CSE6488: Mobile Computing Systems

LBS Query Processing Methods using R-tree

Sungwon Jung

Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Tel: +82-2-705-8930
Email : jungsung@sogang.ac.kr

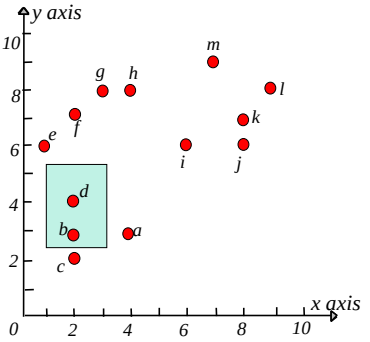
1

Content

- The R-tree
  - Range Query
  - Aggregation Query
- NN Query
- RNN Query
- Closest Pair Query
- Close Pair Query
- Skyline Query

2

R-Tree Motivation

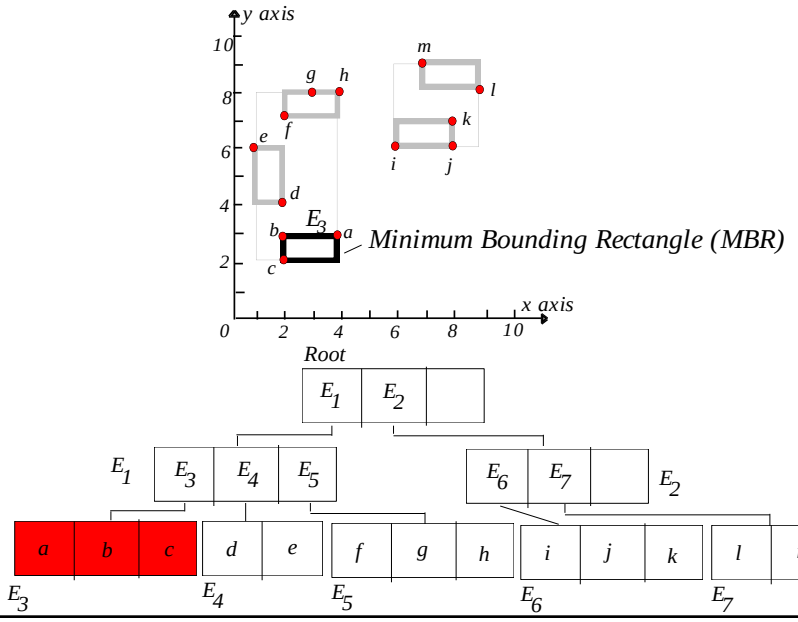


***Range query**: find the objects in a given range.*
E.g. find all hotels in Boston.

No index: scan through all objects. NOT EFFICIENT!

