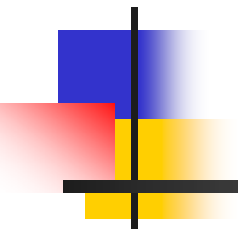


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΤΜΗΥΠ
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΙΙ

Β. Μεγαλοοικονόμου

Μέθοδοι Προσπέλασης Χωρικών Δεδομένων ΙΙ
Spatial Access Methods (SAMs) II

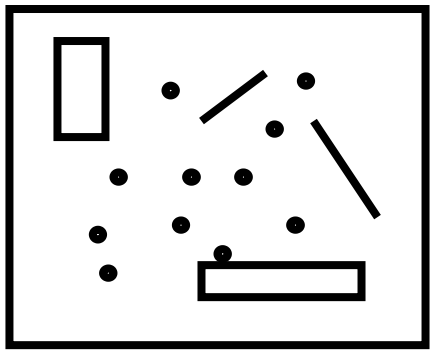
(παρουσίαση βασισμένη εν μέρη σε σημειώσεις των Silberchatz, Korth και Sudarshan και του C. Faloutsos)

Γενική επισκόπηση

- Δεικτοδότηση πολυμέσων
  - spatial access methods
    - πρόβλημα dfn
    - k-d trees
    - point quadtrees
    - MX-quadtrees
    - z-ordering
    - R-trees

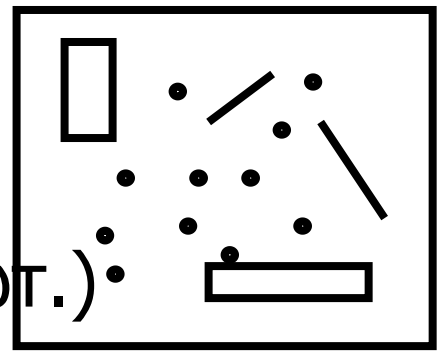
Spatial Access Methods - πρόβλημα

- Δοθείσας μίας συλλογής γεωμετρικών σχημάτων (σημεία, γραμμές, πολύγωνα ...)
- οργανώστε τα στο δίσκο, ώστε να απαντηθούν spatial ερωτήματα(όπως;)



Spatial Access Methods - πρόβλημα

- Δοθείσας μίας συλλογής γεωμετρικών σχημάτων (σημεία, γραμμές, πολύγωνα ...)
- οργανώστε τα στο δίσκο, ώστε να απαντήσετε
  - range queries
  - k-nn queries
  - spatial joins ('όλα τα ζεύγη' ερωτ.)

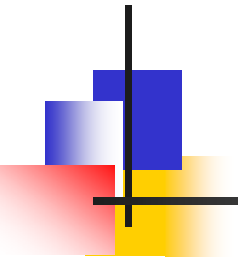


SAMs – Λεπτομερές διάγραμμα

- spatial access methods
  - problem dfn
  - k-d trees
  - point quadtrees
  - MX-quadtrees
  - z-ordering
  - R-trees

SAMs - Λεπτομερές διάγραμμα

- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)

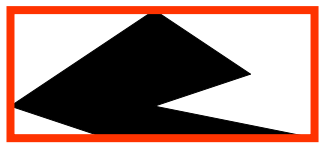


R-trees

- z-ordering: σπάει τις περιοχές σε κομμάτια -> εξάλειψη διπλοτυπιών
- πώς μπορούμε να το αποφύγουμε;
- Ιδέα: Minimum Bounding Rectangles

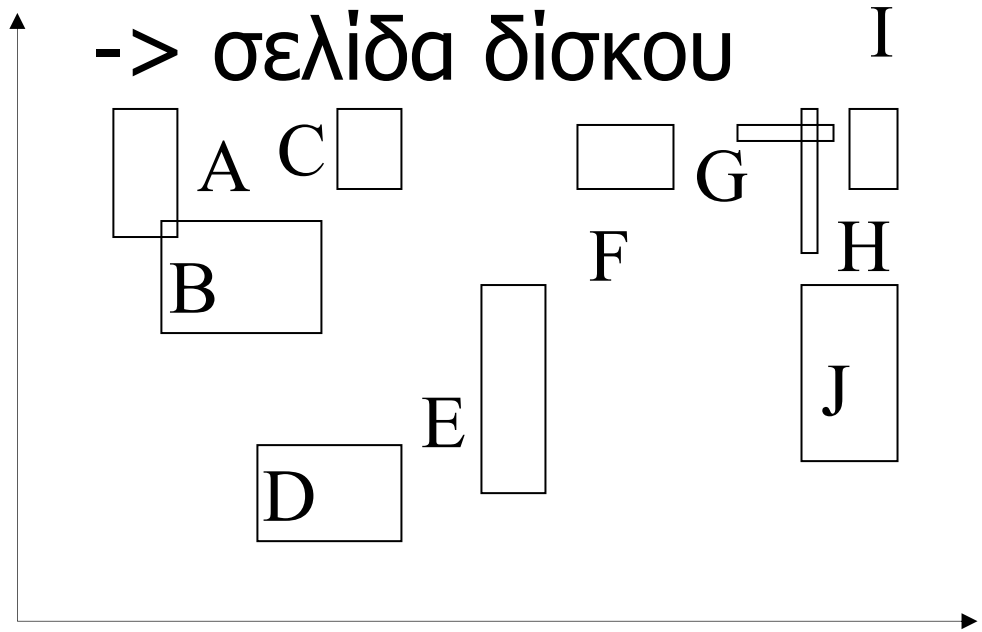
R-trees

- [Guttman 84] Κεντρική ιδέα: επιτρέπει στους γονείς να επικαλύπτουν!
  - => εγγυημένη 50% αξιοποίηση
  - => ευκολότεροι αλγόριθμοι εισαγωγής/split.
  - (χειρίζεται μόνο Minimum Bounding Rectangles - **MBR**s)



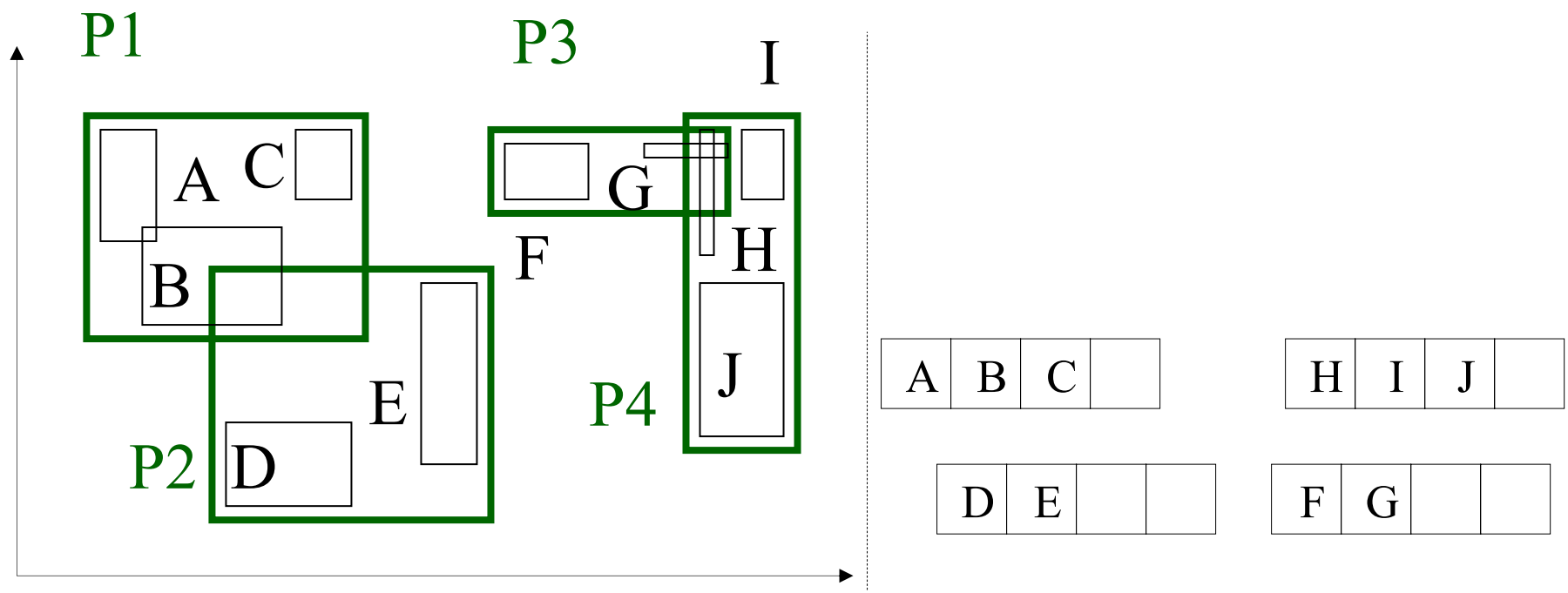
R-trees

- πχ., με fanout 4: ομαδοποίησε κοντινά ορθογώνια στους γονείς MBRs. Κάθε ομάδα -> σελίδα δίσκου



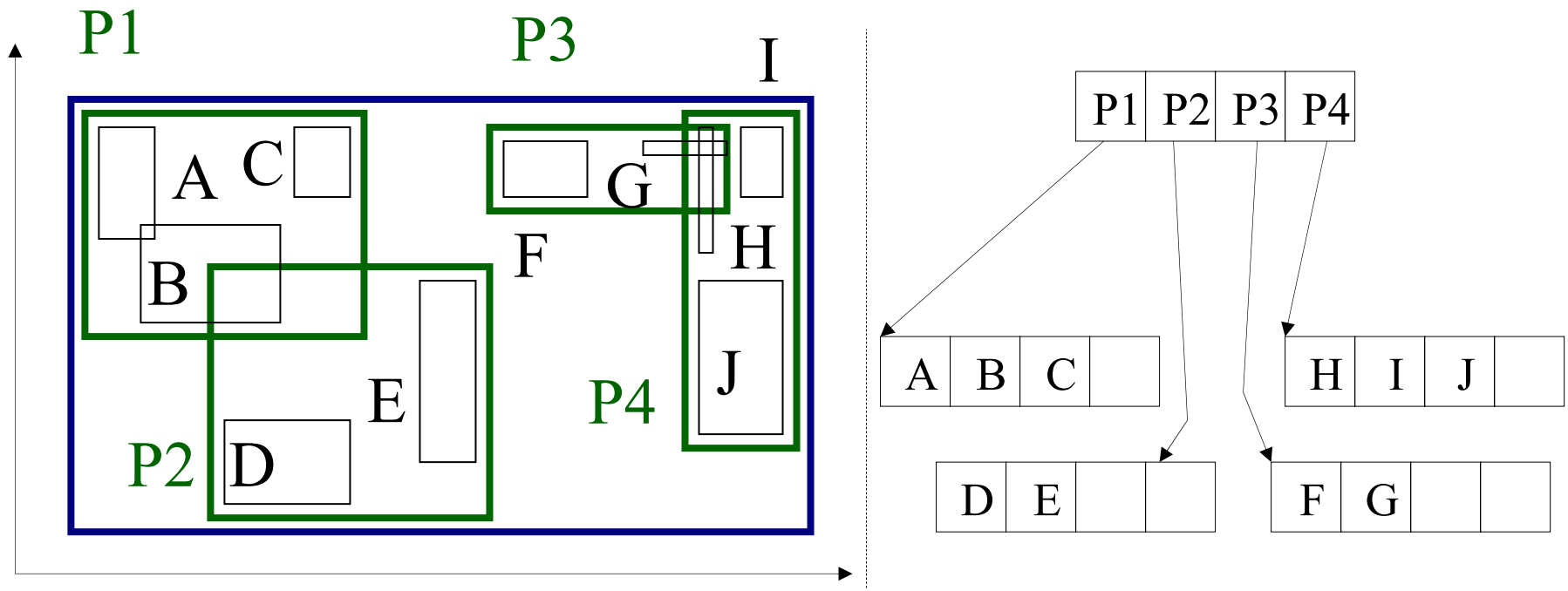
R-trees

- πχ., με fanout 4:



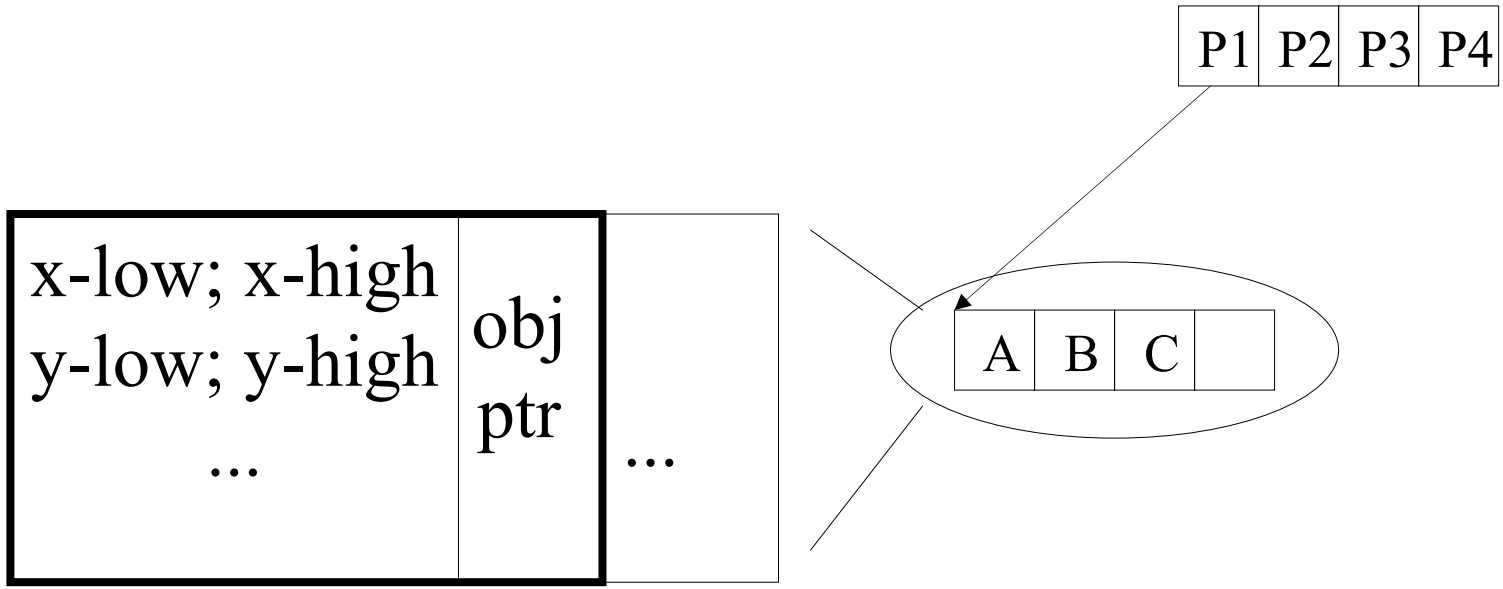
R-trees

- πχ., με fanout 4:



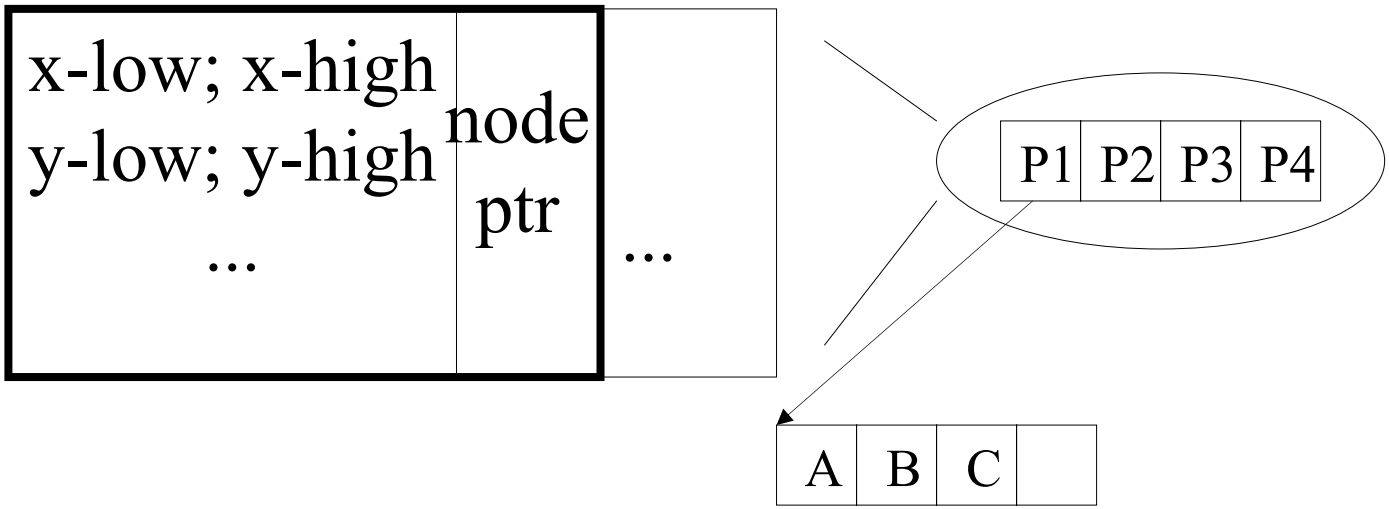
R-trees - οργάνωση κόμβων

- {(MBR; obj-ptr)} για φύλλα

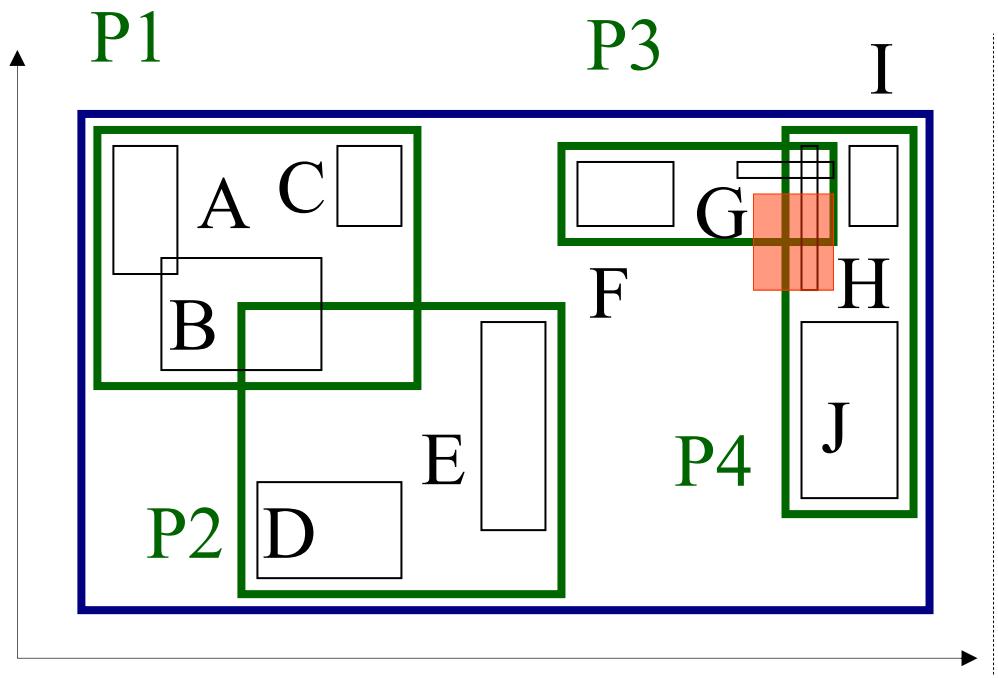


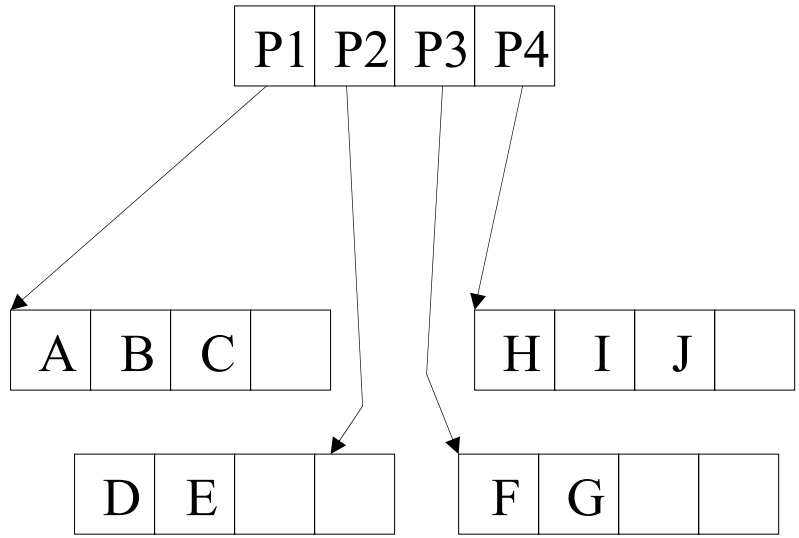
R-trees - οργάνωση κόμβων

- {(MBR; node-ptr)} για μη-φύλλα

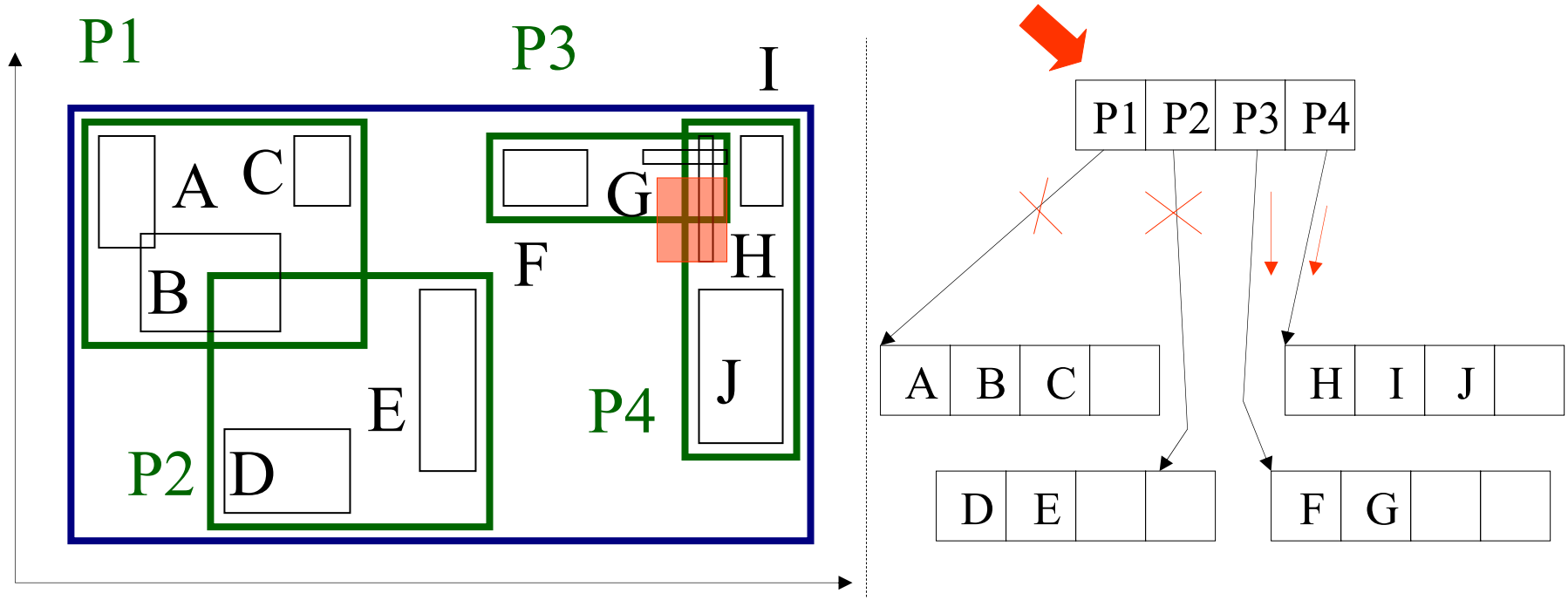


R-trees - αναζήτηση εύρους;





R-trees - αναζήτηση εύρους;



R-trees - αναζήτηση εύρους

Παρατηρήσεις:

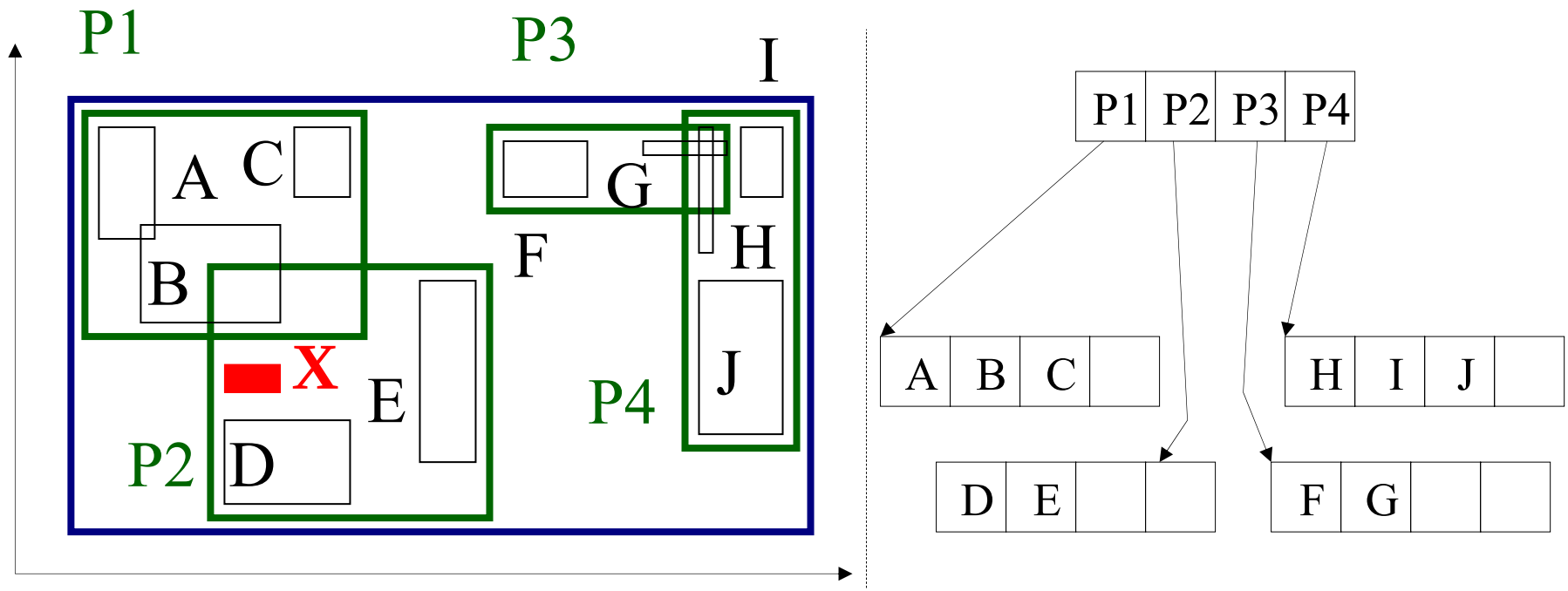
- Κάθε κόμβος γονέας καλύπτει πλήρως τα παιδιά του
- ένα MBR παιδί μπορεί να καλύπτετε από περισσότερους από έναν πατέρα – αποθηκεύεται κάτω από ΜΟΝΟ ΕΝΑΝ από αυτούς. (π.χ., δεν χρειάζεται εξάληψη διπλοτύπων)
- ένα σημειακό ερώτημα μπορεί να ακολουθεί πολλές διακλαδώσεις.
- όλα δουλεύουν για **κάθε** διάσταση

SAMs - Λεπτομερές διάγραμμα

- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)

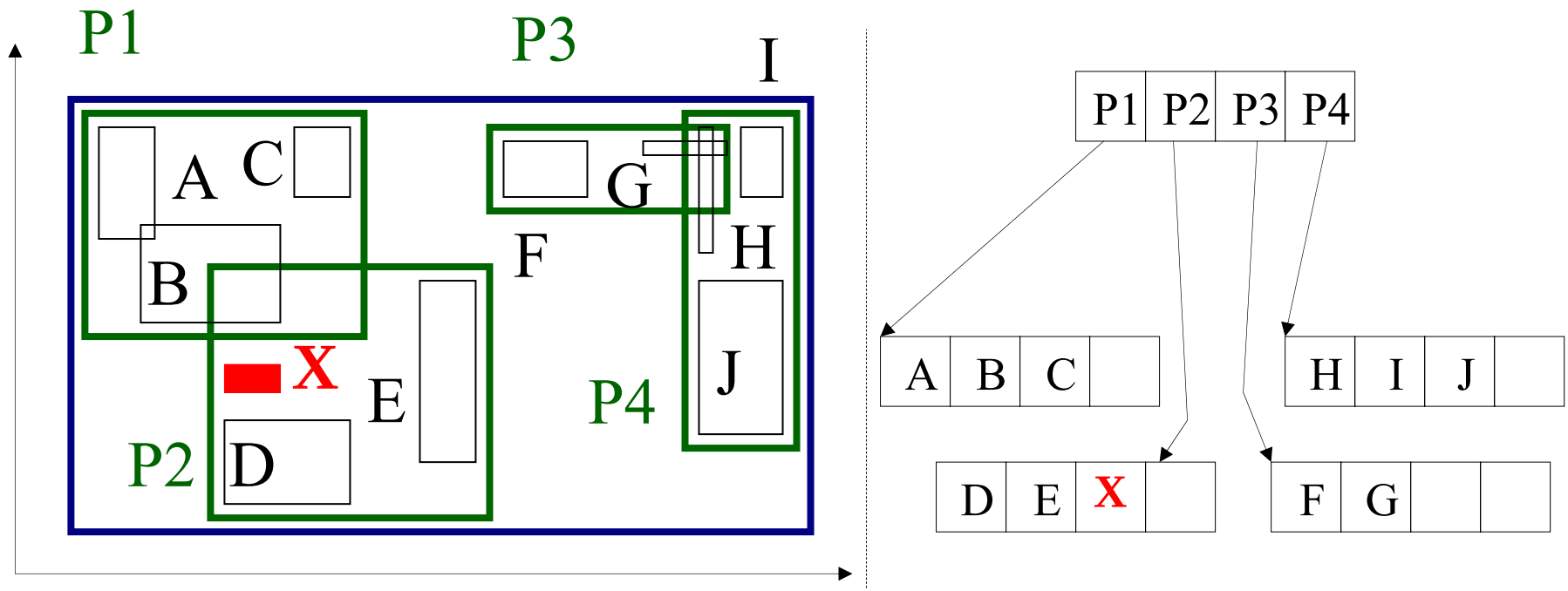
R-trees - εισαγωγή

- πχ., ορθογώνιο 'X'



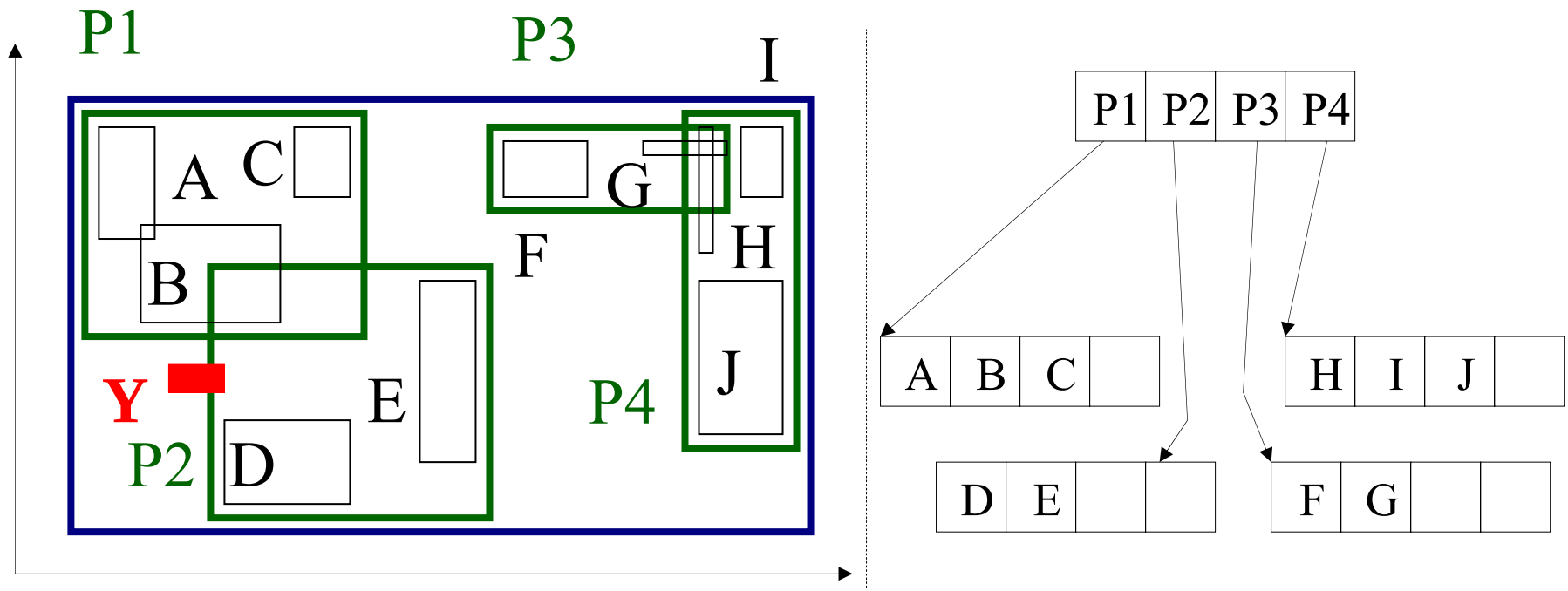
R-trees - εισαγωγή

- πχ., ορθογώνιο 'X'



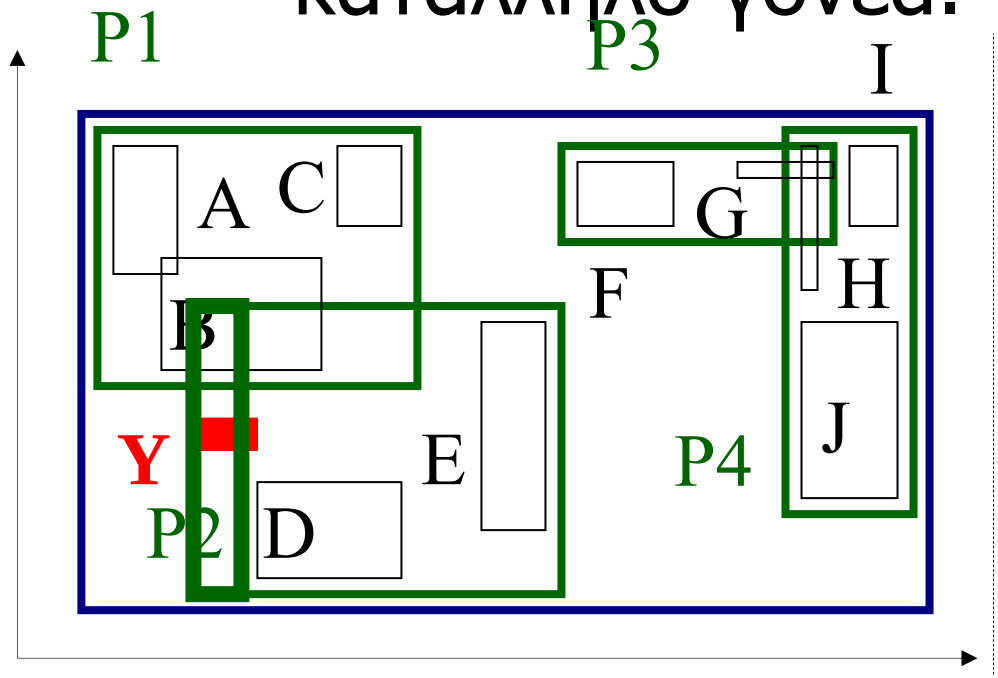
R-trees - εισαγωγή

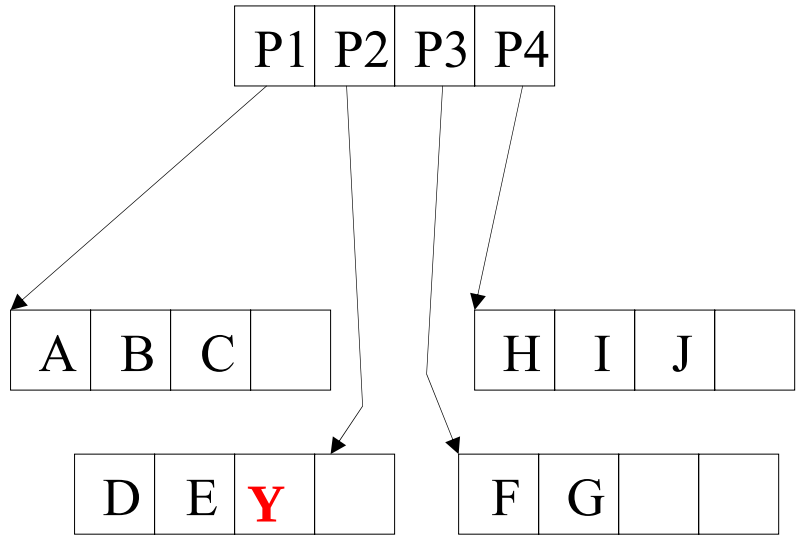
- πχ., ορθογώνιο 'Υ'



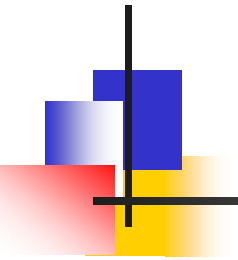
R-trees - εισαγωγή

- πχ., ορθογώνιο 'Υ' : επέκτεινε τον κατάλληλο γονέα.



