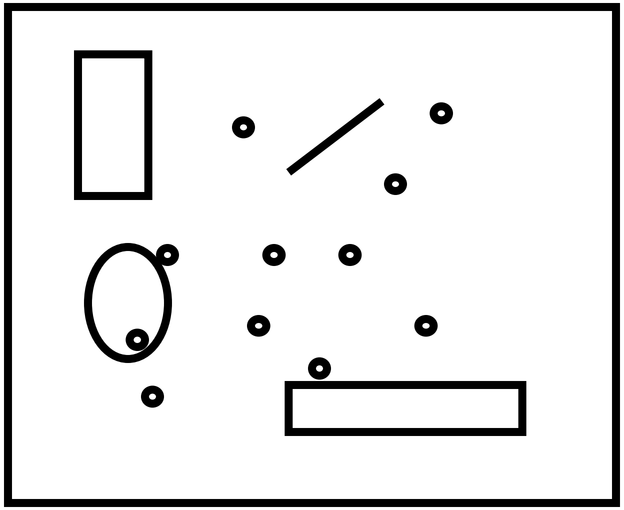
USC Viterbi School of Engineering

Spatial Index Structures
(R-tree Family)

Instructor: Cyrus Shahabi

Problem

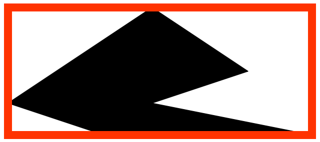
- Given a collection of geometric objects (points, lines, polygons, ...)
- organize them on disk, to answer spatial queries (range, nn, etc)



USC Viterbi School of Engineering

R-trees

- [Guttman 84] Main idea: extend B+-tree to multi-dimensional spaces!
  - (only deal with Minimum Bounding Rectangles - **MBR**s)



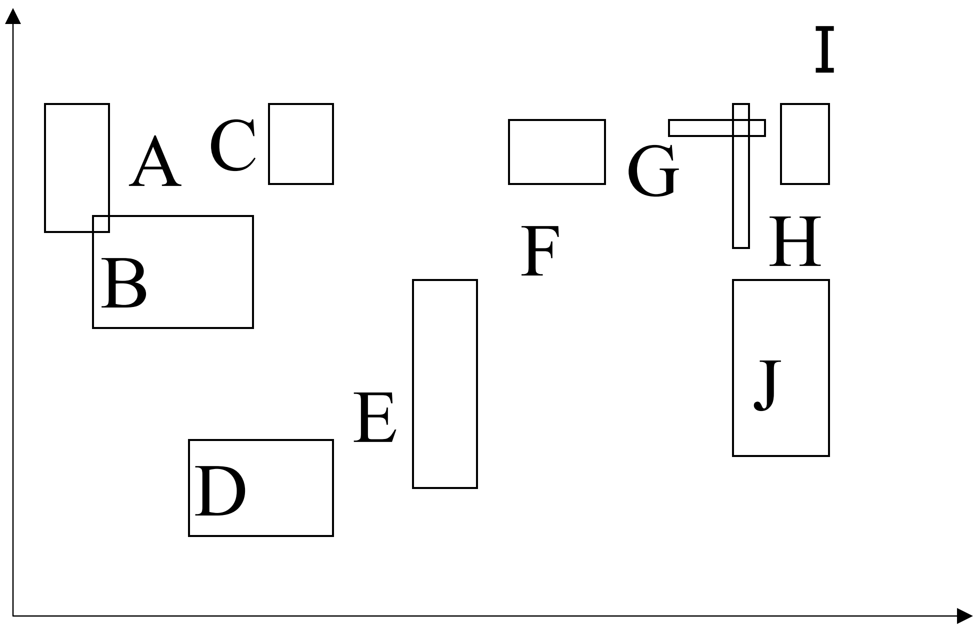
USC Viterbi
School of Engineering

R-trees

- A multi-way external memory tree
- Index nodes and data (leaf) nodes
- All leaf nodes appear on the same level
- Every node contains between m and M entries
- The root node has at least 2 entries (children)

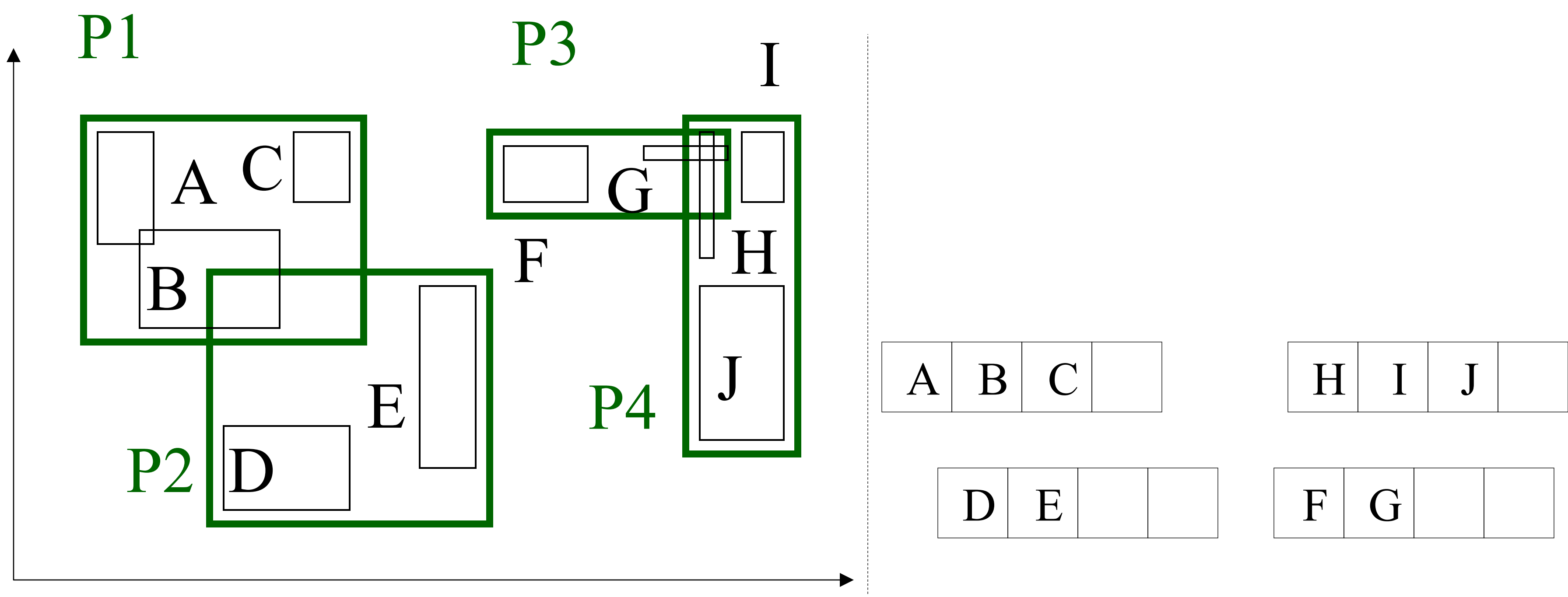
Example

- eg., w/ fanout 4: group nearby rectangles to parent MBRs; each group -> disk page