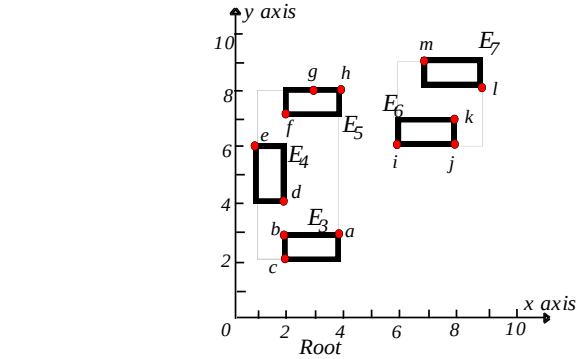
3

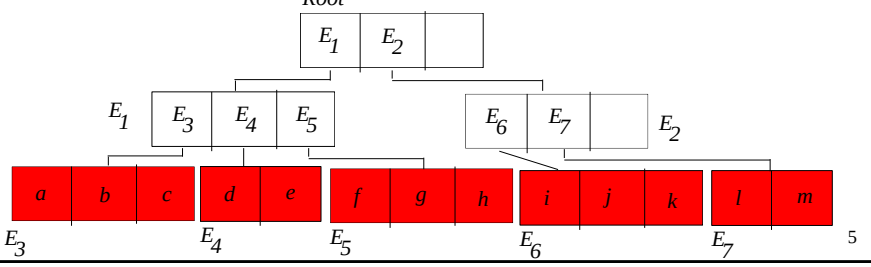
R-Tree: Clustering by Proximity



4

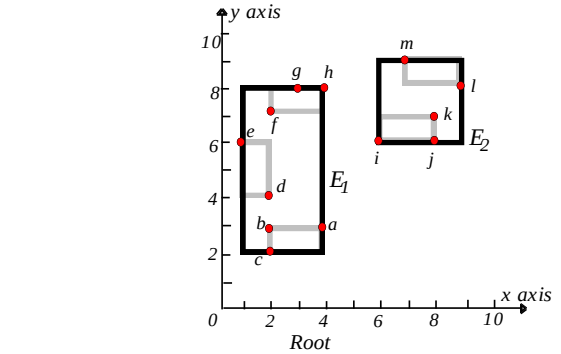
R-Tree

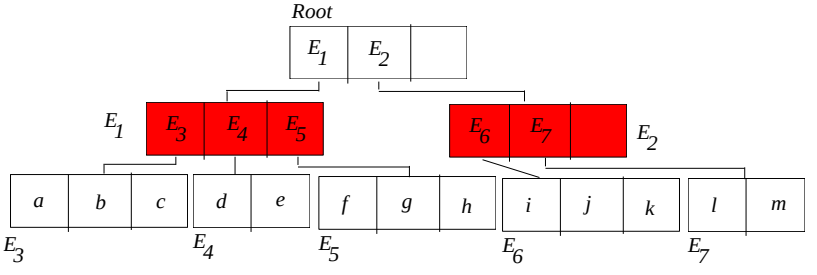




5

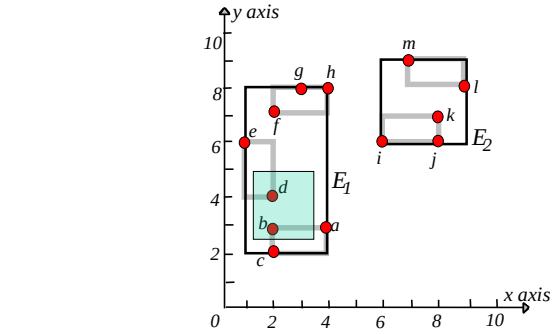
R-Tree

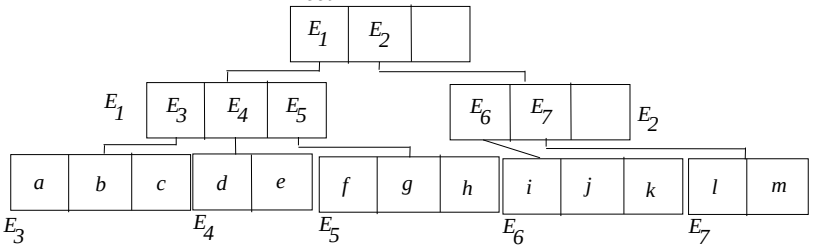




6

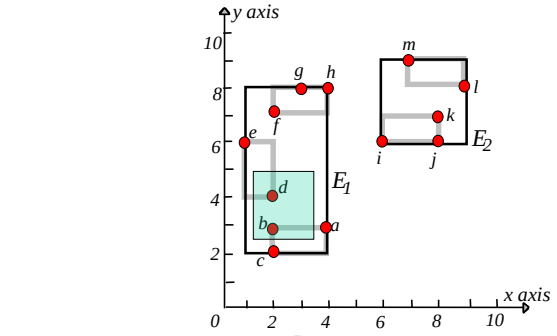
Range Query

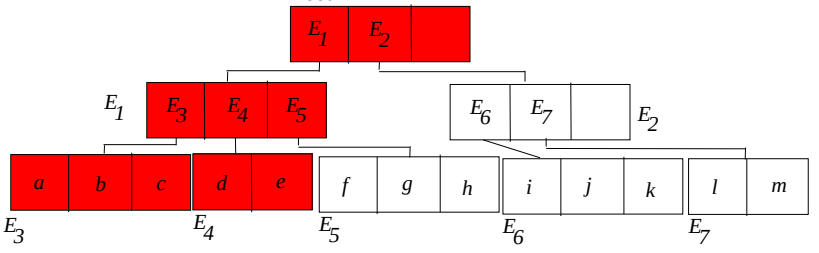




7

Range Query





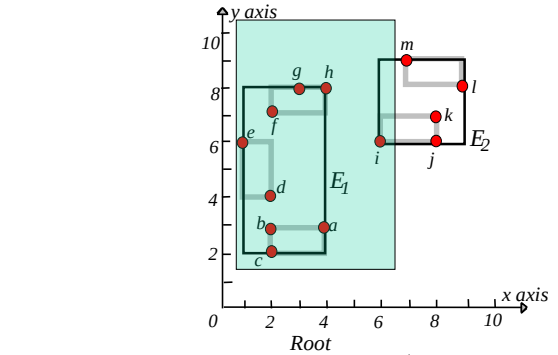
8

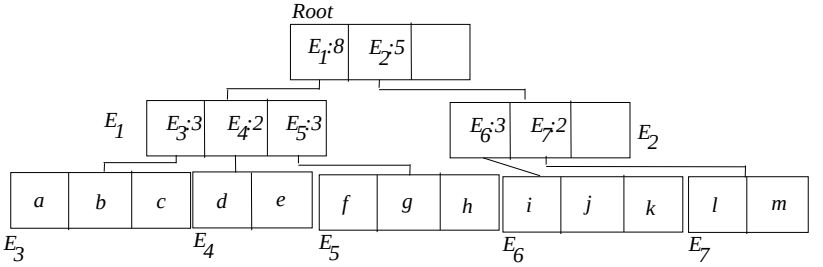
Aggregation Query

- *Given a range, find some aggregate value of objects in this range.*
- COUNT, SUM, AVG, MIN, MAX
- E.g. find the total number of hotels in Massachusetts.
- Straightforward approach: reduce to a range query.
- Better approach: along with each index entry, store aggregate of the sub-tree.

9

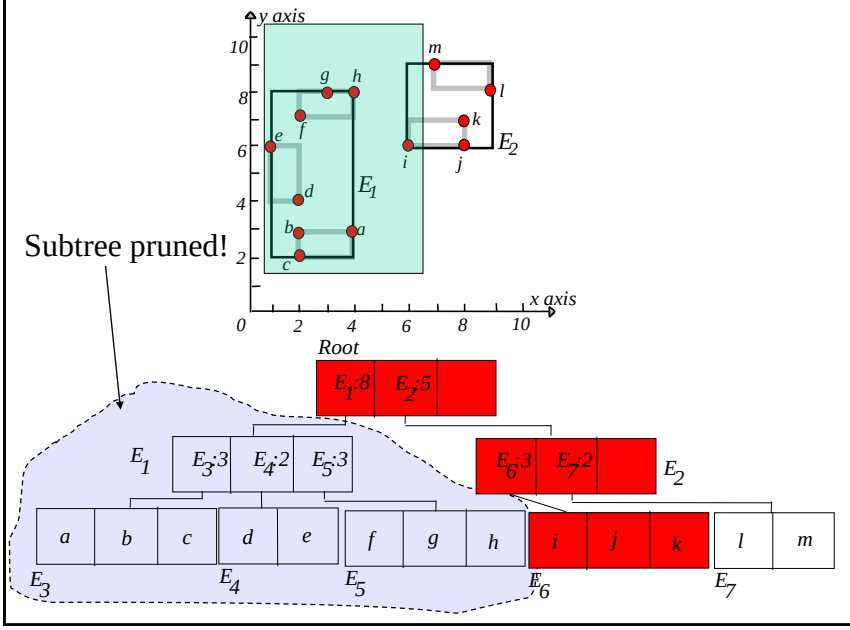
Aggregation Query





10

Aggregation Query

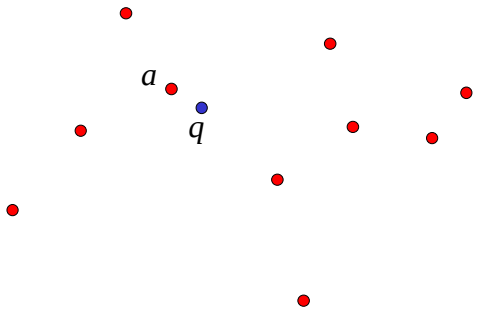


Content

- The R-tree
  - Range Query
  - Aggregation Query
- **NN Query**
- RNN Query
- Closest Pair Query
- Close Pair Query
- Skyline Query

Nearest Neighbor (NN) Query

- Given a query location *q*, find the nearest object.

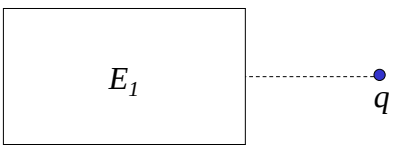


- E.g.: given a hotel, find its nearest bar.

13

A Useful Metric: MINDIST

- Minimum distance between *q* and an MBR.

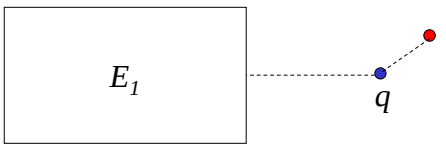


- It is an lower bound of $d(o, q)$ for every object *o* in *E1*.
- MINDIST$(o, q) = d(o, q)$.

