PR-Quadtrees

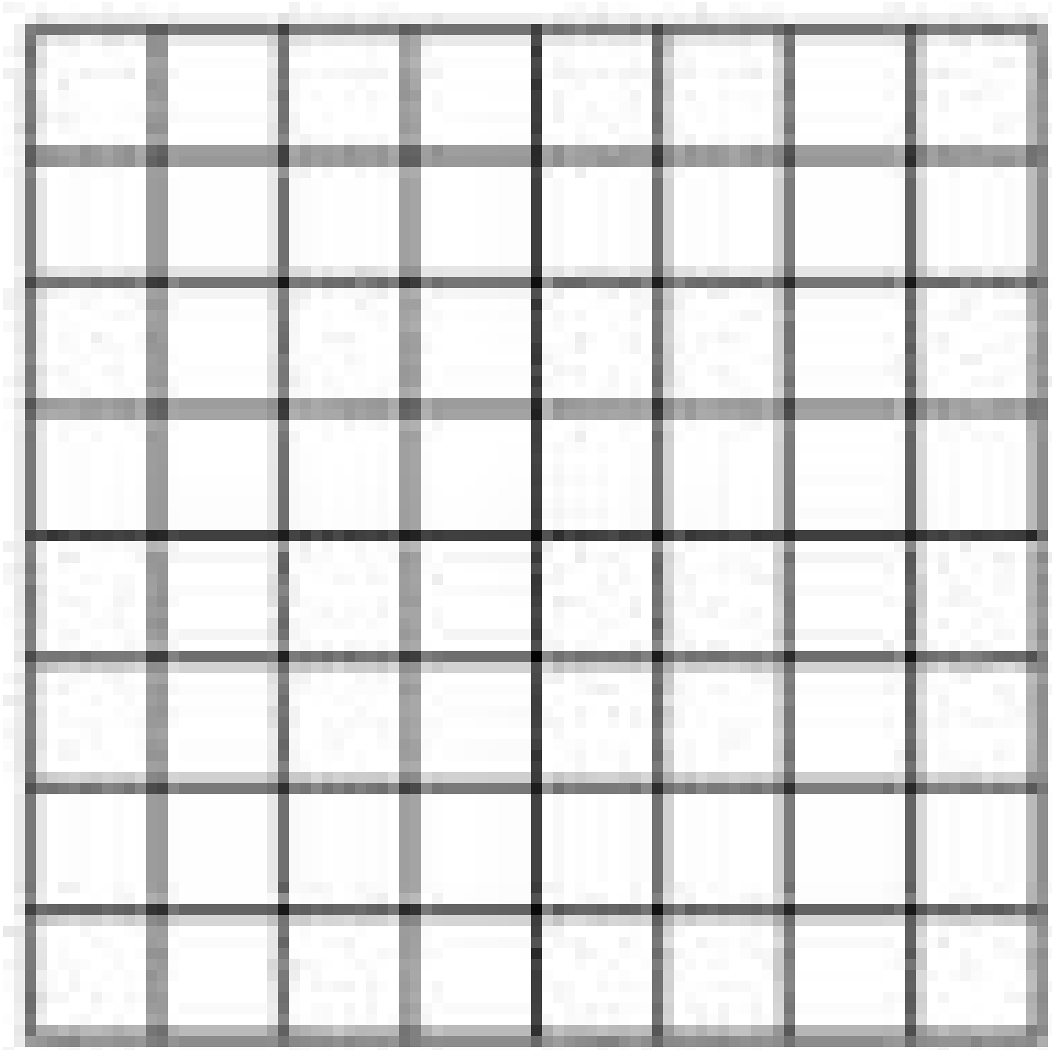
V.S. Subrahmanian

Spring 2013

MX-Quadtrees

- PR-Quadtrees are just like MX-Quadtrees. Nodes <u>always</u> split regions into equal-sized subregions.
- Like MX-quadtrees, PR-quadtrees assume that
  - The overall region is a 2^n x 2^n region for some n
  - All coordinates x,y are <u>integers</u> ranging from 0 up to (and including) $2^n – 1$.
- All data is stored in leaf nodes.
- <u>**But, unlike MX-Quadtrees, PR-Quadtrees do not require that leaves be at level n in the tree.**</u>
- <u>**Pr-Quadtrees do NOT split a node if the region represented by that node only contains one point.**</u>

