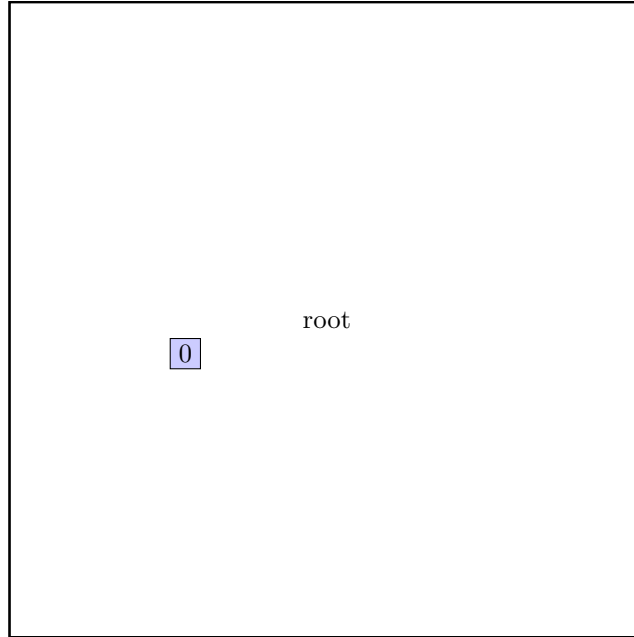


call INSERT with rectangle #S(P :X 1063/1000 :Y 71/40), R+.

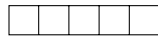
structure view:



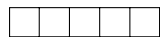
data view:



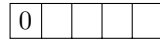
a leaf is found: root



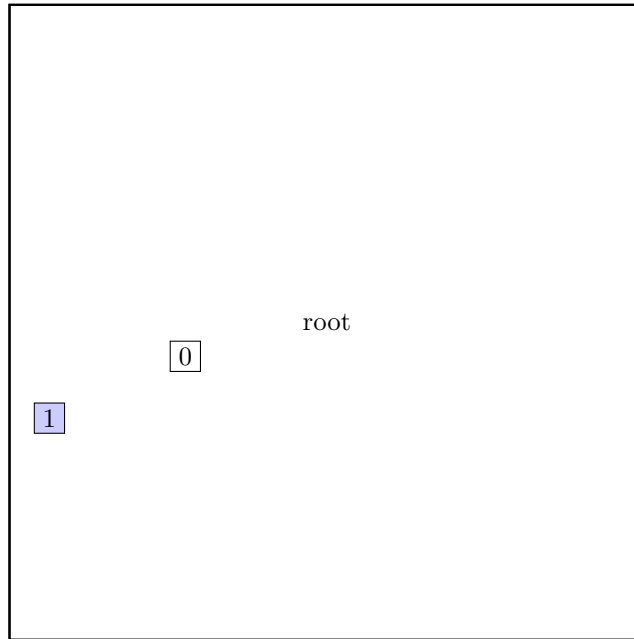
the node root is not full, add the record.



call INSERT with rectangle #S(P :X 41/250 :Y 683/500), R_+ .
structure view:



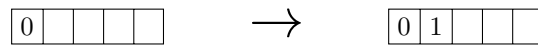
data view:



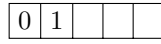
a leaf is found: root



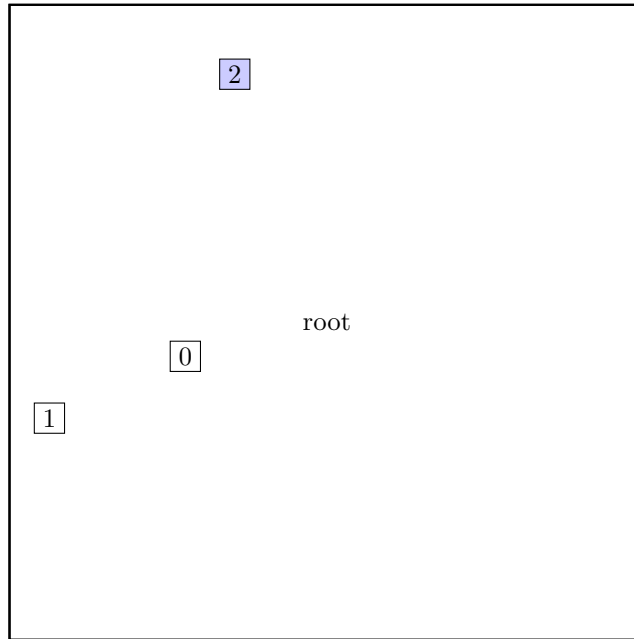
the node root is not full, add the record.



call INSERT with rectangle #S(P :X 139/100 :Y 3641/1000), R+.
structure view:



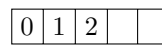
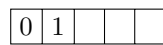
data view:



a leaf is found: root

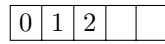


the node root is not full, add the record.

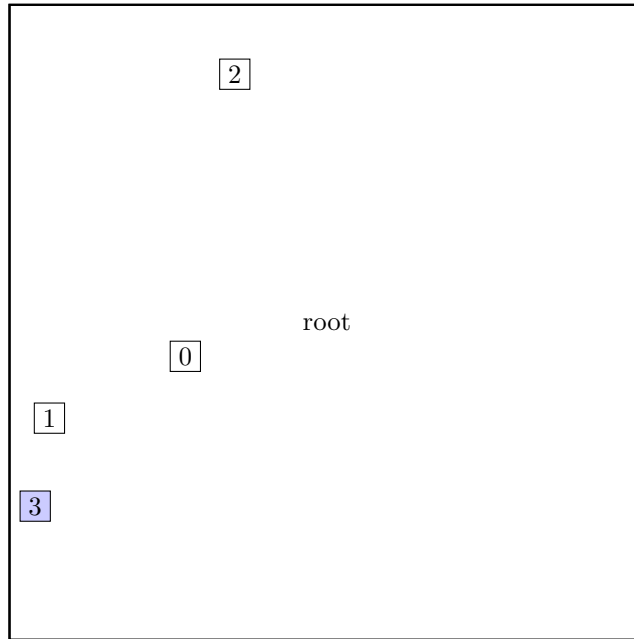


call INSERT with rectangle #S(P :X 7/100 :Y 98/125), R+.

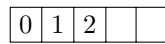
structure view:



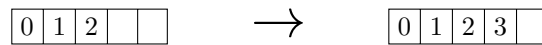
data view:



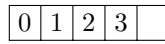
a leaf is found: root



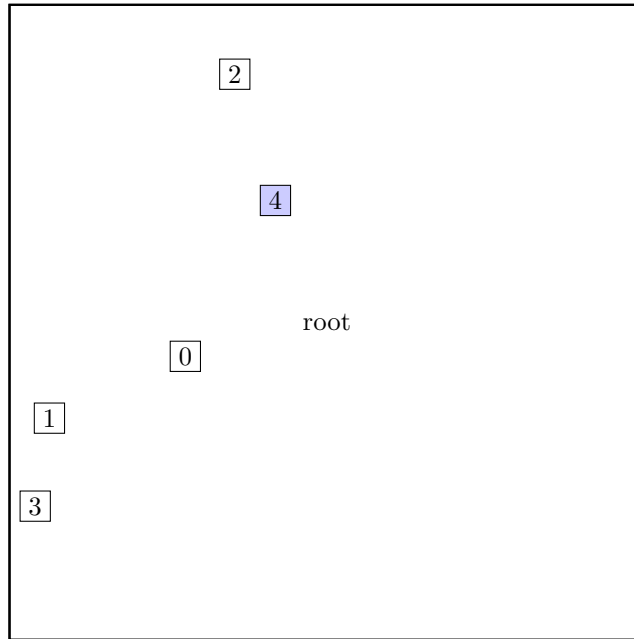
the node root is not full, add the record.