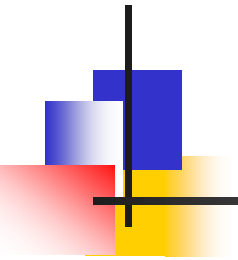


R-trees - εισαγωγή



- πχ., ορθογώνιο ‘Υ’: επέκτεινε τον κατάλληλο γονέα.
- Ε: πώς μετράμε την ‘καταλληλότητα’?

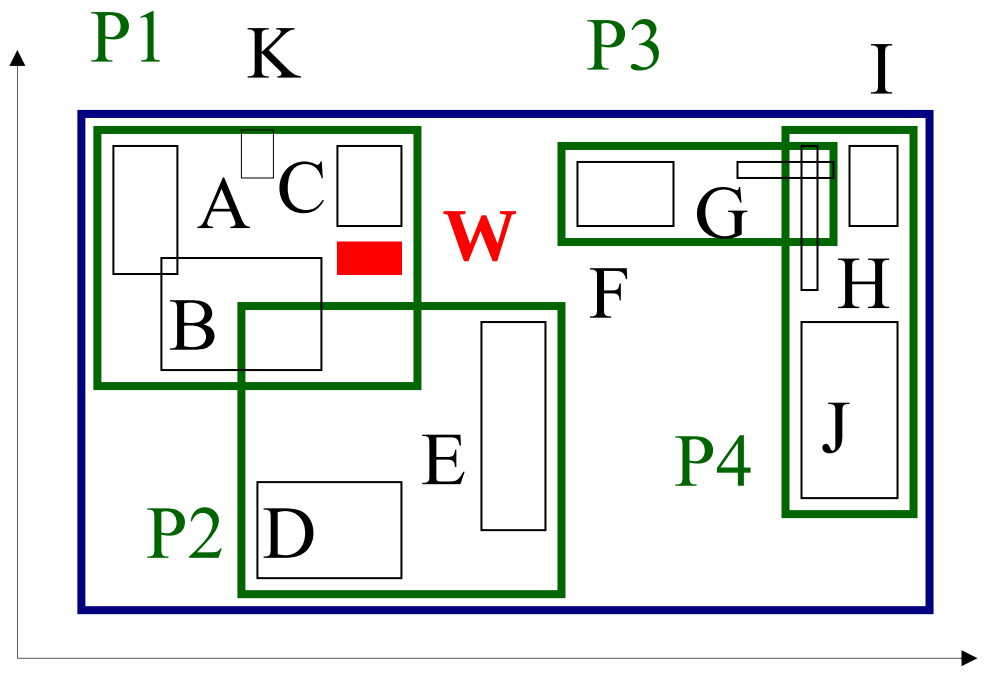
R-trees - εισαγωγή

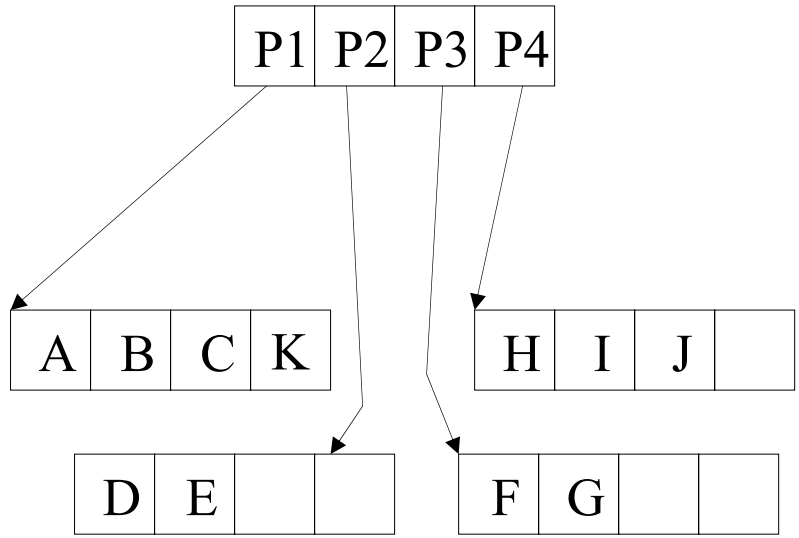


- πχ., ορθογώνιο 'Υ': επέκτεινε τον κατάλληλο γονέα.
- Ε: πώς μετράμε την 'καταλληλότητα'?
- Α: αυξάνοντας σε περιοχή (όγκο) (λεπτομερές: αργότερα, βλέπε 'ανάλυση απόδοσης')
- Ε: τι γίνεται αν δεν υπάρχει χώρος. Πώς να διαχωριστεί;

R-trees - εισαγωγή

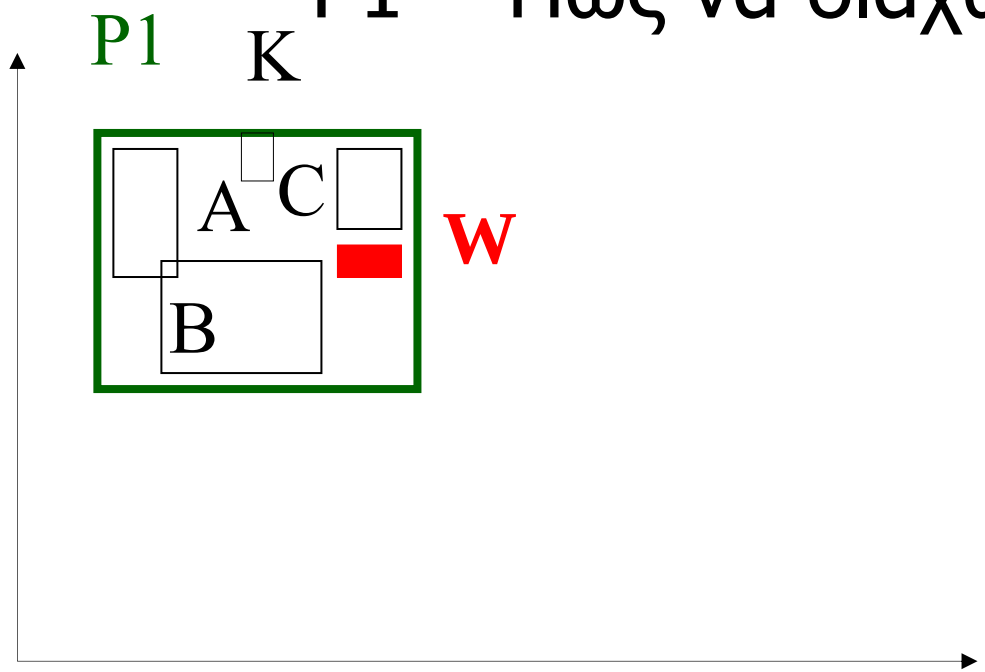
- πχ., ορθογώνιο 'W'





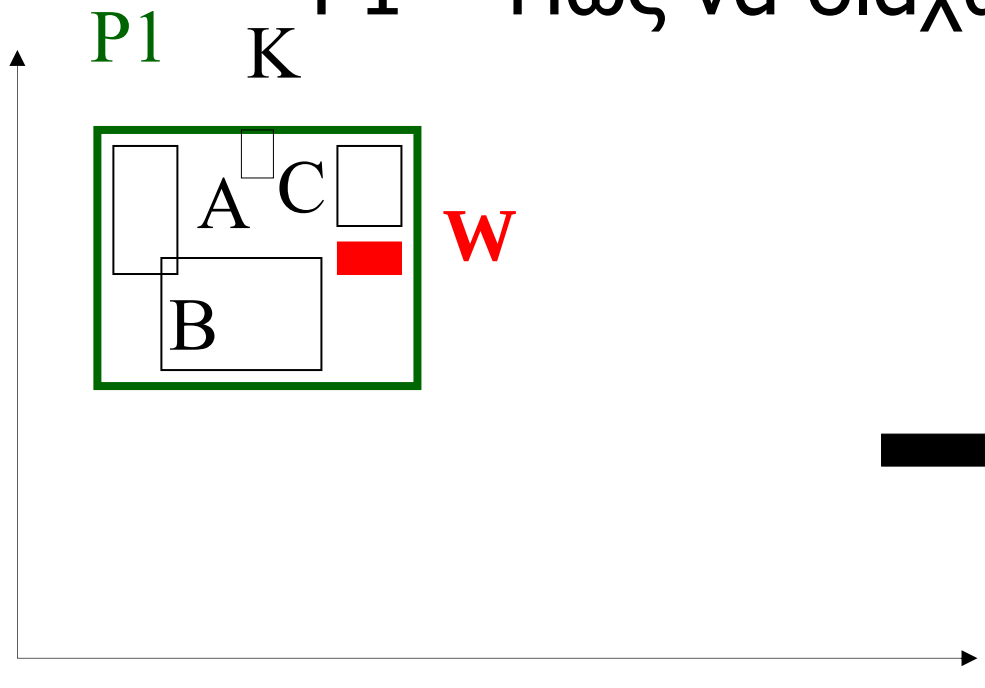
R-trees - εισαγωγή

- πχ., ορθογώνιο 'W' – επικεντρώστε στο 'P1' - Πώς να διαχωριστεί;



R-trees - εισαγωγή

- πχ., ορθογώνιο 'W' - επικεντρώστε στο 'P1' - Πώς να διαχωριστεί;

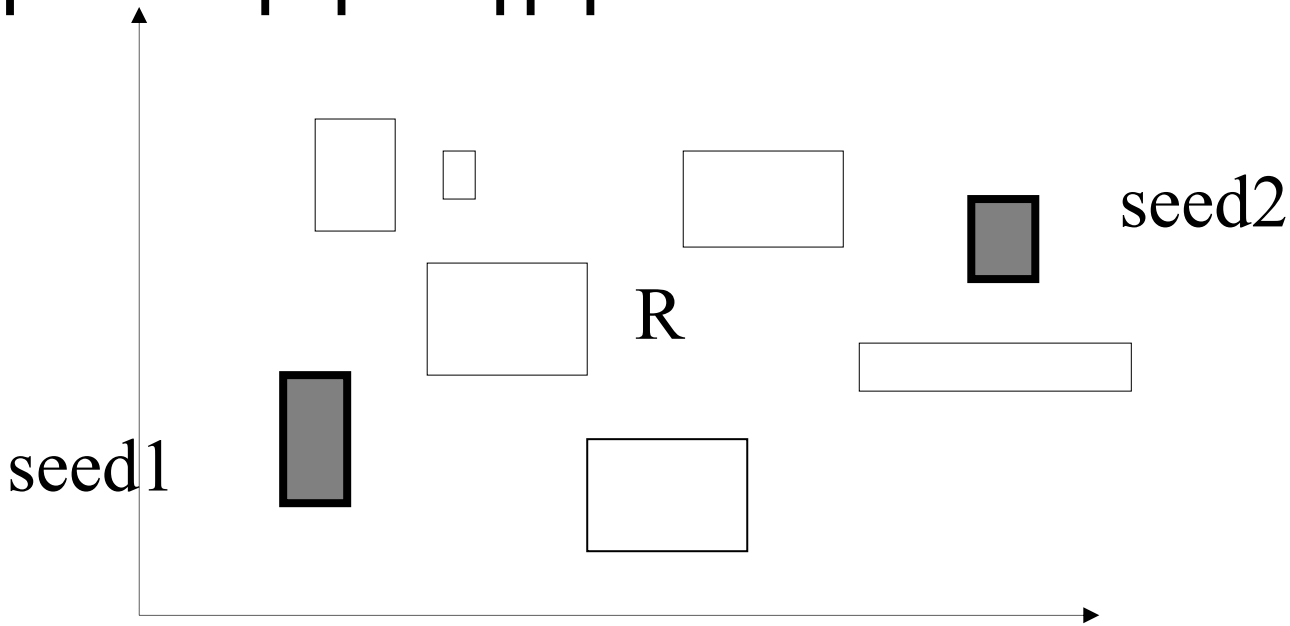


- (A1: plane sweep,

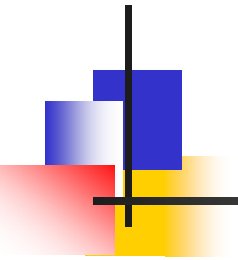
  Μέχρι το 50% του ορθογωνίου)
- A2: 'linear' split
- A3: quadratic split
- A4: exponential split