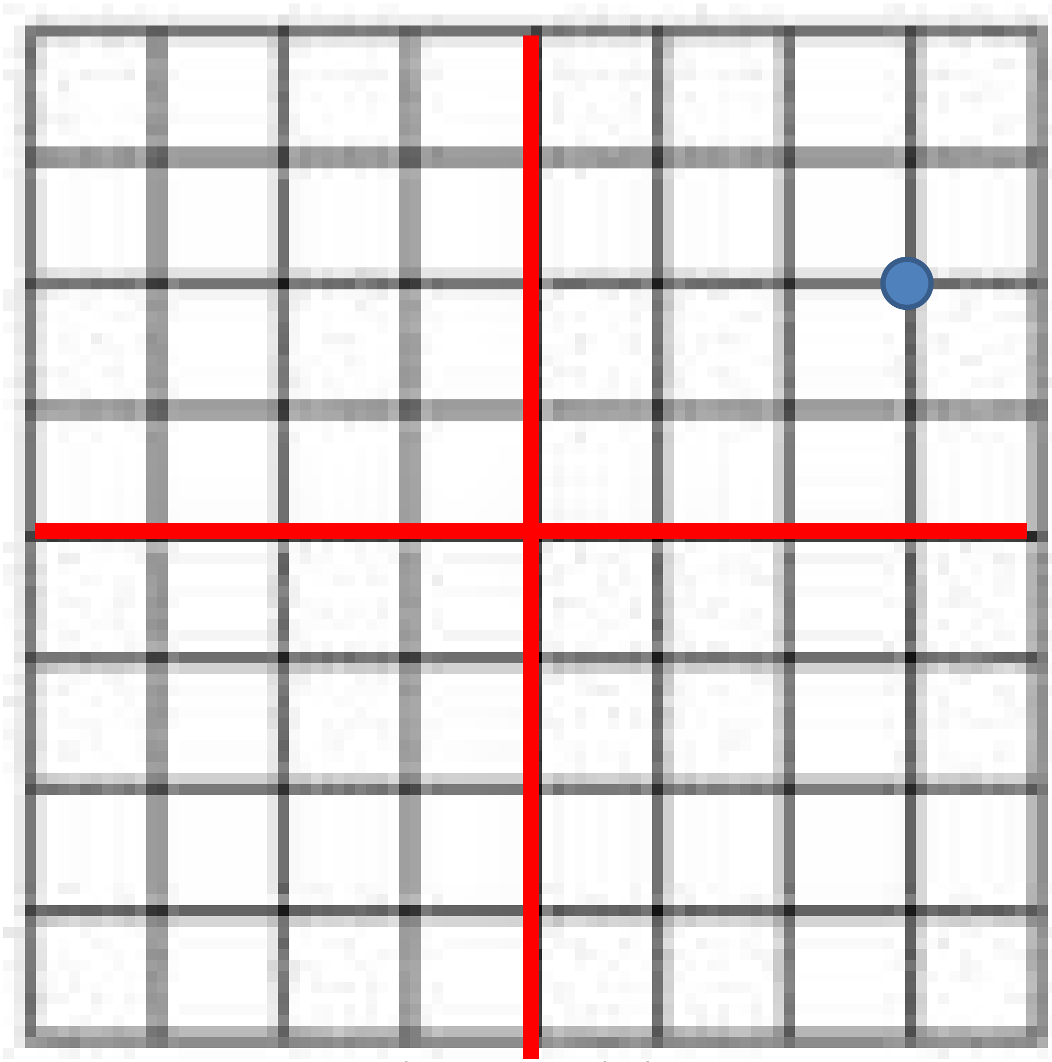
Example: Insert (7,6)



Insert (7,6)

The region associated with the root only contains one point. So insert (7,6) there.