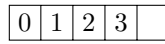
call INSERT with rectangle #S(P :X 1659/1000 :Y 1403/500), R+.
structure view:



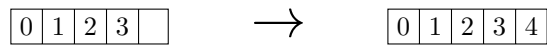
data view:



a leaf is found: root

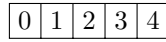


the node root is not full, add the record.

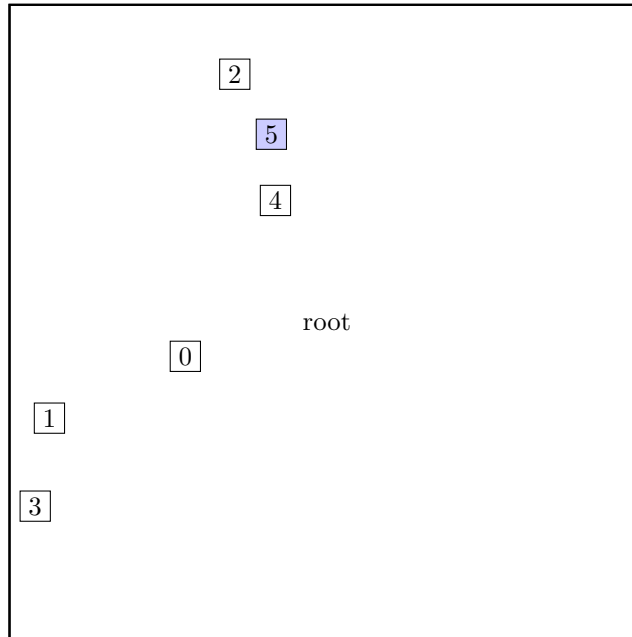


call INSERT with rectangle #S(P :X 204/125 :Y 811/250), R_+ .

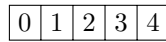
structure view:



data view:



a leaf is found: root



the leaf root is full, need to split.

call SPLIT-NODE with R_+ node newnode

generate partition rectangles.

sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.

(3 1 0) — (2 5 4)

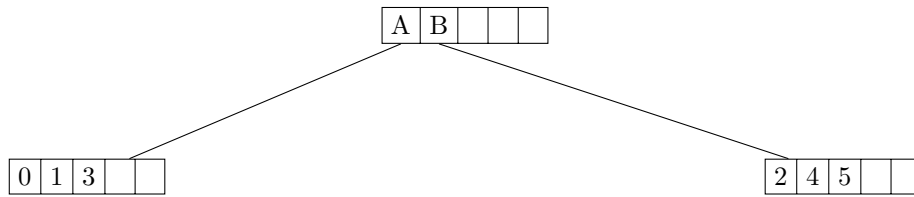
sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

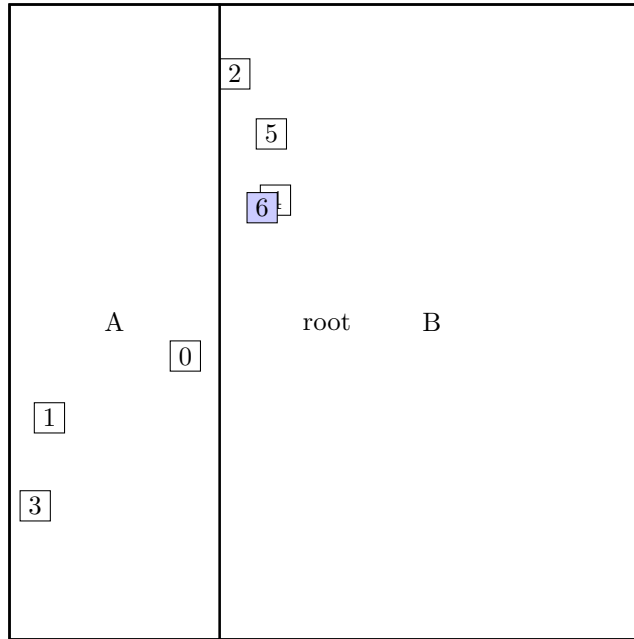
(3 1 0) — (4 5 2)

Split the entries into the two areas.

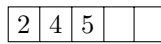
call INSERT with rectangle #S(P :X 1571/1000 :Y 689/250), R+.
 structure view:



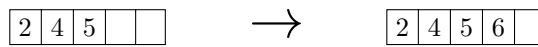
data view:



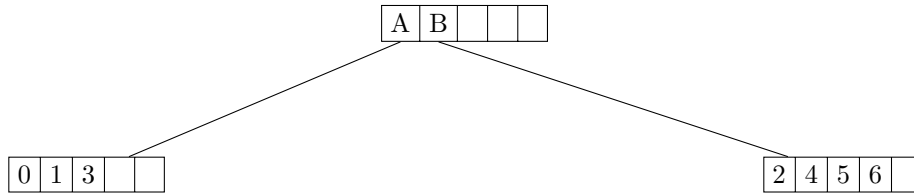
it is an interval node, we descent to all childs which intersect the rectangle 6:
 a leaf is found: B



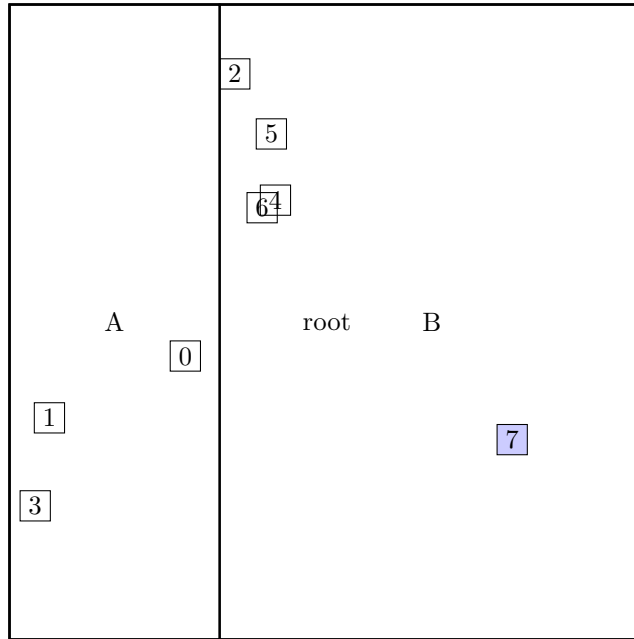
the node B is not full, add the record.



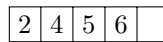
call INSERT with rectangle #S(P :X 129/40 :Y 611/500), R_+ .
 structure view:



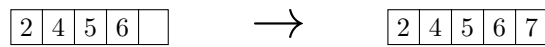
data view:



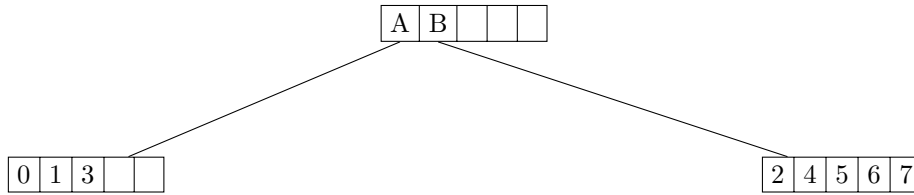
it is an interval node, we descent to all childs which intersect the rectangle 7:
 a leaf is found: B



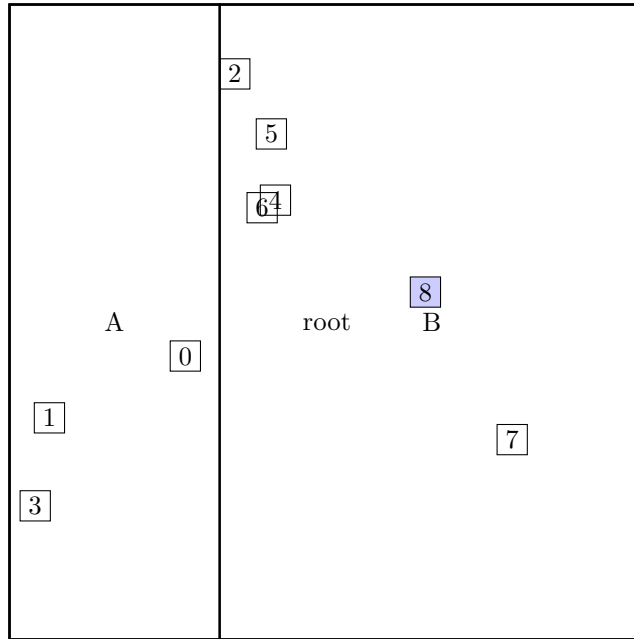
the node B is not full, add the record.