14

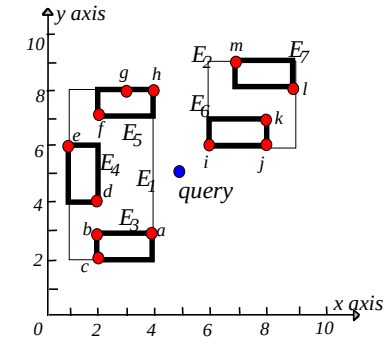
NN Basic Algorithm

- Keep a heap *H* of index entries and objects, ordered by MINDIST.
- Initially, *H* contains the root.
- While $H \neq \phi$
  - Extract the element with minimum MINDIST
  - If it is an index entry, insert its children into *H*.
  - If it is an object, return it as NN.
- End while

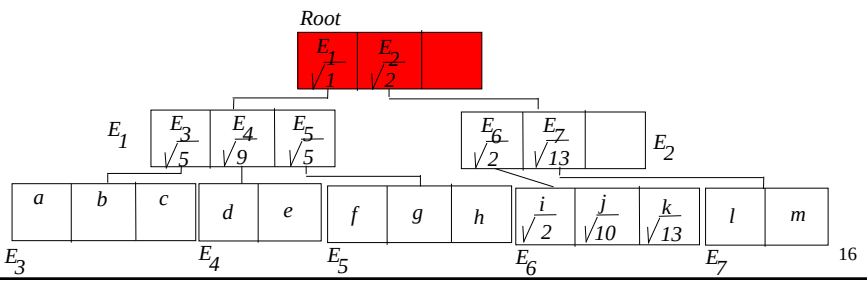


15

NN Query Example

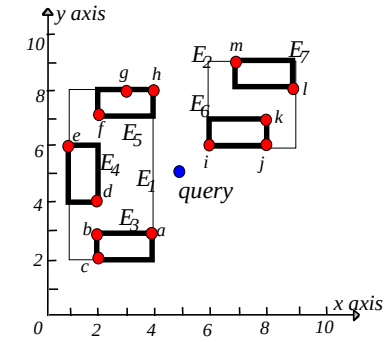


Action	Heap
Visit Root	$E_1\sqrt{1}$ $E_2\sqrt{2}$

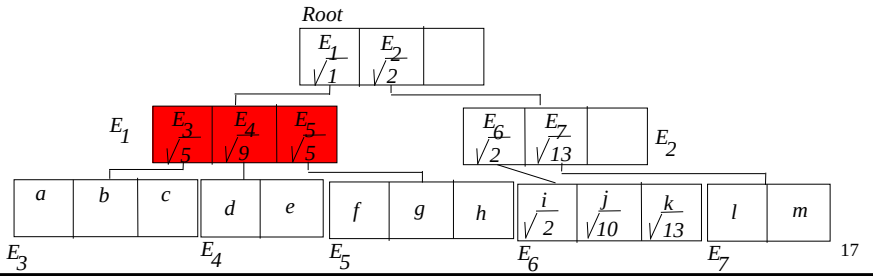


16

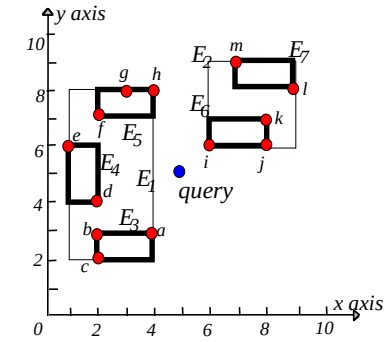
NN Query Example



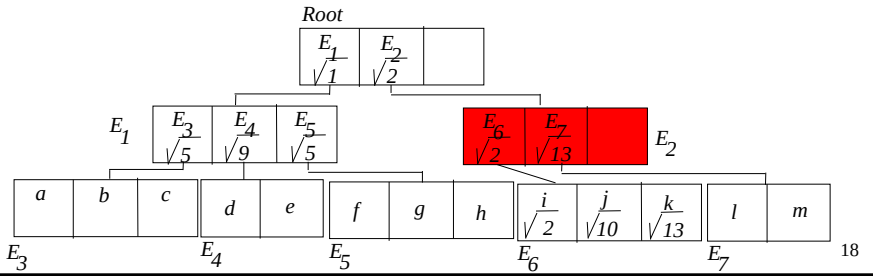
Action	Heap
Visit Root	$E_1\sqrt{1}$ $E_2\sqrt{2}$
follow E_1	$E_2\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$



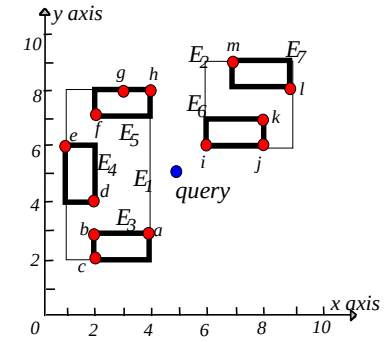
NN Query Example



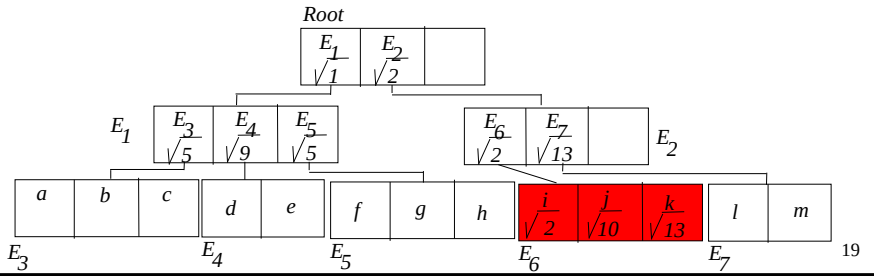
Action	Heap
Visit Root	$E_1\sqrt{1}$ $E_2\sqrt{2}$
follow E_1	$E_2\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$
follow E_2	$E_6\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$ $E_7\sqrt{13}$



NN Query Example

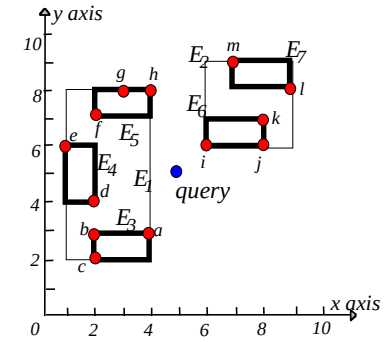


Action	Heap
Visit Root	$E_1\sqrt{1}$ $E_2\sqrt{2}$
follow E_1	$E_2\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$
follow E_2	$E_6\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$ $E_7\sqrt{13}$
follow E_6	$i\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$ $j\sqrt{10}$ $E_7\sqrt{13}$ $k\sqrt{13}$