Example: Insert (7,6)

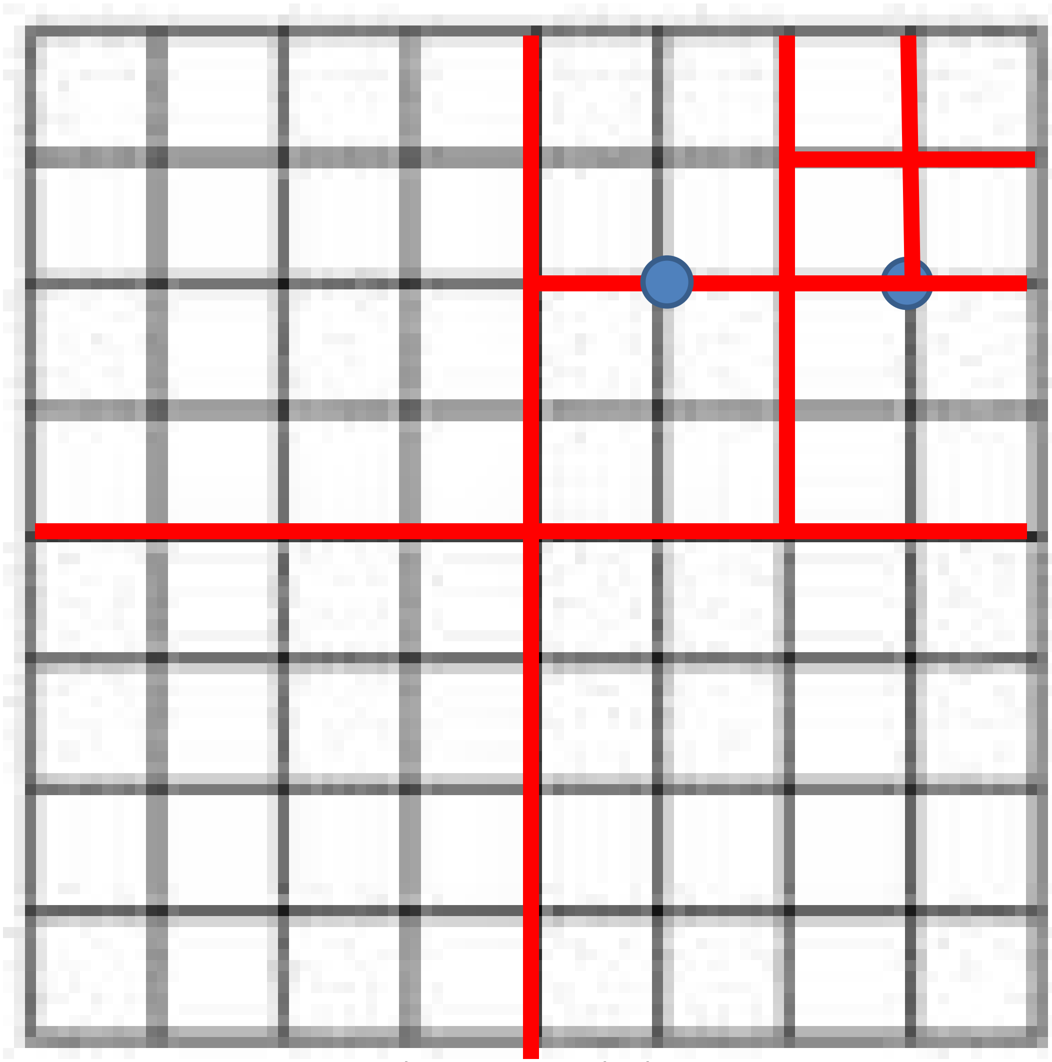


Insert (7,6)

(7,6)

The region associated with the root only contains one point. So insert (7,6) there.

