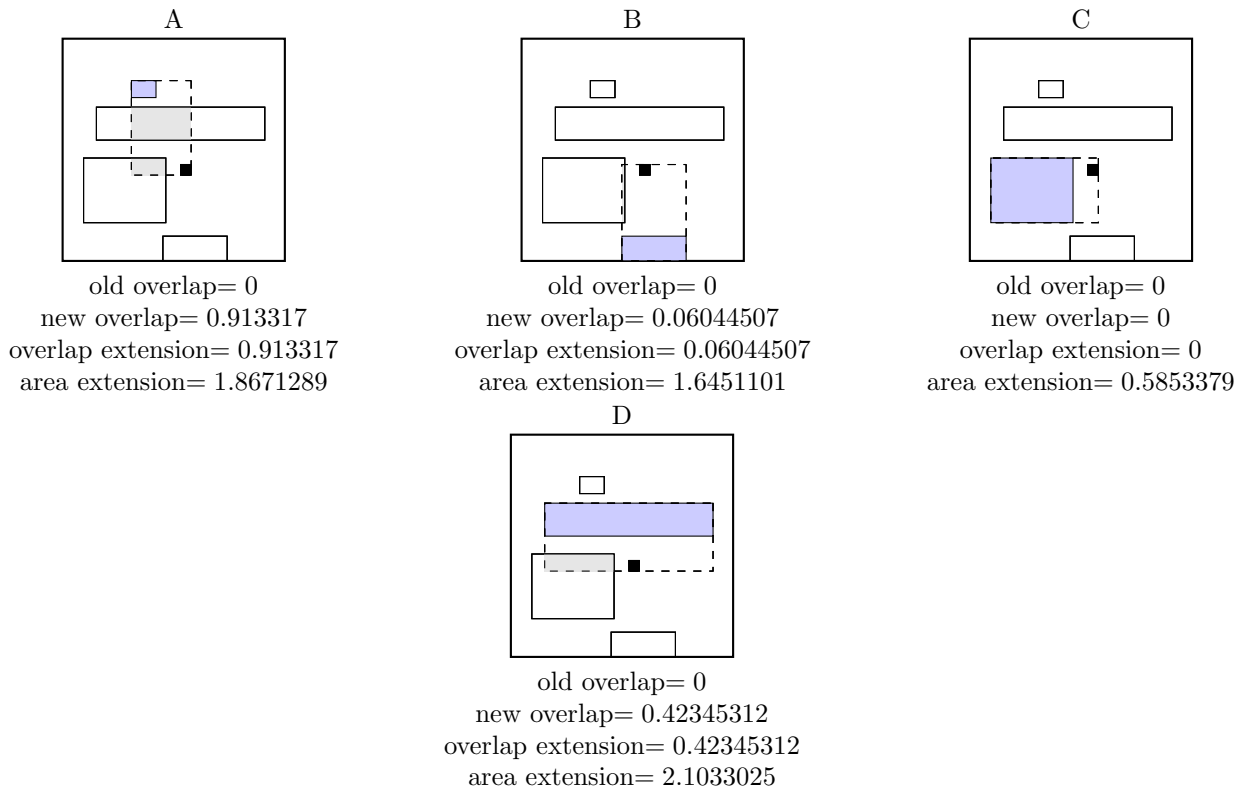


data view:

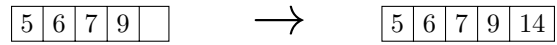


call CHOOSE-SUBTREE with 14, node *root*, level 1

Next level are leaf nodes



Node  $C$  is chosen  
the node  $C$  is not full, add the record.

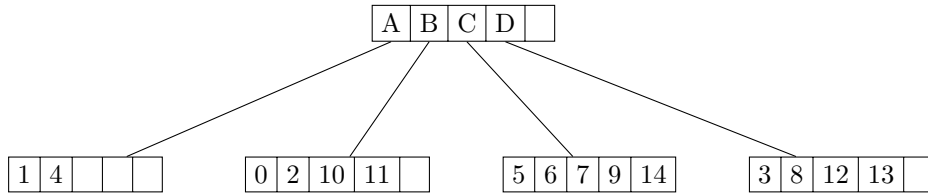


call ADJUST-TREE with  $R$ , node  $C$   
update MBR of node  $C$ .  
continue by adjusting the parent node root

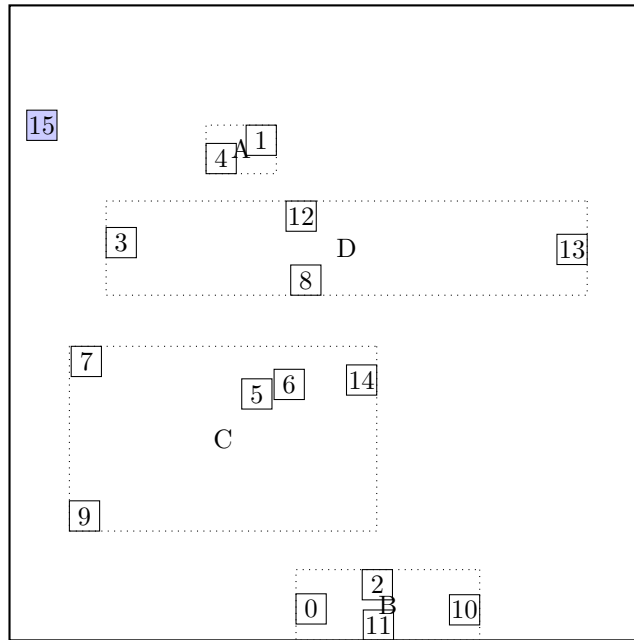
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 57/500 :Y 3307/1000)

structure view:

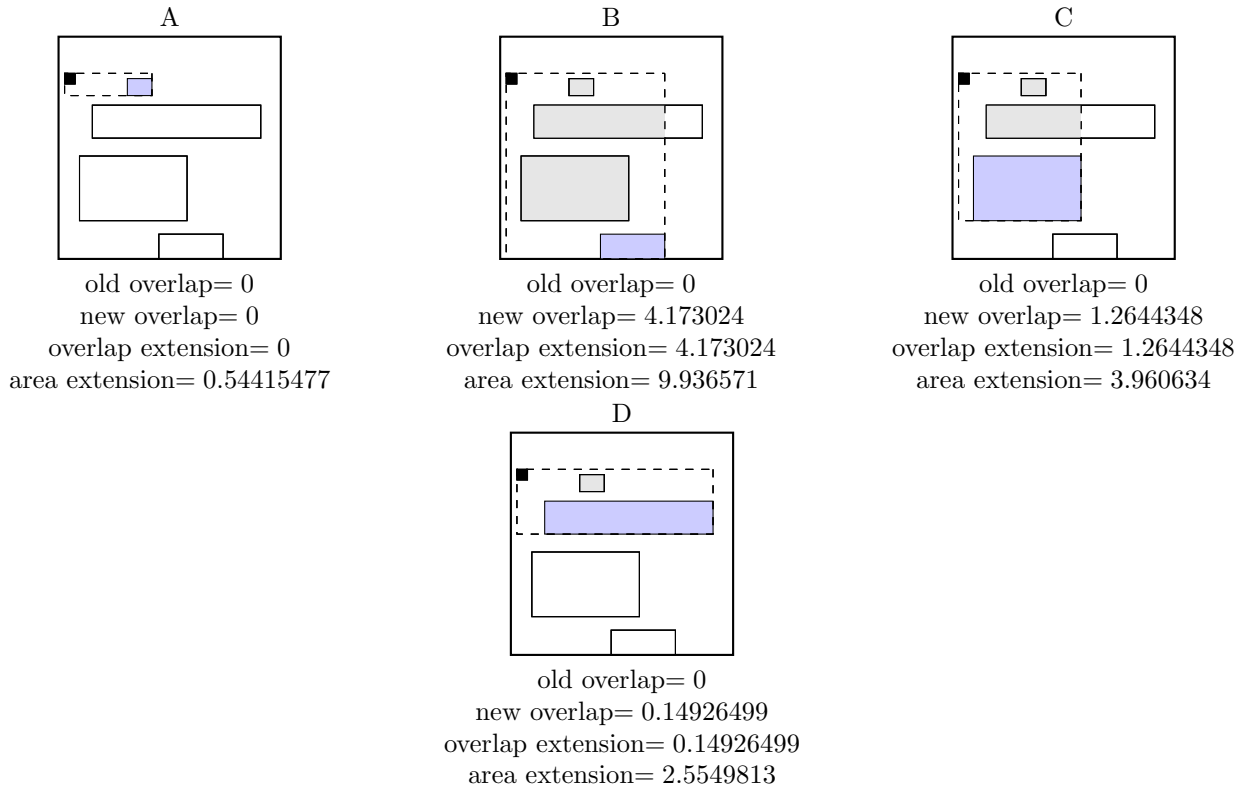


data view:

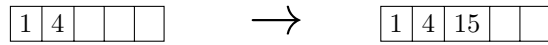


call CHOOSE-SUBTREE with 15, node *root*, level 1

Next level are leaf nodes



Node  $A$  is chosen  
the node  $A$  is not full, add the record.

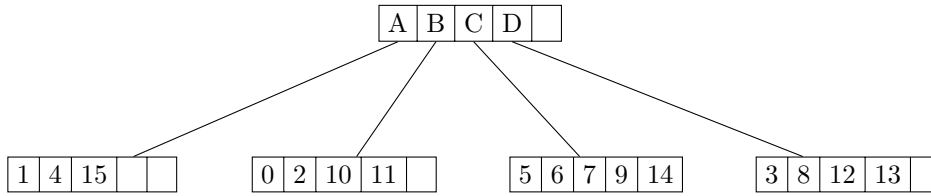


call ADJUST-TREE with  $R$ , node  $A$   
update MBR of node  $A$ .  
continue by adjusting the parent node root

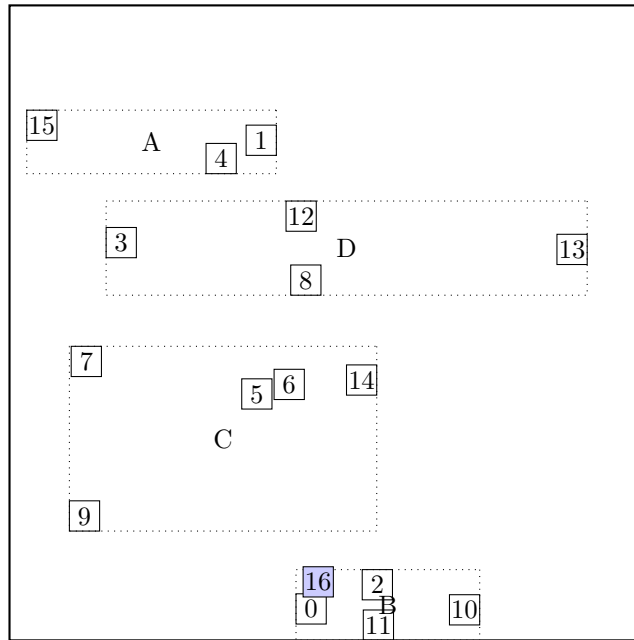
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 971/500 :Y 287/1000)

structure view:

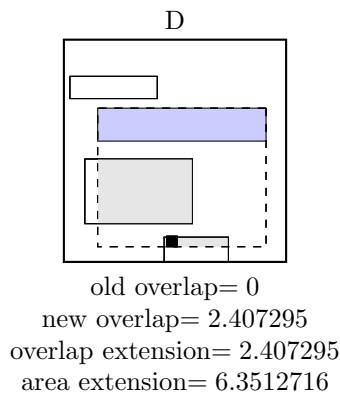
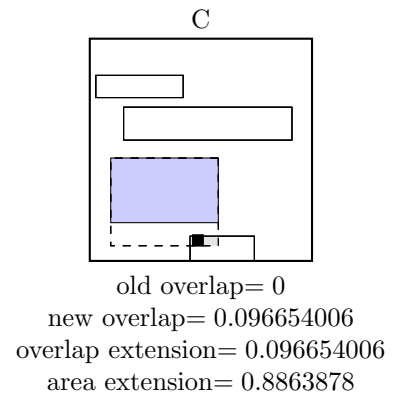
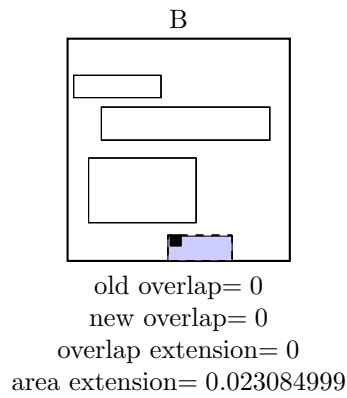
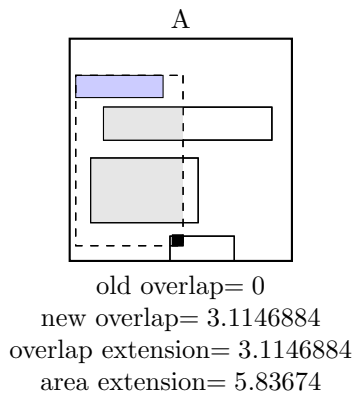


data view:

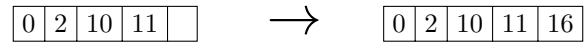


call CHOOSE-SUBTREE with 16, node *root*, level 1

Next level are leaf nodes



Node  $B$  is chosen  
the node  $B$  is not full, add the record.