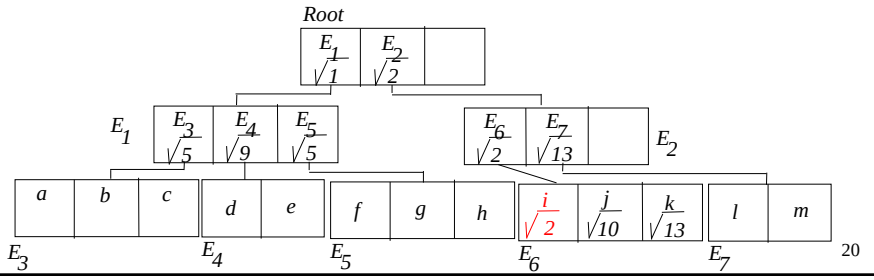
19

NN Query Example



Action	Heap
Visit Root	$E_1\sqrt{1}$ $E_2\sqrt{2}$
follow E_1	$E_2\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$
follow E_2	$E_6\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$ $E_7\sqrt{13}$
follow E_6	$i\sqrt{2}$ $E_3\sqrt{5}$ $E_5\sqrt{5}$ $E_4\sqrt{9}$ $j\sqrt{10}$ $E_7\sqrt{13}$ $k\sqrt{13}$

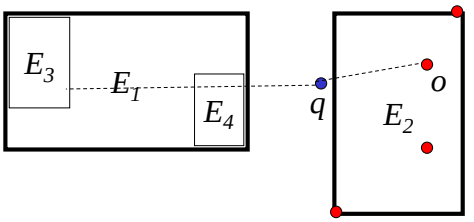
Report i and terminate



20

Pruning 1 in NN Query

- If we see an object *o*, prune every MBR whose MINDIST > *d*(*o*, *q*).

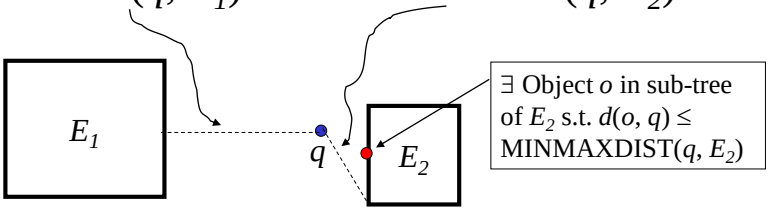


- Side notice: at most one object in *H*!

21

Pruning 2 using MINMAXDIST

- Prune even before we see an object!
- Prune E_1 if exists E_2 s.t. MINDIST(q, E_1) > MINMAXDIST(q, E_2).



- MINMAXDIST: compute max dist between q and each edge of E_2, then take min.

22

NN Full-Blown Algorithm

- Keep a heap H of index entries and objects, ordered by MINDIST.
- Initially, H contains the root.
- Set $\delta = +\infty$.
- While $H \neq \phi$
  - Extract the element e with minimum MINDIST.
  - If it is an object, return it as NN.
  - For every entry se in PAGE(e) whose MINDIST $\leq \delta$
    - Insert se into H.
    - Decrease δ to MINMAXDIST(q, se) if possible.
- End while

23

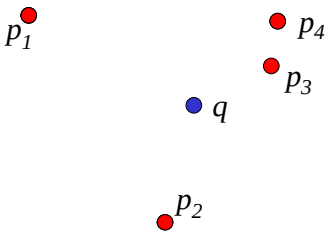
Content

- The R-tree
  - Range Query
  - Aggregation Query
- NN Query
- **RNN Query**
- Closest Pair Query
- Close Pair Query
- Skyline Query

24

Reverse NN: Definition

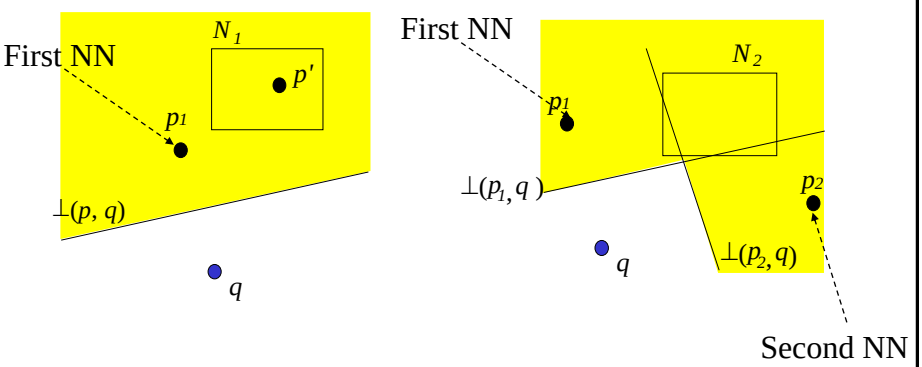
- Given a set of points, and a query location q.
- Find the points whose NN is q.



- RNN(q)=$\{p_1, p_2\}$, NN(q)= p_3.

25

Half-plane pruning



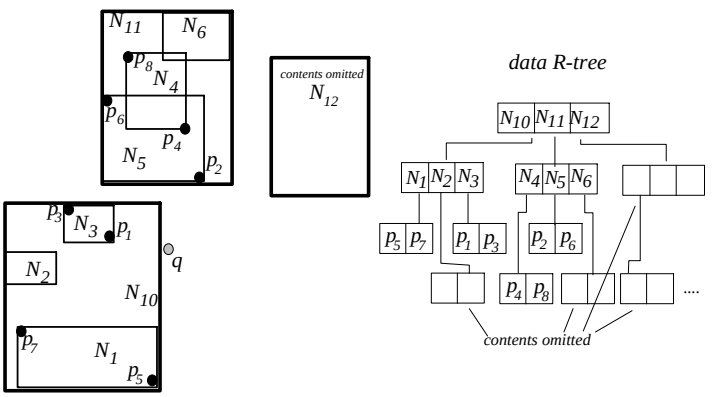
26

(TPL) Algorithm

- Two logical steps:
  - Filter step: Find the set S_{cnd} of candidate points
    - Find NN;
    - Prune space;
    - Find NN in unpruned space;
    - …
    - Till no more object left.
  - Refinement step: eliminate false positives
    - For each point p in S_{cnd}, check whether its NN is not q.
- The two steps are combined in a single tree traversal, by keeping all pruned MBRs/objects in S_{rfn}.