Example: Insert (5,6)

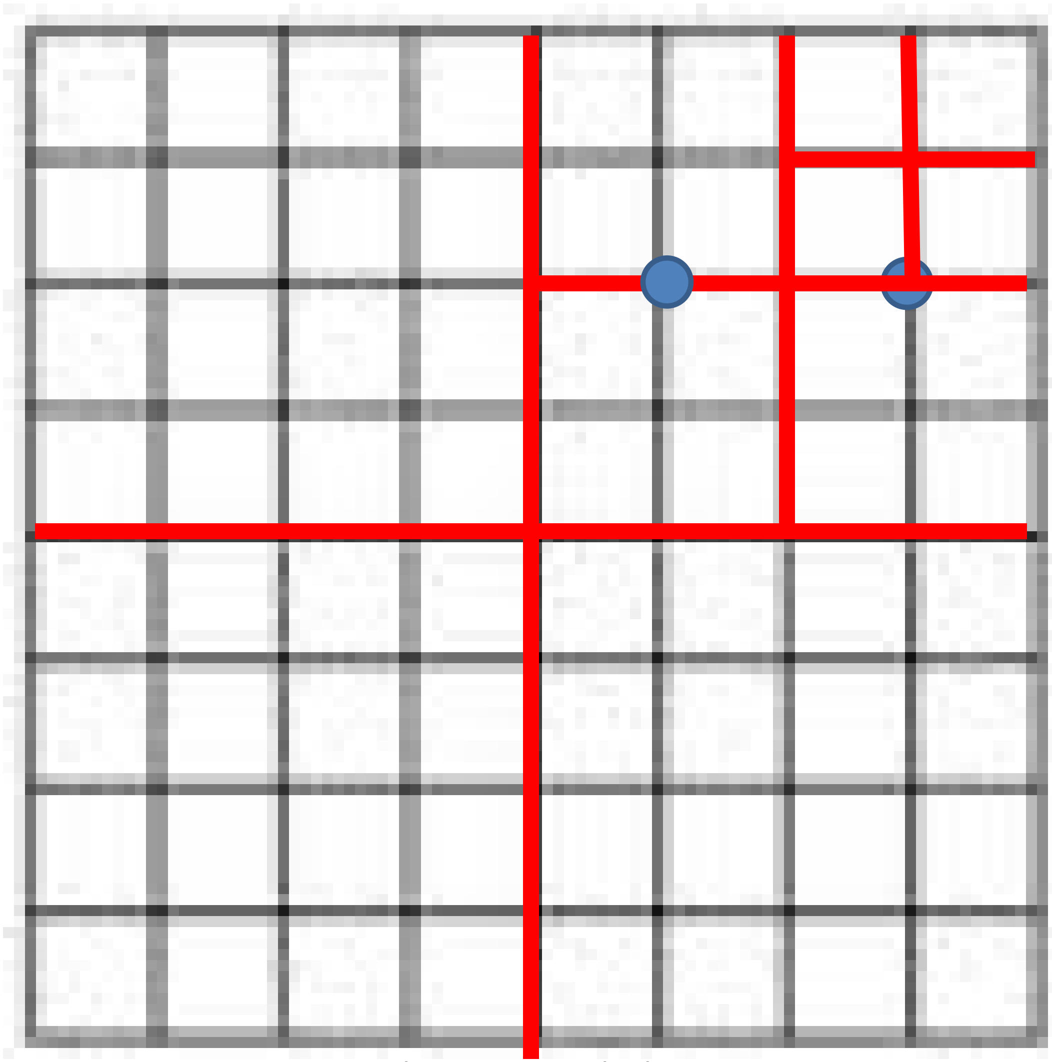


Insert (7,6)

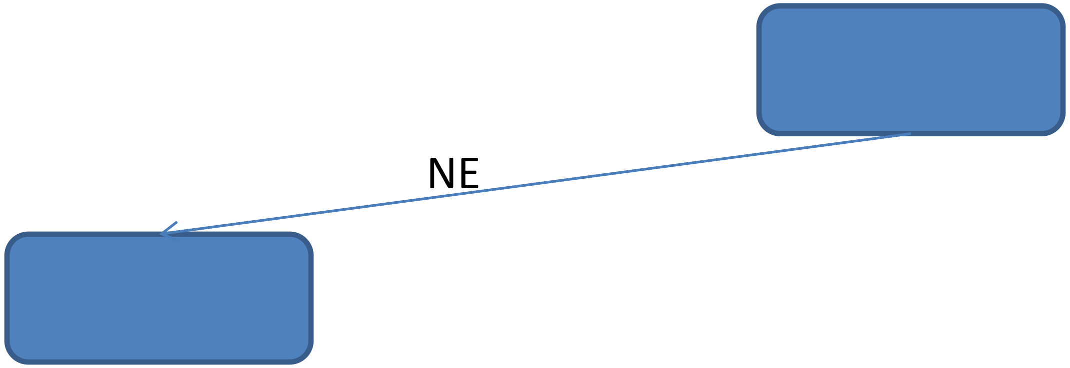
(7,6)