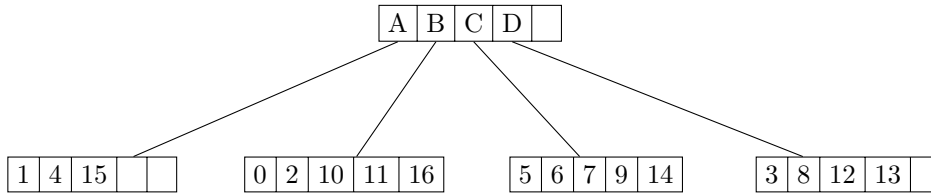
call ADJUST-TREE with  $R$ , node  $B$   
update MBR of node  $B$ .  
continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

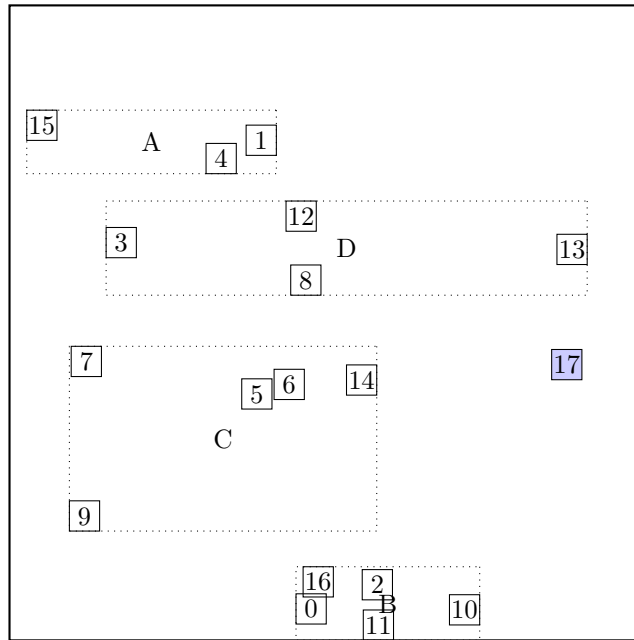


call INSERT, #S(P :X 1793/500 :Y 431/250)

structure view:

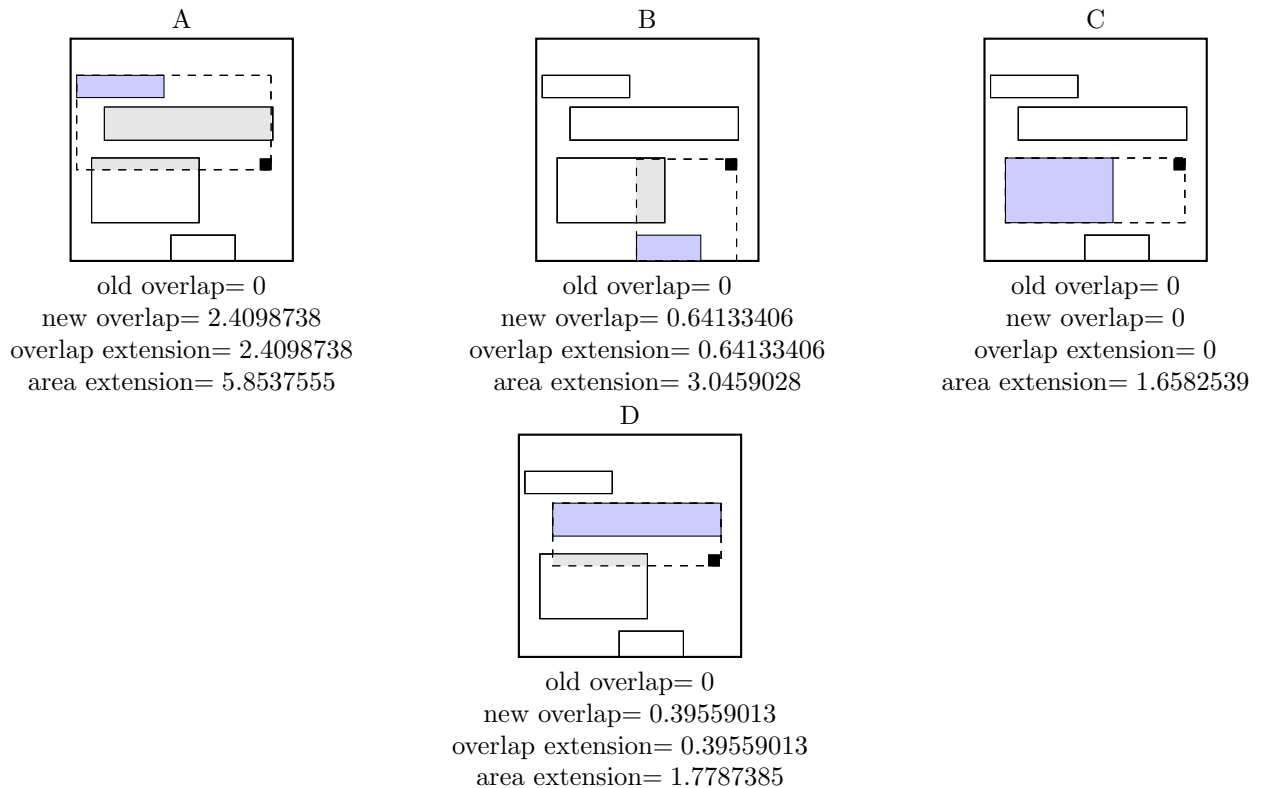


data view:



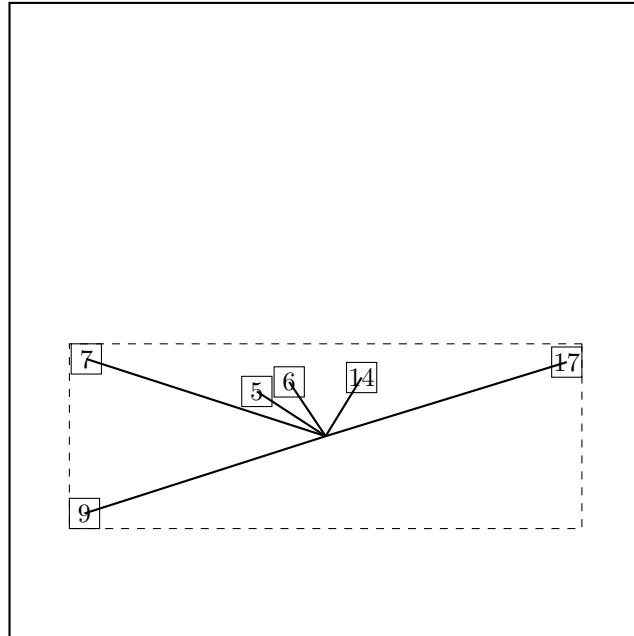
call CHOOSE-SUBTREE with 17, node *root*, level 1

Next level are leaf nodes



Node *C* is chosen  
 call OVERFLOW-TREATMENT

call REINSERT



The two most distant nodes (9 17) are removed and reinserted in reversed order.

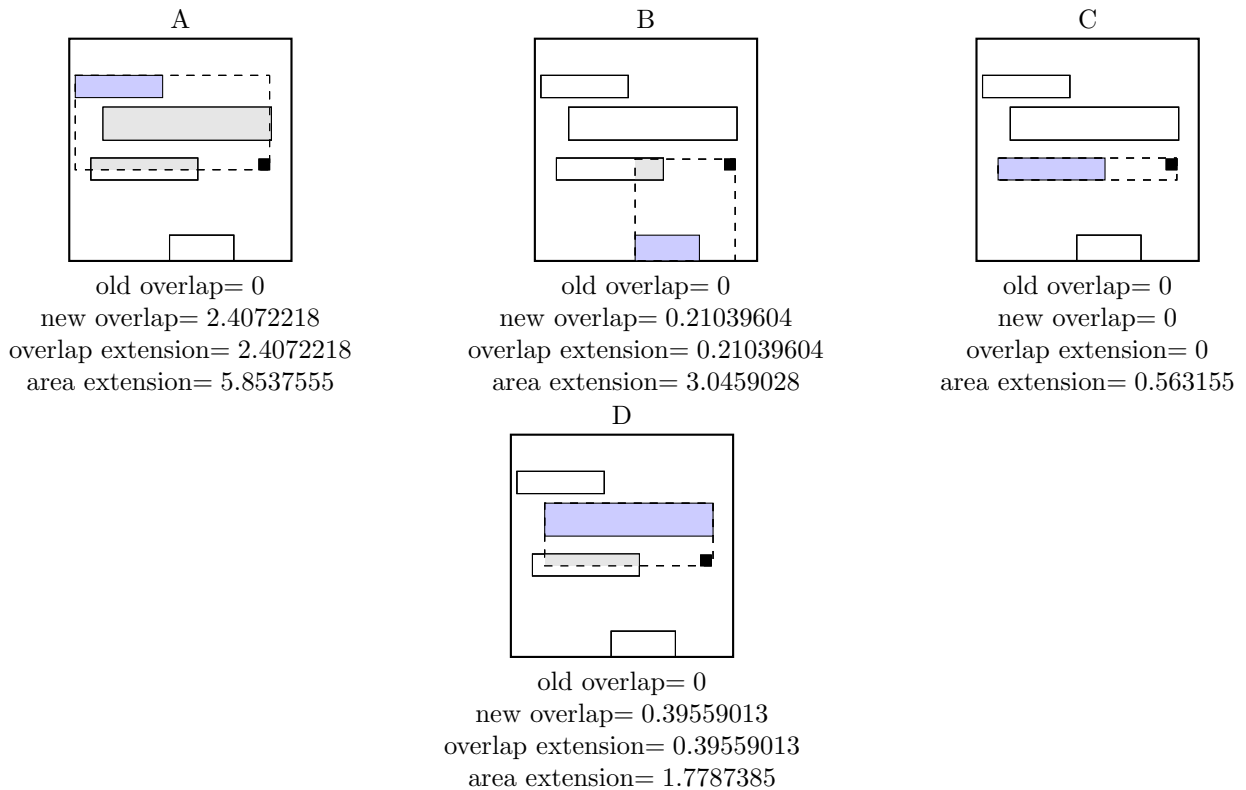
The rest is kept.

Adjust MBR of *C*.

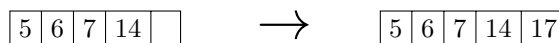
Reinserting 17.

call CHOOSE-SUBTREE with 17, node *root*, level 1

Next level are leaf nodes

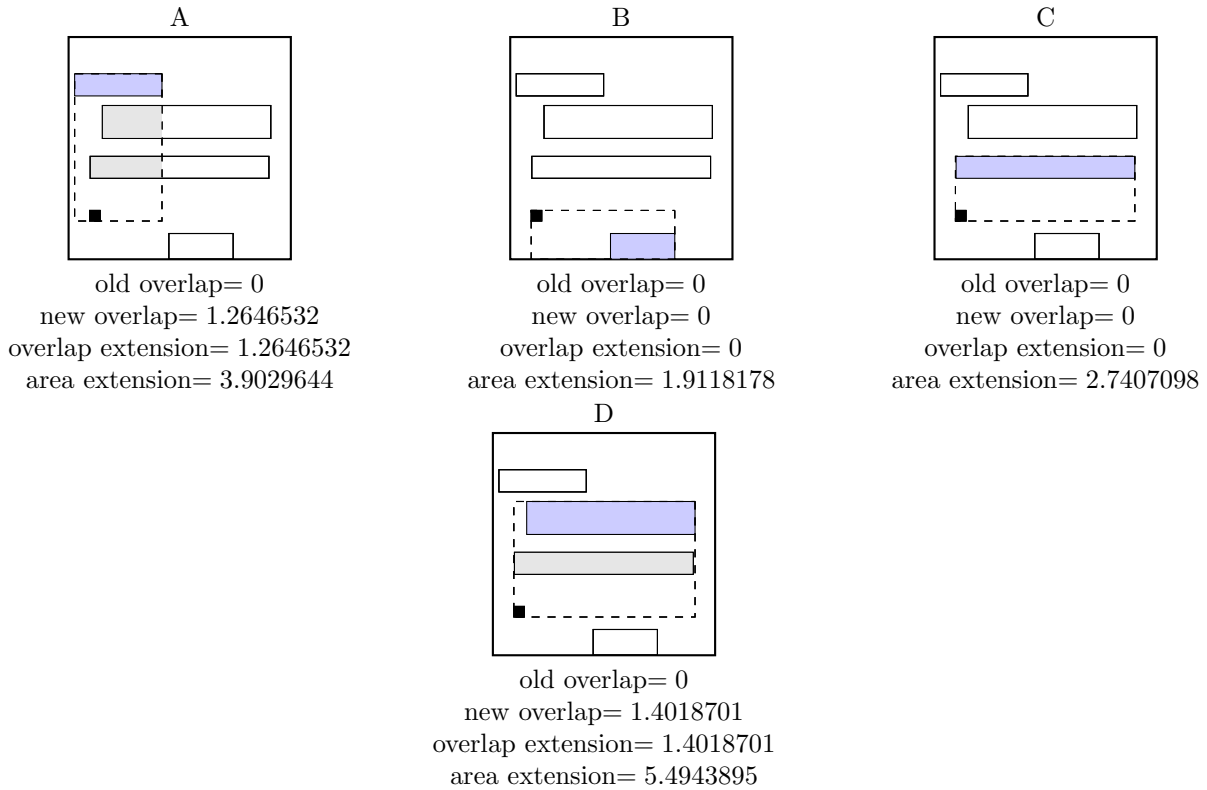


Node *C* is chosen  
 the node *C* is not full, add the record.



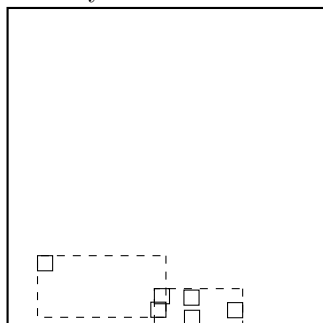
call ADJUST-TREE with  $R$ , node C  
 update MBR of node C.  
 continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT  
 Reinserting 9.  
 call CHOOSE-SUBTREE with 9, node *root*, level 1  
 Next level are leaf nodes

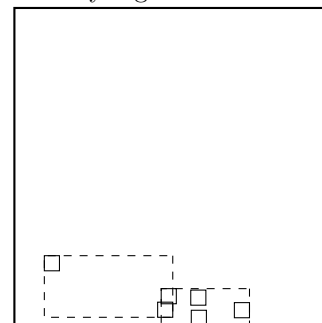


Node  $B$  is chosen  
 call OVERFLOW-TREATMENT  
 Second overflow on the same level during one insert — make a split.  
 call SPLIT-NODE with nodes B and 9  
 call CHOOSE-SPLIT-AXIS with ...  
 Axis:  $x$   
 Entries sorted by low: 9 0 16 2 11 10  
 Entries sorted by high: 9 0 16 2 11 10