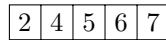
call INSERT with rectangle #S(P :X 2651/1000 :Y 1099/500), R+.
 structure view:



data view:

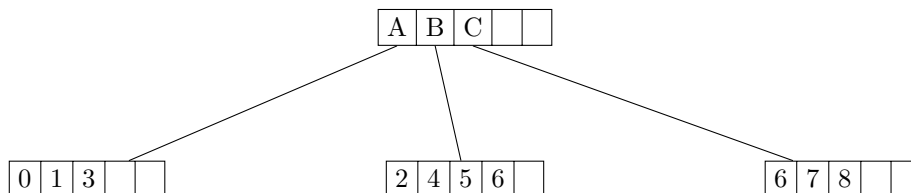


it is an interval node, we descent to all childs which intersect the rectangle 8:
 a leaf is found: B

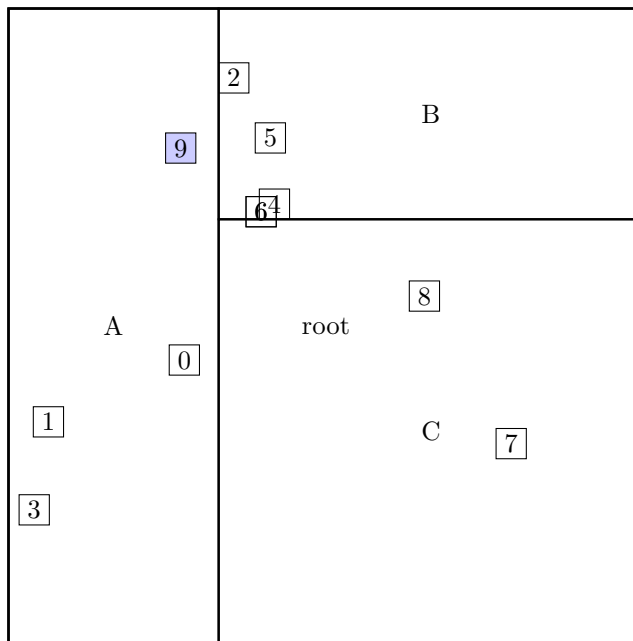


the leaf B is full, need to split.
 call SPLIT-NODE with R+ node newnode
 generate partition rectangles.
 sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.
 (2 6 5) — (4 8 7)
 sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.
 sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.
 (7 8 6) — (4 5 2)
 Split the entries into the two areas.

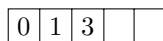
call INSERT with rectangle #S(P :X 1039/1000 :Y 3177/1000), R+.
 structure view:



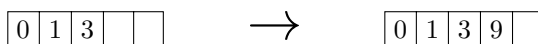
data view:



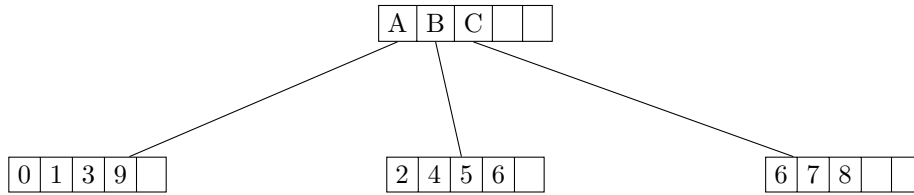
it is an interval node, we descent to all childs which intersect the rectangle 9:
 a leaf is found: A



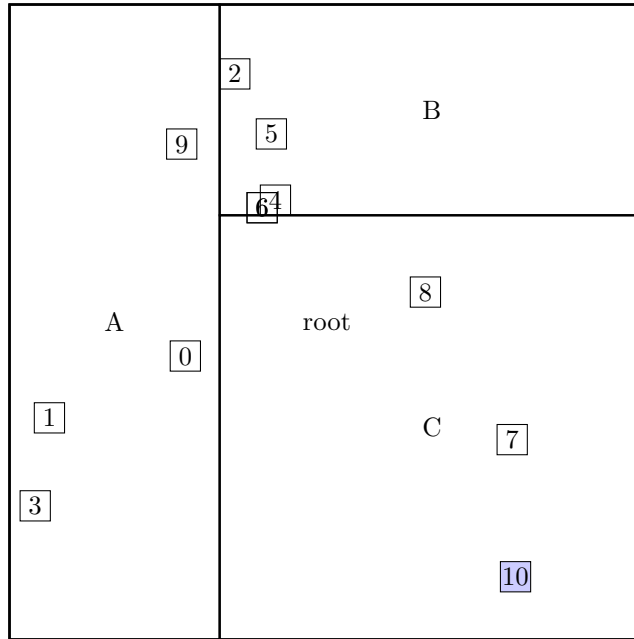
the node A is not full, add the record.



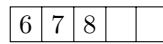
call INSERT with rectangle #S(P :X 3247/1000 :Y 63/200), R+.
 structure view:



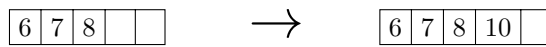
data view:



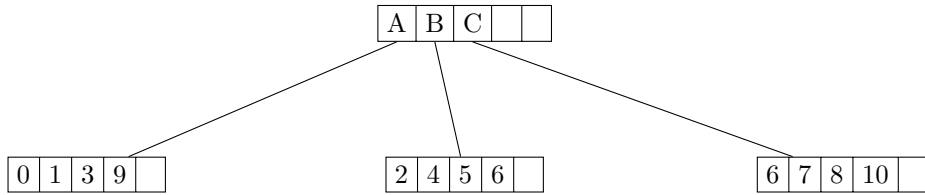
it is an interval node, we descent to all childs which intersect the rectangle 10:
 a leaf is found: C



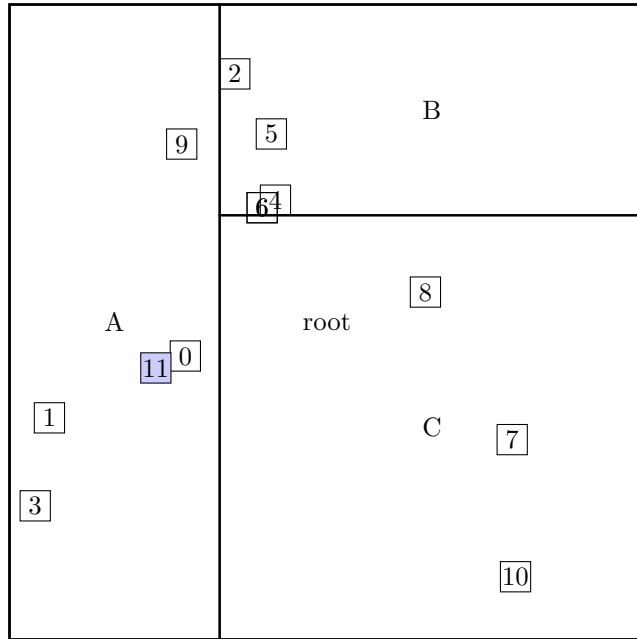
the node C is not full, add the record.