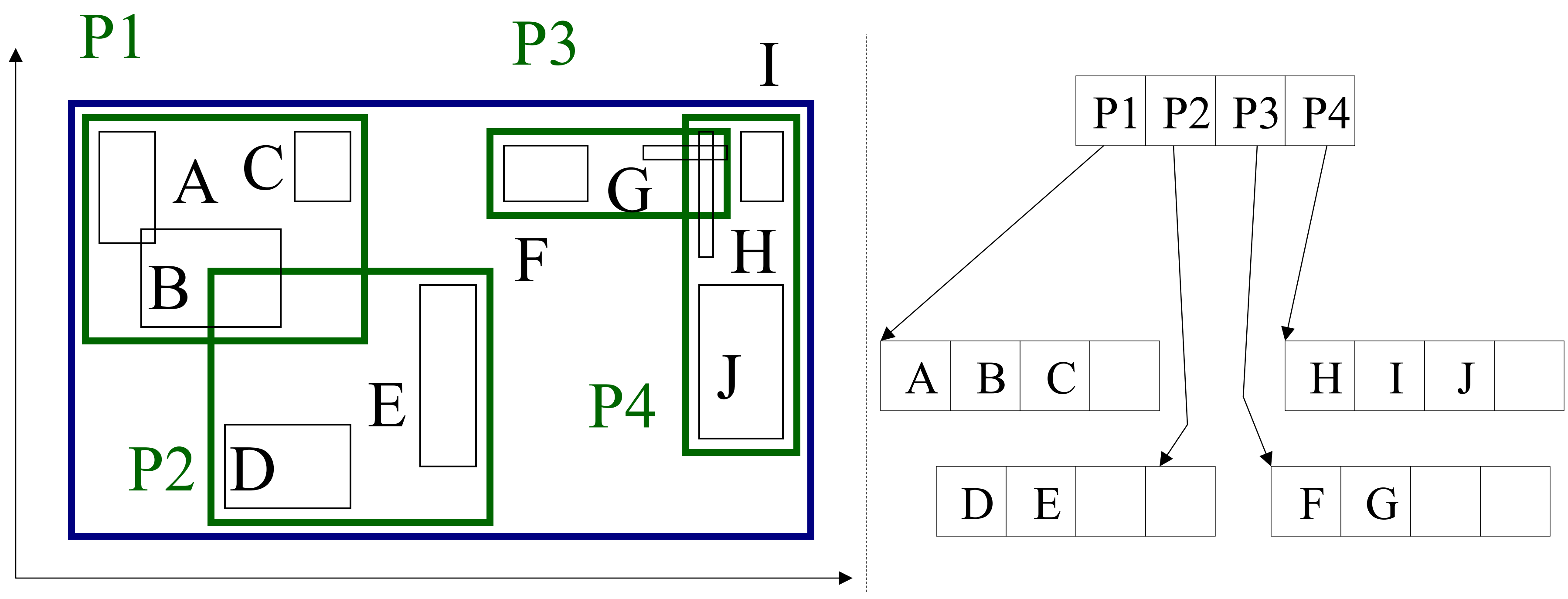
Example

- F=4



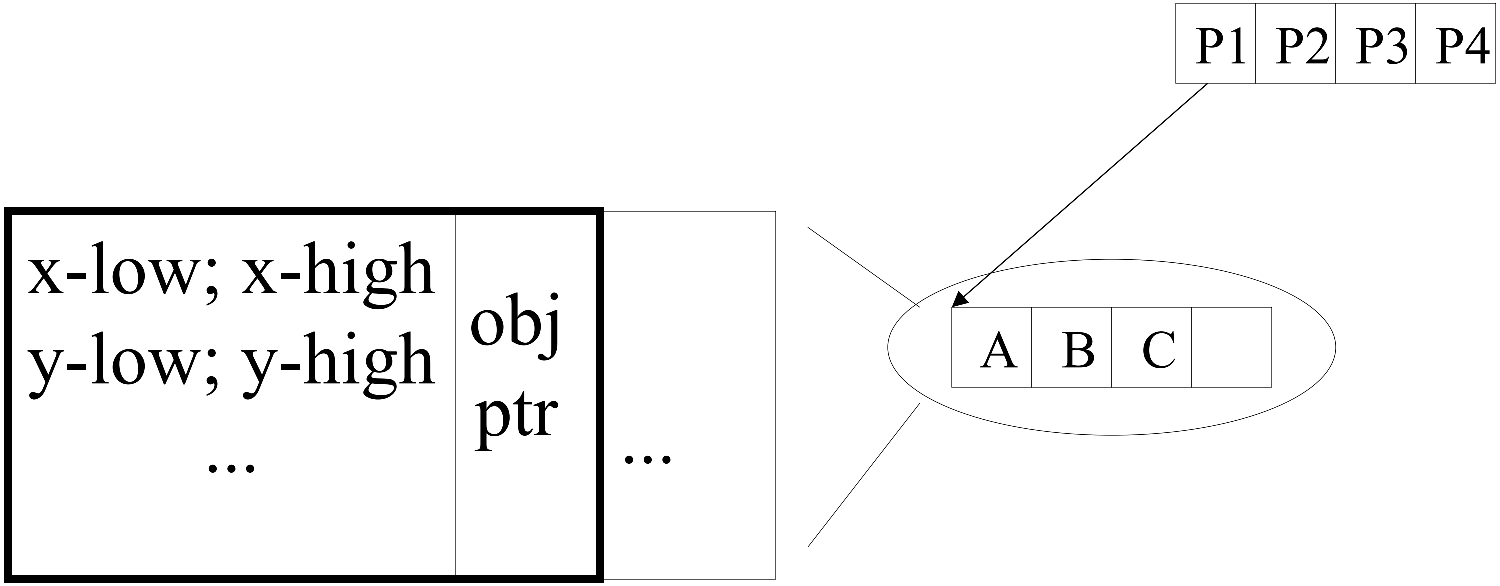
Example

- F=4



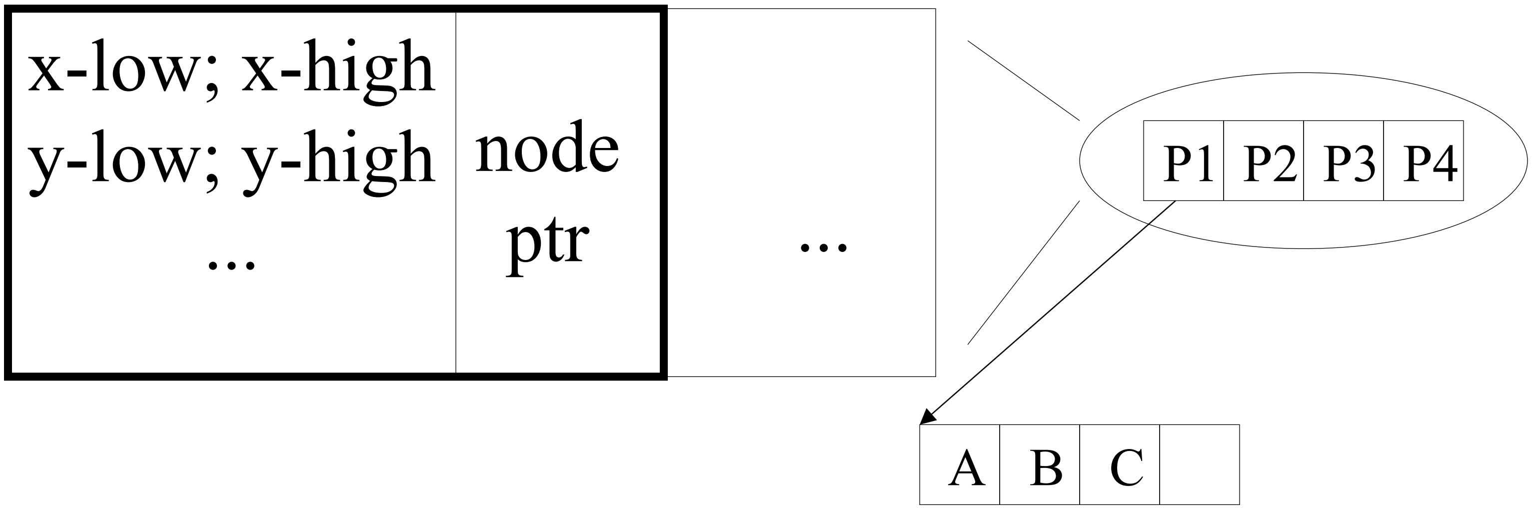
R-trees - format of nodes

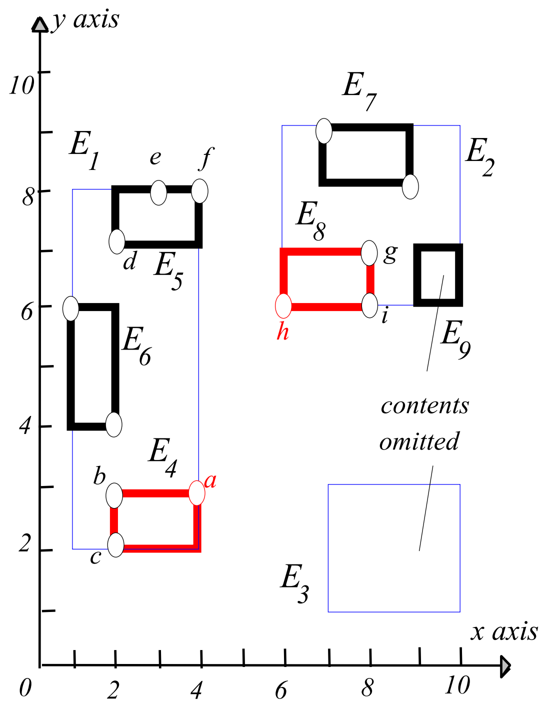
- {(MBR; obj_ptr)} for leaf nodes

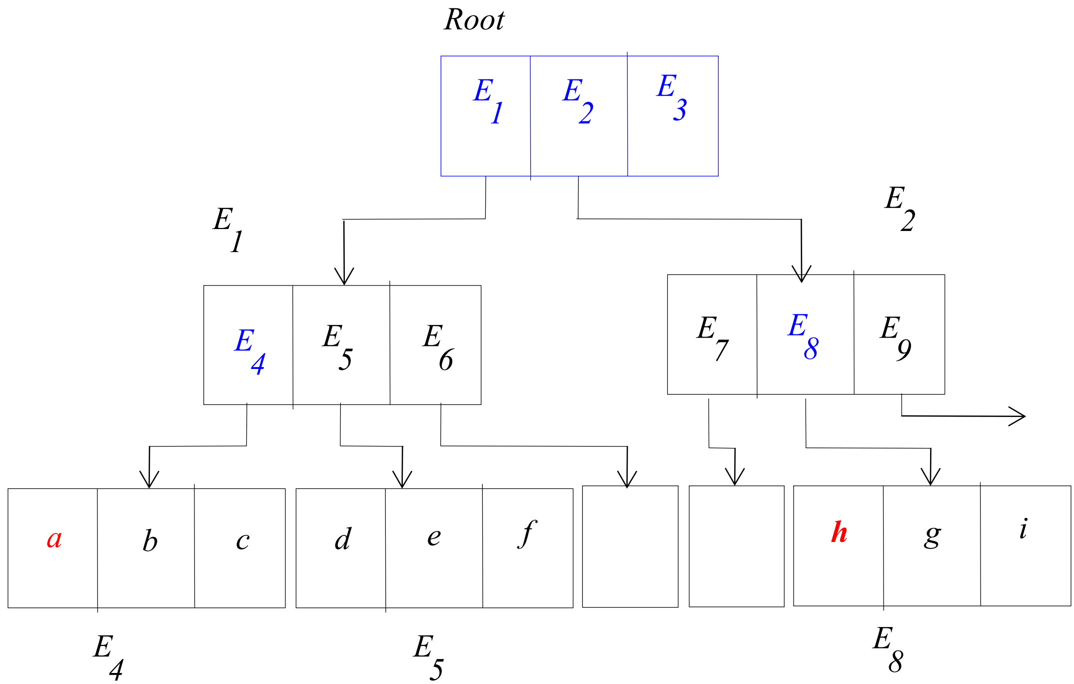


R-trees - format of nodes

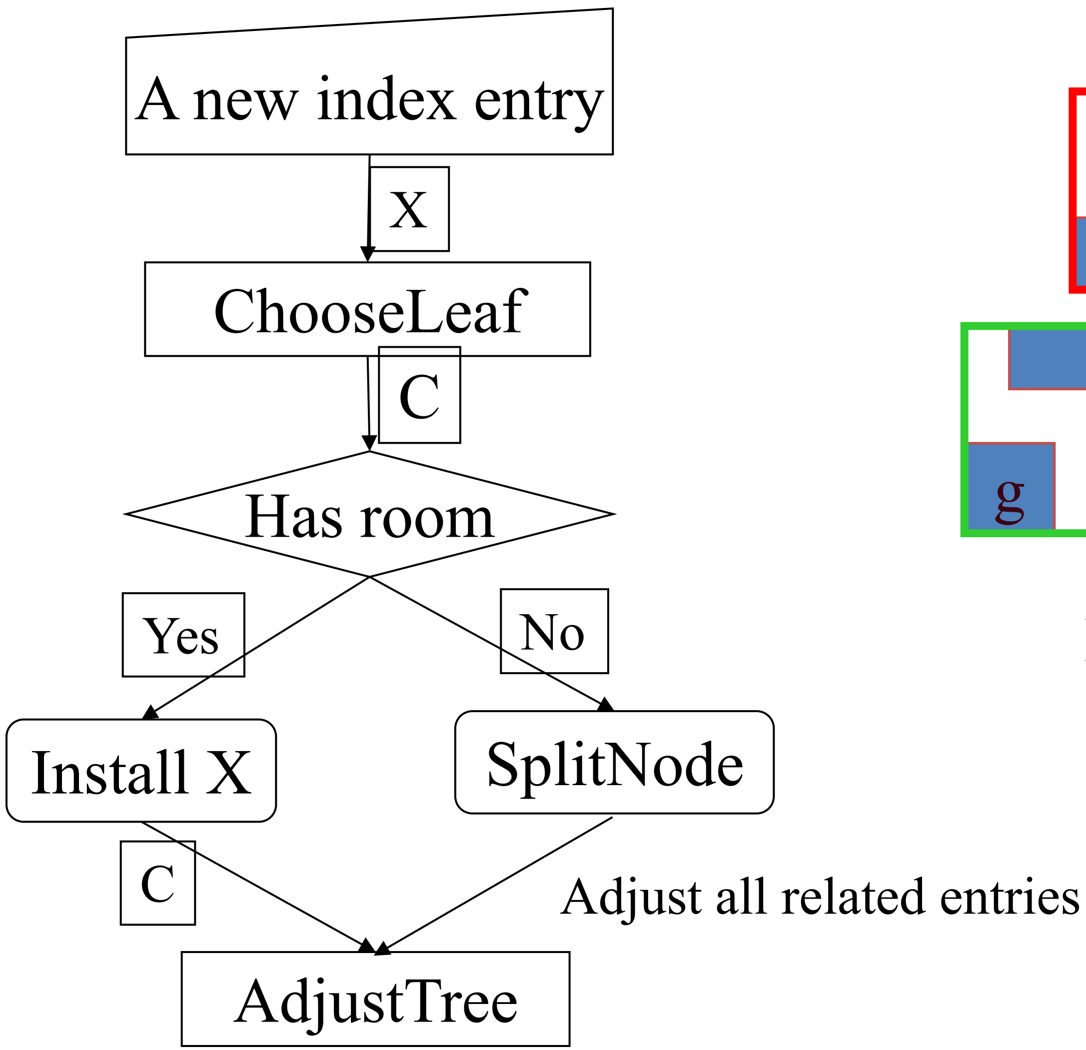