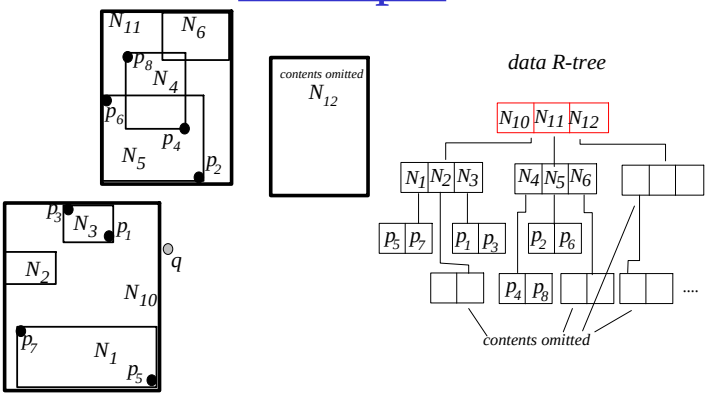
27

Example



28

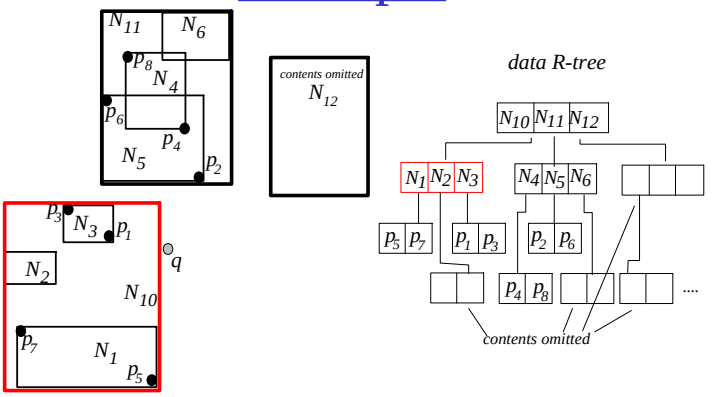
Example



Action	**Heap**	***S*_{*cnd*}**	***S*_{*rfn*}**
visit root	$\{N_{10},N_{11},N_{12}\}$	$\varnothing$	$\varnothing$

29

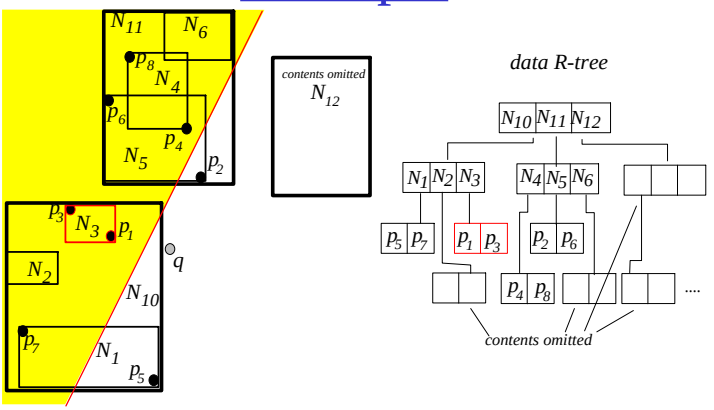
Example



Action	**Heap**	***S*_{*cnd*}**	***S*_{*rfn*}**
visitN_{10}	$\{N_3,N_{11},N_2,N_1,N_{12}\}$	$\varnothing$	$\varnothing$

30

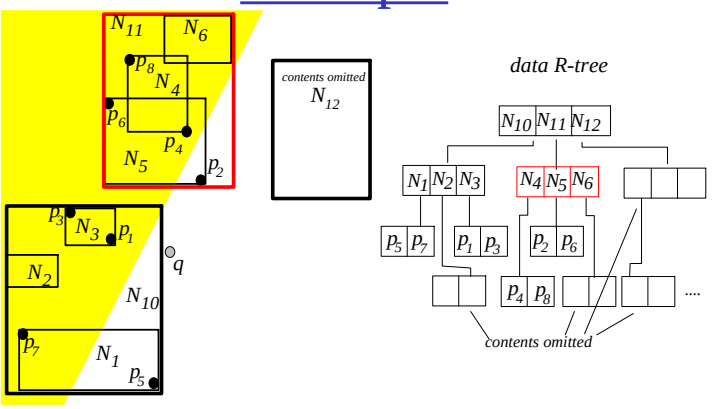
Example



Action	**Heap**	S_{cnd}	S_{rfn}
visitN_3	$\{N_{11},N_2,N_1,N_{12}\}$	$\{p_1\}$	$\{p_3\}$

31

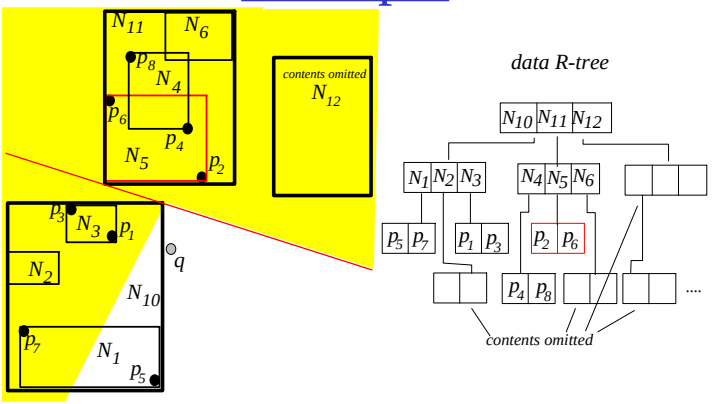
Example



Action	**Heap**	S_{cnd}	S_{rfn}
visitN_{11}	$\{N_5,N_2,N_1,N_{12}\}$	$\{p_1\}$	$\{p_3,N_4,N_6\}$

32

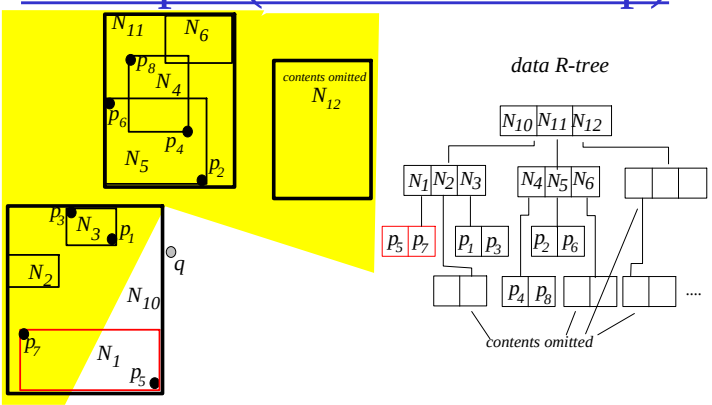
Example



Action	Heap	S_{cnd}	S_{rfn}
visit N_5	$\{N_2,N_1,N_{12}\}$	$\{p_1,p_2\}$	$\{p_3,N_4,N_6,p_6\}$