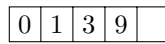
call INSERT with rectangle #S(P :X 217/250 :Y 212/125), R+.
 structure view:



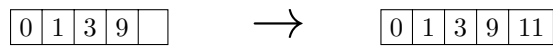
data view:



it is an interval node, we descent to all childs which intersect the rectangle 11:
 a leaf is found: A

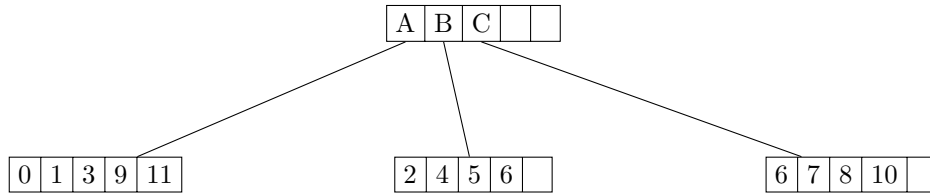


the node A is not full, add the record.

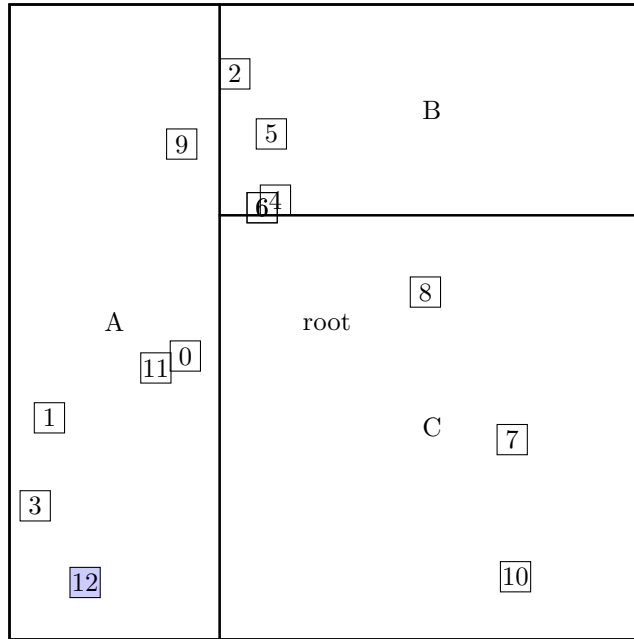


call INSERT with rectangle #S(P :X 2/5 :Y 277/1000), $R+$.

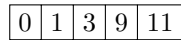
structure view:



data view:



it is an interval node, we descent to all childs which intersect the rectangle 12:
a leaf is found: A



the leaf A is full, need to split.

call SPLIT-NODE with $R+$ node newnode

generate partition rectangles.

sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.

(3 1 12) — (11 9 0)

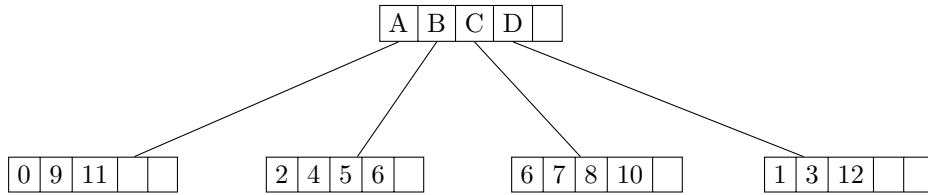
sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

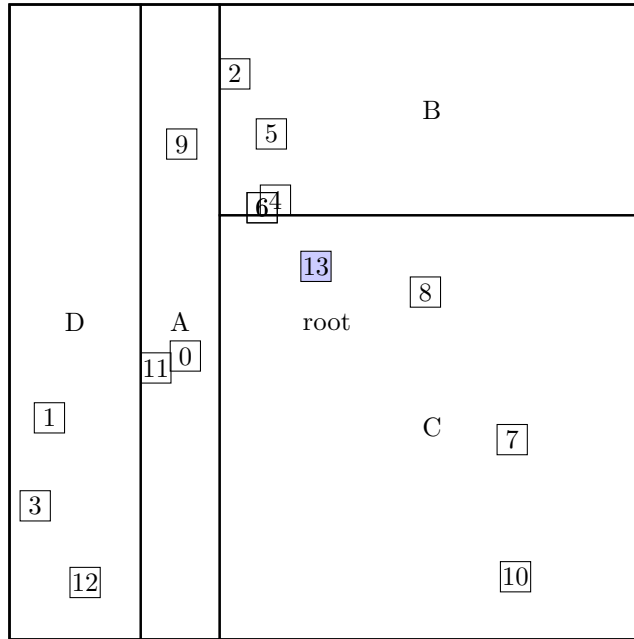
(12 3 1) — (11 0 9)

Split the entries into the two areas.

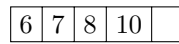
call INSERT with rectangle #S(P :X 1927/1000 :Y 296/125), R+.
 structure view:



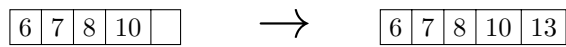
data view:



it is an interval node, we descent to all childs which intersect the rectangle 13:
 a leaf is found: C

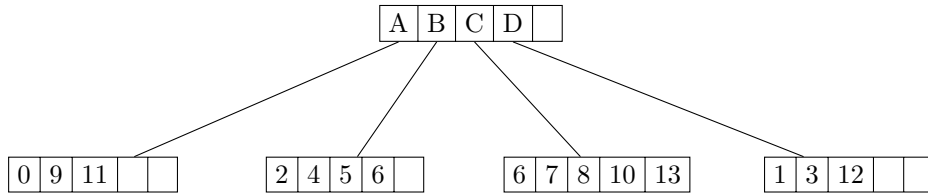


the node C is not full, add the record.

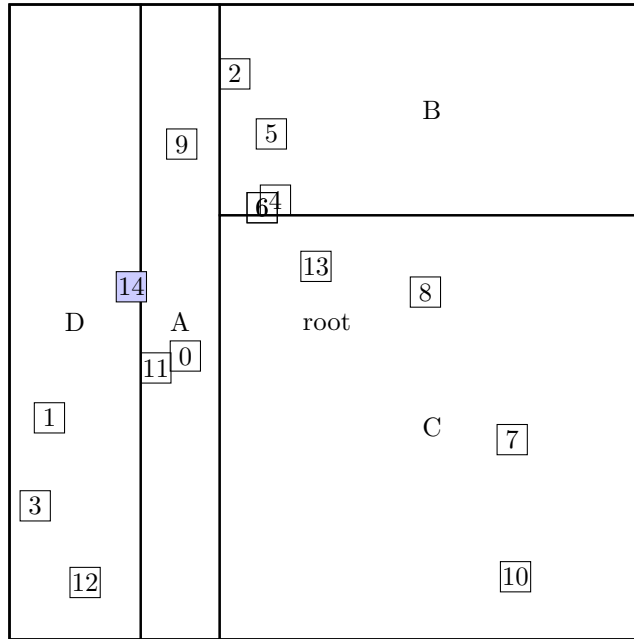


call INSERT with rectangle #S(P :X 353/500 :Y 2233/1000), R+.

structure view:

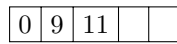


data view:

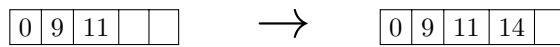


it is an interval node, we descent to all childs which intersect the rectangle 14:

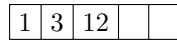
a leaf is found: A



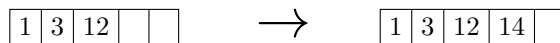
the node A is not full, add the record.



a leaf is found: D

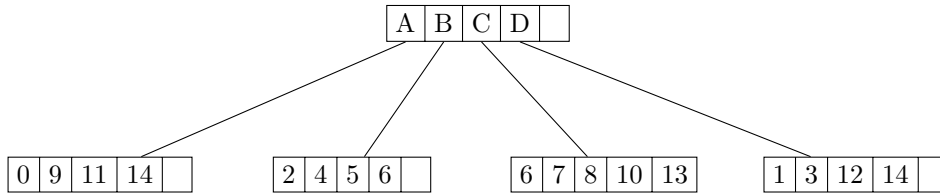


the node D is not full, add the record.