The region associated with the root now contains two points. So we must <u>split</u> the root and put BOTH (7,6) and (5,6) in the right place.

Example: Insert (5,6)

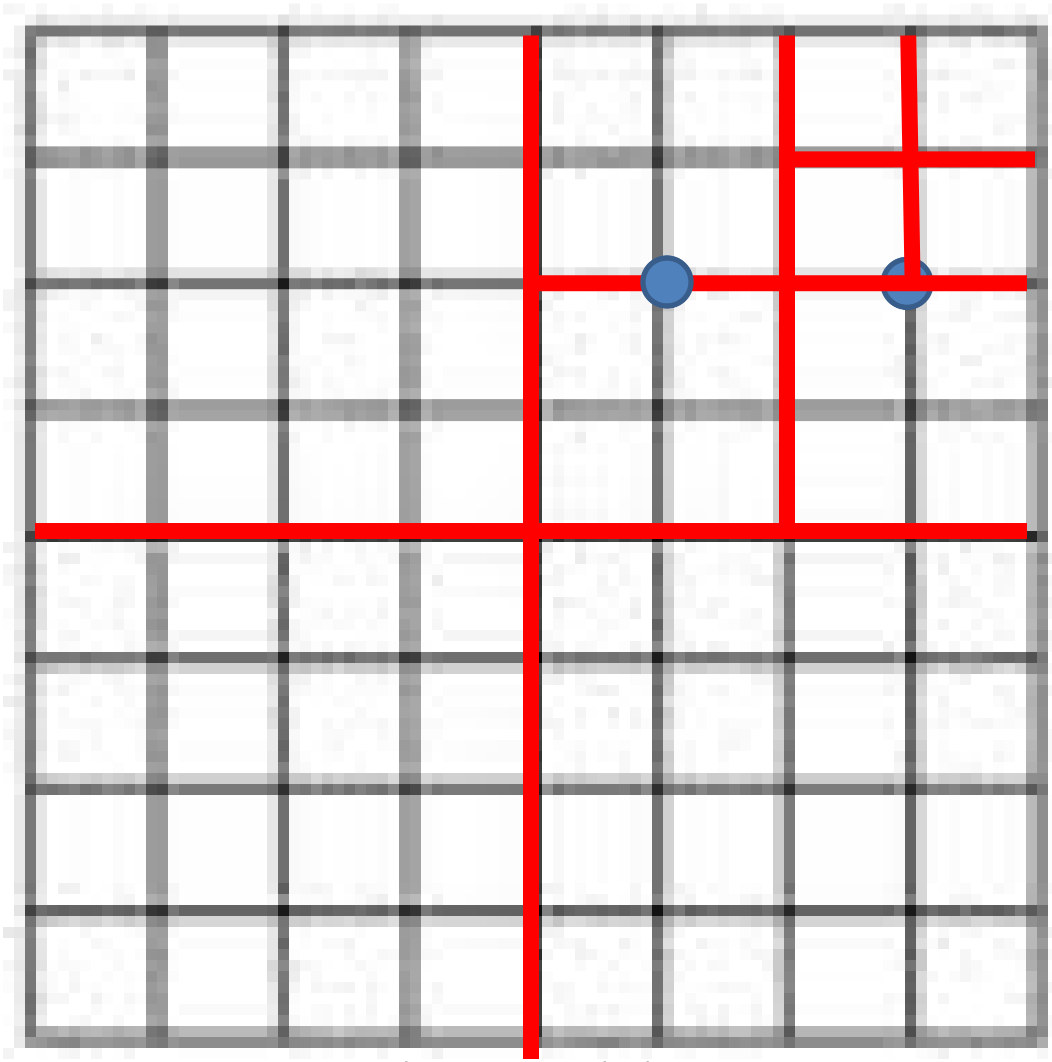


Insert (5,6)