R-trees - εισαγωγή & split

- επιλέξτε δύο ορθογώνια ως 'πηγές'
- αναθέστε κάθε ορθογώνιο 'R' στη 'πλησιέστερη' 'πηγή'



R-trees - εισαγωγή & split



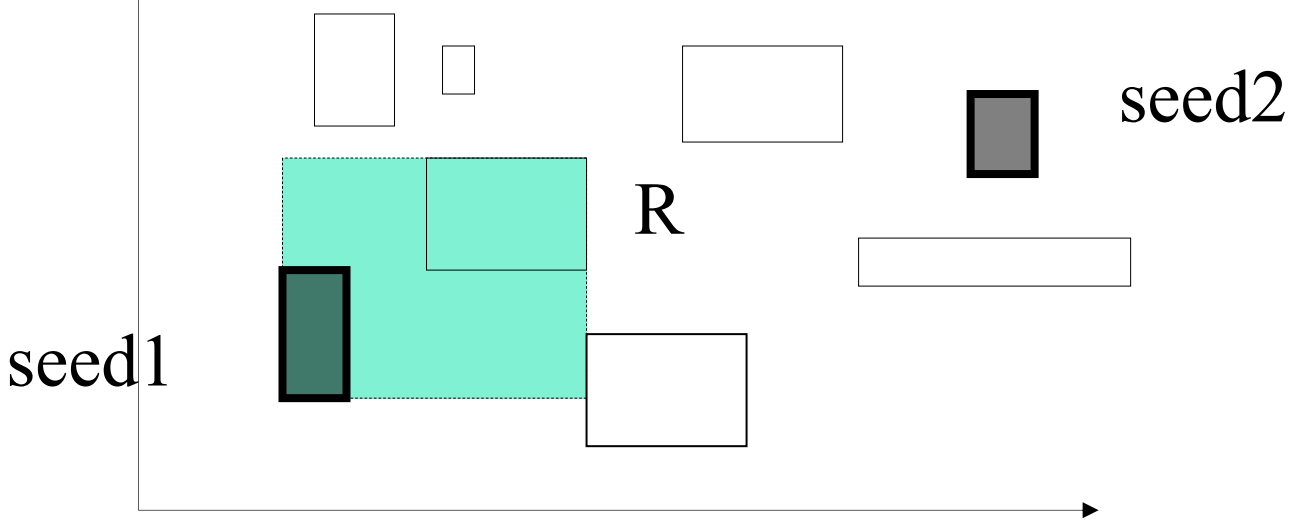
- επιλέξτε δύο ορθογώνια ως ‘πηγές’
- αναθέστε κάθε ορθογώνιο ‘R’ στη ‘πλησιέστερη’ ‘πηγή’
- E: πώς μετράμε την ‘εγγύτητα’?

R-trees - εισαγωγή & split

- επιλέξτε δύο ορθογώνια ως ‘πηγές’
- αναθέστε κάθε ορθογώνιο ‘R’ στη ‘πλησιέστερη’ ‘πηγή’
- E: πώς μετράμε την ‘εγγύτητα’?
- A: αυξάνοντας τη περιοχή (ένταση)

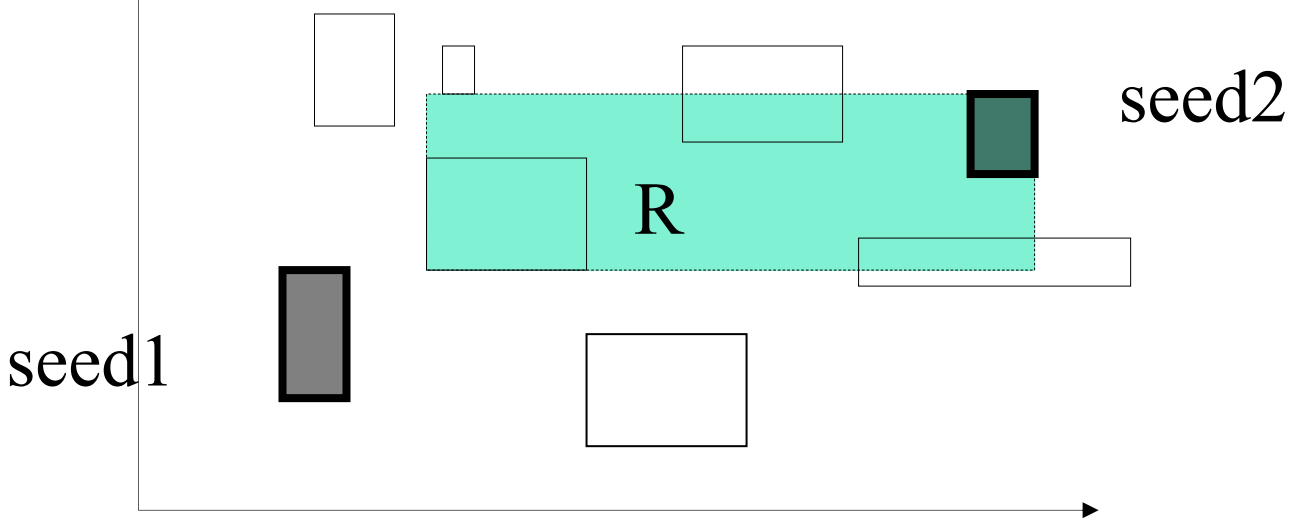
R-trees - εισαγωγή & split

- επιλέξτε δύο ορθογώνια ως 'πηγές'
- αναθέστε κάθε ορθογώνιο 'R' στη 'πλησιέστερη' 'πηγή'



R-trees - εισαγωγή & split

- επιλέξτε δύο ορθογώνια ως 'πηγές'
- αναθέστε κάθε ορθογώνιο 'R' στη 'πλησιέστερη' 'πηγή'



R-trees - εισαγωγή & split

- επιλέξτε δύο ορθογώνια ως ‘πηγές’
- αναθέστε κάθε ορθογώνιο ‘R’ στη ‘πλησιέστερη’ ‘πηγή’
- έξυπνη ιδέα: προ-διατάξτε τα ορθογώνια σύμφωνα με delta της εγγύτητα(πχ., προγραμματίστε πρώτα τις ευκολότερες επιλογές!)

R-trees - εισαγωγή - ψευτοκώδικας

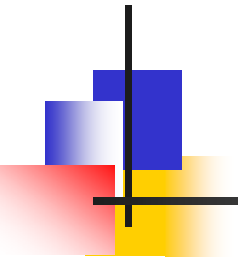
- επιλέξτε σε ποιον πατέρα θα εισάγετε το νέο ορθογώνιο ('πλησιέστερο' πατέρα)
- αν υπερχειλίσει, διαχωρίστε σε δύο, χρησιμοποιώντας (έστω) τον quadratic split αλγόριθμο
  - Διαδώστε το διαχωρισμό προς τα πάνω, αν χρειάζεται
- ενημερώστε τους MBRs των επηρεαζόμενων πατέρων.

R-trees - εισαγωγή - παρατηρήσεις

- υπάρχουν **πολύ** περισσότεροι αλγόριθμοι διαχωρισμού

SAMs - Λεπτομερές διάγραμμα

- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)



R-trees - διαγραφή

- διαγραφή ορθογωνίου
- αν υπερχειλίσει
  - ;;

R-trees - διαγραφή

- διαγραφή ορθογωνίου
- αν υπερχειλίσει
  - προσωρινά διαγράψτε όλα τους συγγενείς (!)
  - διαγράψτε τον πατρικό κόμβο και
  - επανα-εισάγετε τους

SAMs - Λεπτομερές διάγραμμα

- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)

R-trees - range αναζήτηση

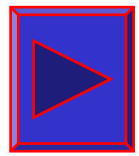
ψευτοκώδικας:

ελέγξτε τη ρίζα

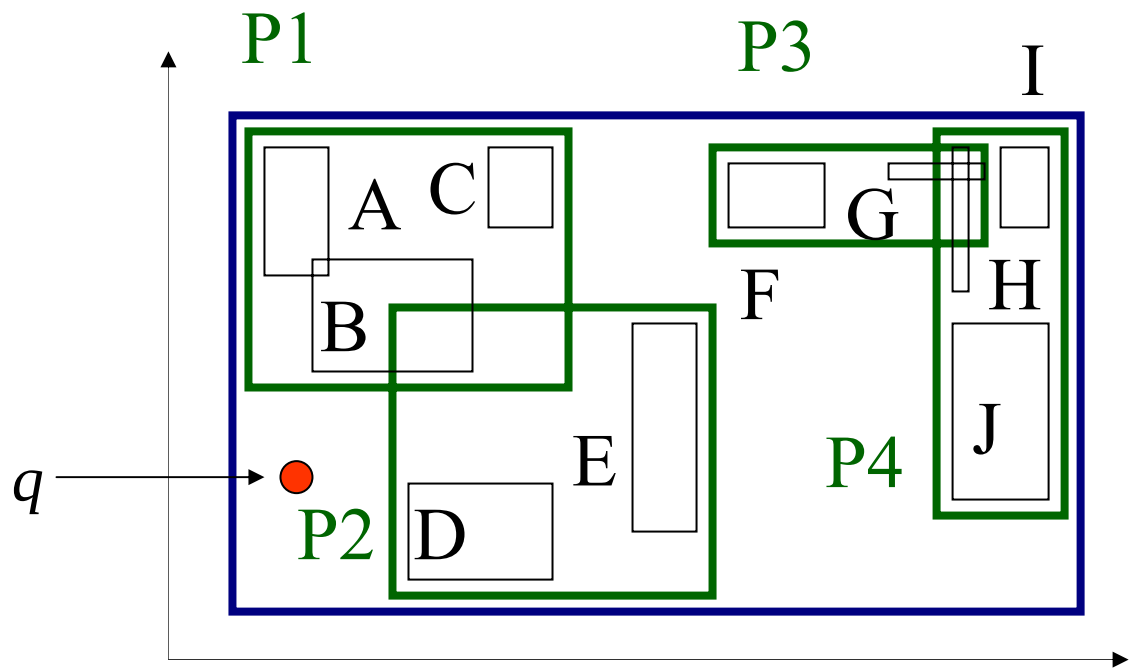
για κάθε κλαδί,

αν το MBR τέμνει το ορθογώνιο του ερωτήματος

εφάρμοσε αναζήτηση-εύρους (ή τύπωσε το, αν είναι φύλλο)