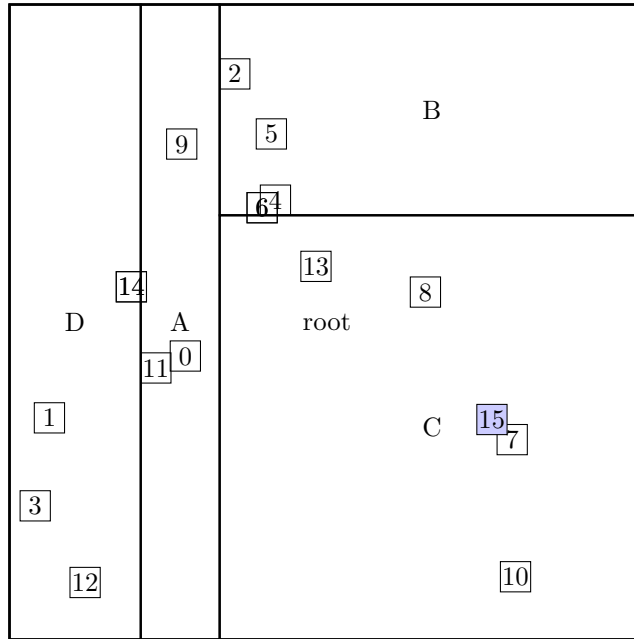


call INSERT with rectangle #S(P :X 3091/1000 :Y 271/200), R+.

structure view:

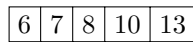


data view:



it is an interval node, we descent to all childs which intersect the rectangle 15:

a leaf is found: C



the leaf C is full, need to split.

call SPLIT-NODE with R+ node newnode

generate partition rectangles.

sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.

(6 13 8) — (15 7 10)

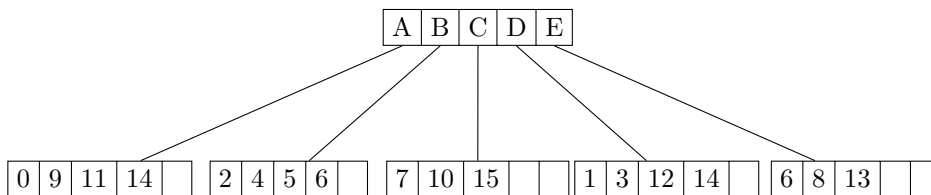
sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

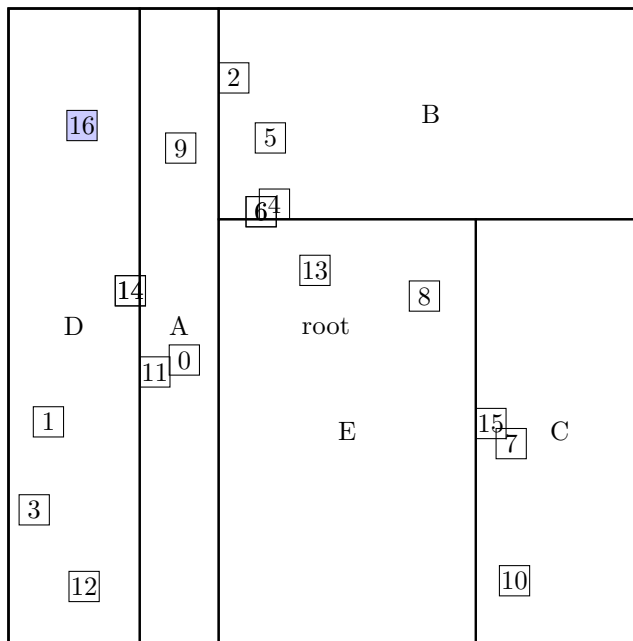
(10 7 15) — (8 13 6)

Split the entries into the two areas.

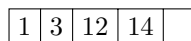
call INSERT with rectangle #S(P :X 193/500 :Y 1663/500), R+.
 structure view:



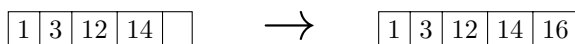
data view:



it is an interval node, we descent to all childs which intersect the rectangle 16:
 a leaf is found: D

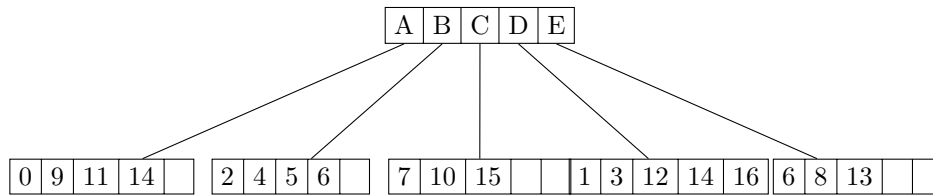


the node D is not full, add the record.

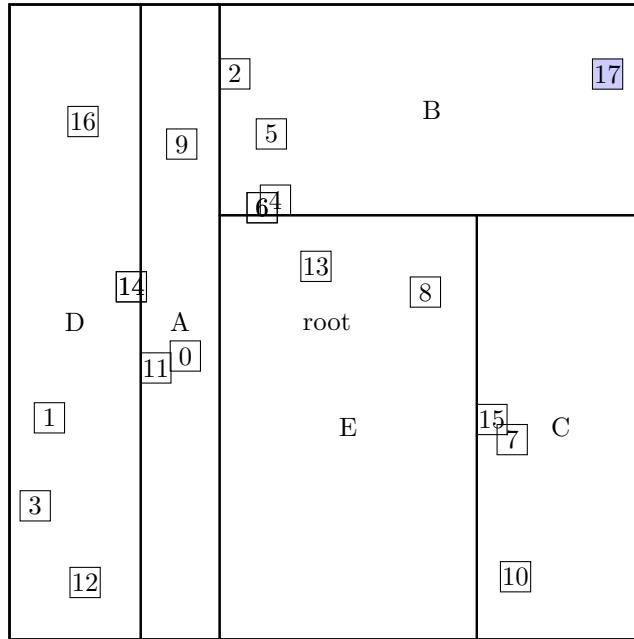


call INSERT with rectangle #S(P :X 482/125 :Y 91/25), R+.

structure view:

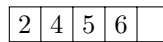


data view:

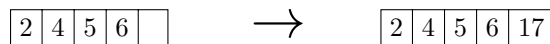


it is an interval node, we descent to all childs which intersect the rectangle 17:

a leaf is found: B

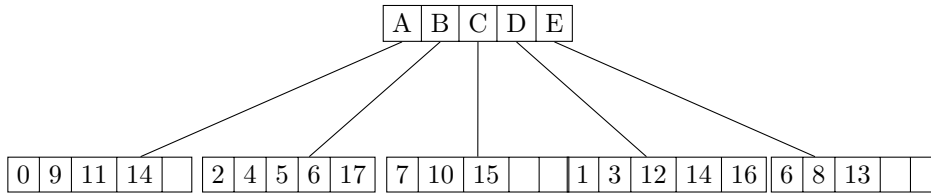


the node B is not full, add the record.