- {(MBR; node_ptr)} for non-leaf nodes

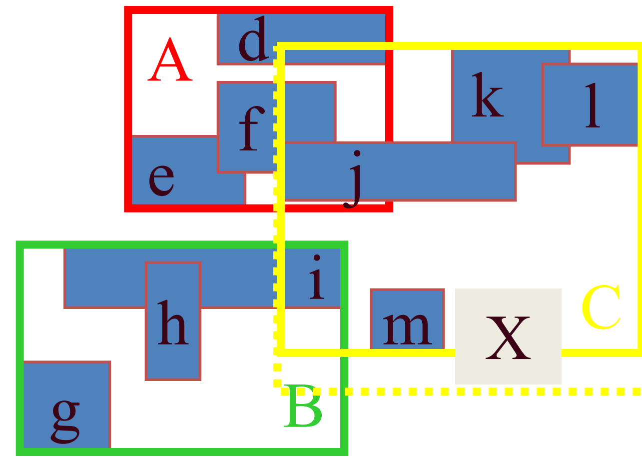



y axis
x axis
10
8
6
4
2
0
2
4
6
8
10
E1
E2
E3
E4
E5
E6
E7
E8
E9
a
b
c
d
e
f
g
h
i
contents
omitted


Root
E1
E2
E3
E1
E4
E5
E6
E2
E7
E8
E9
a
b
c
d
e
f
h
g
i
E4
E5
E8

Insertion Processes





Different variant:

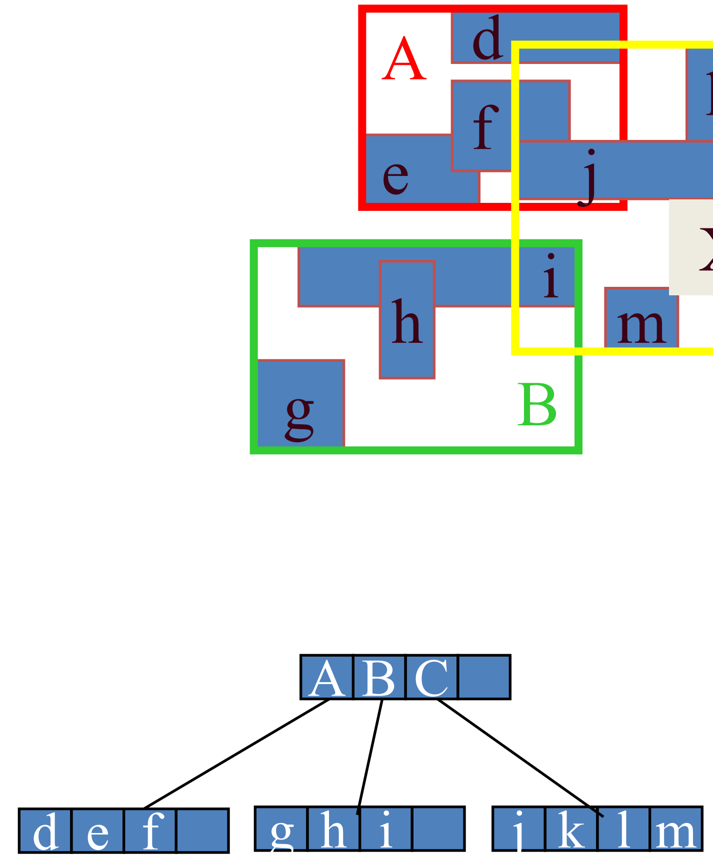
- Exhaustive
- Quadratic
- Linear
- Packed
- Hilbert Packed
- …etc.