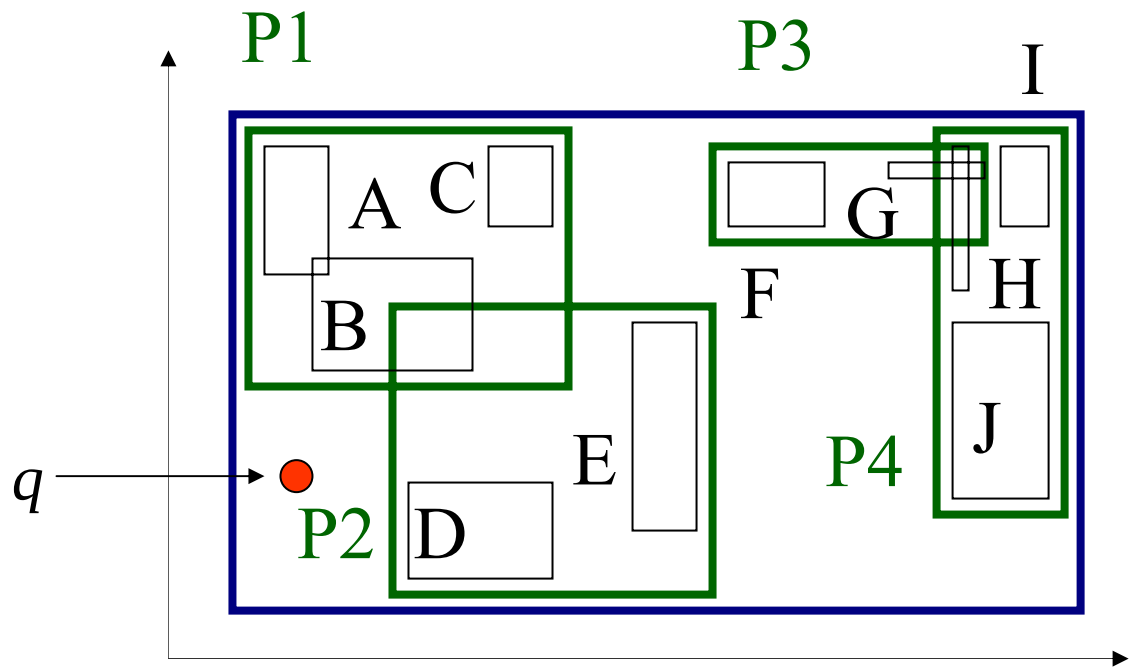
R-trees - nn αναζήτηση



R-trees - nn αναζήτηση

skip

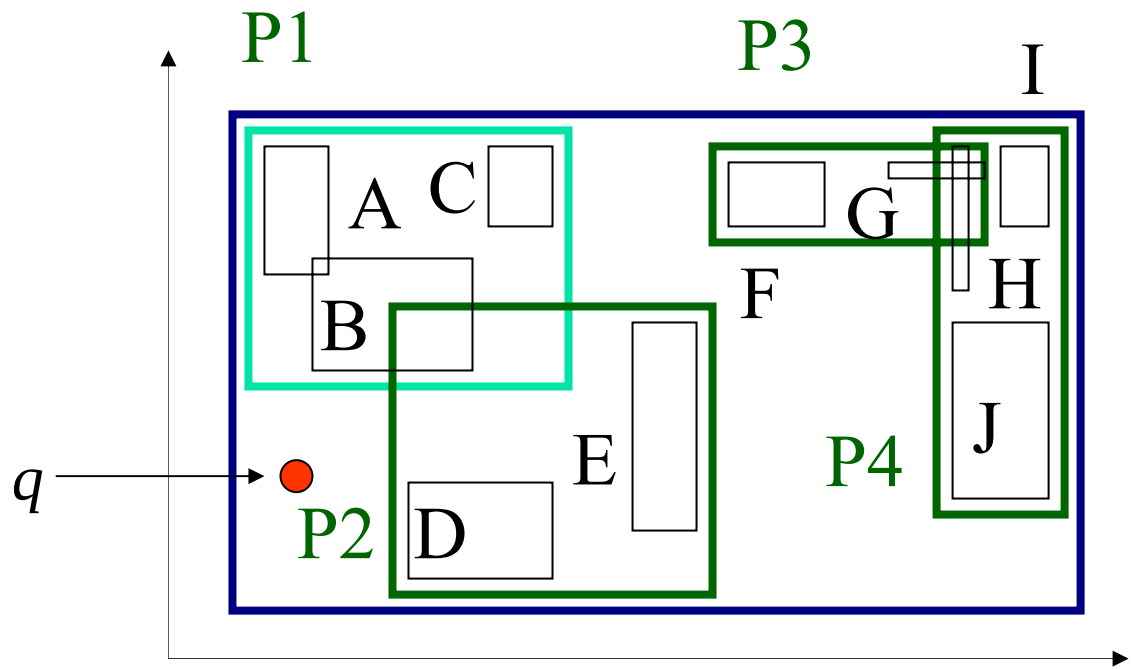
- E: Πώς; (βρες τον πλησιέστερο γείτονα, εκλέπτυνε...)



R-trees - nn αναζήτηση

skip

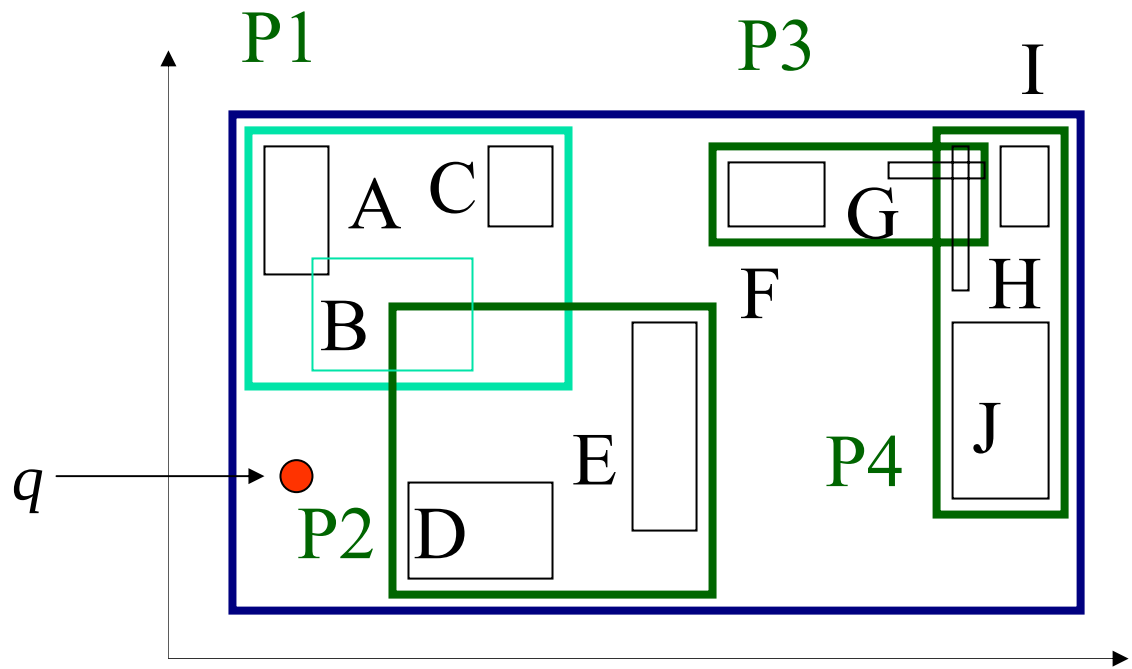
- A1: depth-first αναζήτηση, στη συνέχεια, αναζήτηση εύρους



R-trees - nn αναζήτηση

skip

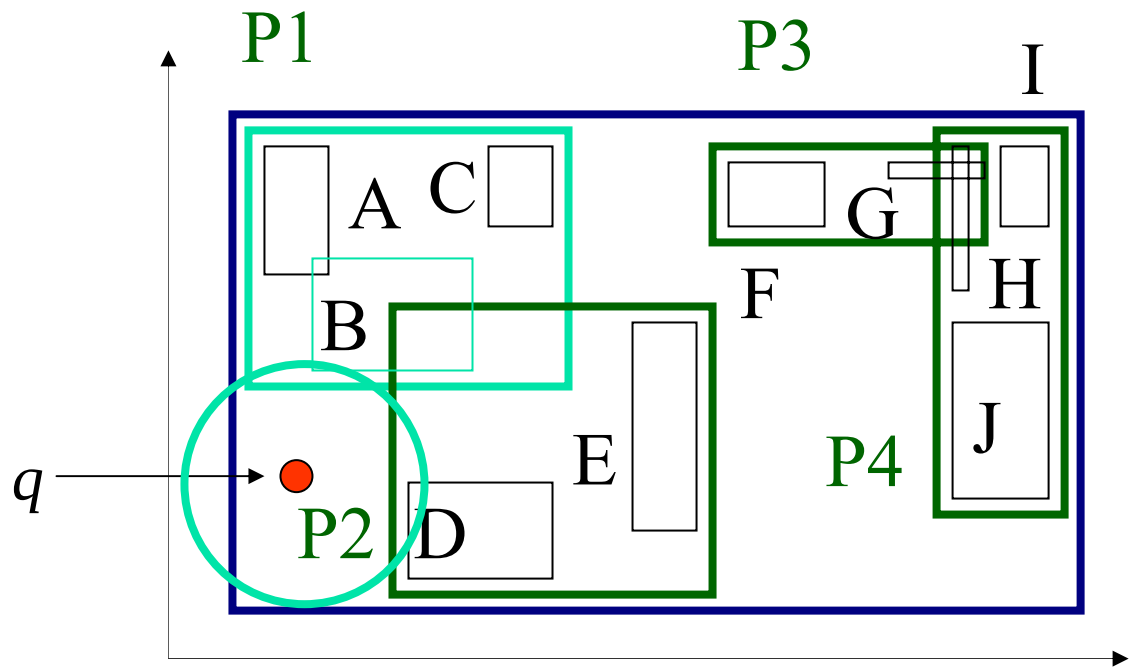
- A1: depth-first αναζήτηση, στη συνέχεια, αναζήτηση εύρους



R-trees - nn αναζήτηση

skip

- A1: depth-first αναζήτηση, στη συνέχεια, αναζήτηση εύρους



R-trees - nn αναζήτηση

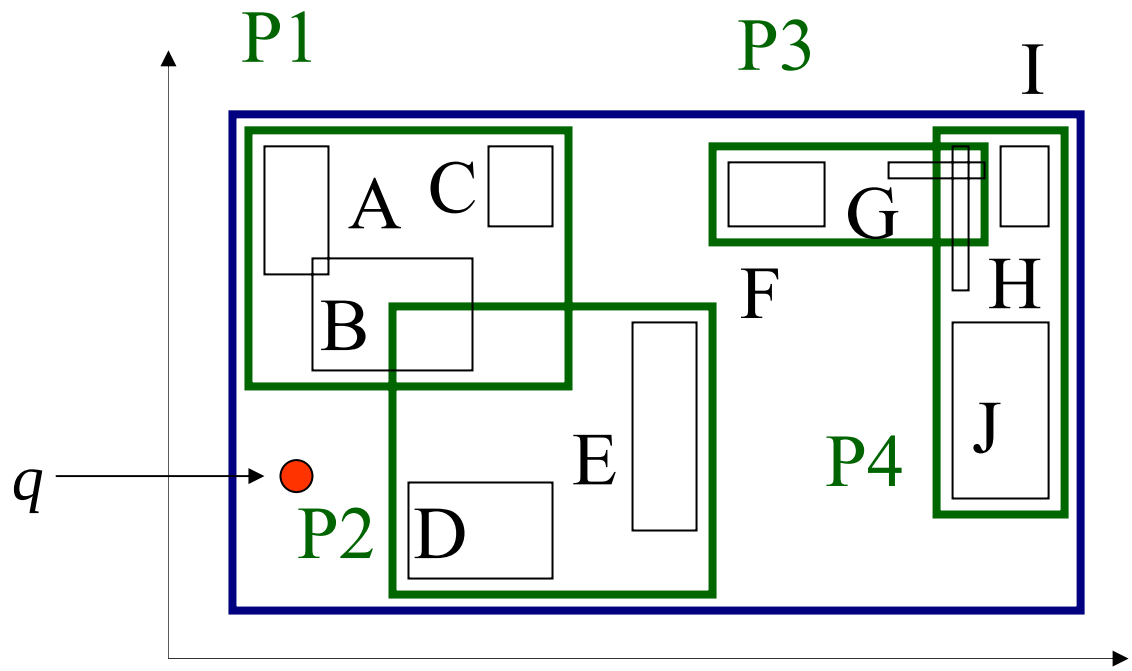
skip

- A2: [Roussopoulos+, sigmod95]:
  - ουρές προτεραιότητας, με υποσχόμενα MBRs και την βέλτιστη και χείριστη απόσταση
- κεντρική ιδέα:

R-trees - nn αναζήτηση

skip

θεωρήστε μόνο τα P2 και P4, χάριν παραδείγματος

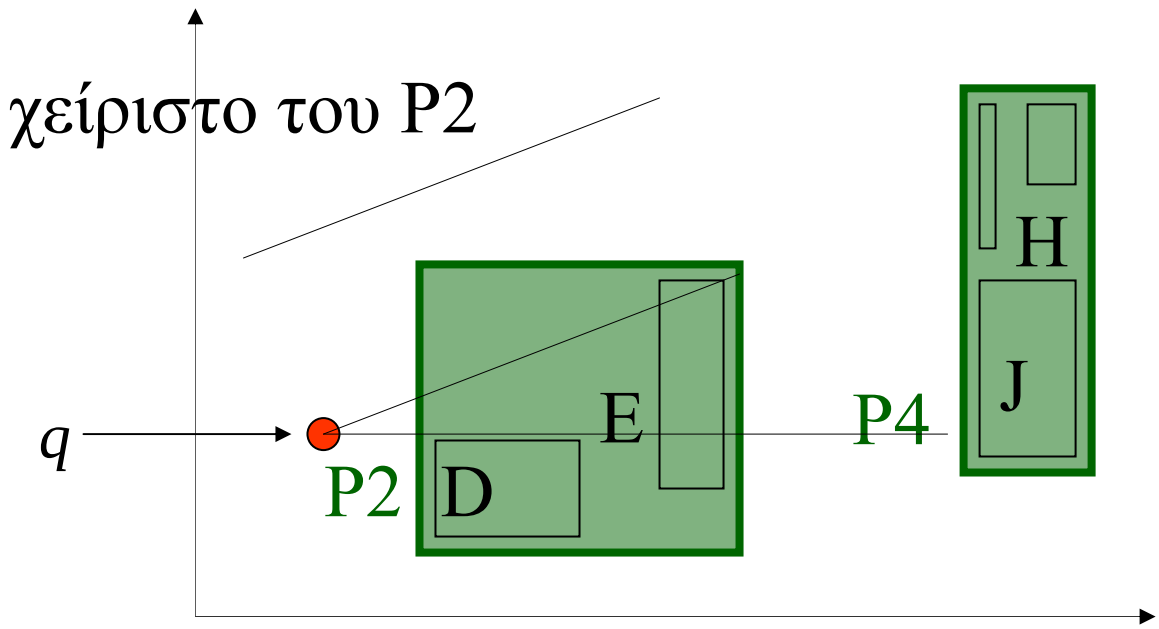


R-trees - nn αναζήτηση

skip

βέλτιστο του P4 => P4 είναι άχρηστο

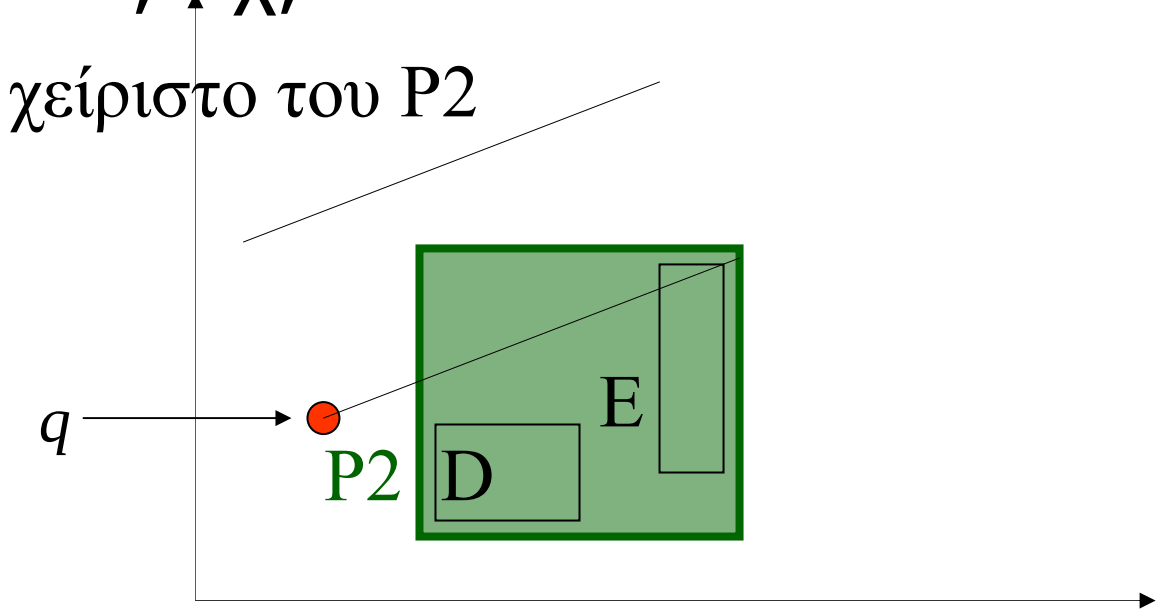
για 1-nn



R-trees - nn αναζήτηση

skip

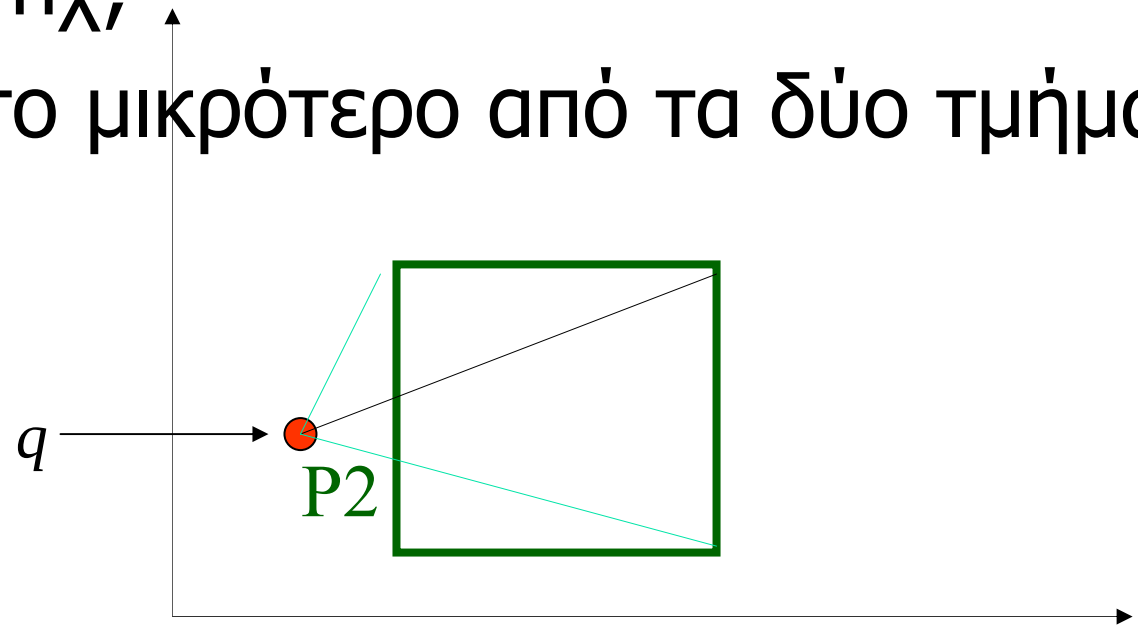
- ποιο είναι πραγματικά το χείριστο για το P2, πχ;



R-trees - nn αναζήτηση

skip

- ποιο είναι πραγματικά το χείριστο για το P2, πχ;
- A: το μικρότερο από τα δύο τμήματα!



R-trees - nn αναζήτηση

skip

- παραλλαγές: [Hjaltason & Samet] αυξητικό nn:
  - φτιάξτε μια ουρά προτεραιότητας (priority queue)
  - ερεύνησε αρκετά από το δένδρο, για να βεβαιωθείς ότι έχεις το *k* nn
  - για να βρεις το (*k+1)*-στό, έλεγξε την ουρά και ερεύνησε λίγο παραπάνω το δένδρο
- 'βέλτιστο' (αλλά, μπορεί να χρειαστεί πολύ μνήμη)

SAMs - Λεπτομερές διάγραμμα

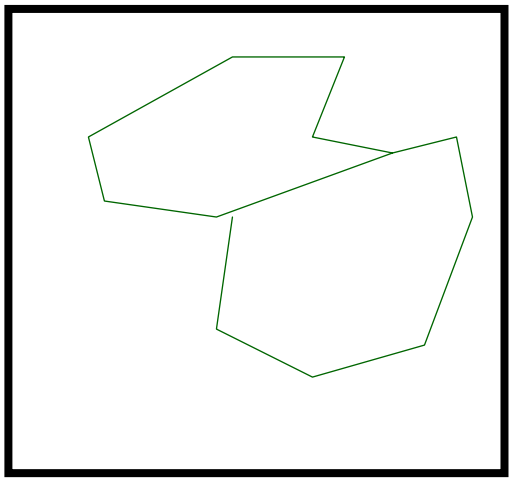
skip

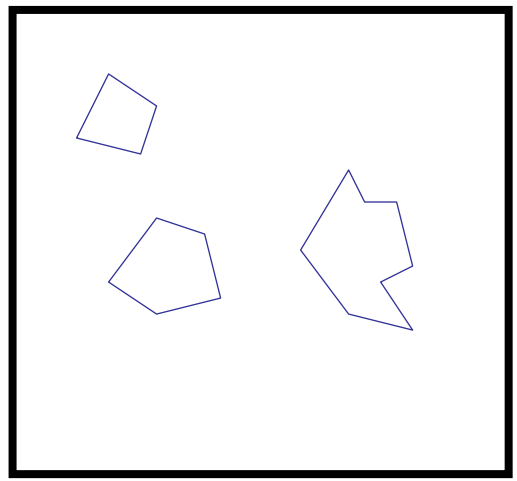
- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)

R-trees - spatial joins

skip

Spatial joins: βρες (γρήγορα) όλους τους νομούς που τέμνουν λίμνες

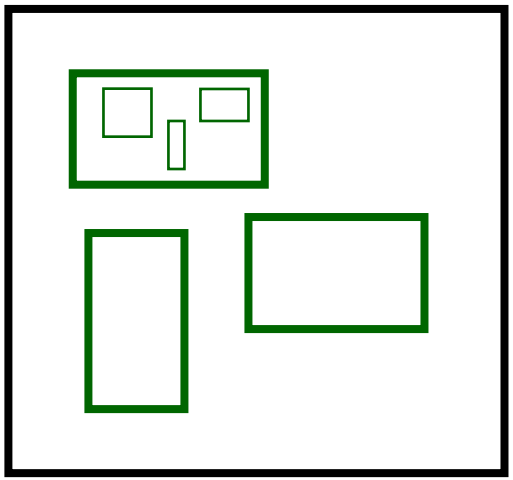


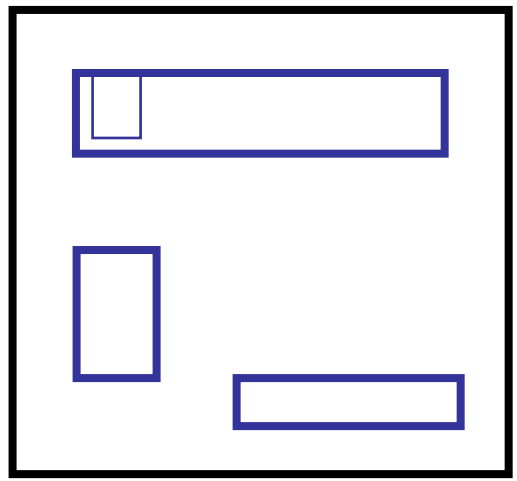


R-trees - spatial joins

skip

Υποθέστε ότι και τα δύο είναι οργανομένα σε R-trees:





R-trees - spatial joins

skip

για κάθε πατέρα P1 του δένδρου T1
  για κάθε πατέρα P2 του δένδρου T2
    αν το MBRs τους τέμνει,
      επεξεργαστείτε τα αναδρομικά (πχ., ελέγξτε τα παιδιά τους)

R-trees - spatial joins

skip

Βελτιώσεις - παραλλαγές:

- [Seeger+, sigmod 92]: κάνε κάποια προ-επεξεργασία, κάνε plane-sweeping για να αποφύγετε *N1 * N2* ελέγχους για τομές
- [Lo & Ravishankar, sigmod 94]: 'seeded' R-trees

(πολλές δημοσιεύσεις πάνω σε spatial joins, χωρίς R-trees: [Koudas+ Sevcik], κτλ.)

SAMs - Λεπτομερές διάγραμμα

- R-trees
  - κύρια ιδέα, δομή αρχείων
  - (αλγόριθμοι: εισαγωγή/split)
  - (διαγραφή)
  - (αναζήτηση: εύρος, nn, spatial joins)
  - ανάλυση απόδοσης
  - παραλλαγές (packed; hilbert;...)