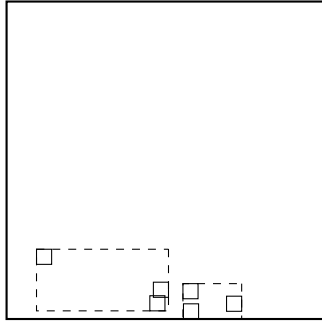
partition by low: 9 0 — 16 2 11 10



partition by high: 9 0 — 16 2 11 10

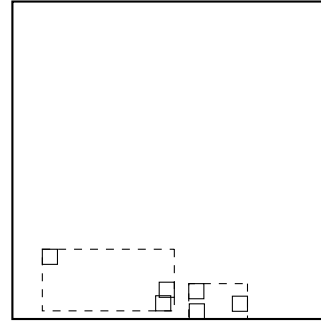


partition by low: 9 0 16 — 2 11 10



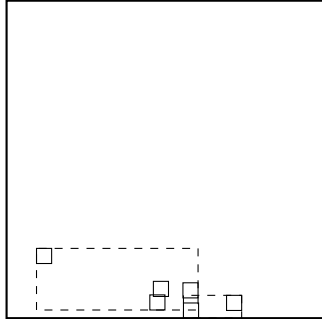
$o = 3.805$

partition by high: 9 0 16 — 2 11 10



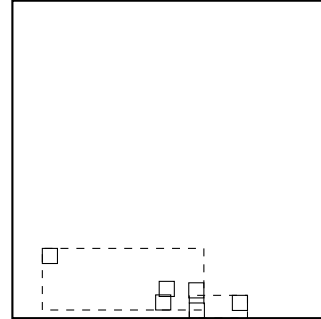
$o = 3.805$

partition by low: 9 0 16 2 — 11 10



$o = 4.023$

partition by high: 9 0 16 2 — 11 10



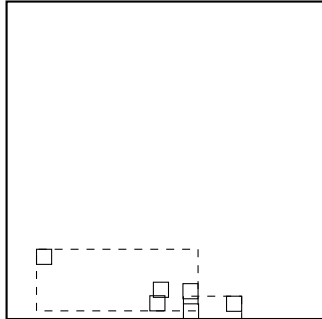
$o = 4.023$

Axis:  $y$

Entries sorted by low: 11 10 0 2 16 9

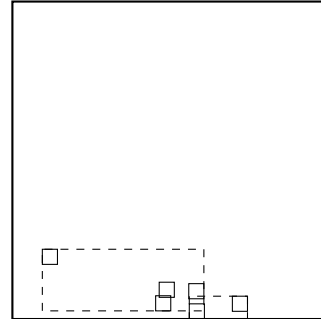
Entries sorted by high: 11 10 0 2 16 9

partition by low: 11 10 — 0 2 16 9



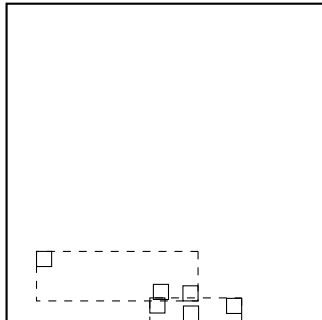
$o = 4.023$

partition by high: 11 10 — 0 2 16 9



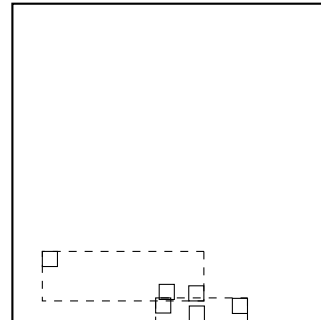
$o = 4.023$

partition by low: 11 10 0 — 2 16 9



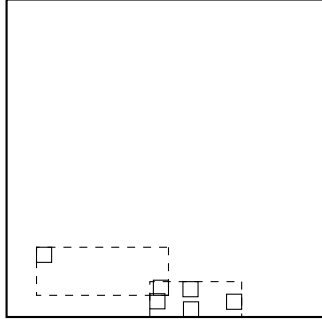
$o = 4.314$

partition by high: 11 10 0 — 2 16 9



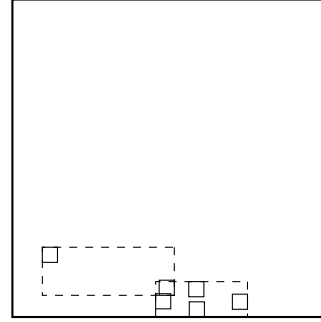
$o = 4.314$

partition by low: 11 10 0 2 — 16 9



$o = 4.064$

partition by high: 11 10 0 2 — 16 9

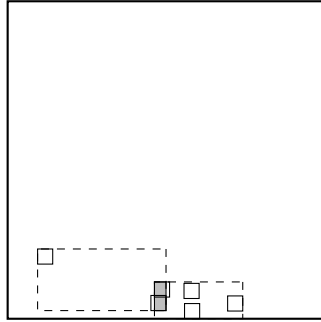


$o = 4.064$

Minimal margin ( $o = 3.805$ ) was reached for axis  $x$ .  
return from CHOOSE-SPLIT-AXIS

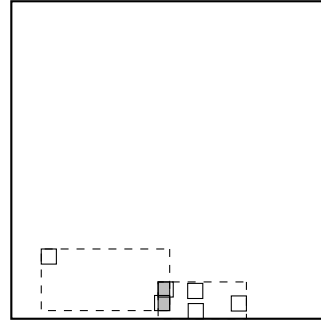
call CHOOSE-SPLIT-INDEX with entries, axis  $x$ ,  $R$   
Entries sorted by low at axis  $x$ : 9 0 16 2 11 10  
Entries sorted by high at axis  $x$ : 9 0 16 2 11 10

Partition by low: 9 0 — 16 2 11 10



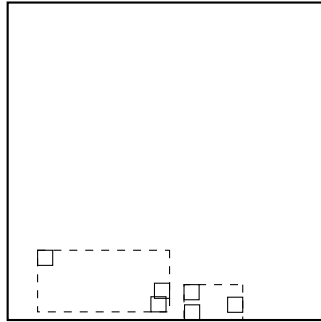
overlap = 0.057987  
 $S = 1.952333$

Partition by high: 9 0 — 16 2 11 10



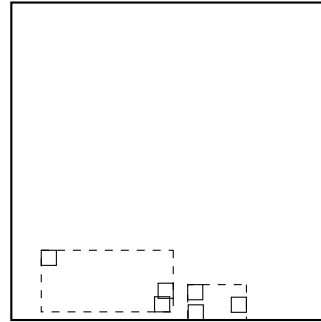
overlap = 0.057987  
 $S = 1.952333$

Partition by low: 9 0 16 — 2 11 10



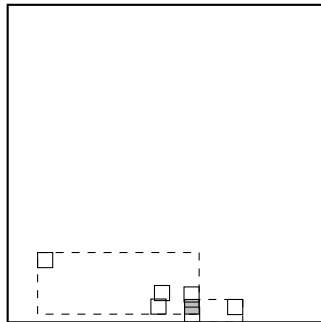
overlap = 0  
 $S = 1.7858491$

Partition by high: 9 0 16 — 2 11 10



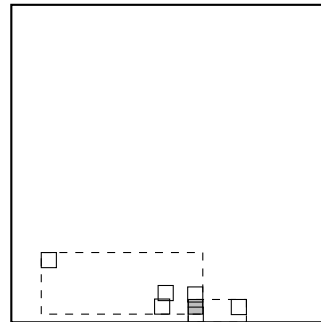
overlap = 0  
 $S = 1.7858491$

Partition by low: 9 0 16 2 — 11 10



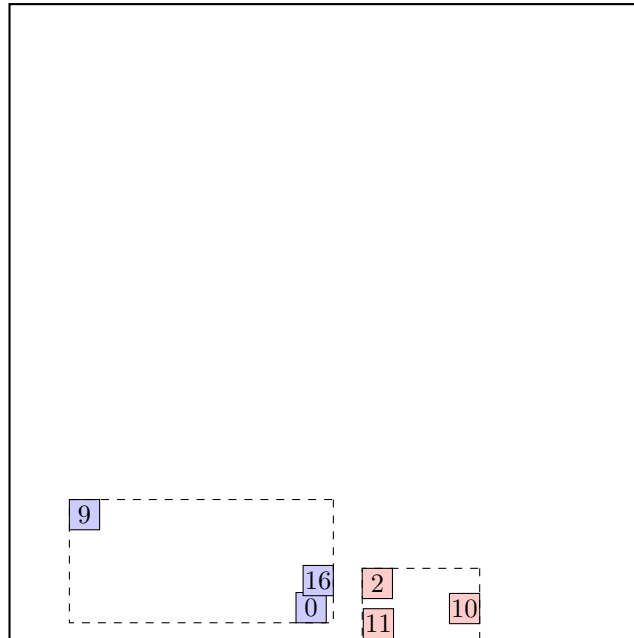
overlap = 0.03744197  
 $S = 1.9734249$

Partition by high: 9 0 16 2 — 11 10



overlap = 0.03744197  
 $S = 1.9734249$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

call ADJUST-TREE with  $R$ , node B and the new node  
update MBR of node B.  
add the new node to the parent node root

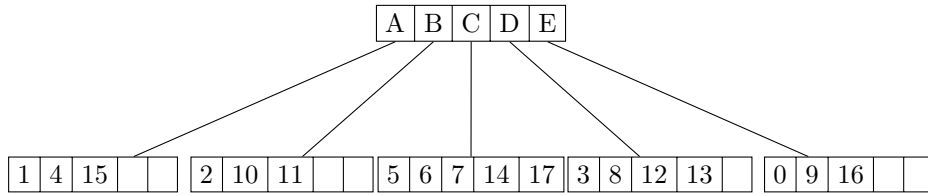
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE  
return from REINSERT

call ADJUST-TREE with  $R$ , node C  
update MBR of node C.  
continue by adjusting the parent node root

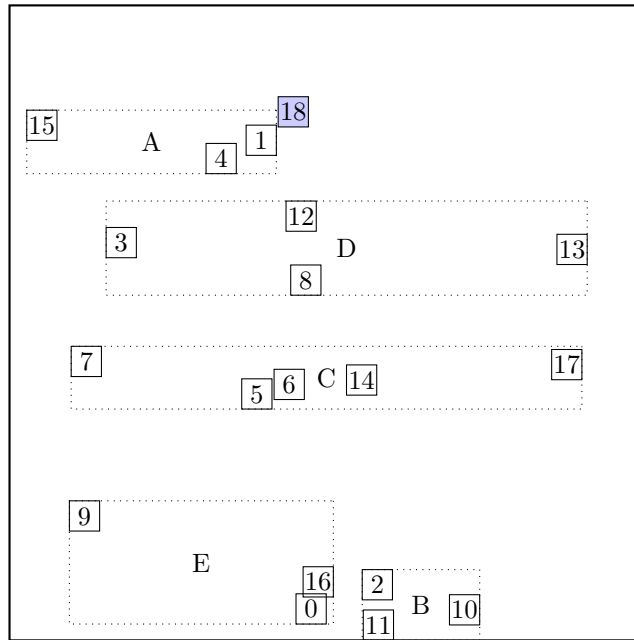
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 1777/1000 :Y 849/250)

structure view:

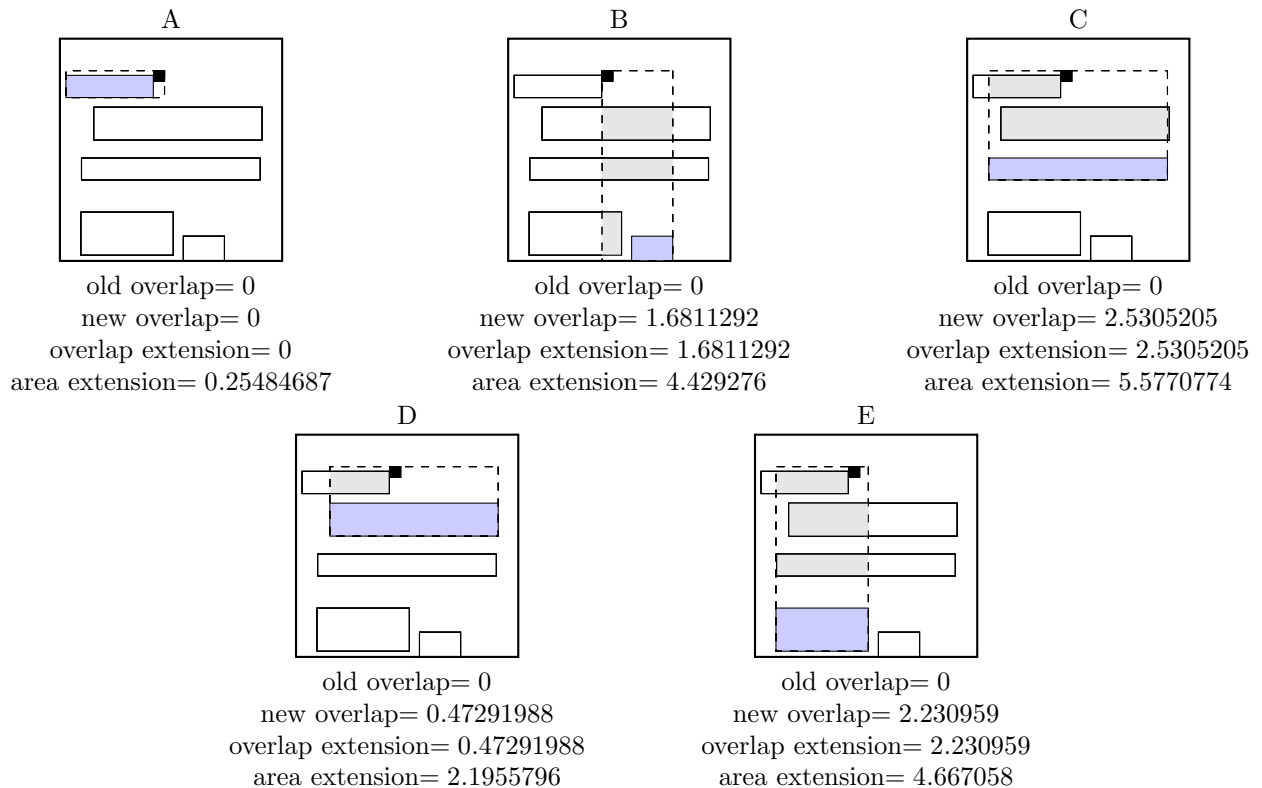


data view:

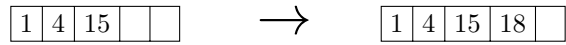


call CHOOSE-SUBTREE with 18, node *root*, level 1

Next level are leaf nodes



Node  $A$  is chosen  
the node  $A$  is not full, add the record.



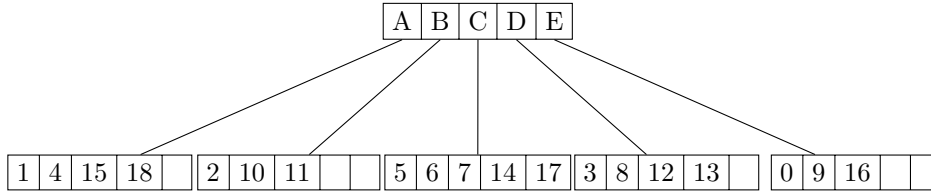
call ADJUST-TREE with  $R$ , node  $A$   
update MBR of node  $A$ .  
continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

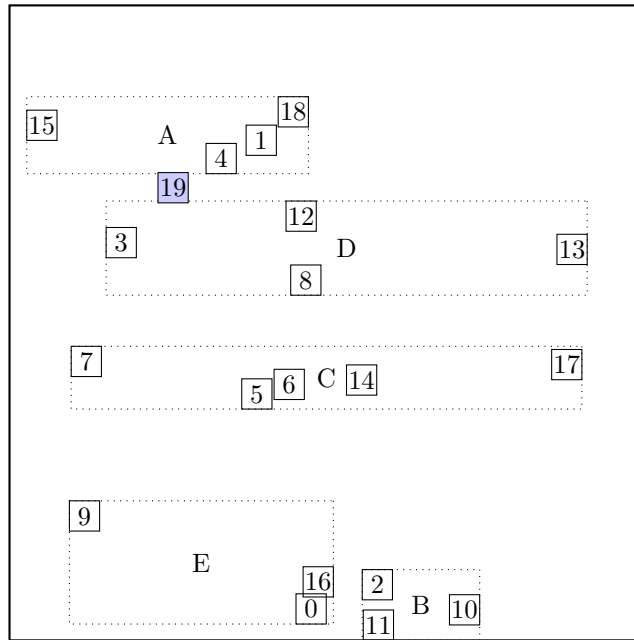


call INSERT, #S(P :X 491/500 :Y 1447/500)