USC Viterbi School of Engineering

Processes of Quadratic Spilt

(page 52 in Guttman's paper [1])

Pick first entry for each group
Run PickSeeds



Processes of Quadratic Spilt

(page 52 in Guttman's paper)

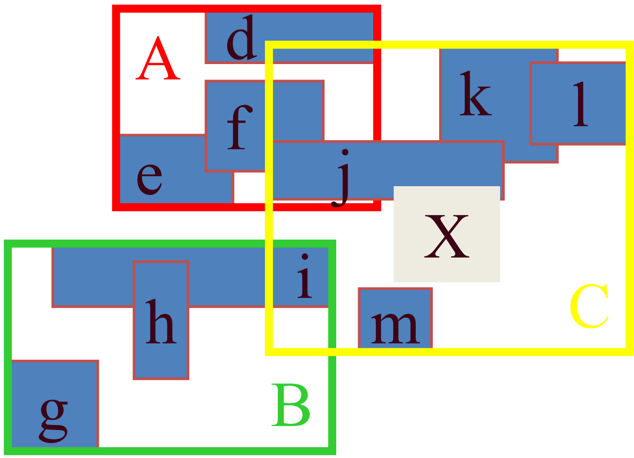
PickSeeds

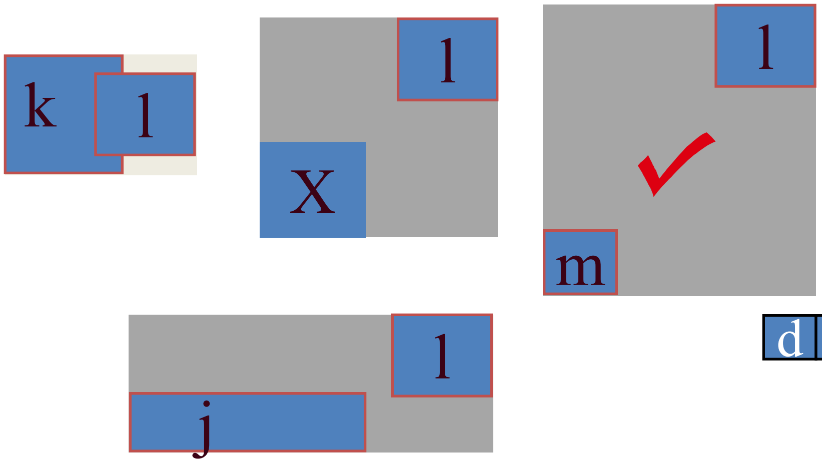
PS1 [Calculate inefficiency of grouping entries together]

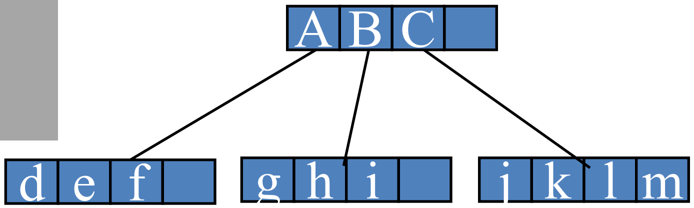
For each pair of E1 and E2, compose a rectangle R including E1 and E2

Calculate d = area(R) - area(E1) - area(E2)

PS2 [Choose the most wasteful pair]
Choose the pair with the largest d

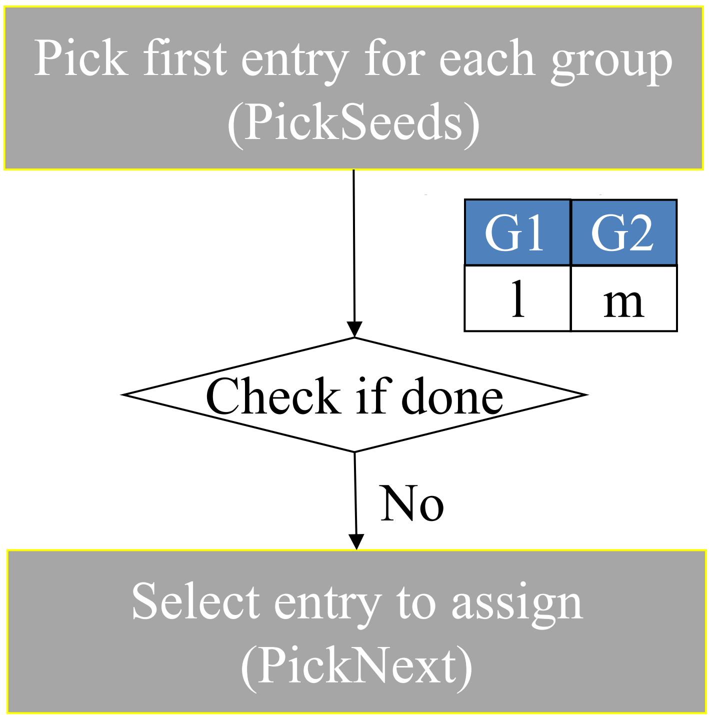




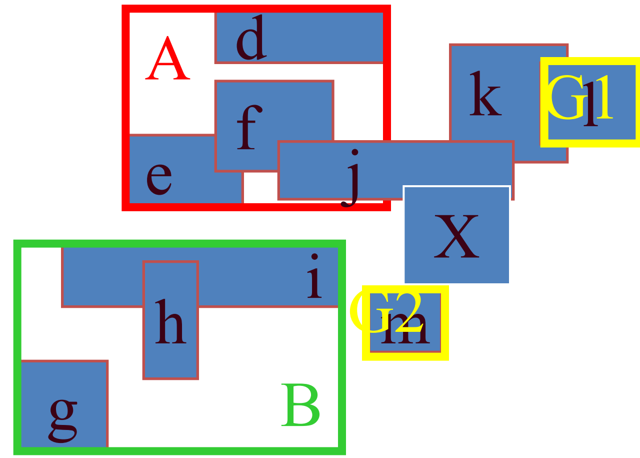


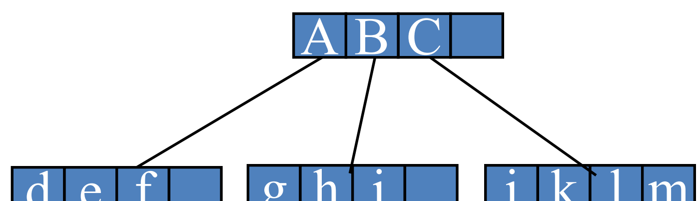
Processes of Quadratic Spilt

(page 52 in Guttman's paper)



G1	G2
l	m





USC Viterbi School of Engineering

Processes of Quadratic Spilt

(page 52 in Guttman's paper)

Pick first entry for each group (PickSeeds)