33

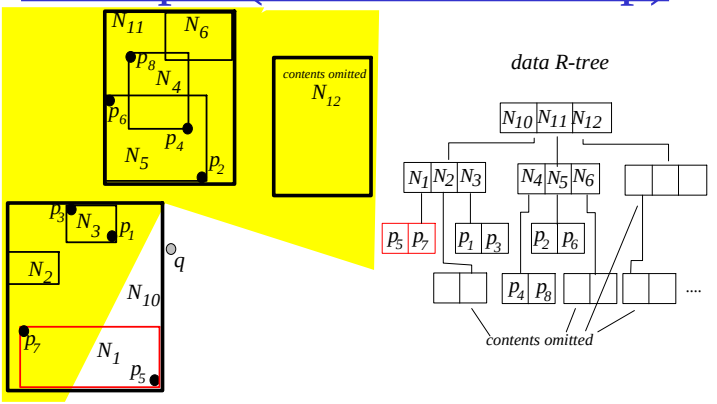
Example (end of filter step)



Action	Heap	S_{cnd}	S_{rfn}
visit N_1	$\{N_{12}\}$	$\{p_1,p_2,p_5\}$	$\{p_3,N_4,N_6,p_6,N_2,p_7\}$

34

Example (end of filter step)

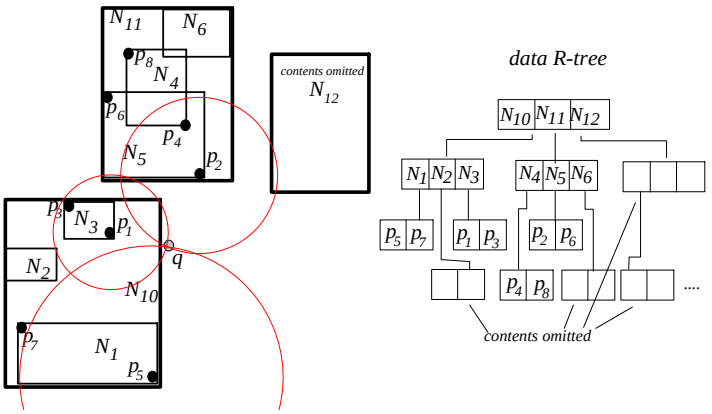


Action	Heap	S_{cnd}	S_{rfn}
	$\varnothing$	$\{p_1,p_2,p_5\}$	$\{p_3,N_4,N_6,p_6,N_2,p_7,N_{12}\}$

35

Example (refinement)

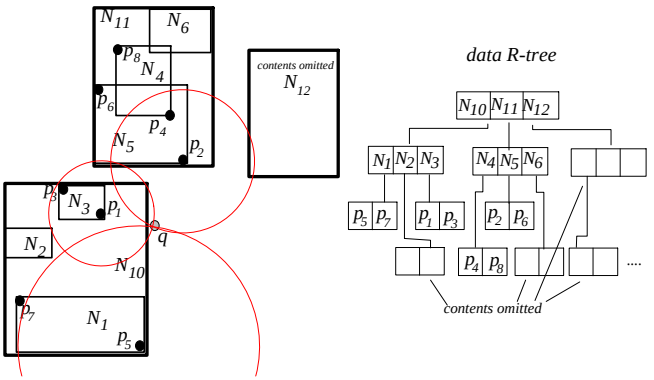
Action	S_{cnd}	S_{rfn}
	$\{p_1,p_2,p_5\}$	$\{p_3, p_6, p_7, N_4,N_6, N_2, N_{12}\}$
invalidate p_1	~~$\{p_1,p_2,p_5\}$~~	$\{N_4,N_6, N_2, N_{12}\}$
remove N_6, N_2	~~$\{p_1,p_2,p_5\}$~~	$\{N_4, N_{12}\}$



36

Example (refinement)

Action	S_{cnd}	S_{rfn}
	2 $\{p_1,p_2,p_5\}$	$\{N_4, N_{12}\}$
access N_4	$\{p_1,p_2,p_5\}$	$\{p_4, p_8, N_{12}\}$
invalidate p_2	$\{p_1,p_2,p_5\}$	$\{N_{12}\}$
remove N_{12}	$\{p_1,p_2,p_5\}$	$\varnothing$



37

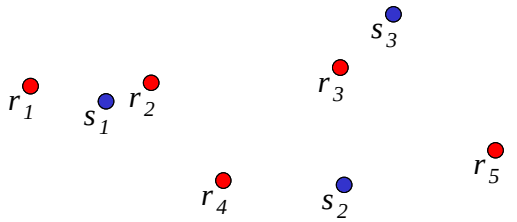
Content

- The R-tree
  - Range Query
  - Aggregation Query
- NN Query
- RNN Query
- **Closest Pair Query**
- Close Pair Query
- Skyline Query

38

Closest Pair (CP) Query

- Given two sets of objects *R* and *S*,
- Find the pair of objects ($r \in R$, $s \in S$) with minimum distance.

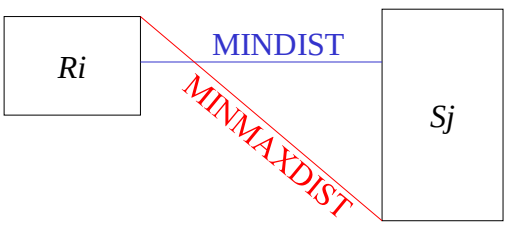


- CP = (r_2, s_1)
- E.g. find the closest pair of hotel-bar.