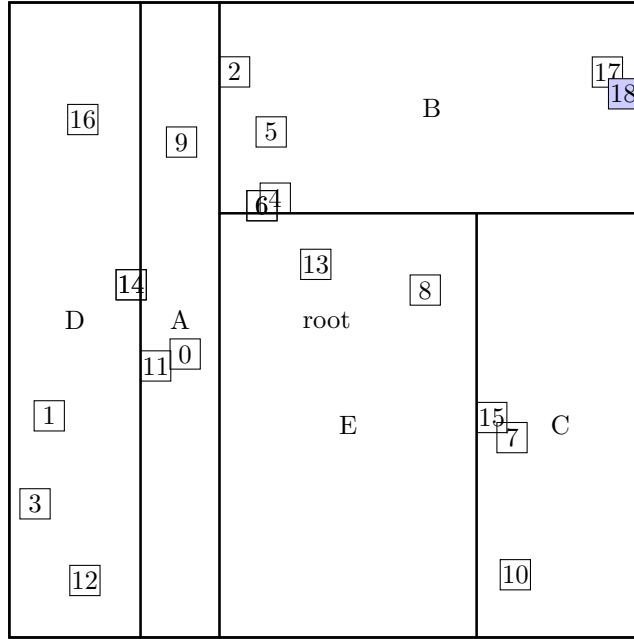


call INSERT with rectangle #S(P :X 3963/1000 :Y 437/125), R+.

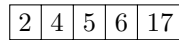
structure view:



data view:



it is an interval node, we descent to all childs which intersect the rectangle 18:
a leaf is found: B



the leaf B is full, need to split.

call SPLIT-NODE with R+ node newnode

generate partition rectangles.

sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.

(2 6 5) — (4 17 18)

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

(6 4 5) — (18 17 2)

Split the entries into the two areas.

call SPLIT-NODE with R+ node newnode

generate partition rectangles.

sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.

(D A F) — (B E C)

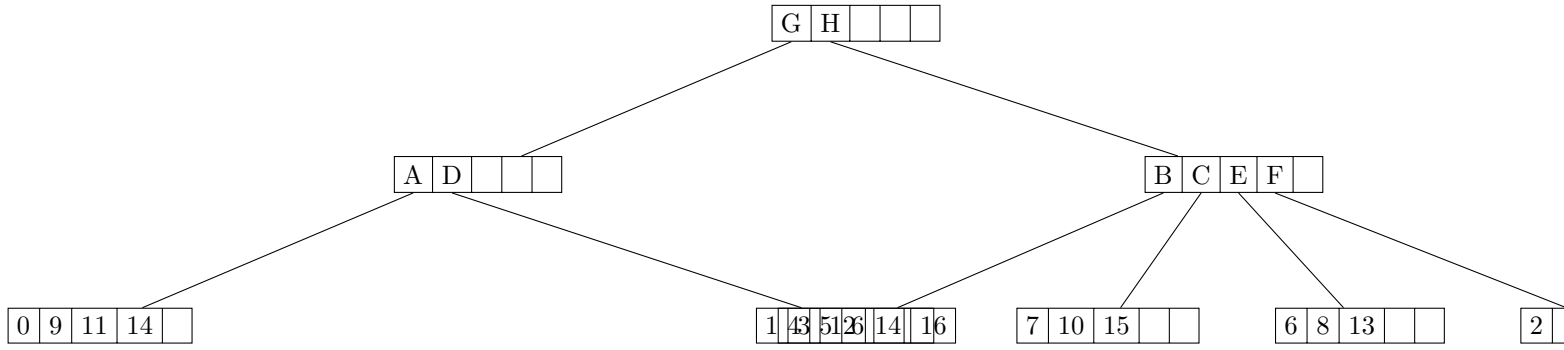
sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.

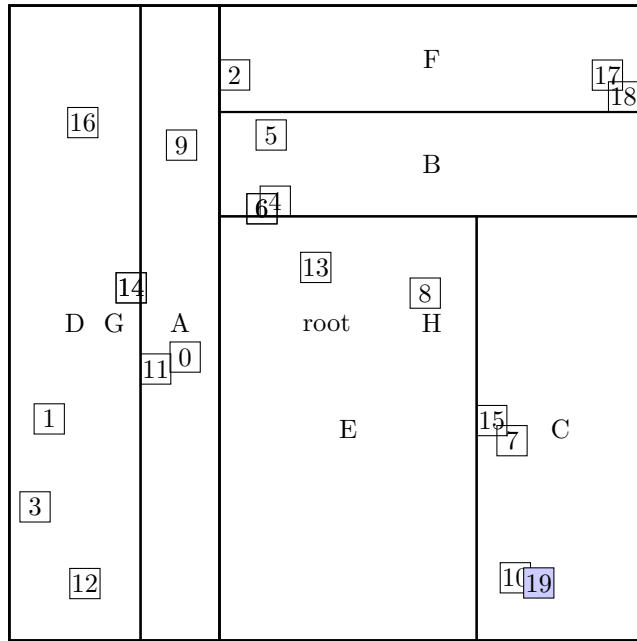
(A C D) — (E B F)

Split the entries into the two areas.

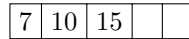
call INSERT with rectangle #S(P :X 1701/500 :Y 281/1000), R+.
 structure view:



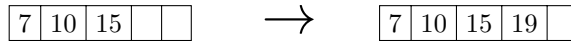
data view:



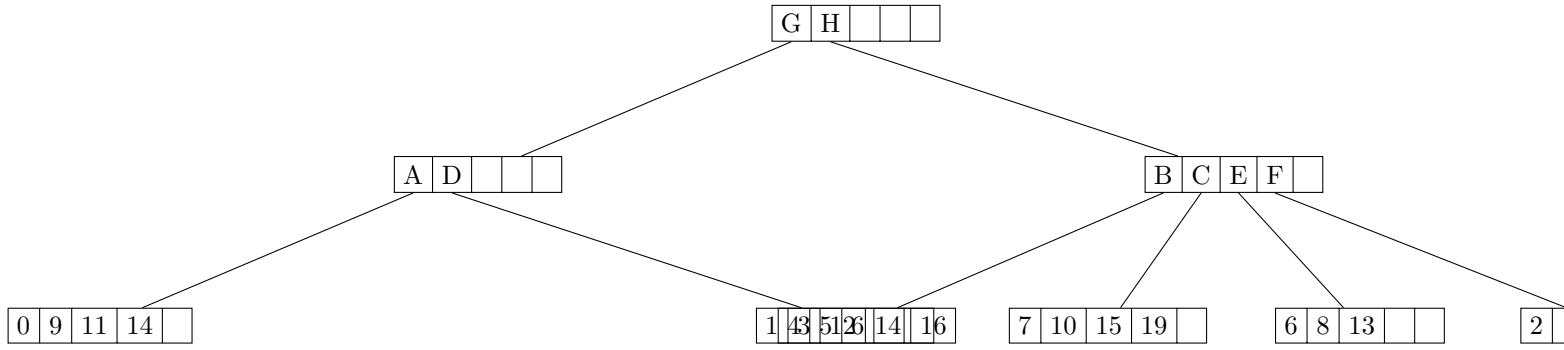
it is an interval node, we descent to all childs which intersect the rectangle 19:
 it is an interval node, we descent to all childs which intersect the rectangle 19:
 a leaf is found: C



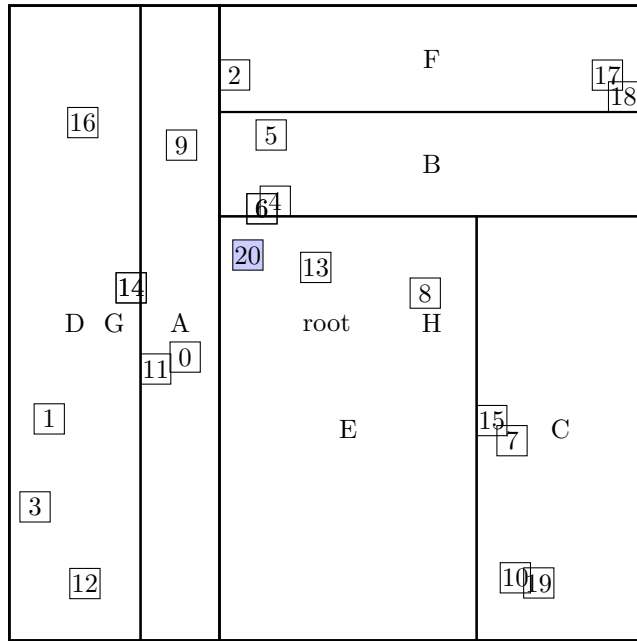
the node C is not full, add the record.