G1	G2
l	m

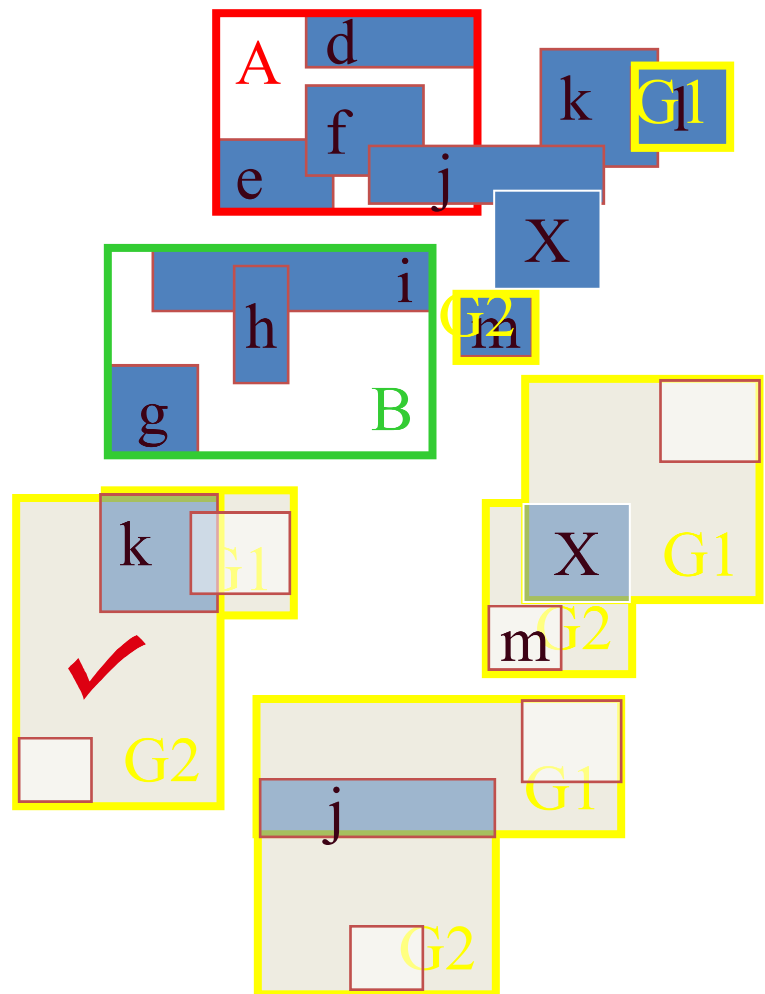
Check if done

No

PickNext

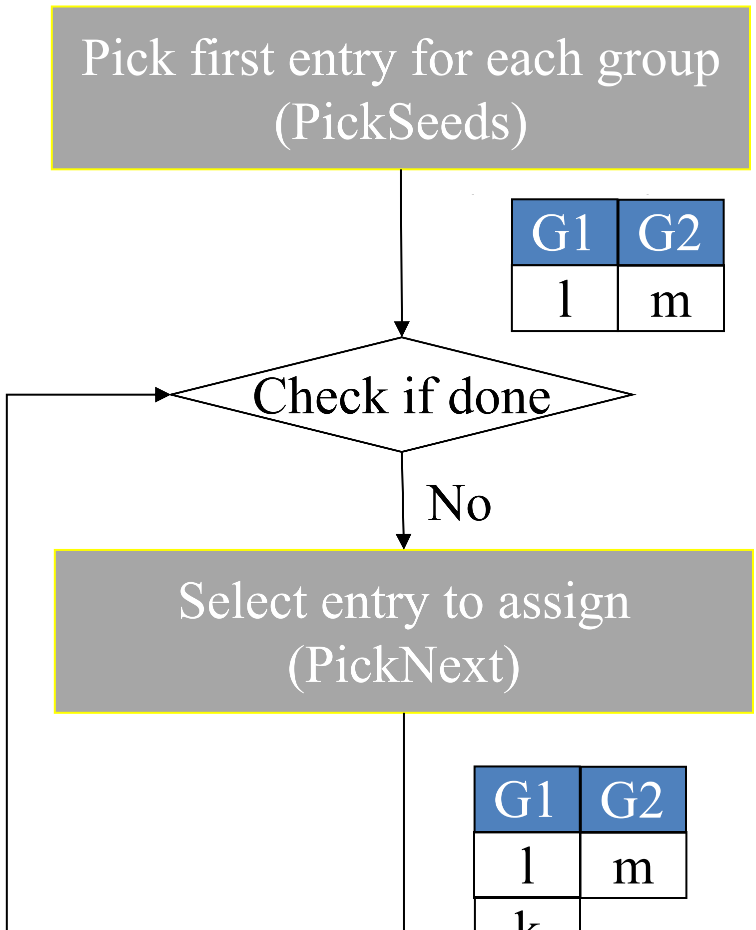
PN1 [Determine cost of putting each entry in each group] For each entry E
calculate d1 = the increased MBR area required for G1
calculate d2 = the increased MBR area required for G2

PN2 [Find entry with greatest preference for one group] Choose the entry with the maximum difference between d1 and d2



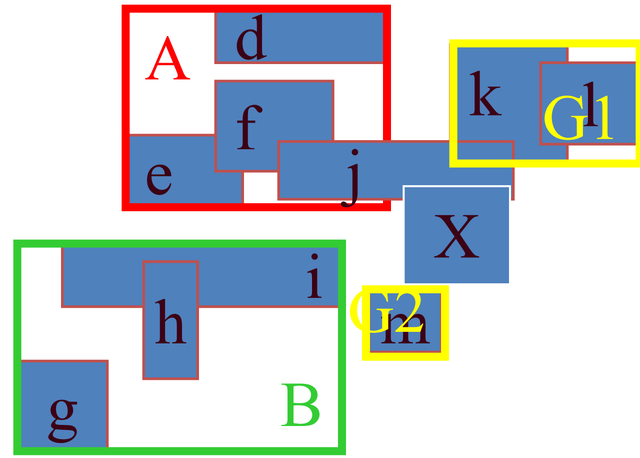
Processes of Quadratic Spilt

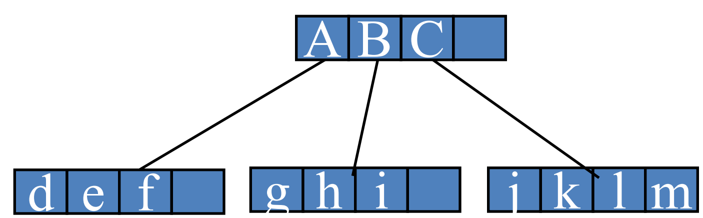
(page 52 in Guttman's paper)