R-trees - παραλλαγές

Guttman's R-trees ενέπνευσαν **περισσότερη** δουλειά

- μπορούμε να κάνουμε καλύτερους διαχωρισμούς;
- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
- τί συμβαίνει με άλλα συνοριακά σχήματα;

R-trees - παραλλαγές

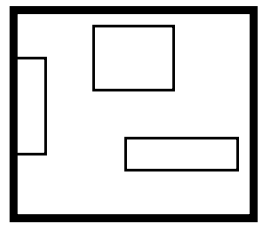
Guttman's R-trees ενέπνευσαν περισσότερη δουλειά

- μπορούμε να κάνουμε καλύτερους διαχωρισμούς;
  - π.χ, defer splits;

R-trees - παραλλαγές

A: R*-trees [Kriegel+, SIGMOD90]

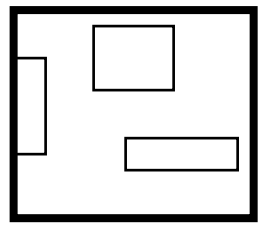
- defer splits, επιβάλλοντας επανα-εισαγωγές, π.χ.: αντί για διαχωρισμό, προσωρινά διάγραψε μερικές οντότητες, συρρικνώνοντας το περίσσιο MBR, και επανα-εισήγαγε αυτές τις οντότητες
- Ποιες να επανα-εισάγετε;
- Πόσες?

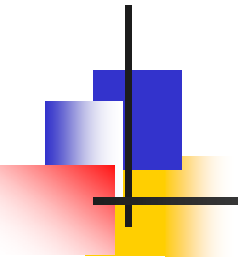


R-trees - παραλλαγές

A: R*-trees [Kriegel+, SIGMOD90]

- defer splits, επιβάλλοντας επανα-εισαγωγές, π.χ.: αντί για διαχωρισμό, προσωρινά διάγραψε μερικές οντότητες, συρρικνώνοντας το περίσσιο MBR, και επανα-εισήγαγε αυτές τις οντότητες
- Ποιες να επανα-εισάγετε;
- Πόσες? A: 30%





R-trees - παραλλαγές

Ε: Άλλοι τρόποι για defer splits;

R-trees - παραλλαγές

Ε: Άλλοι τρόποι για defer splits;

Α: Σπρώξτε μερικά κλειδιά προς το πλησιέστερο αδελφό κόμβο

(πλησιέστερο = ;;)

R-trees - παραλλαγές

R*-trees: Επίσης προσπαθήστε να ελαχιστοποιήσετε τη περιοχή ΚΑΙ το περίμετρο, στο διαχωρισμό.

Απόδοση: υψηλότερη αξιοποίηση του χώρου, γρηγορότερα από τα απλά R-trees. Μία από τις **πιο επιτυχημένες** παραλλαγές του R-tree.

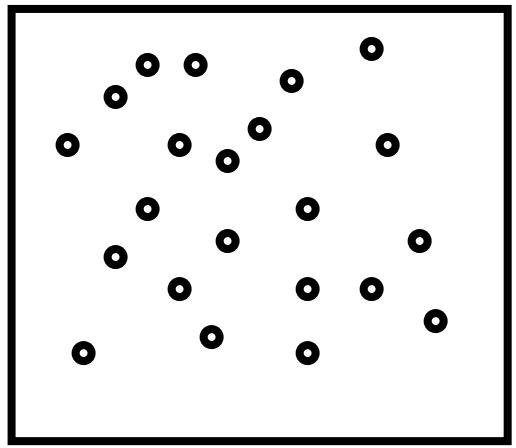
R-trees - παραλλαγές

Guttman's R-trees ενέπνευσαν **περισσότερη** δουλειά

- μπορούμε να κάνουμε καλύτερους διαχωρισμούς;
- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
  - Hilbert R-trees
- τί συμβαίνει με άλλα συνοριακά σχήματα;

R-trees - παραλλαγές

- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
- Ε: Βέλτιστος τρόπος για στοίβασμα σημείων;

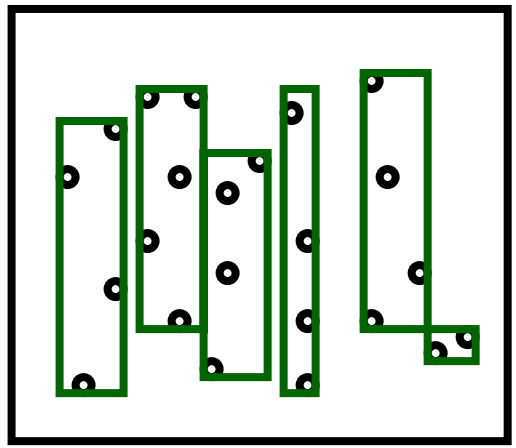


R-trees - παραλλαγές

- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
- E: Βέλτιστος τρόπος για στοίβασμα σημείων;
- A1: plane-sweep

  βέλτιστο για ερωτήματα στο ‘x’;

  χείριστο για ‘y’

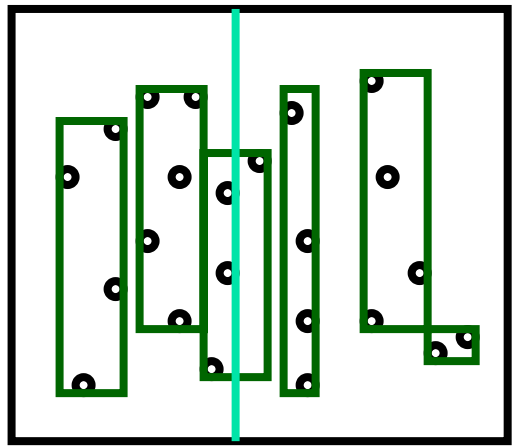


R-trees - παραλλαγές

- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
- E: Βέλτιστος τρόπος για στοίβασμα σημείων;
- A1: plane-sweep

  βέλτιστο για ερωτήματα στο ‘x’;

  χείριστο για ‘y’

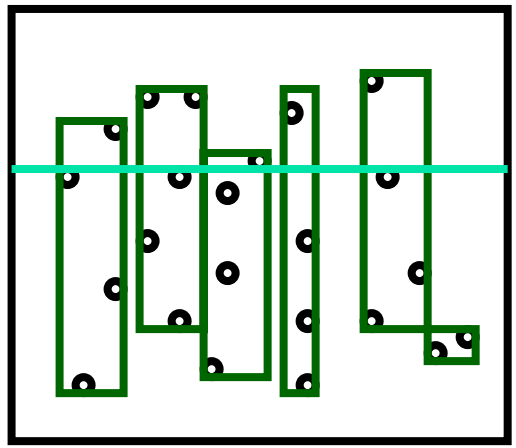


R-trees - παραλλαγές

- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
- E: Βέλτιστος τρόπος για στοίβασμα σημείων;
- A1: plane-sweep

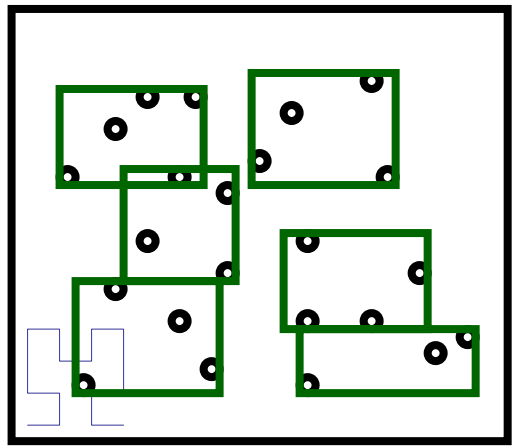
  βέλτιστο για ερωτήματα στο ‘x’;

  χείριστο για ‘y’
- E: Πώς βελτιώνεται;



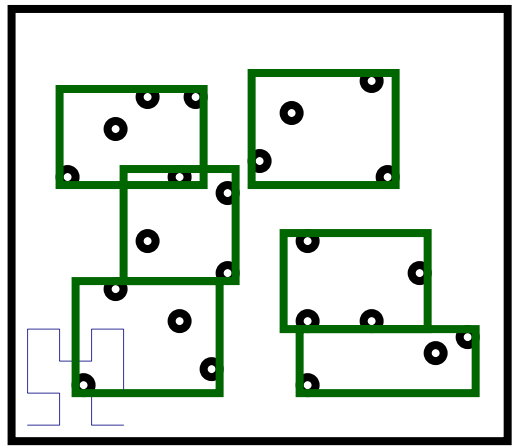
R-trees - παραλλαγές

- A: plane-sweep on HILBERT curve!



R-trees - παραλλαγές

- A: plane-sweep on HILBERT curve!
- Στη πράξη, μπορεί να γίνει δυναμικά (πώς;), επίσης και να χειριστεί εύρη (πώς;)
- A: [Kamel+, VLDB94]



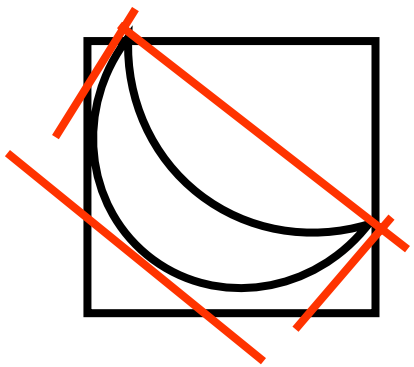
R-trees - παραλλαγές

Guttman's R-trees ενέπνευσαν **περισσότερη** δουλειά

- μπορούμε να κάνουμε καλύτερους διαχωρισμούς;
- τί συμβαίνει με στατικά δεδομένα (χωρίς εισ/διαγ/ενημ);
  - Hilbert R-trees
- τί συμβαίνει με άλλα συνοριακά σχήματα;

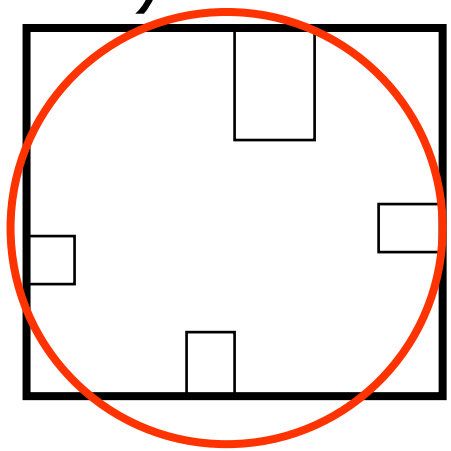
R-trees - παραλλαγές

- τί συμβαίνει με άλλα συνοριακά σχήματα; (και γιατί;)
- A1: arbitrary-orientation lines (cell-tree, [Guenther]
- A2: P-trees (πολύγωνα δένδρα) (MB πολύγωνο: 0, 90, 45, 135 μοιρών γραμμές)



R-trees - παραλλαγές

- A3: L-shapes; τρύπες (hB-tree)
- A4: TV-trees [Lin+, VLDB-Journal 1994]
- A5: SR-trees [Katayama+, SIGMOD97] (χρησιμοποιείτε στη Informedia)

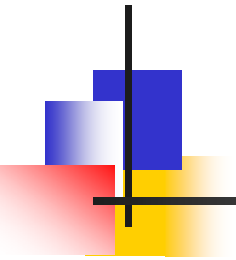


R-trees - συμπεράσματα

- Δημοφιλής μέθοδος, like multi-d B-trees
- εγγυημένη αξιοποίηση
- καλοί χρόνοι αναζήτησης (για μικρές διαστάσεις τουλάχιστον)
- R*-, Hilbert- and SR-trees: still used
- Informix ships DataBlade with R-trees

Αναφορές

- Guttman, A. (June 1984). R-Trees: A Dynamic Index Structure for Spatial Searching. Proc. ACM SIGMOD, Boston, Mass.
- Jagadish, H. V. (May 23-25, 1990). Linear Clustering of Objects with Multiple Attributes. ACM SIGMOD Conf., Atlantic City, NJ.
- Lin, K.-I., H. V. Jagadish, et al. (Oct. 1994). "The TV-tree - An Index Structure for High-dimensional Data." VLDB Journal 3: 517-542.



Αναφορές, συνέχεια

- Pagel, B., H. Six, et al. (May 1993). Towards an Analysis of Range Query Performance. Proc. of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Washington, D.C.
- Robinson, J. T. (1981). The k-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes. Proc. ACM SIGMOD.
- Roussopoulos, N., S. Kelley, et al. (May 1995). Nearest Neighbor Queries. Proc. of ACM-SIGMOD, San Jose, CA.