39

CP Solution Idea

- Assume *R* and *S* are indexed by R-trees with same height.
- Similar to the NN query algorithm.
- MINDIST, MINMAXDIST for a pair of MBRs:



40

CP Basic Algorithm

- Keep a heap H of pairs of index entries and pairs of objects, ordered by MINDIST.
- Initially, H contains the pair of roots.
- While $H \neq \phi$
  - Extract the pair (e_R, e_S) with minimum MINDIST.
  - If it is a pair of objects, return it as CP.
  - For every entry se_R in PAGE(e_R) and every entry se_S in PAGE(e_S)
    - Insert(e_R, e_S) into H.
- End while

41

CP Full-Blown Algorithm

- Keep a priority queue H of pairs of index entries and pairs of objects, ordered by MINDIST.
- Initially, H contains the pair of roots.
- Set $\delta = +\infty$.
- While $H \neq \phi$
  - Extract the pair (e_R, e_S) with minimum MINDIST.
  - If it is a pair of objects, return it as CP.
  - For every entry se_R in PAGE(e_R) and every entry se_S in PAGE(e_S) whose MINDIST $\leq \delta$
    - Insert(se_R, se_S) into H.
    - Decrease δ to MINMAXDIST(se_R, se_S) if possible.
- End while

42

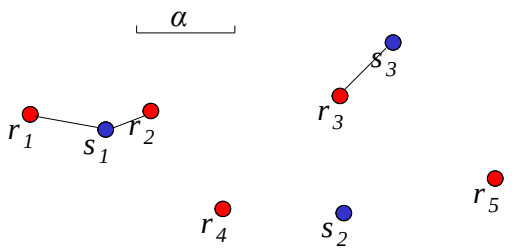
Content

- The R-tree
  - Range Query
  - Aggregation Query
- NN Query
- RNN Query
- Closest Pair Query
- **Close Pair Query**
- Skyline Query

43

Close Pair Query

- Given two sets of objects R and S, plus a threshold α,
- Find every pair of objects ($r \in R$, $s \in S$) with distance $< \alpha$.



- Close pairs = (r_1, s_1), (r_2, s_1), and (r_3, s_3).

44

Close Pair Solution Idea

- Observation: if $d(r, s) < \alpha$, $\forall mbr_R, mbr_S$ that contain r and s, respectively, we have: MINDIST(mbr_R, mbr_S) $< \alpha$.
- Solution idea:
  - start with the pair of root nodes,
  - Join pairs of index entries whose MINDIST $< \alpha$,
  - Till we reach leaf level.

45

Close Pair Algorithm

- Push the pair of root nodes into *stack*.
- While $stack \neq \phi$
  - Pop a pair (e_R, e_S) from *stack*.
  - For every entry se_R in PAGE(e_R) and se_S in PAGE(e_S) where MINDIST(se_R, se_S) $< \alpha$
    - Push (se_R, se_S) into stack if se_R is an index entry;
    - Otherwise report (se_R, se_S) as one close pair.
- End while

46

Content

- The R-tree
  - Range Query
  - Aggregation Query
- NN Query
- Closest Pair Query
- Close Pair Query
- **Skyline Query**

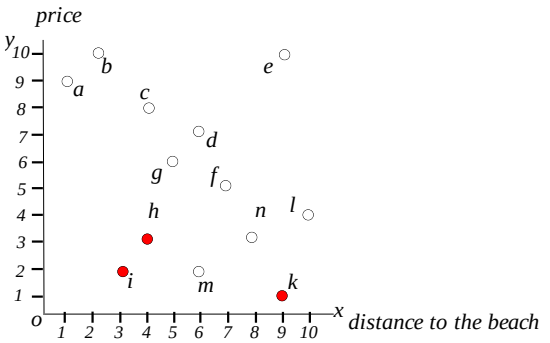
47

Skyline of Manhattan

- Which buildings can we see?
  - *Higher or nearer*

48

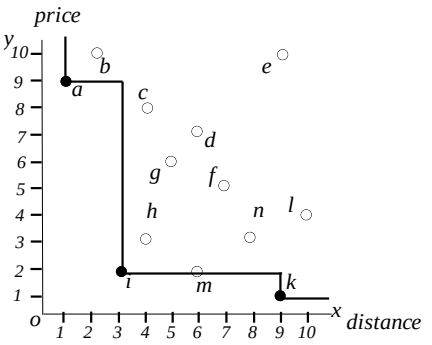
Finding A Hotel Close to the Beach



- Which one is better?
  - *i* or *h*? (*i*, because its price and distance dominate those of *h*)
  - *i* or *k*?

49

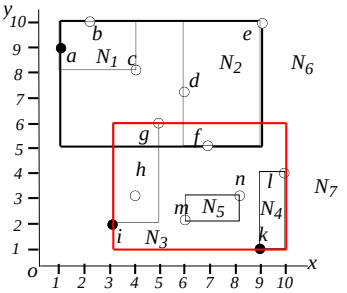
Skyline Queries



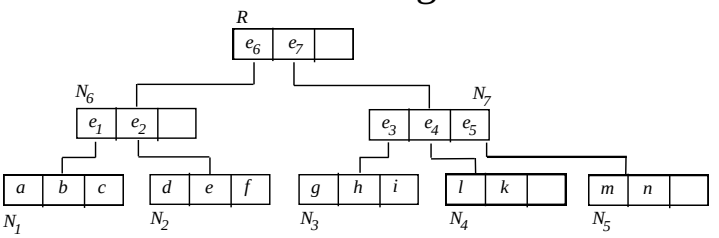
- Retrieve points not dominated by any other point.

50

Branched and Bound Skyline (BBS)

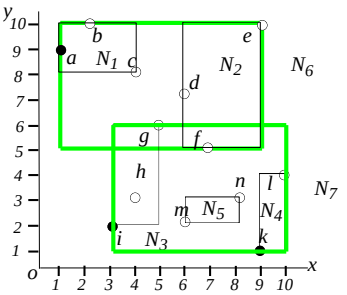


- Assume all points are indexed in an R-tree.
- mindist(MBR) = the L_1 distance between its lower-left corner and the origin.



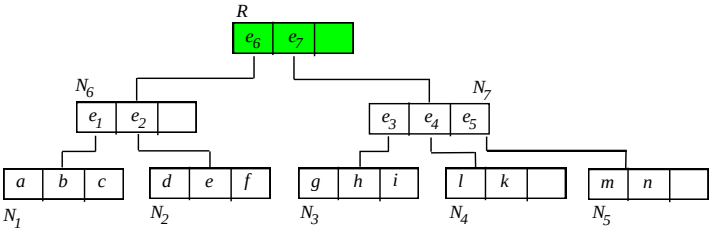
51

Branched and Bound Skyline (BBS)



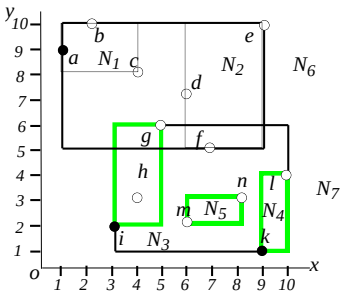
- Each heap entry keeps the mindist of the MBR.