G1	G2
l	m

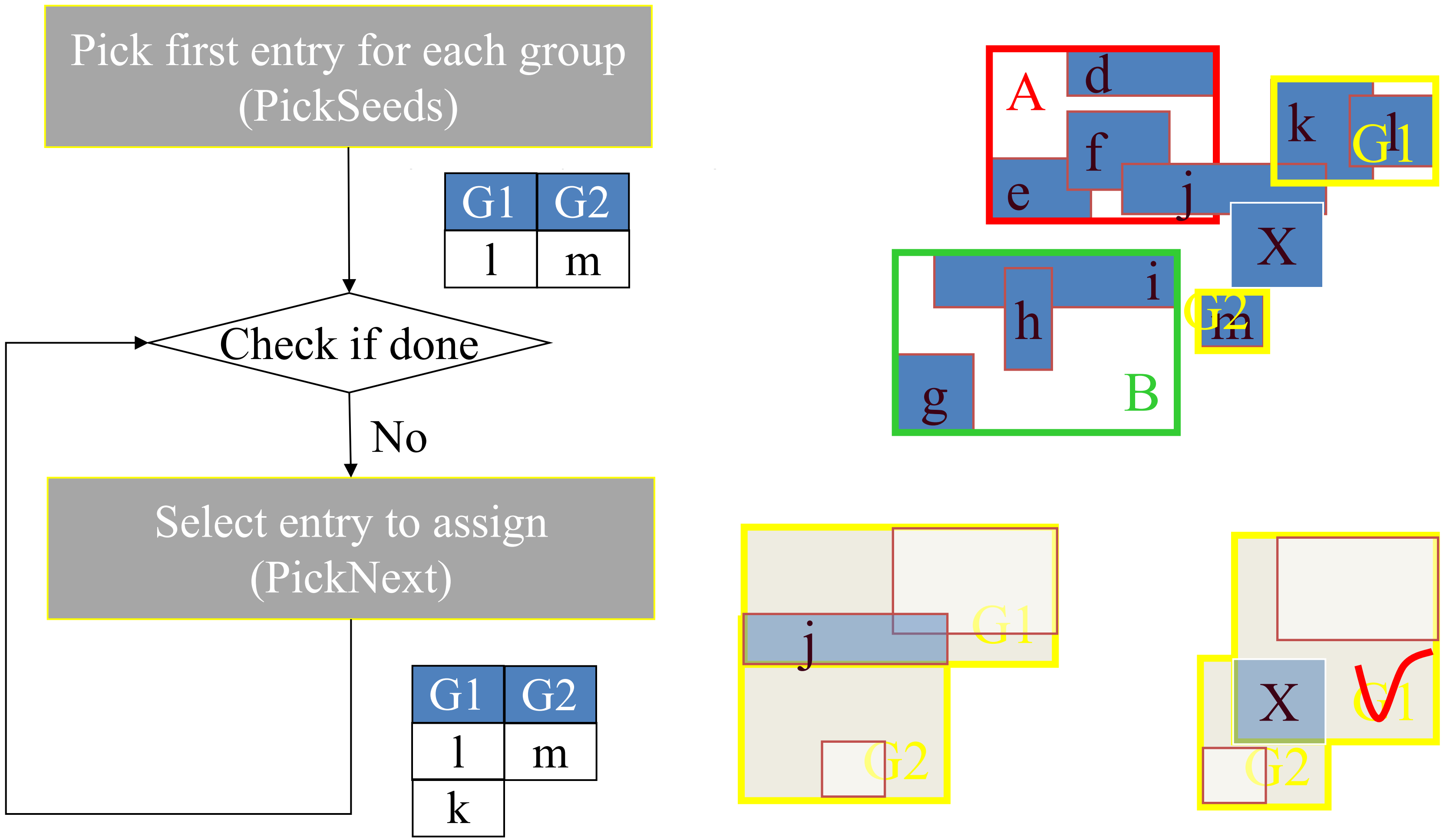
G1	G2
l	m
k	





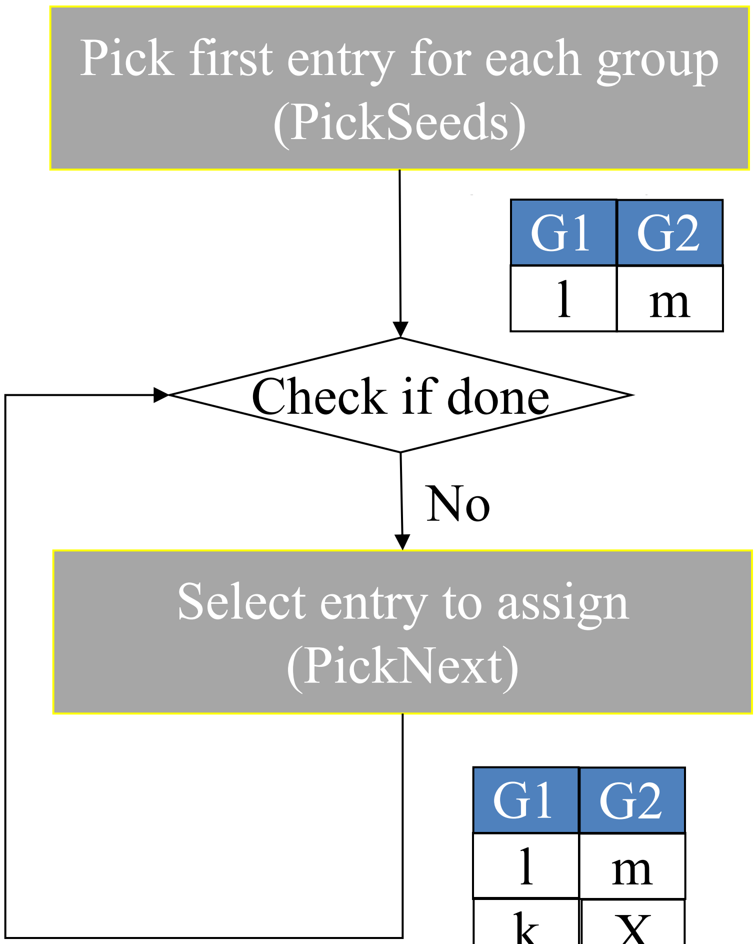
Processes of Quadratic Spilt

(page 52 in Guttman's paper)



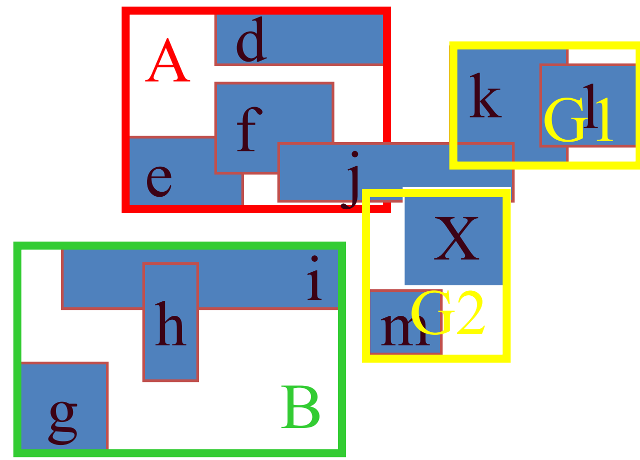
Processes of Quadratic Spilt

(page 52 in Guttman's paper)



G1	G2
l	m

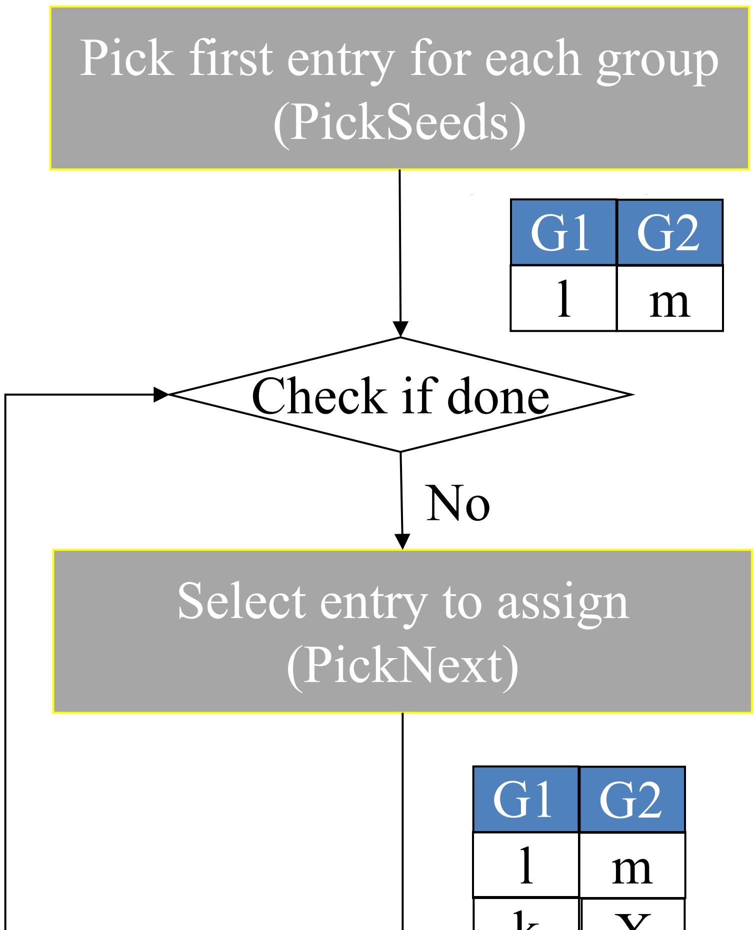
G1	G2
l	m
k	X



USC Viterbi School of Engineering

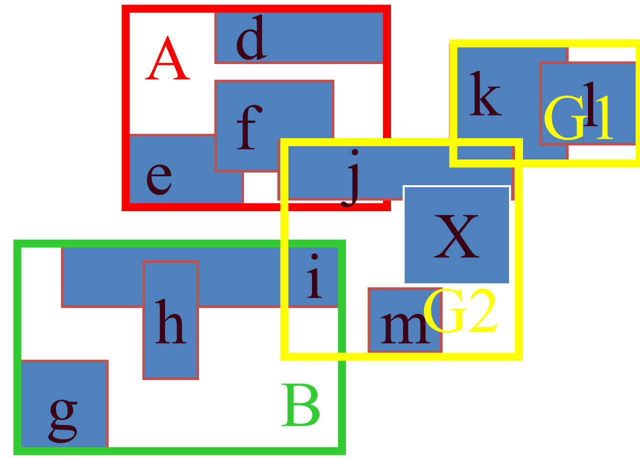
Processes of Quadratic Spilt

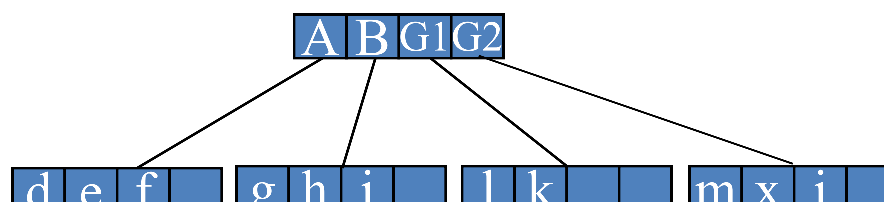
(page 52 in Guttman's paper)