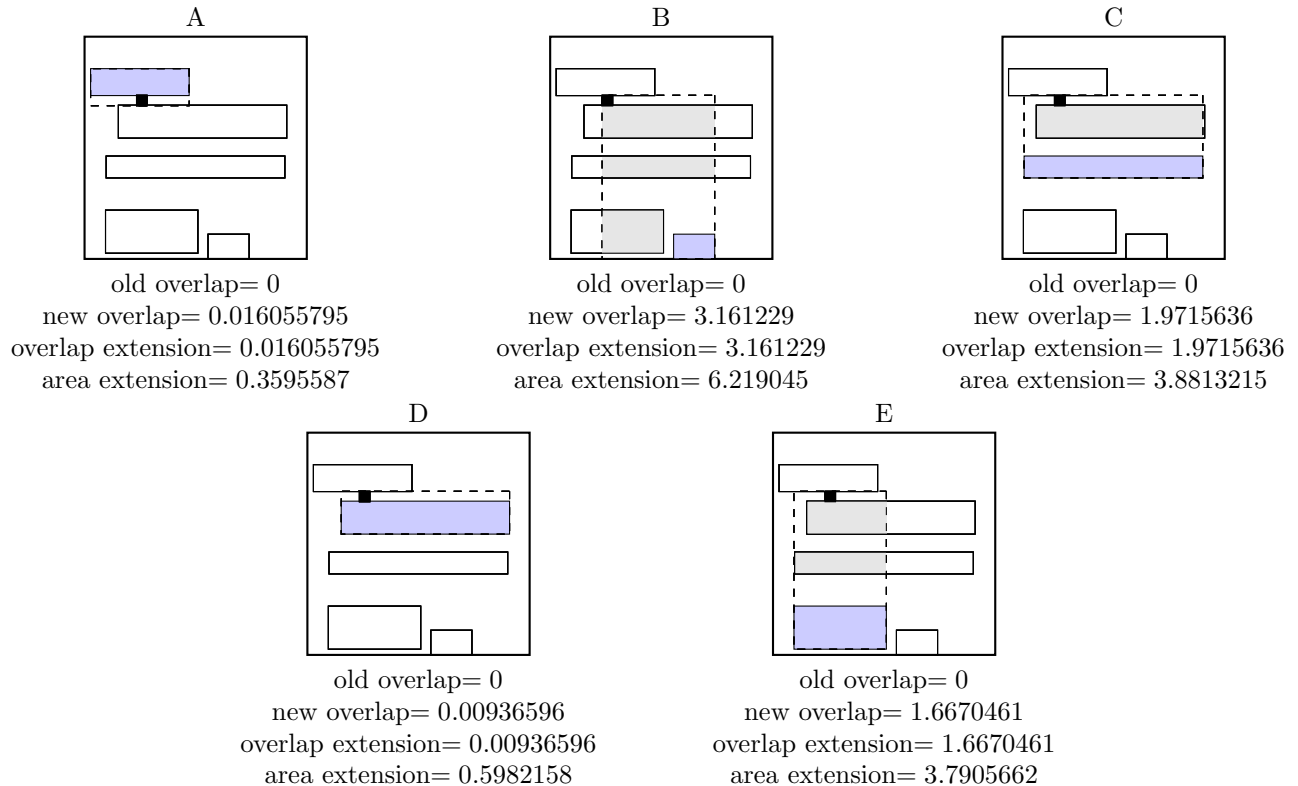
structure view:



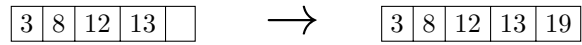
data view:



call CHOOSE-SUBTREE with 19, node *root*, level 1  
Next level are leaf nodes



Node  $D$  is chosen  
the node  $D$  is not full, add the record.

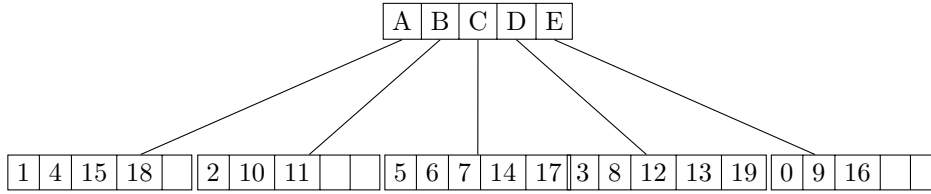


call ADJUST-TREE with  $R$ , node  $D$   
update MBR of node  $D$ .  
continue by adjusting the parent node root

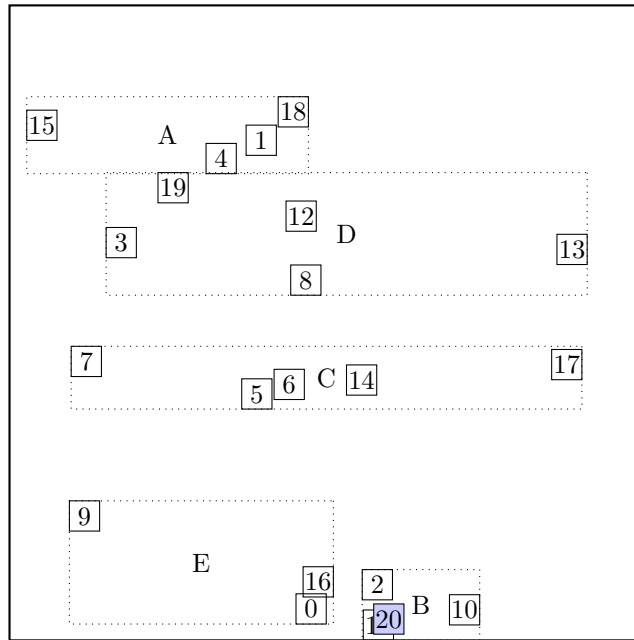
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 2409/1000 :Y 41/1000)

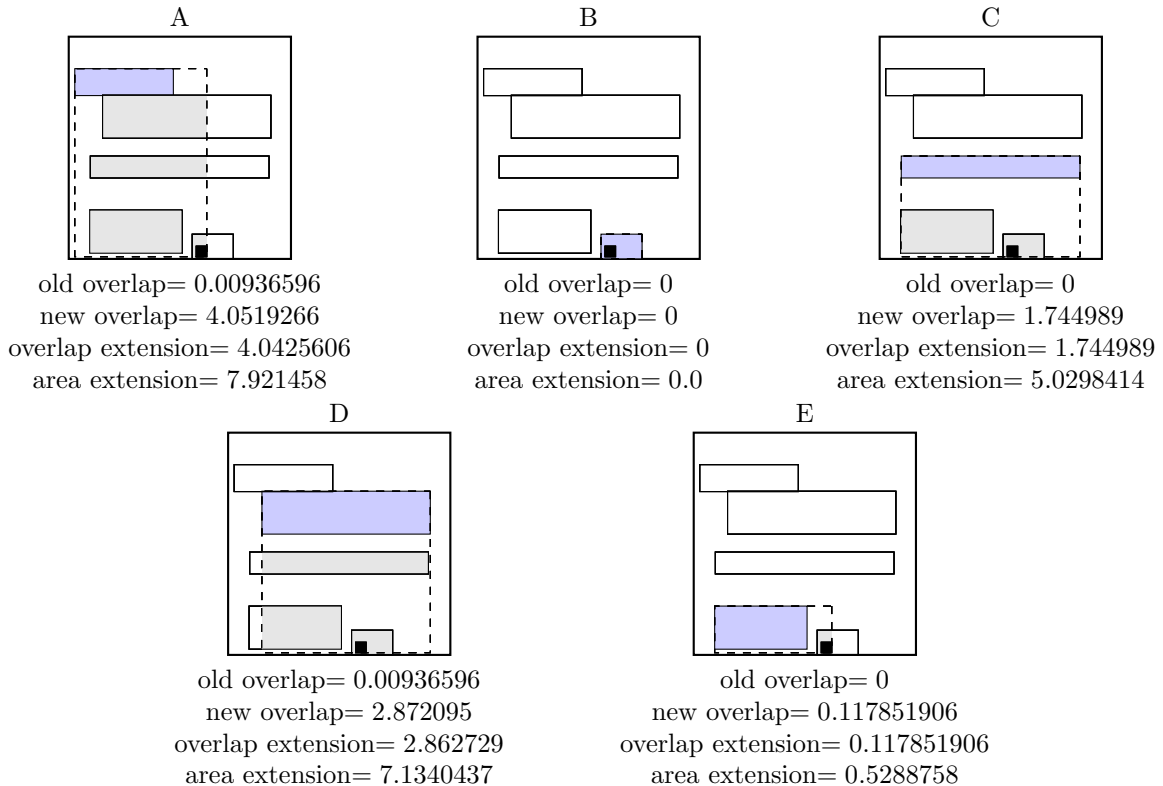
structure view:



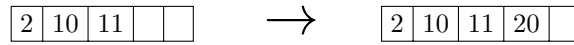
data view:



call CHOOSE-SUBTREE with 20, node *root*, level 1  
Next level are leaf nodes



Node  $B$  is chosen  
the node  $B$  is not full, add the record.

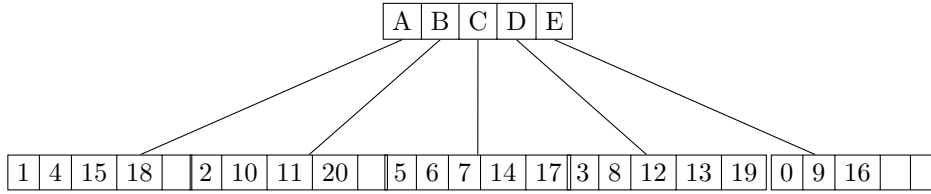


call ADJUST-TREE with  $R$ , node  $B$   
update MBR of node  $B$ .  
continue by adjusting the parent node root

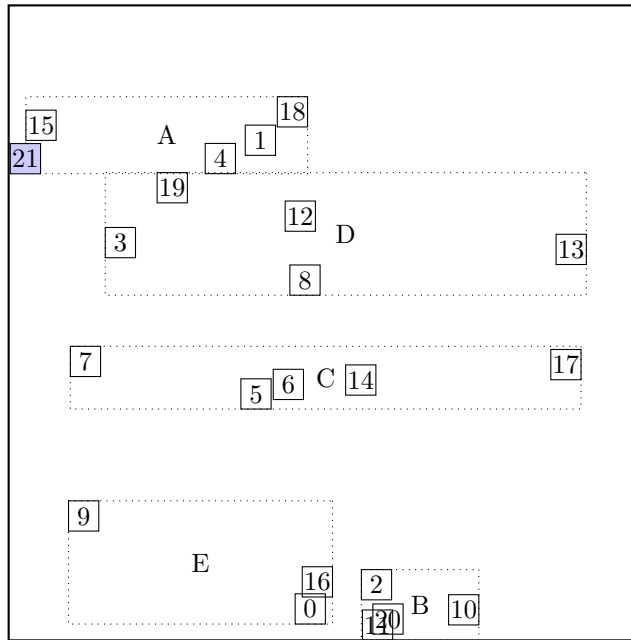
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 11/1000 :Y 1543/500)

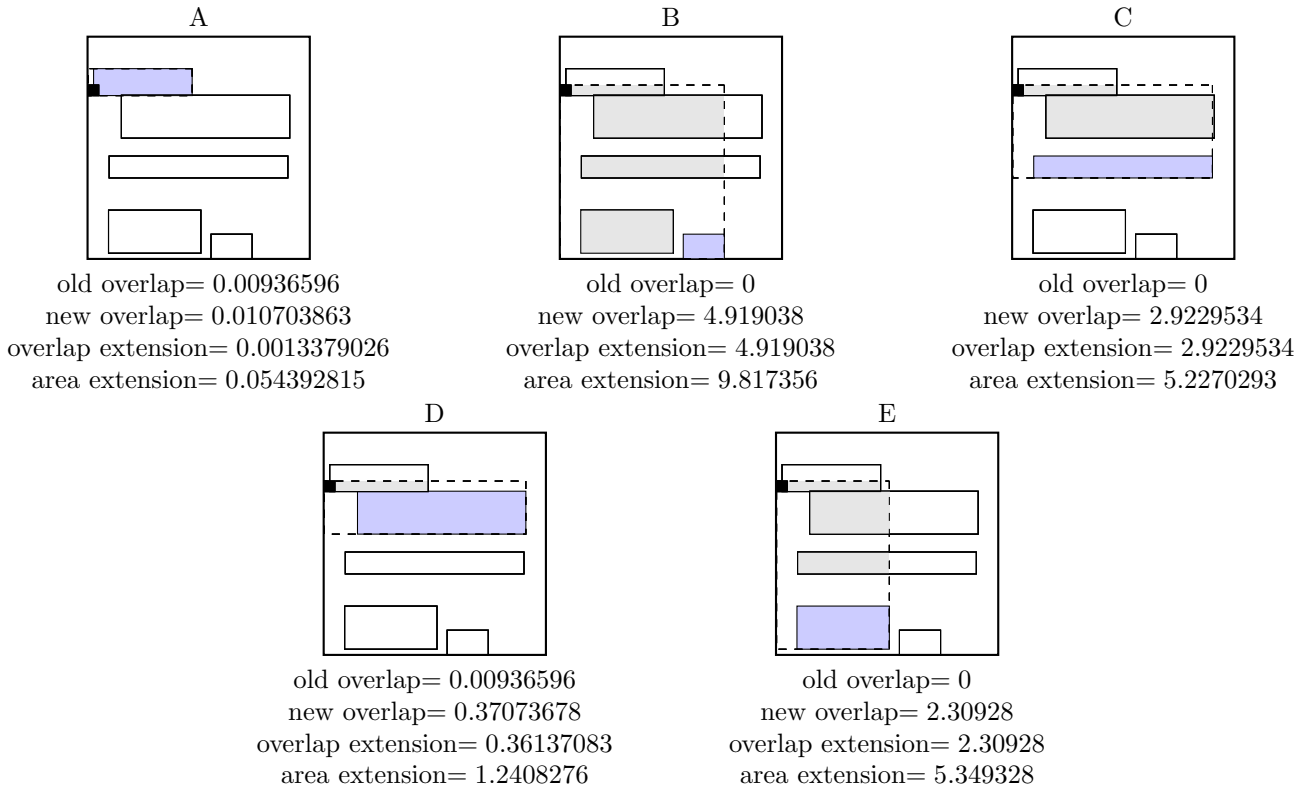
structure view:



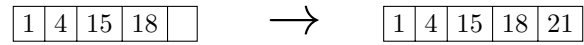
data view:



call CHOOSE-SUBTREE with 21, node *root*, level 1  
Next level are leaf nodes



Node  $A$  is chosen  
the node  $A$  is not full, add the record.