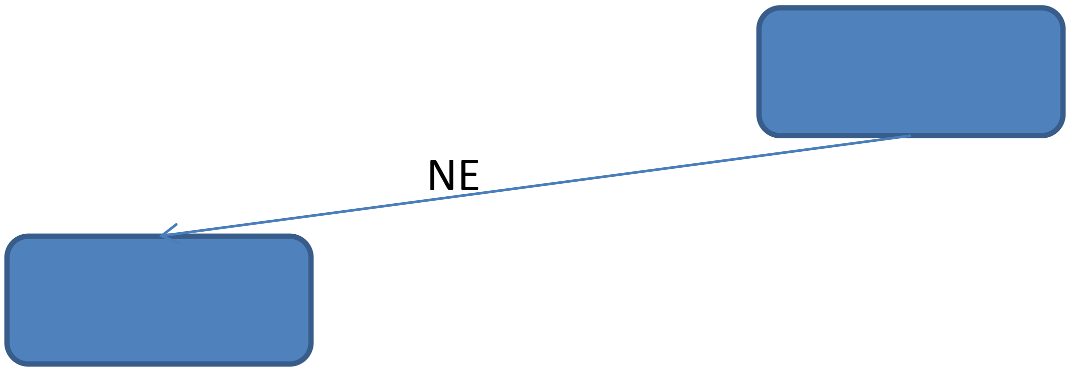


(7,6) and (5,6) are both in the NE quadrant of the region associated with the root. So we must split that region as well.

Example: Insert (5,6)