G1	G2
l	m

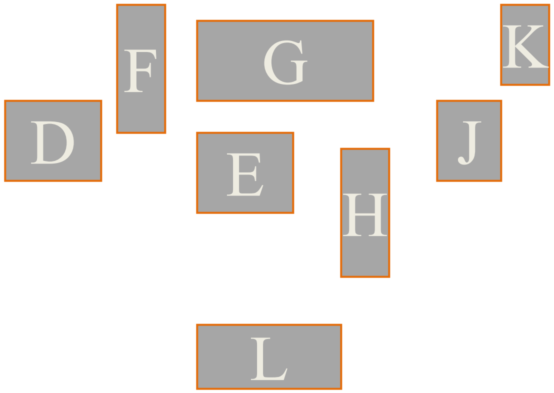
G1	G2
l	m
k	X
	j





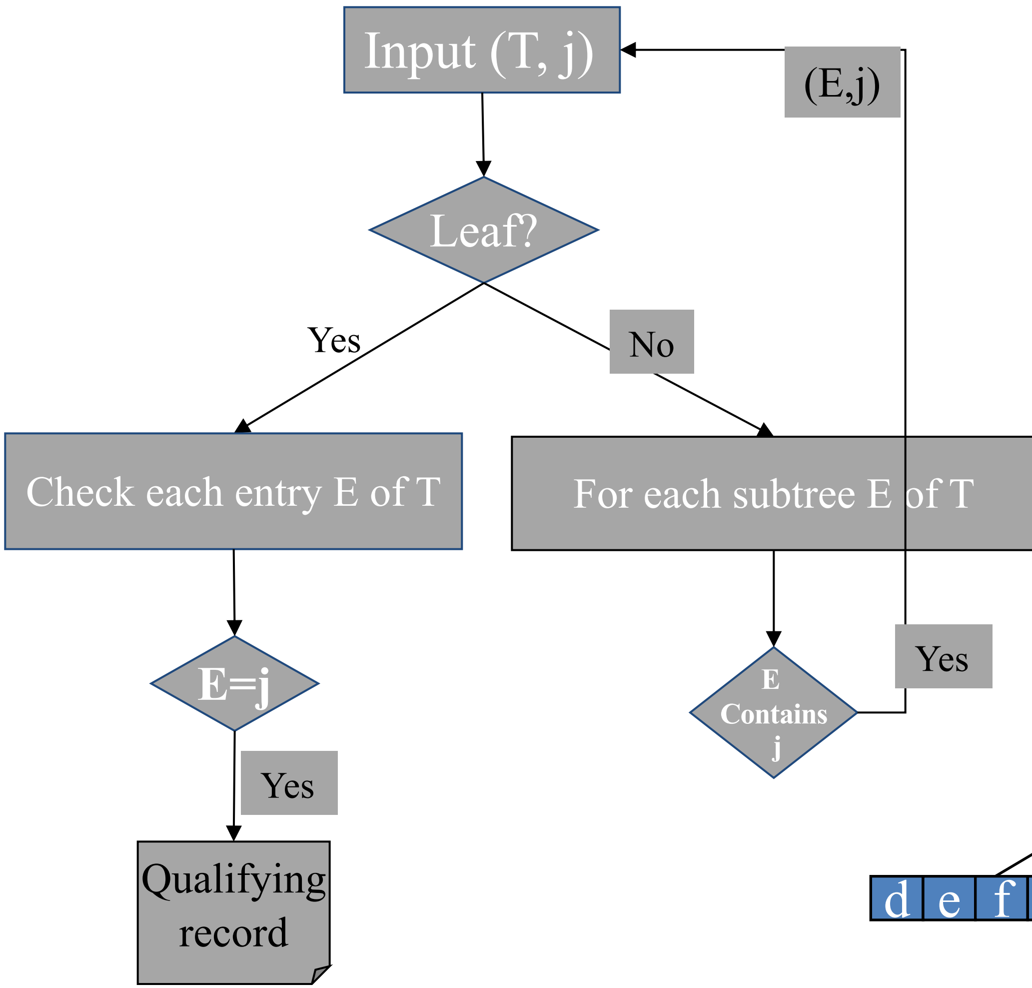
Excercise

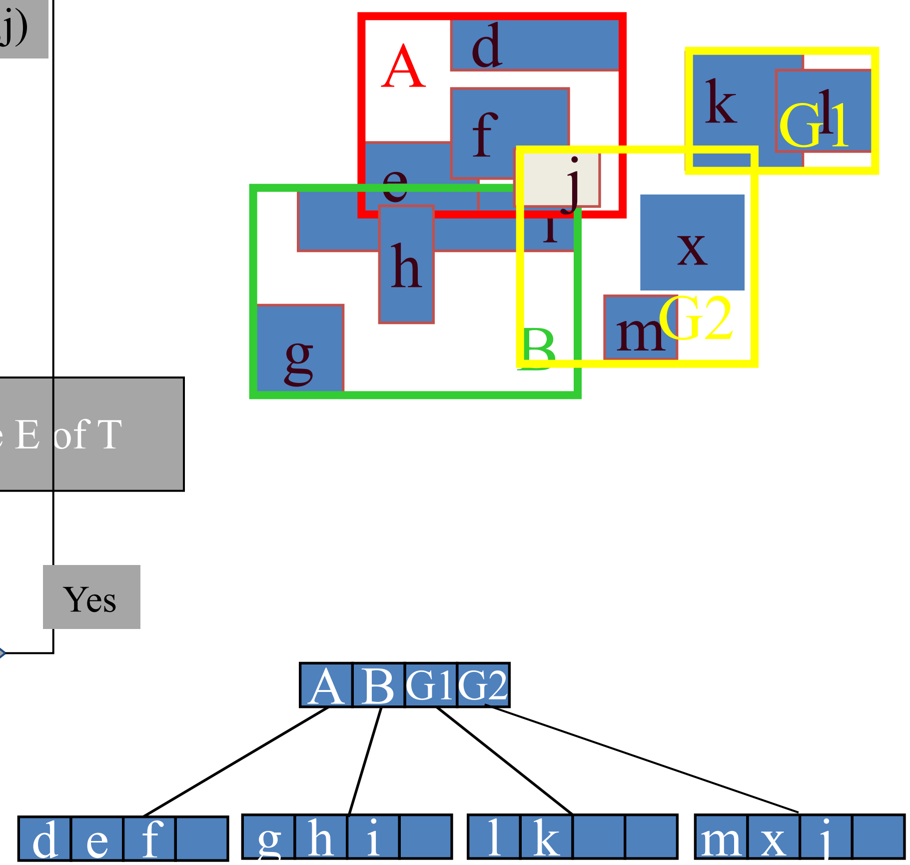
- (m,M)=(2,4)



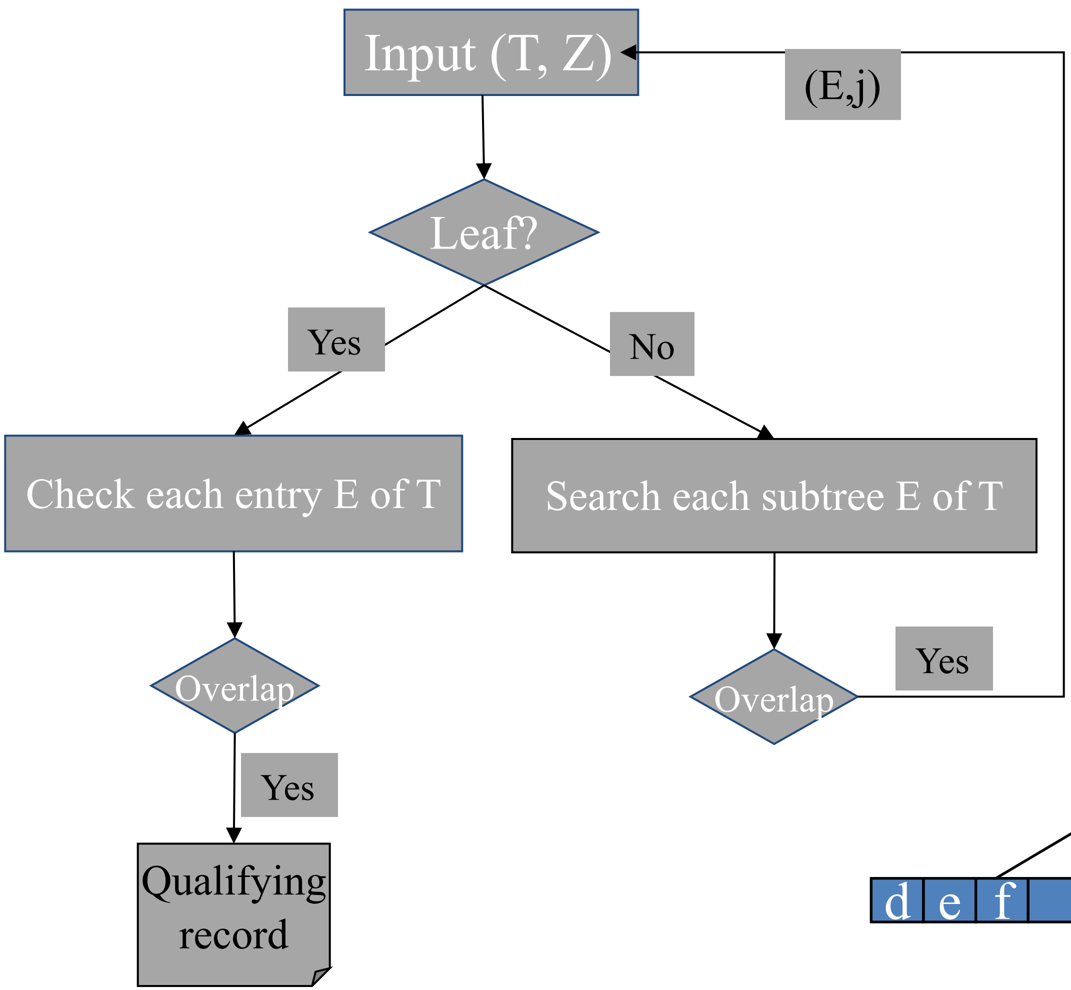
- Build a R-Tree for these spatial data
- Hint: You could use the Spatial index structures demo application step by step