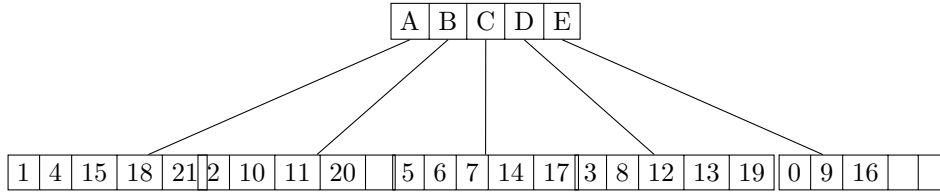


call ADJUST-TREE with  $R$ , node  $A$   
update MBR of node  $A$ .  
continue by adjusting the parent node root

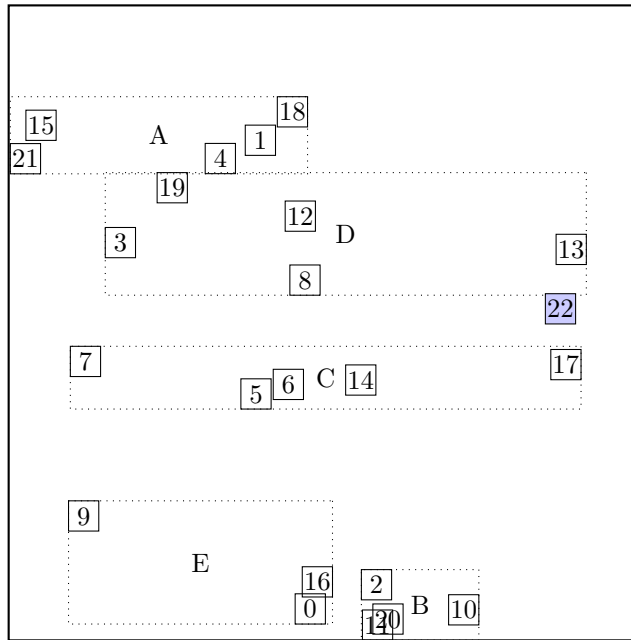
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 71/20 :Y 2093/1000)

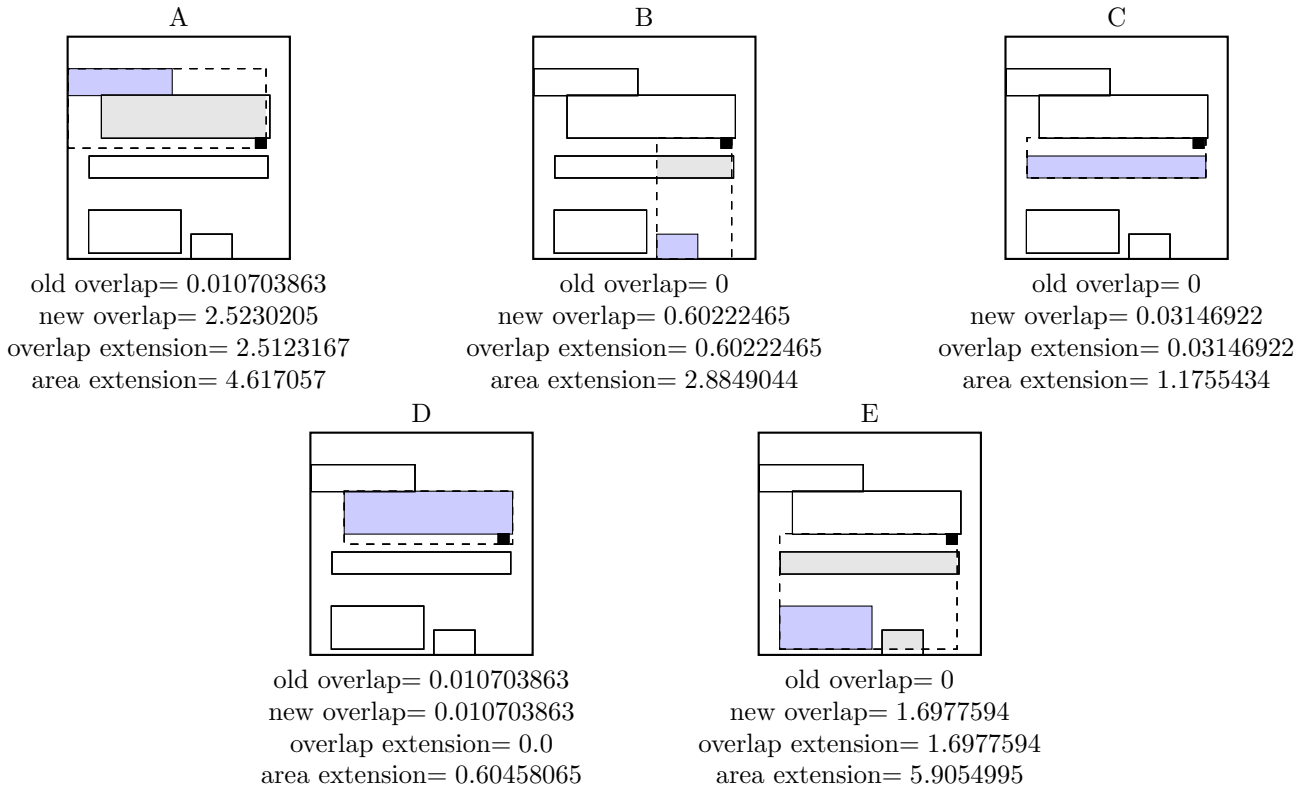
structure view:



data view:

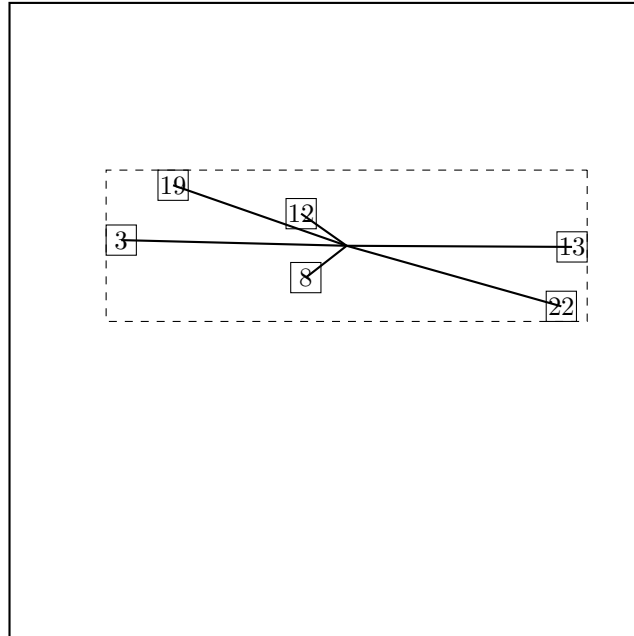


call CHOOSE-SUBTREE with 22, node *root*, level 1  
Next level are leaf nodes



Node *D* is chosen  
 call OVERFLOW-TREATMENT

call REINSERT



The two most distant nodes (3 13) are removed and reinserted in reversed order.

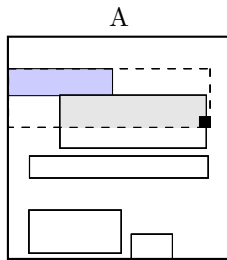
The rest is kept.

Adjust MBR of *D*.

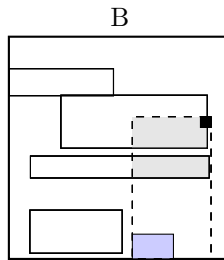
Reinserting 13.

call CHOOSE-SUBTREE with 13, node *root*, level 1

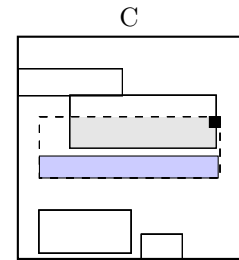
Next level are leaf nodes



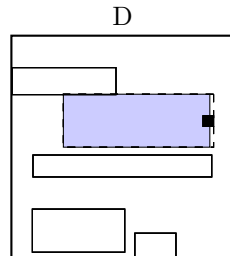
old overlap= 0.007959898  
 new overlap= 1.6829434  
 overlap extension= 1.6749835  
 area extension= 3.2264395



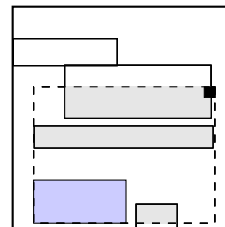
old overlap= 0  
 new overlap= 1.4432759  
 overlap extension= 1.4432759  
 area extension= 3.6324208



old overlap= 0  
 new overlap= 1.6414241  
 overlap extension= 1.6414241  
 area extension= 2.5435577



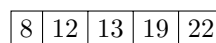
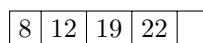
old overlap= 0.007959898  
 new overlap= 0.007959898  
 overlap extension= 0.0  
 area extension= 0.07107115



old overlap= 0  
 new overlap= 3.3230143  
 overlap extension= 3.3230143  
 area extension= 7.40666

Node *D* is chosen

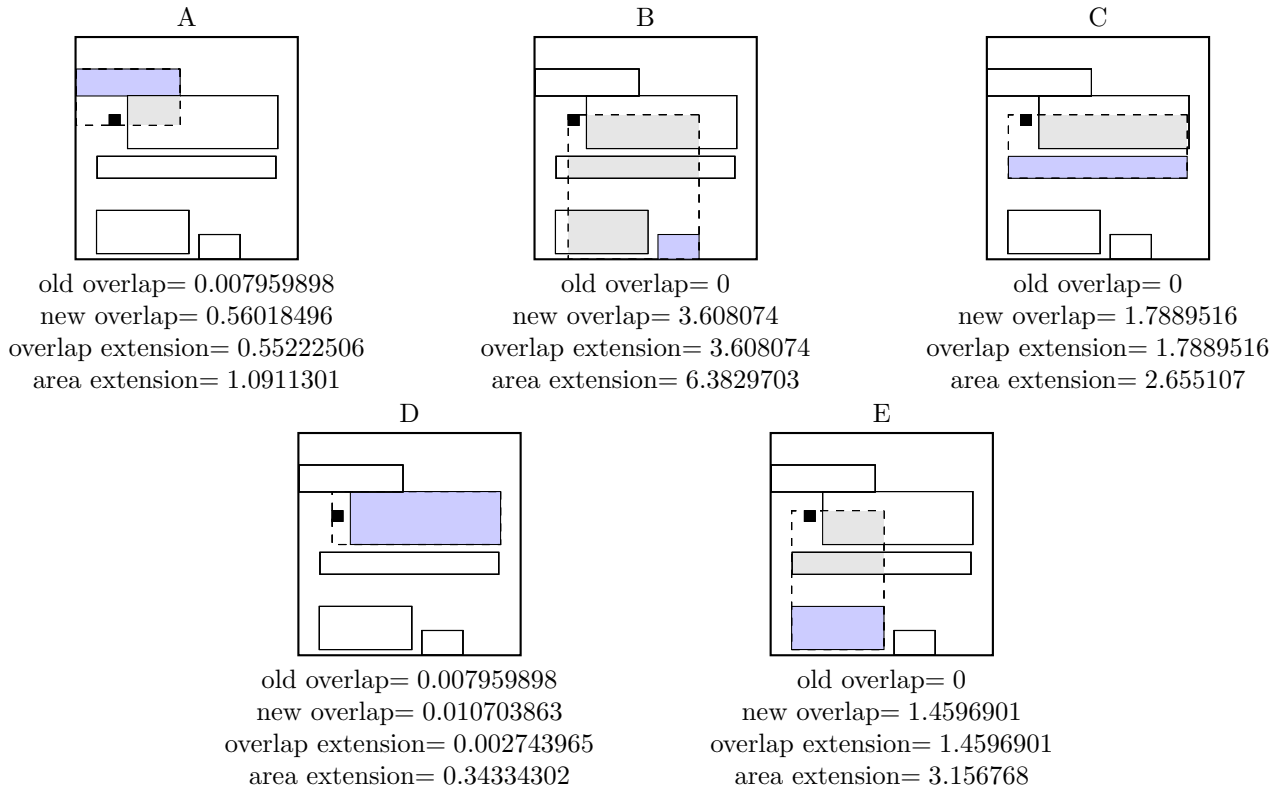
the node *D* is not full, add the record.





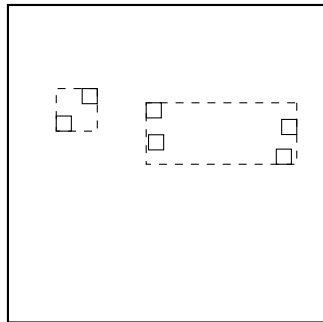
call ADJUST-TREE with  $R$ , node D  
 update MBR of node D.  
 continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT  
 Reinserting 3.  
 call CHOOSE-SUBTREE with 3, node *root*, level 1  
 Next level are leaf nodes



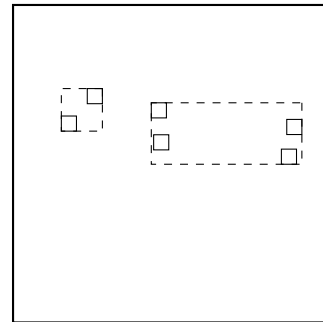
Node  $D$  is chosen  
 call OVERFLOW-TREATMENT  
 Second overflow on the same level during one insert — make a split.  
 call SPLIT-NODE with nodes D and 3  
 call CHOOSE-SPLIT-AXIS with ...  
 Axis:  $x$   
 Entries sorted by low: 3 19 12 8 22 13  
 Entries sorted by high: 3 19 12 8 22 13

partition by low: 3 19 — 12 8 22 13



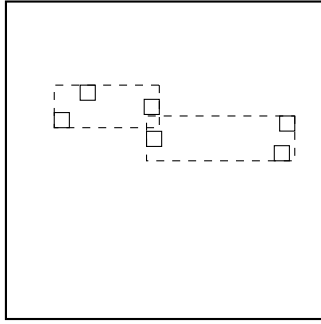
$$o = 3.911$$

partition by high: 3 19 — 12 8 22 13



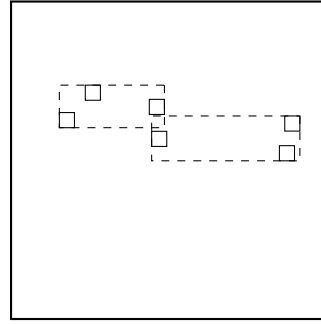
$$o = 3.911$$

partition by low: 3 19 12 — 8 22 13



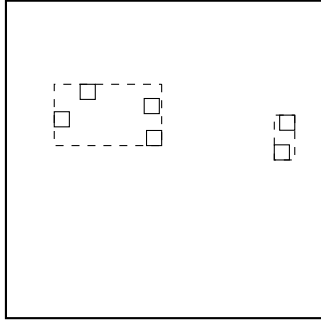
$o = 4.507$

partition by high: 3 19 12 — 8 22 13



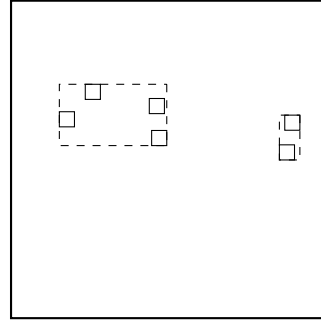
$o = 4.507$

partition by low: 3 19 12 8 — 22 13



$o = 3.0959997$

partition by high: 3 19 12 8 — 22 13



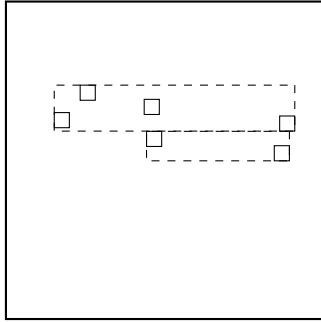
$o = 3.0959997$

Axis:  $y$

Entries sorted by low: 22 8 13 3 12 19

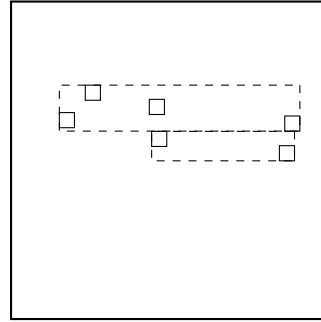
Entries sorted by high: 22 8 13 3 12 19

partition by low: 22 8 — 13 3 12 19



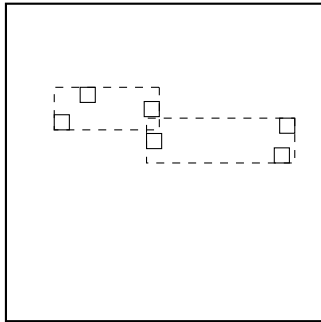
$o = 6.0699997$

partition by high: 22 8 — 13 3 12 19



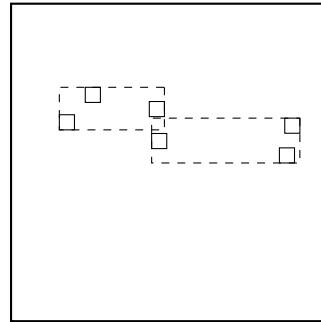
$o = 6.0699997$

partition by low: 22 8 13 — 3 12 19



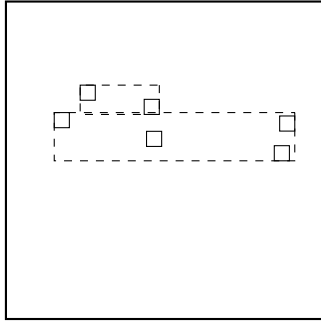
$o = 4.507$

partition by high: 22 8 13 — 3 12 19



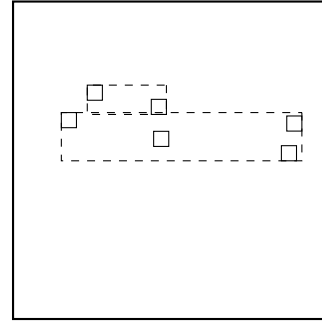
$o = 4.507$

partition by low: 22 8 13 3 — 12 19



$o = 5.2549996$

partition by high: 22 8 13 3 — 12 19

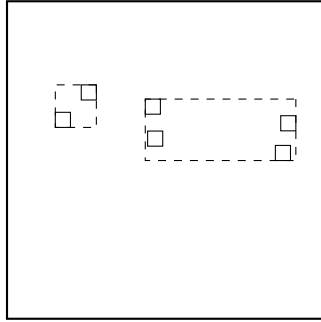


$o = 5.2549996$

Minimal margin ( $o = 3.0959997$ ) was reached for axis  $x$ .  
return from CHOOSE-SPLIT-AXIS