action	heap contents	S
access root	<e_7,4><e_6,6>	∅



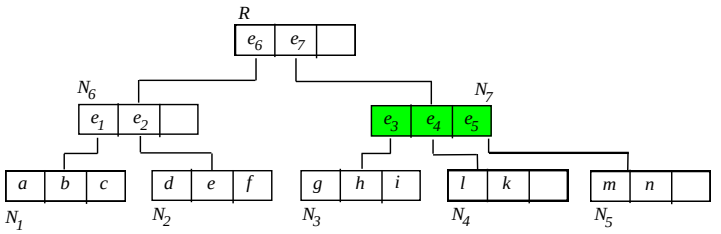
52

Example of BBS



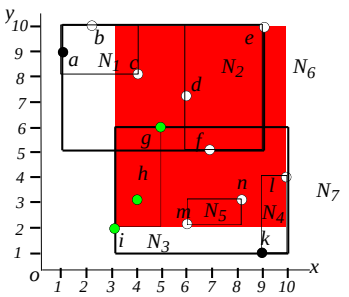
- Process entries in ascending order of their mindists.

action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅

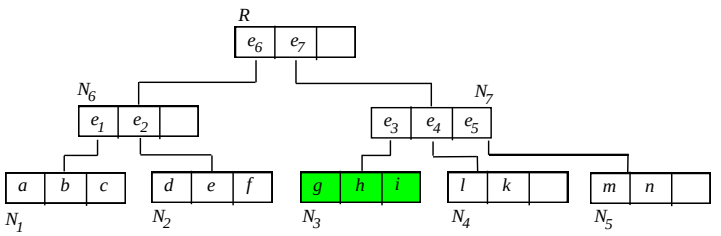


53

Example of BBS

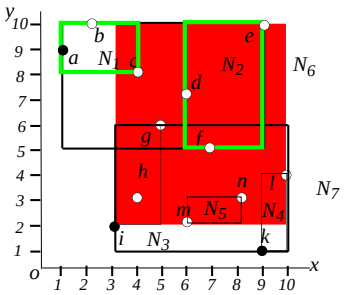


action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅
expand e_3	<***i*,5**><e_6,6><e_5,8><e_4,10>	{*i*}

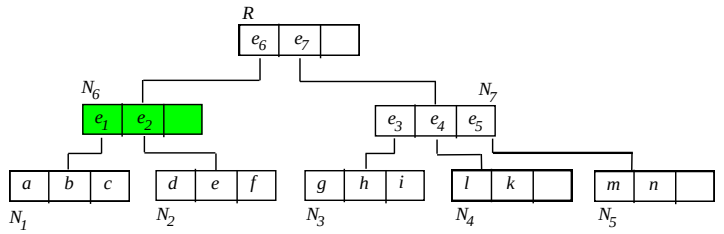


54

Example of BBS

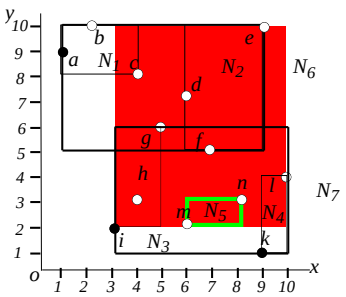


action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅
expand e_3	**<*i*,5>**<e_6,6><e_5,8> <e_4,10>	{*i*}
expand e_6	<e_5,8><e_1,9><e_4,10>	{*i*}

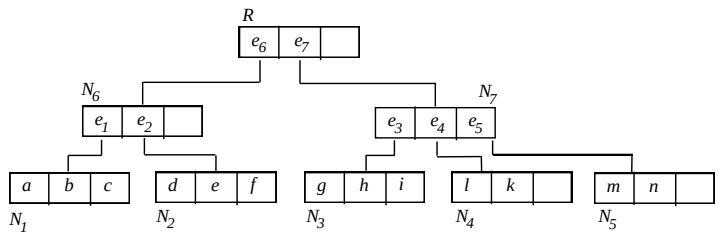


55

Example of BBS

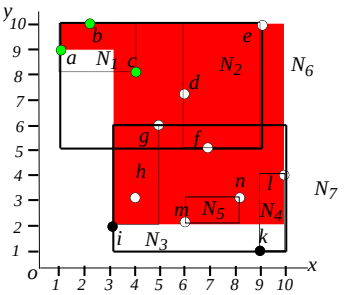


action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅
expand e_3	**<*i*,5>**<e_6,6><e_5,8> <e_4,10>	{*i*}
expand e_6	<e_5,8><e_1,9><e_4,10>	{*i*}
remove e_5	<e_1,9><e_4,10>	{*i*}

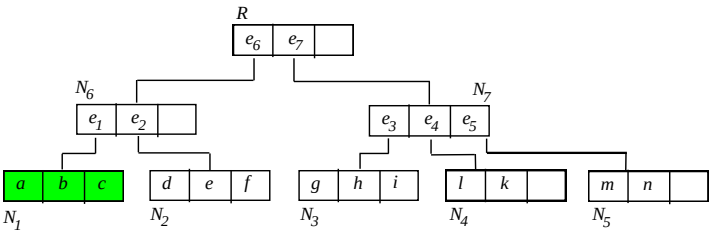


56

Example of BBS

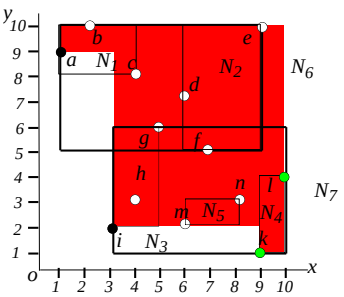


action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅
expand e_3	**<*i*,5>**<e_6,6><e_5,8> <e_4,10>	{*i*}
expand e_6	<e_5,8><e_1,9><e_4,10>	{*i*}
remove e_5	<e_1,9><e_4,10>	{*i*}
expand e_1	**<*a*,10>**<e_4,10>	{*i*,*a*}

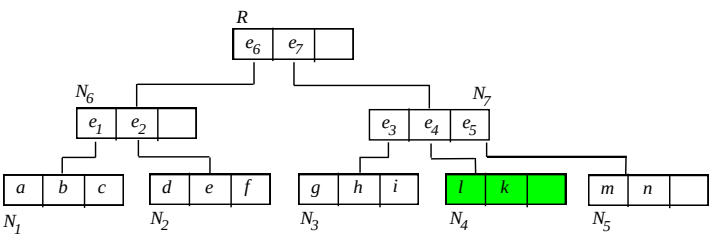


57

Example of BBS



action	heap contents	S
access root	<e_7,4><e_6,6>	∅
expand e_7	<e_3,5><e_6,6><e_5,8><e_4,10>	∅
expand e_3	**<*i*,5>**<e_6,6><e_5,8> <e_4,10>	{*i*}
expand e_6	<e_5,8><e_1,9><e_4,10>	{*i*}
remove e_5	<e_1,9><e_4,10>	{*i*}
expand e_1	**<*a*,10>**<e_4,10>	{*i*,*a*}
expand e_4	**<*k*,10>**	{*i*,*a*,*k*}



58