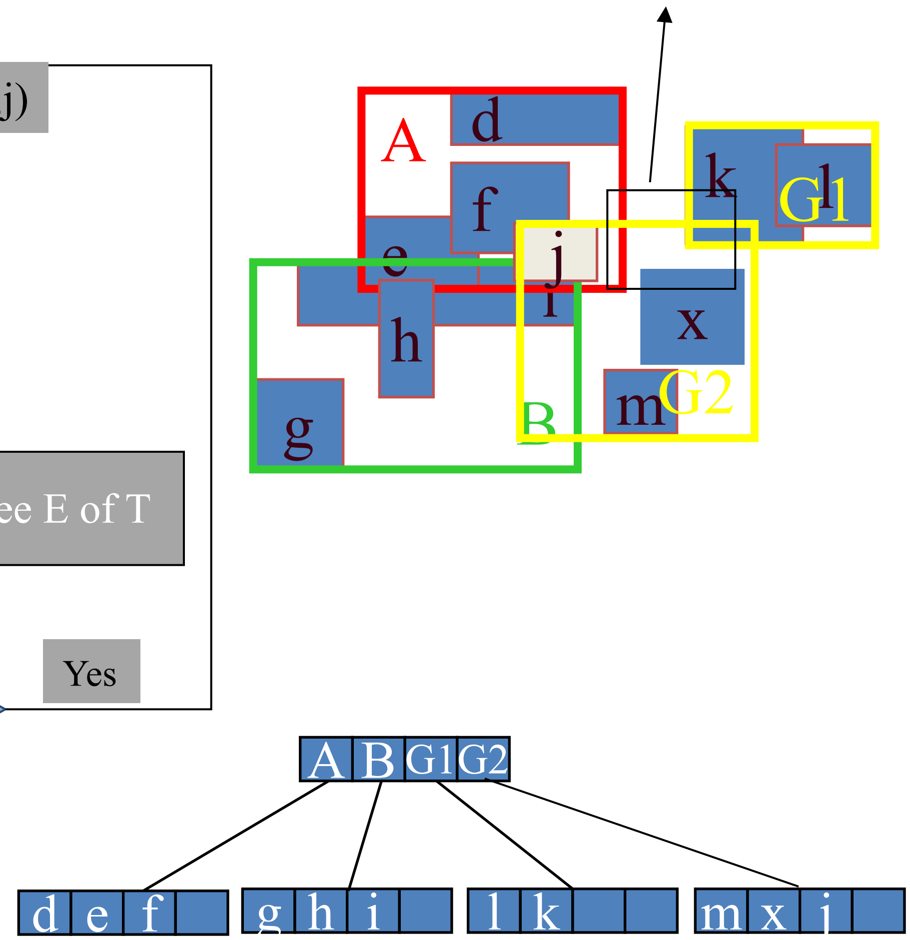
Search Object in R-Tree





Overlap query in R-Tree (find objects that overlap with Z)





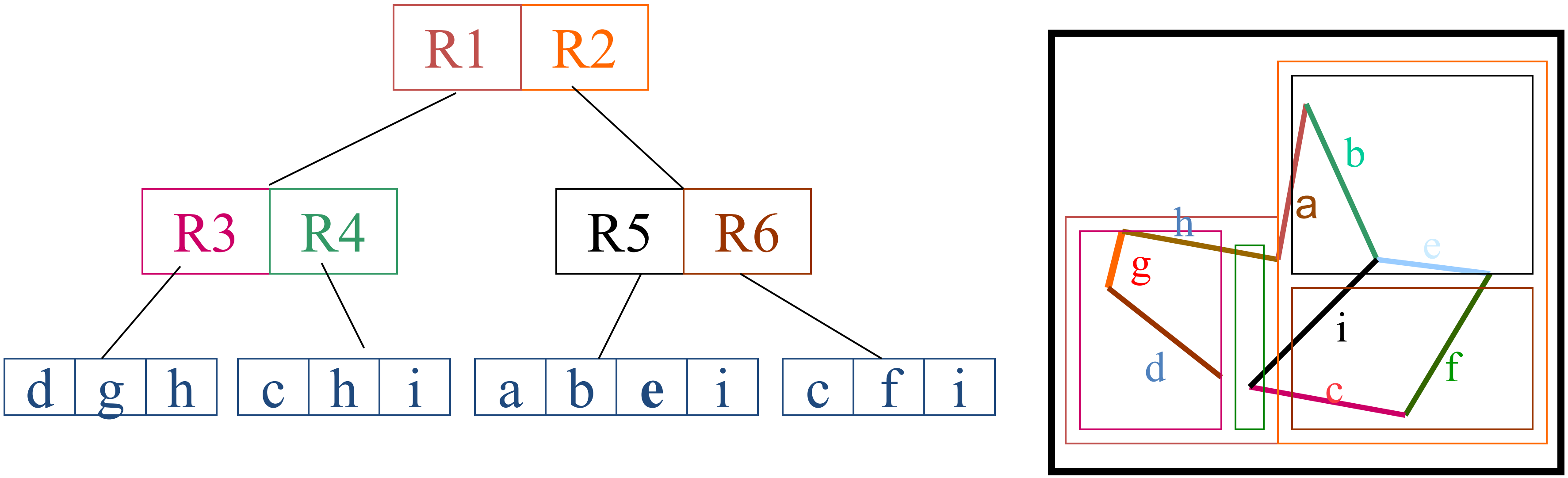
Main Drawbacks of R-Tree

- R-tree is not unique, rectangles depend on how objects are inserted and deleted from the tree.
- In order to search some object you might have to go through several rectangles or the whole database
  - Why?
  - Solution?

USC Viterbi School of Engineering

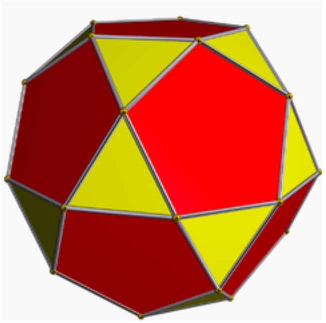
R+-Tree

- Overcome problems with R-Tree
- If node overlaps with several rectangles insert the node in all
- Decompose the space into disjoint cells



R^+-Tree Properties

- R+-tree and cell-trees used approach of discomposing space into cells
  - R+-trees deals with collection of objects bounded by rectangles
  - Cell tree deals with collection of objects bounded by convex polyhedron



- R+-trees is extension of k-d-B-tree
- Retrieval times are smaller
- When summing the objects, needs eliminate duplicates
- Again, it is data-dependent

R-tree

- The original R-tree tries to minimize the area of each enclosing rectangle in the index nodes.
- Is there any other property that can be optimized?

R*-tree → Yes!

USC Viterbi
School of Engineering

R*-tree

- Optimization Criteria:
  - (O1) Area covered by an index MBR
  - (O2) Overlap between index MBRs
  - (O3) Margin (perimeter) of an index rectangle
  - (O4) Storage utilization
- Sometimes it is impossible to optimize all the above criteria at the same time!

USC Viterbi School of Engineering

R*-tree

- ChooseSubtree:
  - If next node is a leaf node, choose the node using the following criteria:
    - Least overlap enlargement
    - Least area enlargement
    - Smaller area
  - Else
    - Least area enlargement
    - Smaller area

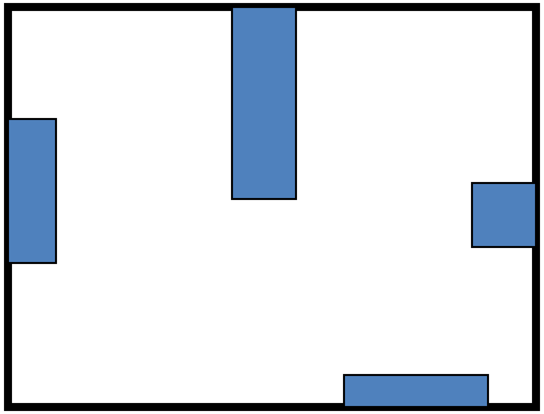
R*-tree

- SplitNode
  - Choose the axis to split
  - Choose the two groups along the chosen axis
- ChooseSplitAxis
  - Along each axis, sort rectangles and break them into two groups (M-2m+2 possible ways where one group contains at least m rectangles). Compute the sum S of all <u>margin-values (perimeters)</u> of each pair of groups. Choose the one that minimizes S
- ChooseSplitIndex
  - Along the chosen axis, choose the grouping that gives the minimum <u>overlap-value</u>

USC Viterbi School of Engineering

R*-tree

- Forced Reinsert:
  - defer splits, by forced-reinsert, i.e.: instead of splitting, temporarily delete some entries, shrink overflowing MBR, and re-insert those entries (hopefully they'll end up in an adjacent node so need for split)
- Which ones to re-insert?
- How many? A: 30%



USC Viterbi
School of Engineering

References

- Antonin Guttman, R-trees: a dynamic index structure for spatial searching, Proceedings of the 1984 ACM SIGMOD international conference on Management of data, June 18-21, 1984, Boston, Massachusetts
- Norbert Beckmann, et al. , The R*-tree: an efficient and robust access method for points and rectangles, SIGMOD 1990
- Roussopoulos et al. , The R+-Tree: A Dynamic Index for Multi-Dimensional Objects, VLDB 1987
- National Technical University of Athens , Theoretical Computer Science II: Advanced Data Structures