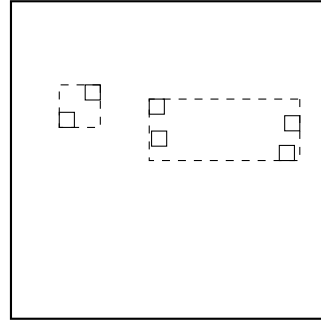
call CHOOSE-SPLIT-INDEX with entries, axis  $x$ ,  $R$   
Entries sorted by low at axis  $x$ : 3 19 12 8 22 13  
Entries sorted by high at axis  $x$ : 3 19 12 8 22 13

Partition by low: 3 19 — 12 8 22 13



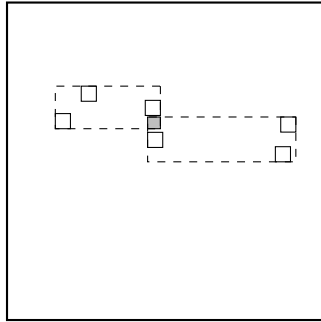
overlap = 0  
 $S = 1.9252051$

Partition by high: 3 19 — 12 8 22 13



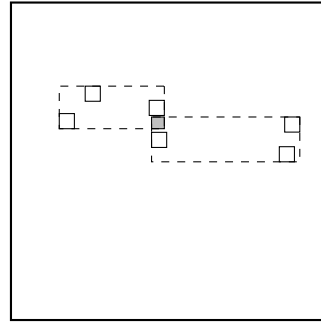
overlap = 0  
 $S = 1.9252051$

Partition by low: 3 19 12 — 8 22 13



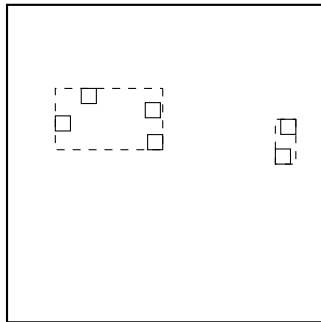
overlap = 0.026194999  
 $S = 1.945443$

Partition by high: 3 19 12 — 8 22 13



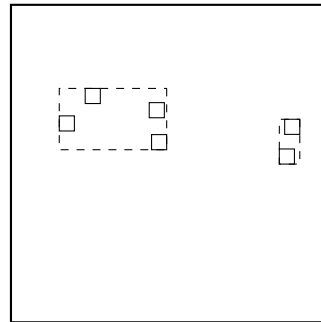
overlap = 0.026194999  
 $S = 1.945443$

Partition by low: 3 19 12 8 — 22 13



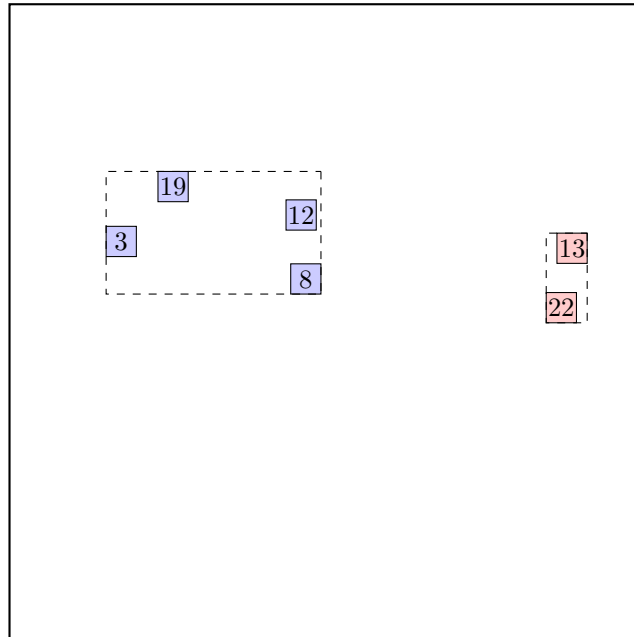
overlap = 0  
 $S = 1.3131337$

Partition by high: 3 19 12 8 — 22 13



overlap = 0  
 $S = 1.3131337$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

call ADJUST-TREE with  $R$ , node D and the new node  
 update MBR of node D.

add the new node to the parent node root

Parent node root is full, promote split (create a new parent)

call SPLIT-NODE with nodes root and NIL

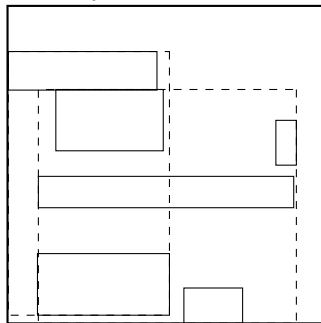
call CHOOSE-SPLIT-AXIS with ...

Axis:  $x$

Entries sorted by low: A E C D B F

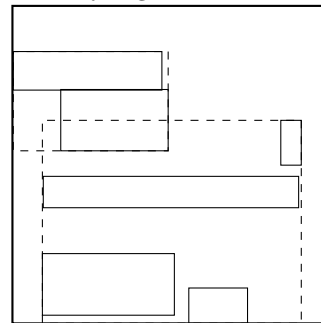
Entries sorted by high: A D E B C F

partition by low: A E — C D B F



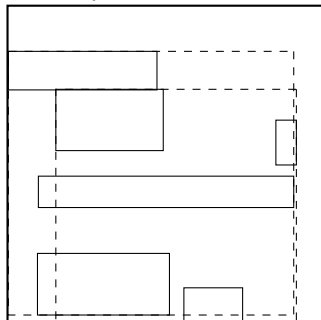
$$o = 12.125$$

partition by high: A D — E B C F



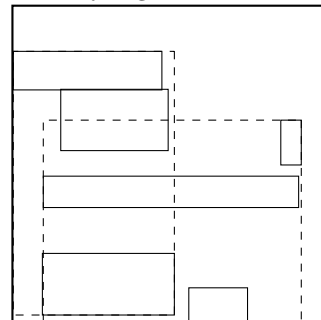
$$o = 9.471999$$

partition by low: A E C — D B F



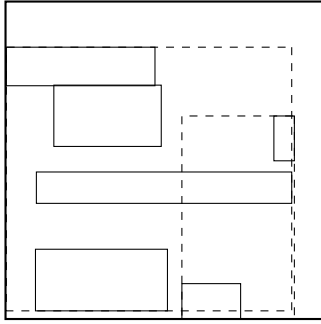
$$o = 13.537999$$

partition by high: A D E — B C F



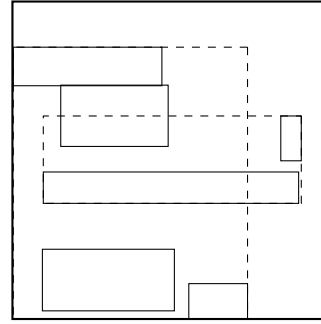
$$o = 11.716999$$

partition by low: A E C D — B F



$o = 11.435999$

partition by high: A D E B — C F



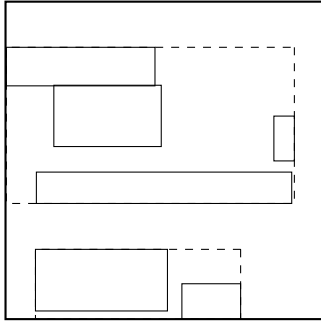
$o = 11.263$

Axis:  $y$

Entries sorted by low: B E C F D A

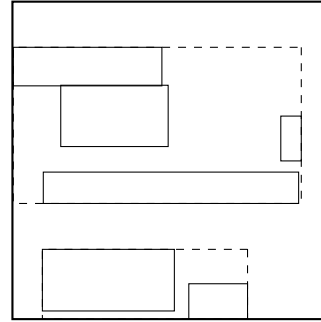
Entries sorted by high: B E C F D A

partition by low: B E — C F D A



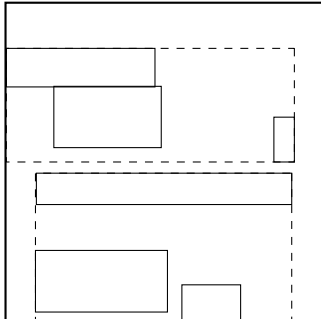
$o = 9.512$

partition by high: B E — C F D A



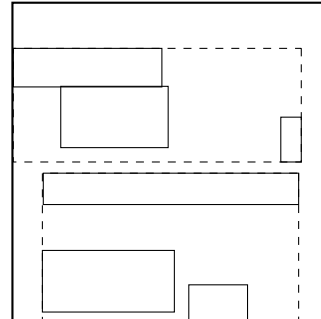
$o = 9.512$

partition by low: B E C — F D A



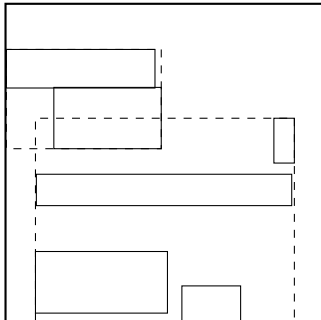
$o = 10.646999$

partition by high: B E C — F D A



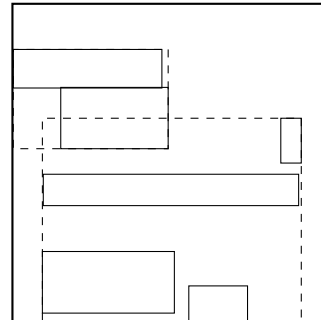
$o = 10.646999$

partition by low: B E C F — D A



$o = 9.471999$

partition by high: B E C F — D A



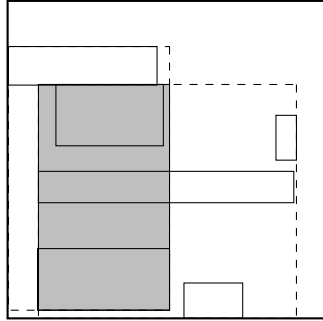
$o = 9.471999$

Minimal margin ( $o = 9.471999$ ) was reached for axis  $x$ .  
return from CHOOSE-SPLIT-AXIS

call CHOOSE-SPLIT-INDEX with entries, axis  $x$ ,  $R$   
Entries sorted by low at axis  $x$ : A E C D B F

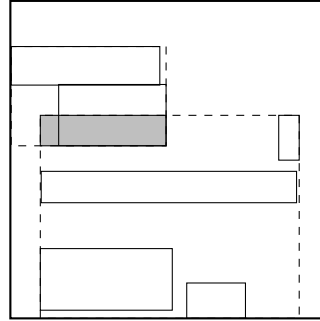
Entries sorted by high at axis  $x$ : A D E B C F

Partition by low: A E — C D B F



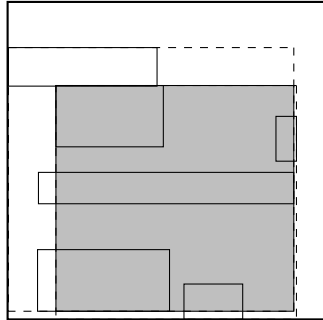
overlap = 5.1777244  
 $S = 17.989336$

Partition by high: A D — E B C F



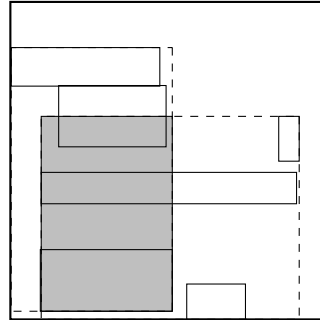
overlap = 0.67059184  
 $S = 11.886461$

Partition by low: A E C — D B F



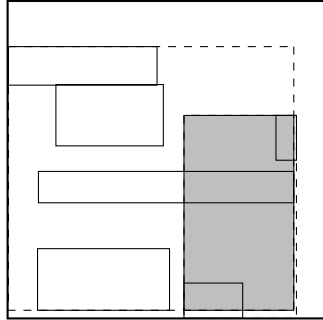
overlap = 9.396942  
 $S = 23.009125$

Partition by high: A D E — B C F



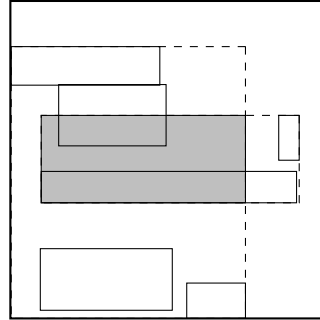
overlap = 4.4702525  
 $S = 16.596833$

Partition by low: A E C D — B F



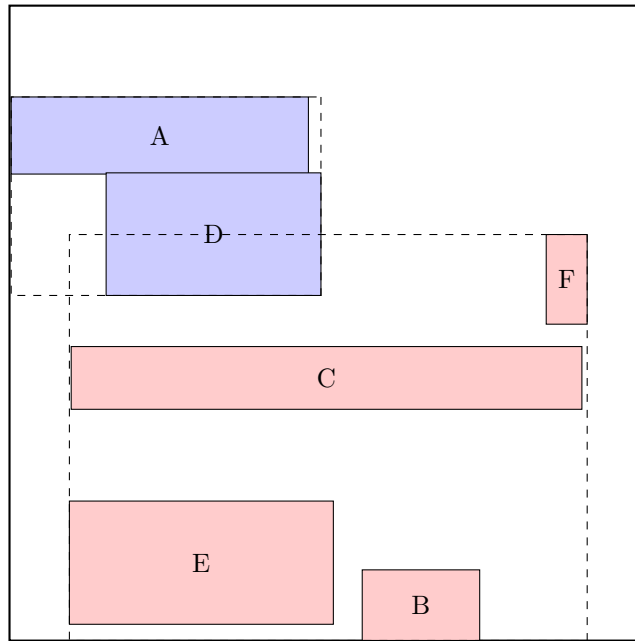
overlap = 3.7458337  
 $S = 17.16248$

Partition by high: A D E B — C F



overlap = 3.123512  
 $S = 15.086332$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.  
 continue by adjusting the parent node NIL, the new parent

call ADJUST-TREE with  $R$ , node B and the new node  
 we are at the root  
 return from ADJUST-TREE  
 create a new root. return from REINSERT