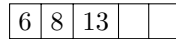
call INSERT with rectangle #S(P :X 739/500 :Y 49/20), R+.
 structure view:



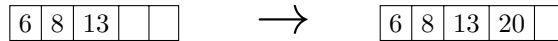
data view:



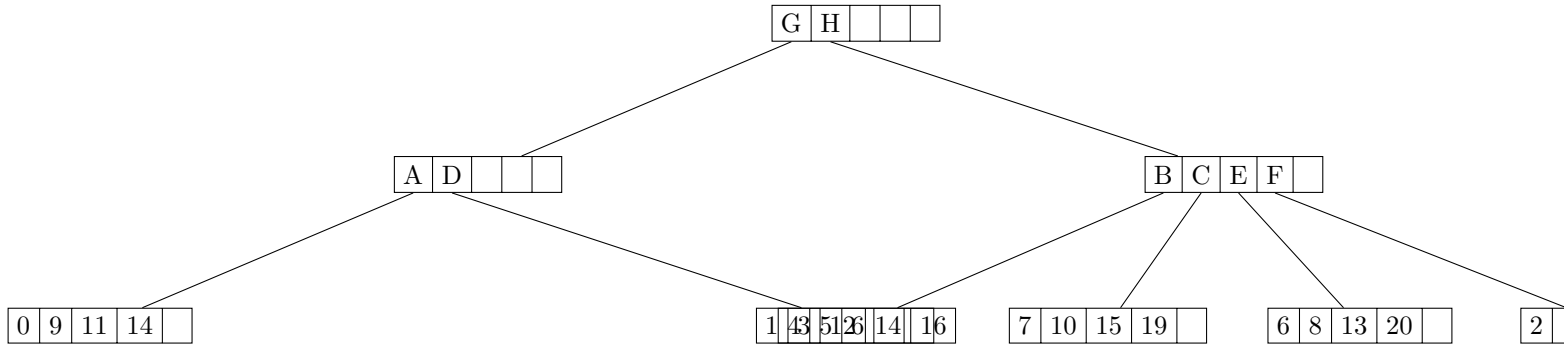
it is an interval node, we descent to all childs which intersect the rectangle 20:
 it is an interval node, we descent to all childs which intersect the rectangle 20:
 a leaf is found: E



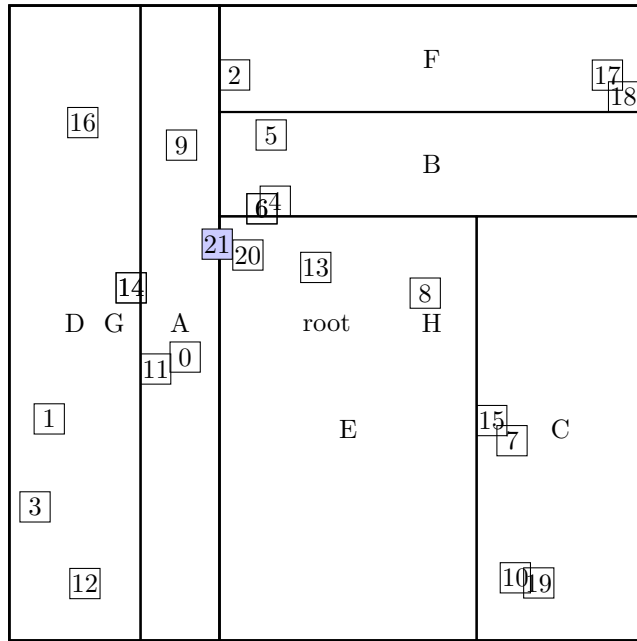
the node E is not full, add the record.



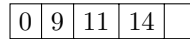
call INSERT with rectangle #S(P :X 51/40 :Y 1261/500), R_+ .
 structure view:



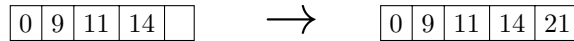
data view:



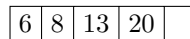
it is an interval node, we descent to all childs which intersect the rectangle 21:
 it is an interval node, we descent to all childs which intersect the rectangle 21:
 a leaf is found: A



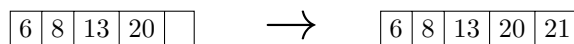
the node A is not full, add the record.



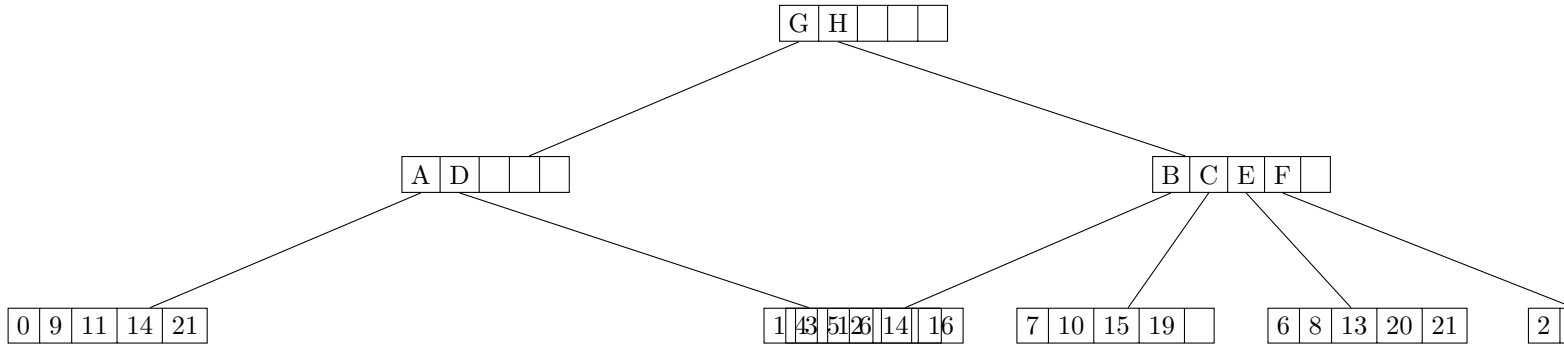
it is an interval node, we descent to all childs which intersect the rectangle 21:
 a leaf is found: E



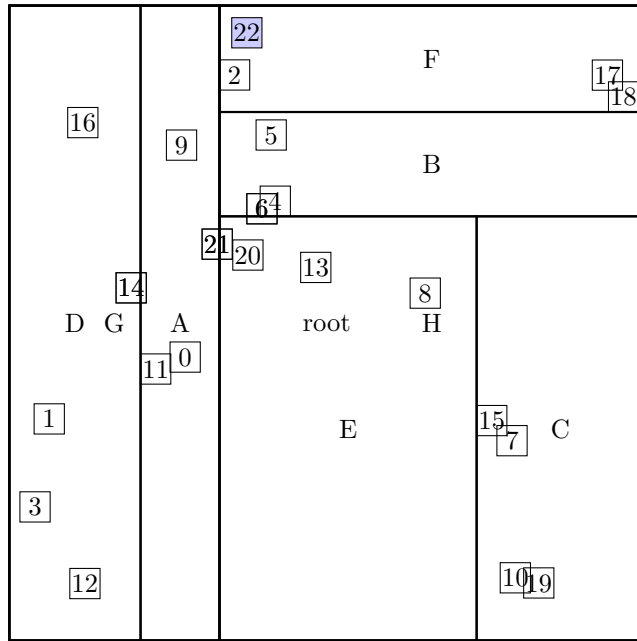
the node E is not full, add the record.



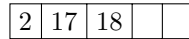
call INSERT with rectangle #S(P :X 1471/1000 :Y 98/25), R+.
 structure view:



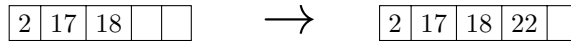
data view:



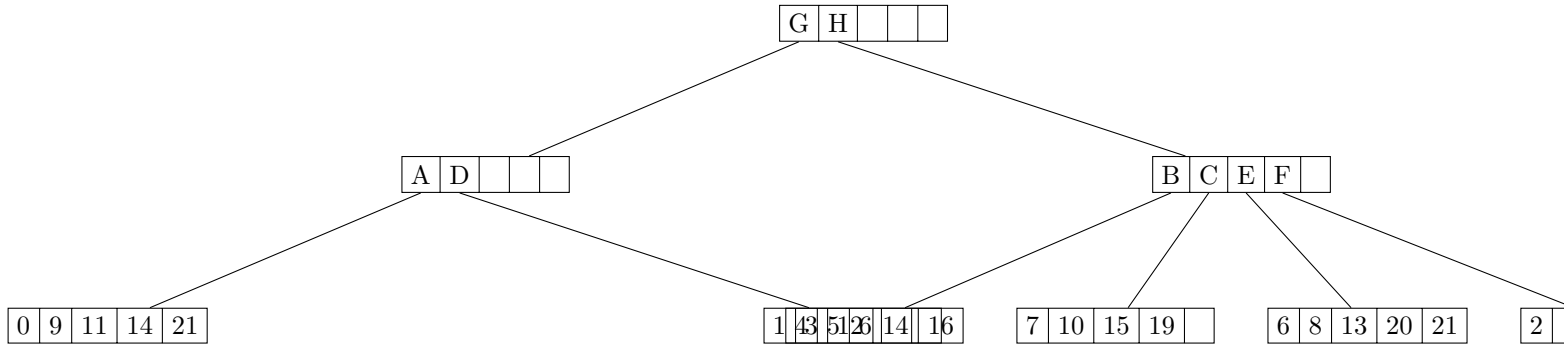
it is an interval node, we descent to all childs which intersect the rectangle 22:
 it is an interval node, we descent to all childs which intersect the rectangle 22:
 a leaf is found: F



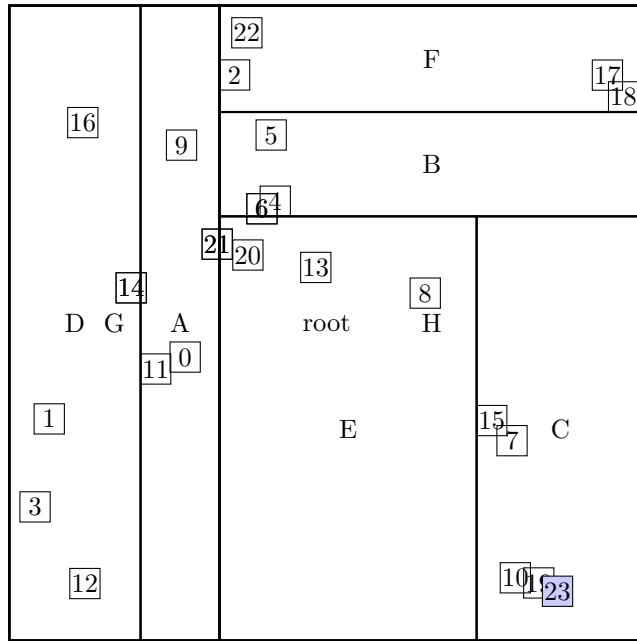
the node F is not full, add the record.



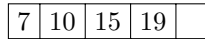
call INSERT with rectangle #S(P :X 1763/500 :Y 113/500), R+.
 structure view:



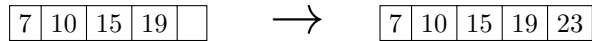
data view:



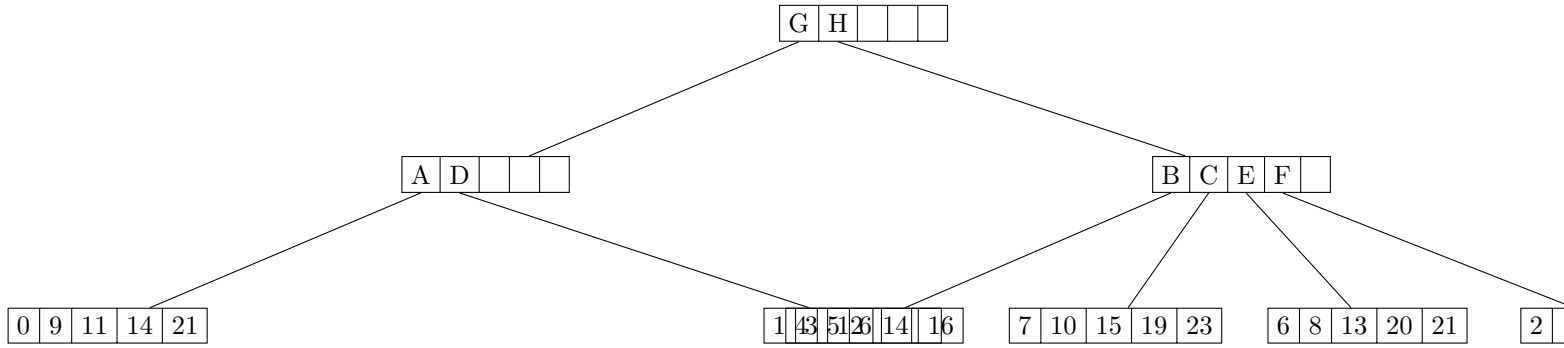
it is an interval node, we descent to all childs which intersect the rectangle 23:
 it is an interval node, we descent to all childs which intersect the rectangle 23:
 a leaf is found: C



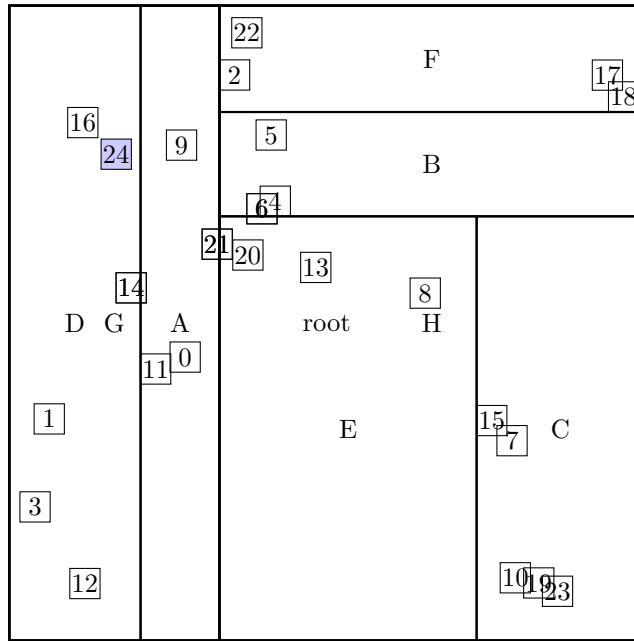
the node C is not full, add the record.



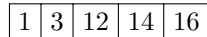
call INSERT with rectangle #S(P :X 607/1000 :Y 3117/1000), $R+$.
 structure view:



data view:



it is an interval node, we descent to all childs which intersect the rectangle 24:
 it is an interval node, we descent to all childs which intersect the rectangle 24:
 a leaf is found: D



the leaf D is full, need to split.
 call SPLIT-NODE with $R+$ node newnode
 generate partition rectangles.
 sort the rectangle by their low bounds wrt. axis x and divide them w.r.t fill factor 3.
 (3 1 16) — (12 24 14)
 sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.
 sort the rectangle by their low bounds wrt. axis y and divide them w.r.t fill factor 3.
 (12 3 1) — (14 24 16)
 Split the entries into the two areas.