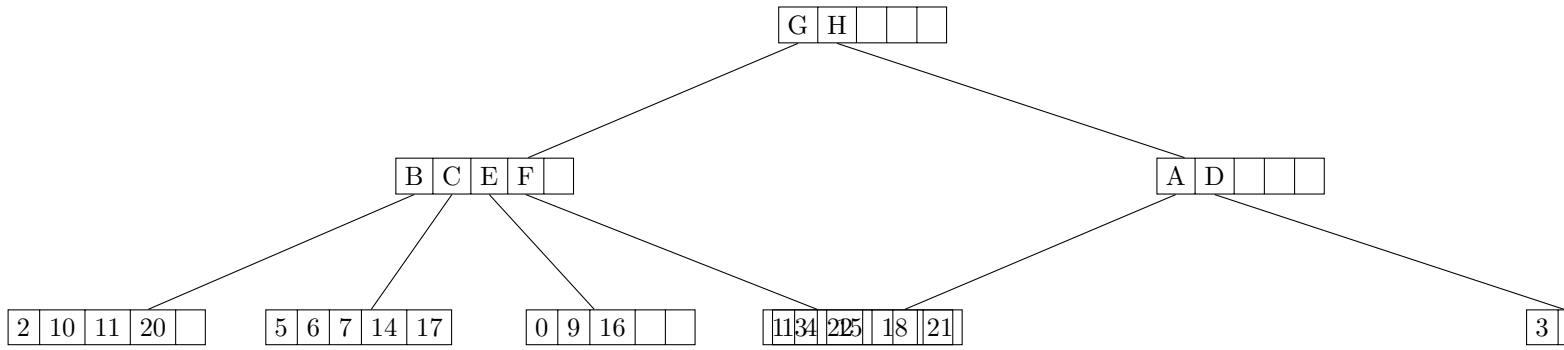
call ADJUST-TREE with  $R$ , node D  
 update MBR of node D.  
 continue by adjusting the parent node B

call ADJUST-TREE with  $R$ , node B  
 update MBR of node B.  
 continue by adjusting the parent node root

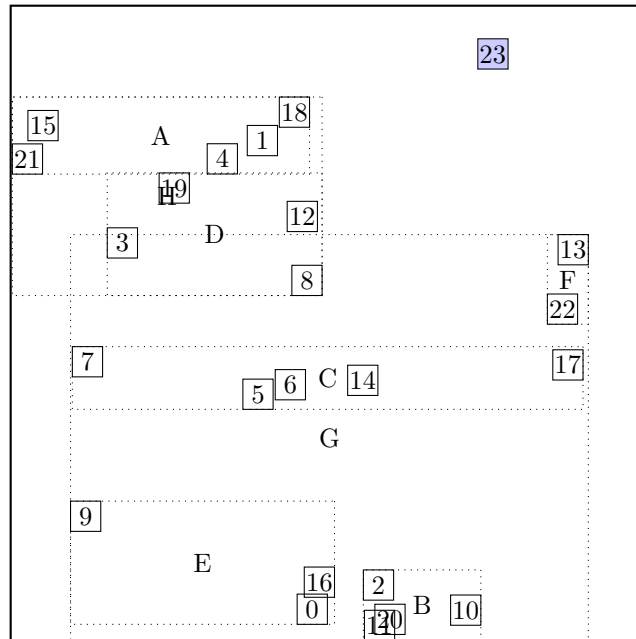
call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE

call INSERT, #S(P :X 309/100 :Y 3781/1000)

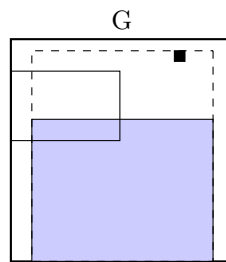
structure view:



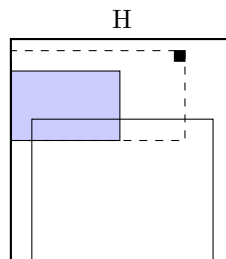
data view:



call CHOOSE-SUBTREE with 23, node root, level 2  
 Next level are not leaf nodes



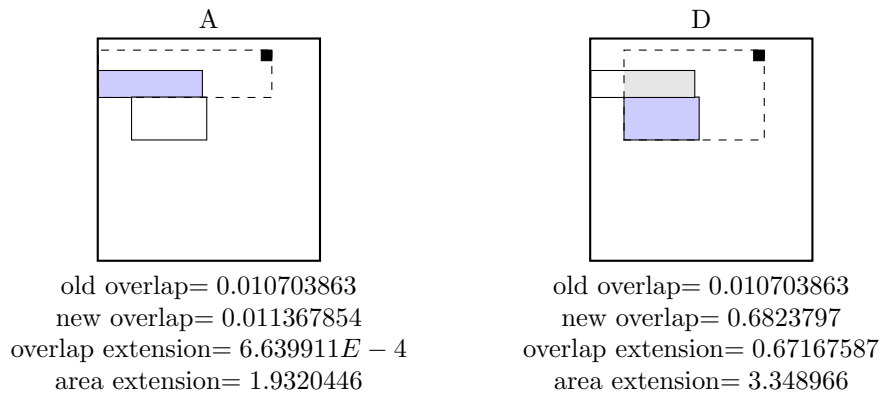
overlap extension= 4.4353743  
 area extension= 4.4353743



overlap extension= 2.8774045  
 area extension= 2.8774045

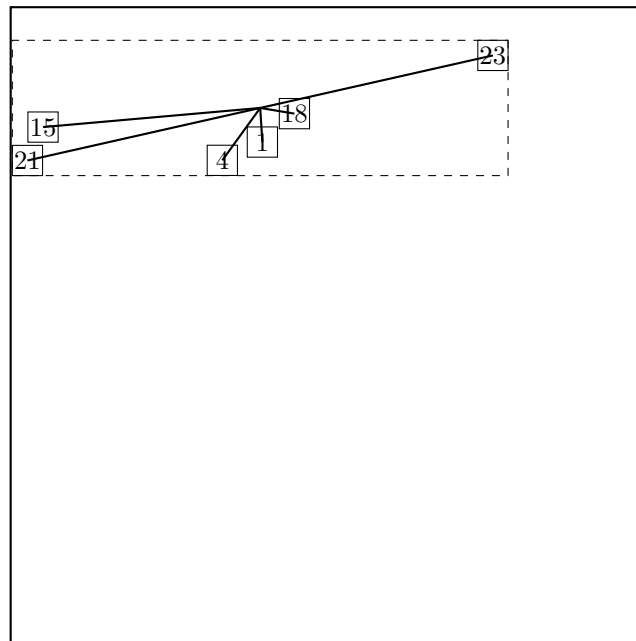


Node  $H$  is chosen  
 call CHOOSE-SUBTREE with 23, node  $H$ , level 1  
 Next level are leaf nodes

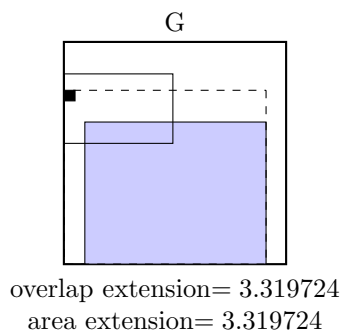


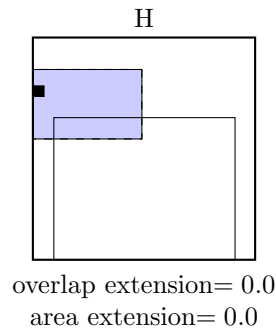
Node  $A$  is chosen  
 call OVERFLOW-TREATMENT

call REINSERT

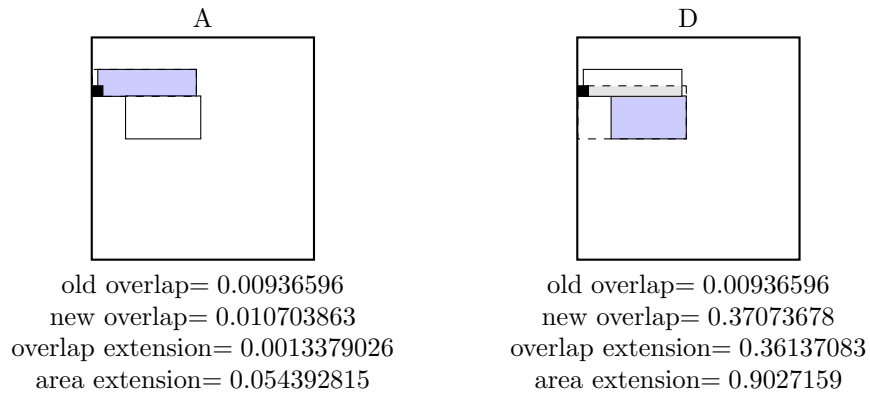


The two most distant nodes (23 21) are removed and reinserted in reversed order.  
 The rest is kept.  
 Adjust MBR of  $A$ .  
 Reinserting 21.  
 call CHOOSE-SUBTREE with 21, node  $root$ , level 2  
 Next level are not leaf nodes

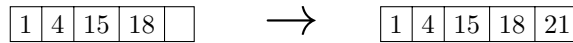




Node *H* is chosen  
 call CHOOSE-SUBTREE with 21, node *H*, level 1  
 Next level are leaf nodes



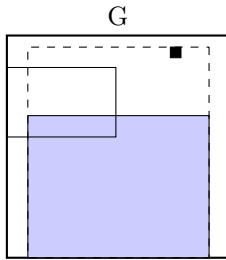
Node *A* is chosen  
 the node *A* is not full, add the record.



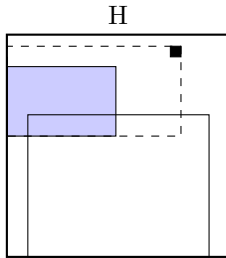
call ADJUST-TREE with *R*, node *A*  
 update MBR of node *A*.  
 continue by adjusting the parent node *H*

call ADJUST-TREE with *R*, node *H*  
 update MBR of node *H*.  
 continue by adjusting the parent node *root*

call ADJUST-TREE with *R*, node *root*  
 we are at the root  
 return from ADJUST-TREE  
 return from REINSERT  
 Reinserting 23.  
 call CHOOSE-SUBTREE with 23, node *root*, level 2  
 Next level are not leaf nodes

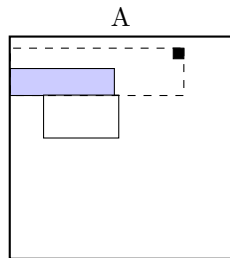


overlap extension= 4.4353743  
 area extension= 4.4353743

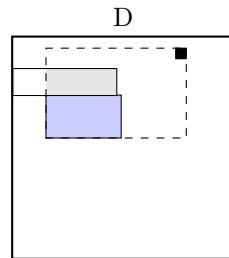


overlap extension= 2.8774045  
 area extension= 2.8774045

Node *H* is chosen  
 call CHOOSE-SUBTREE with 23, node *H*, level 1  
 Next level are leaf nodes



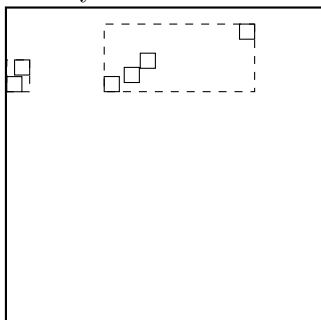
old overlap= 0.010703863  
 new overlap= 0.011367854  
 overlap extension=  $6.639911E - 4$   
 area extension= 1.9320446



old overlap= 0.010703863  
 new overlap= 0.6823797  
 overlap extension= 0.67167587  
 area extension= 3.348966

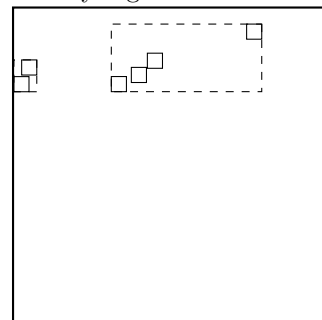
Node *A* is chosen  
 call OVERFLOW-TREATMENT  
 Second overflow on the same level during one insert — make a split.  
 call SPLIT-NODE with nodes *A* and 23  
 call CHOOSE-SPLIT-AXIS with ...  
 Axis: *x*  
 Entries sorted by low: 21 15 4 1 18 23  
 Entries sorted by high: 21 15 4 1 18 23

partition by low: 21 15 — 4 1 18 23



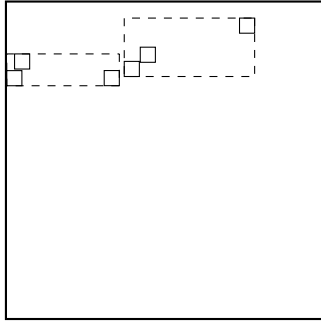
$o = 3.6079996$

partition by high: 21 15 — 4 1 18 23



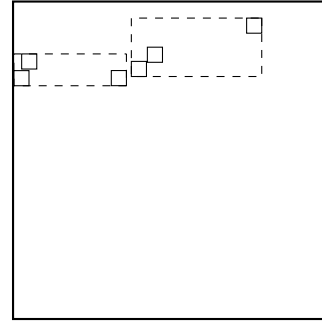
$o = 3.6079996$

partition by low: 21 15 4 — 1 18 23



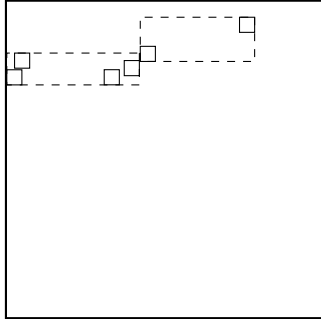
$o = 4.4079995$

partition by high: 21 15 4 — 1 18 23



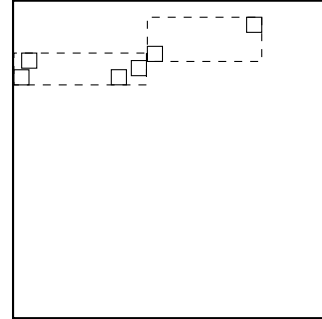
$o = 4.4079995$

partition by low: 21 15 4 1 — 18 23



$o = 4.2729993$

partition by high: 21 15 4 1 — 18 23



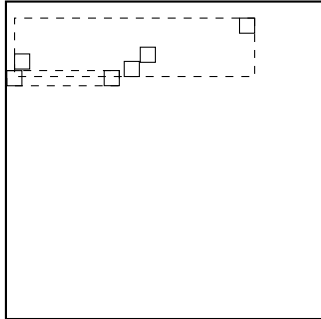
$o = 4.2729993$

Axis:  $y$

Entries sorted by low: 21 4 1 15 18 23

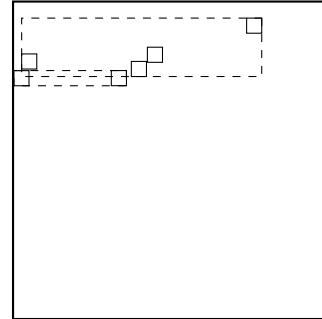
Entries sorted by high: 21 4 1 15 18 23

partition by low: 21 4 — 1 15 18 23



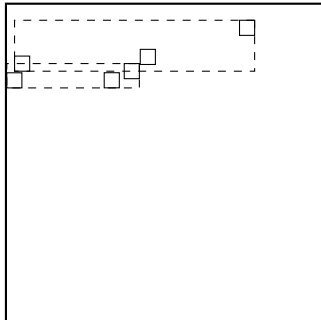
$o = 5.6389995$

partition by high: 21 4 — 1 15 18 23



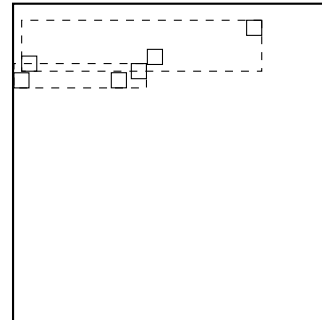
$o = 5.6389995$

partition by low: 21 4 1 — 15 18 23



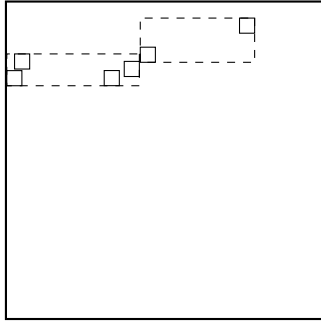
$o = 5.9259997$

partition by high: 21 4 1 — 15 18 23



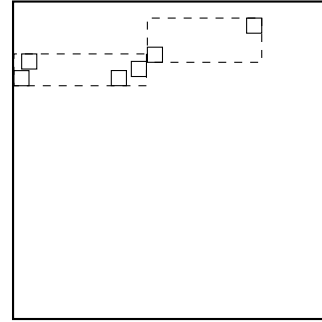
$o = 5.9259997$

partition by low: 21 4 1 15 — 18 23



$o = 4.2729993$

partition by high: 21 4 1 15 — 18 23



$o = 4.2729993$

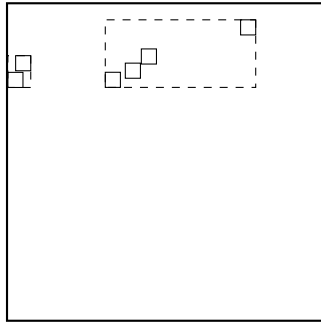
Minimal margin ( $o = 3.6079996$ ) was reached for axis  $x$ .  
return from CHOOSE-SPLIT-AXIS

call CHOOSE-SPLIT-INDEX with entries, axis  $x$ ,  $R$

Entries sorted by low at axis  $x$ : 21 15 4 1 18 23

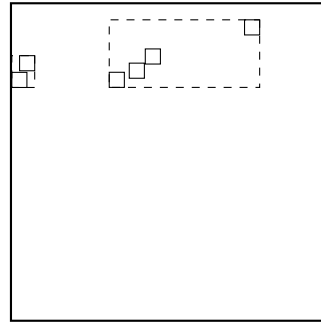
Entries sorted by high at axis  $x$ : 21 15 4 1 18 23

Partition by low: 21 15 — 4 1 18 23



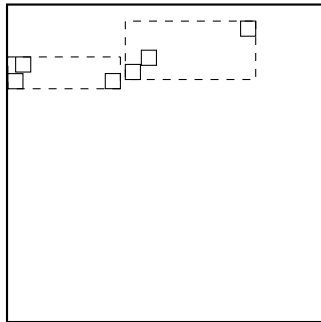
overlap = 0  
 $S = 1.9066224$

Partition by high: 21 15 — 4 1 18 23



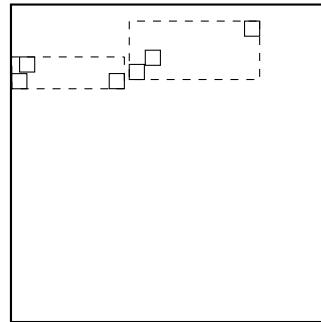
overlap = 0  
 $S = 1.9066224$

Partition by low: 21 15 4 — 1 18 23



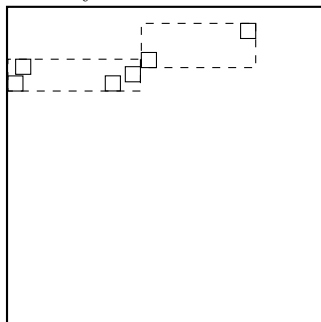
overlap = 0  
 $S = 1.960293$

Partition by high: 21 15 4 — 1 18 23



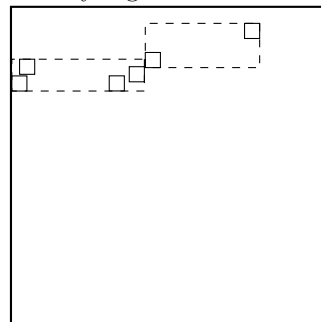
overlap = 0  
 $S = 1.960293$

Partition by low: 21 15 4 1 — 18 23



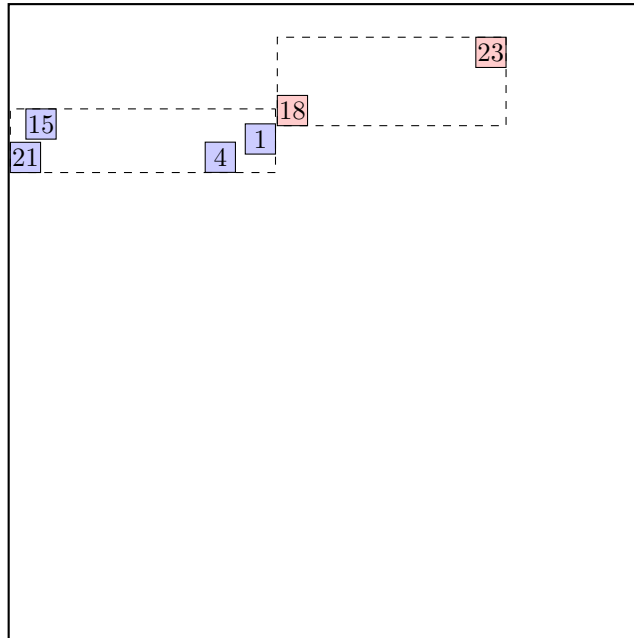
overlap = 0  
 $S = 1.6235383$

Partition by high: 21 15 4 1 — 18 23



overlap = 0  
 $S = 1.6235383$

... and the winner is:



return from CHOOSE-SPLIT-INDEX.

call ADJUST-TREE with  $R$ , node A and the new node  
update MBR of node A.  
add the new node to the parent node H

call ADJUST-TREE with  $R$ , node H  
update MBR of node H.  
continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE  
return from REINSERT

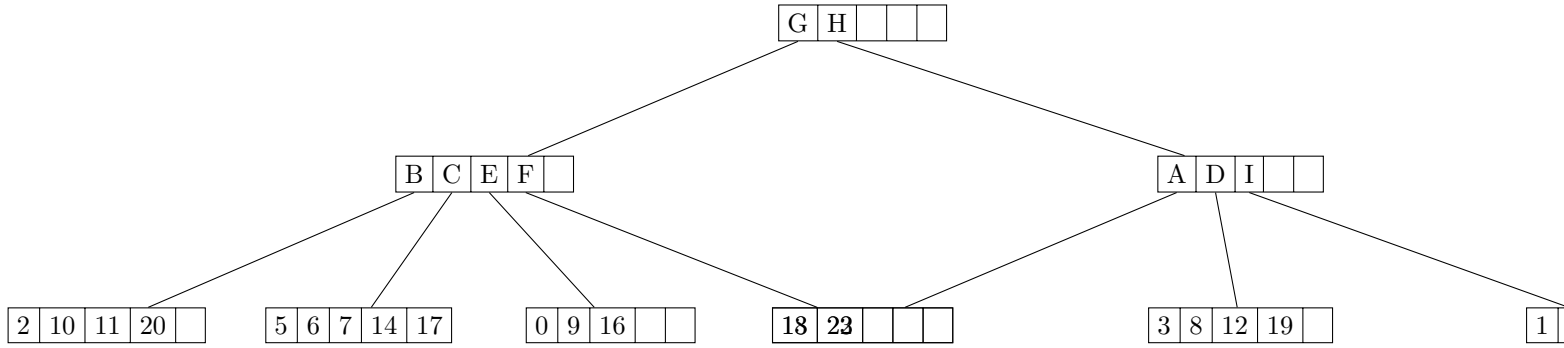
call ADJUST-TREE with  $R$ , node I  
update MBR of node I.  
continue by adjusting the parent node H

call ADJUST-TREE with  $R$ , node H  
update MBR of node H.  
continue by adjusting the parent node root

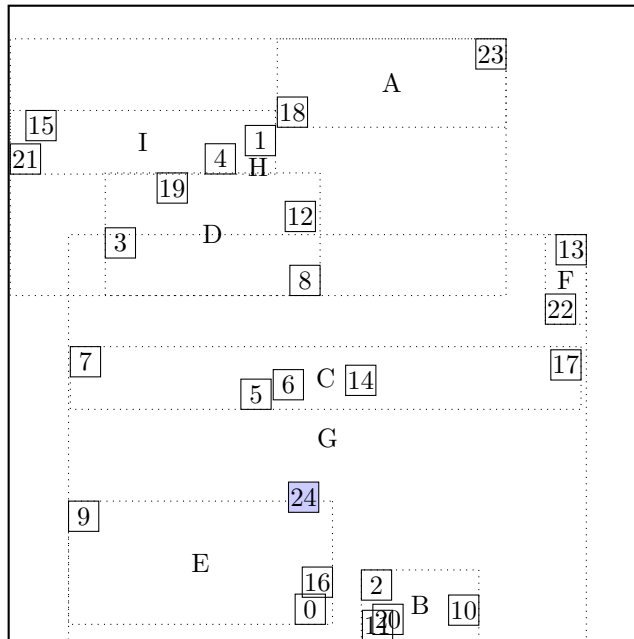
call ADJUST-TREE with  $R$ , node root  
we are at the root  
return from ADJUST-TREE

call INSERT, #S(P :X 1851/1000 :Y 849/1000)

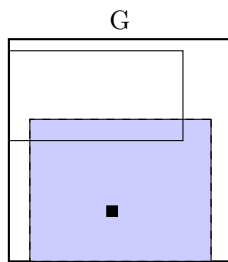
structure view:



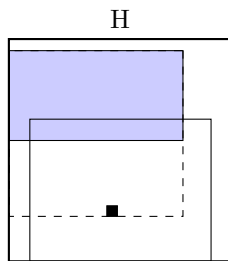
data view:



call CHOOSE-SUBTREE with 24, node *root*, level 2  
 Next level are not leaf nodes

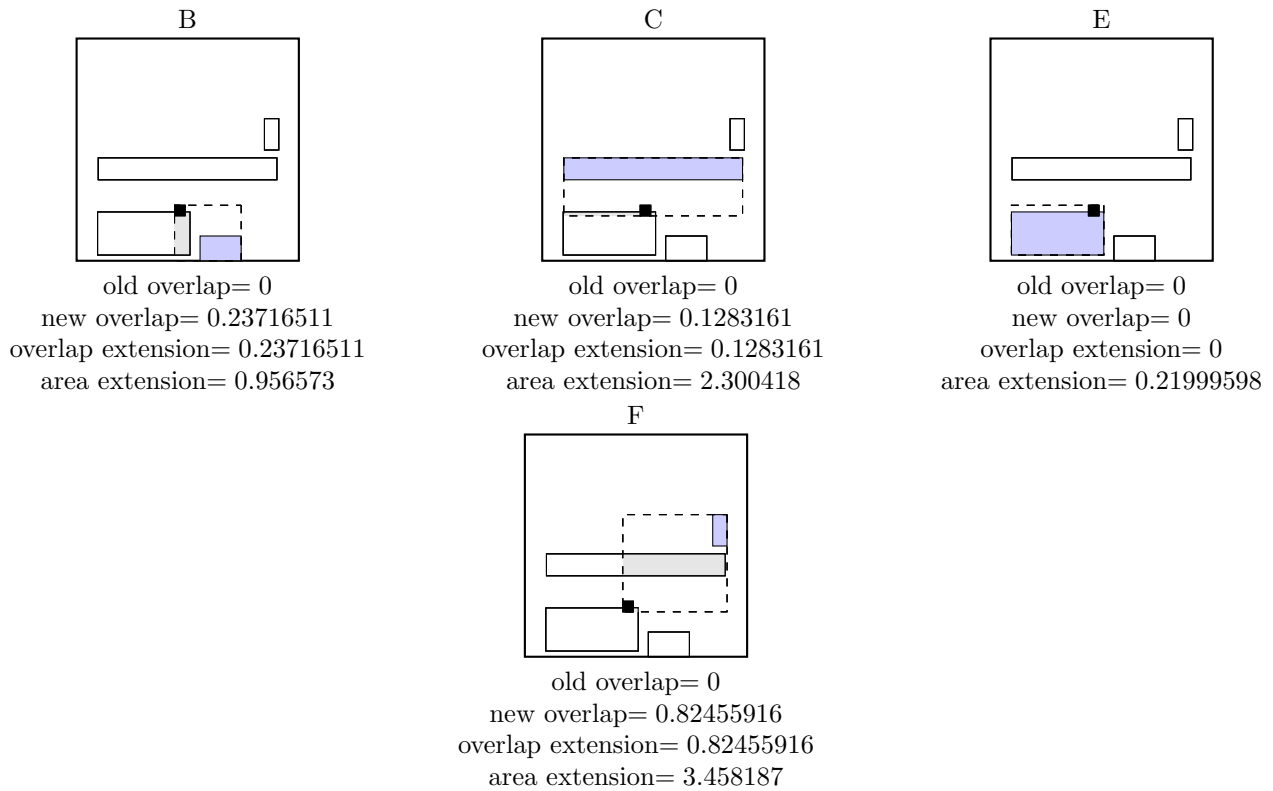


overlap extension= 0.0  
 area extension= 0.0

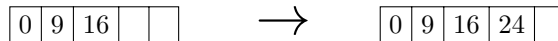


overlap extension= 4.702086  
 area extension= 4.702086

Node  $G$  is chosen  
 call CHOOSE-SUBTREE with 24, node  $G$ , level 1  
 Next level are leaf nodes



Node  $E$  is chosen  
 the node  $E$  is not full, add the record.



call ADJUST-TREE with  $R$ , node  $E$   
 update MBR of node  $E$ .  
 continue by adjusting the parent node  $G$

call ADJUST-TREE with  $R$ , node  $G$   
 update MBR of node  $G$ .  
 continue by adjusting the parent node root

call ADJUST-TREE with  $R$ , node root  
 we are at the root  
 return from ADJUST-TREE