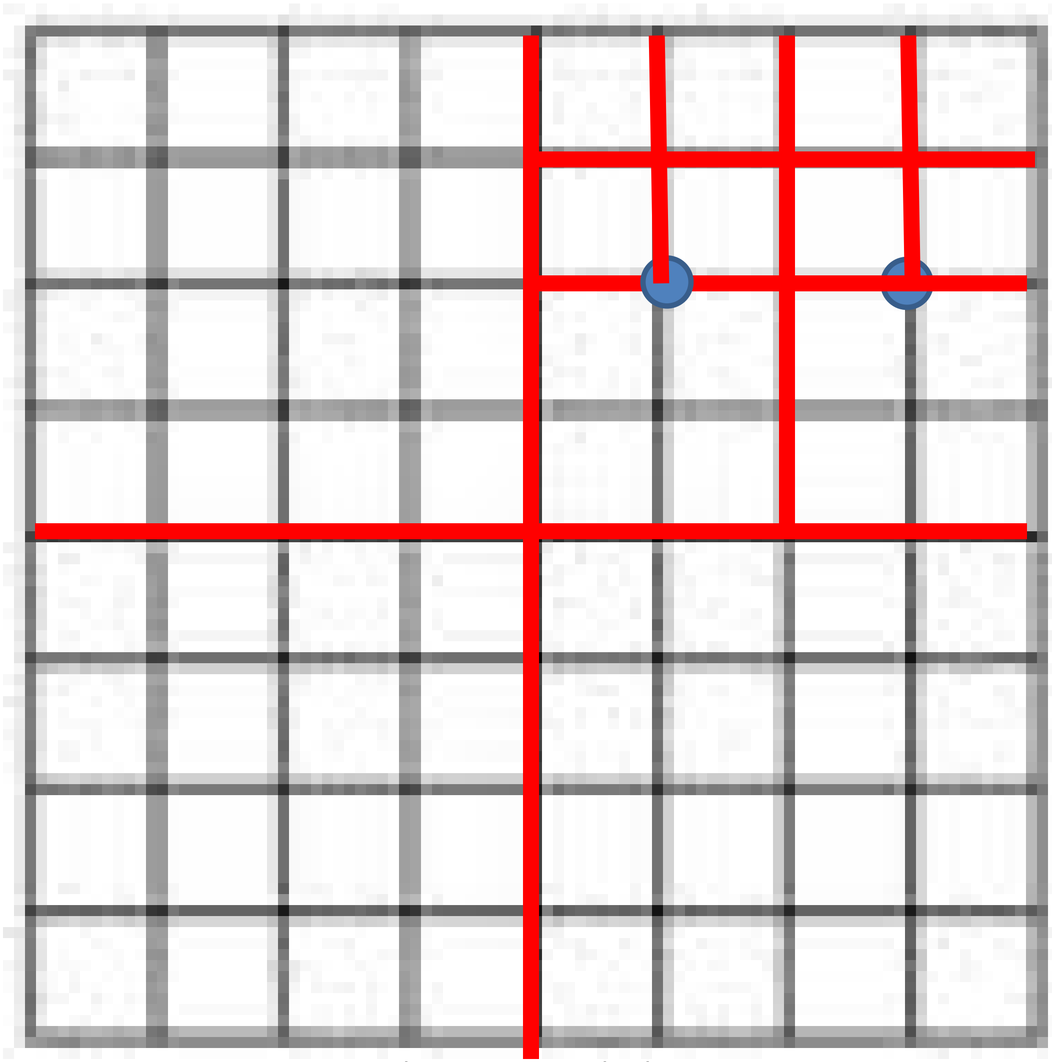


Insert (5,6)

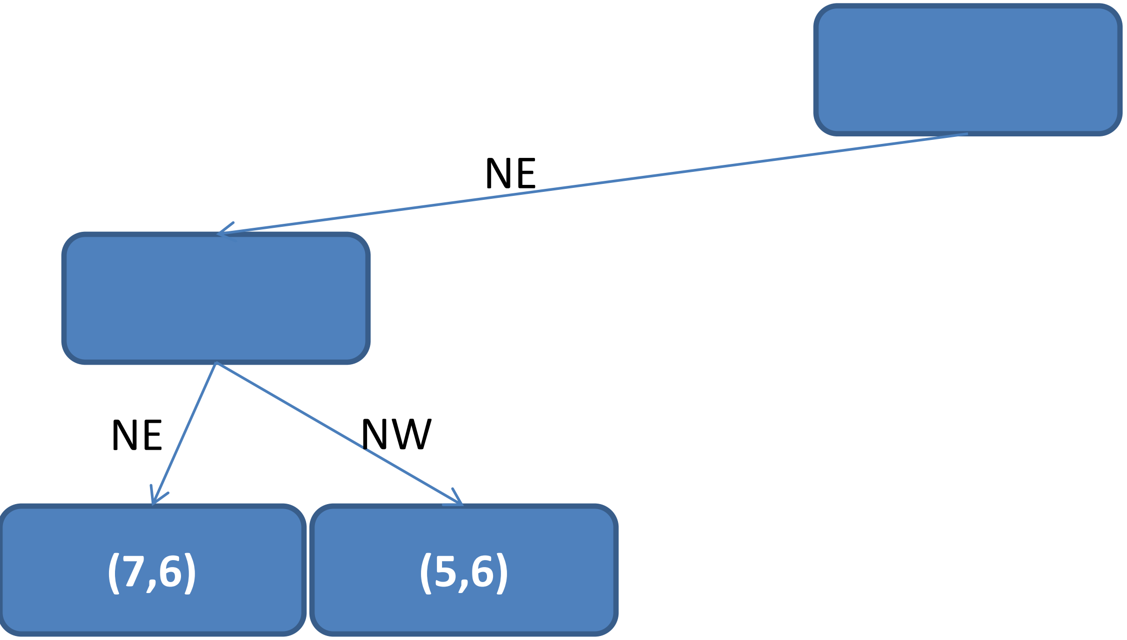


(7,6) and (5,6) are both in the NE quadrant of the region associated with the root. So we must split that region as well.

Example: Insert (5,6)

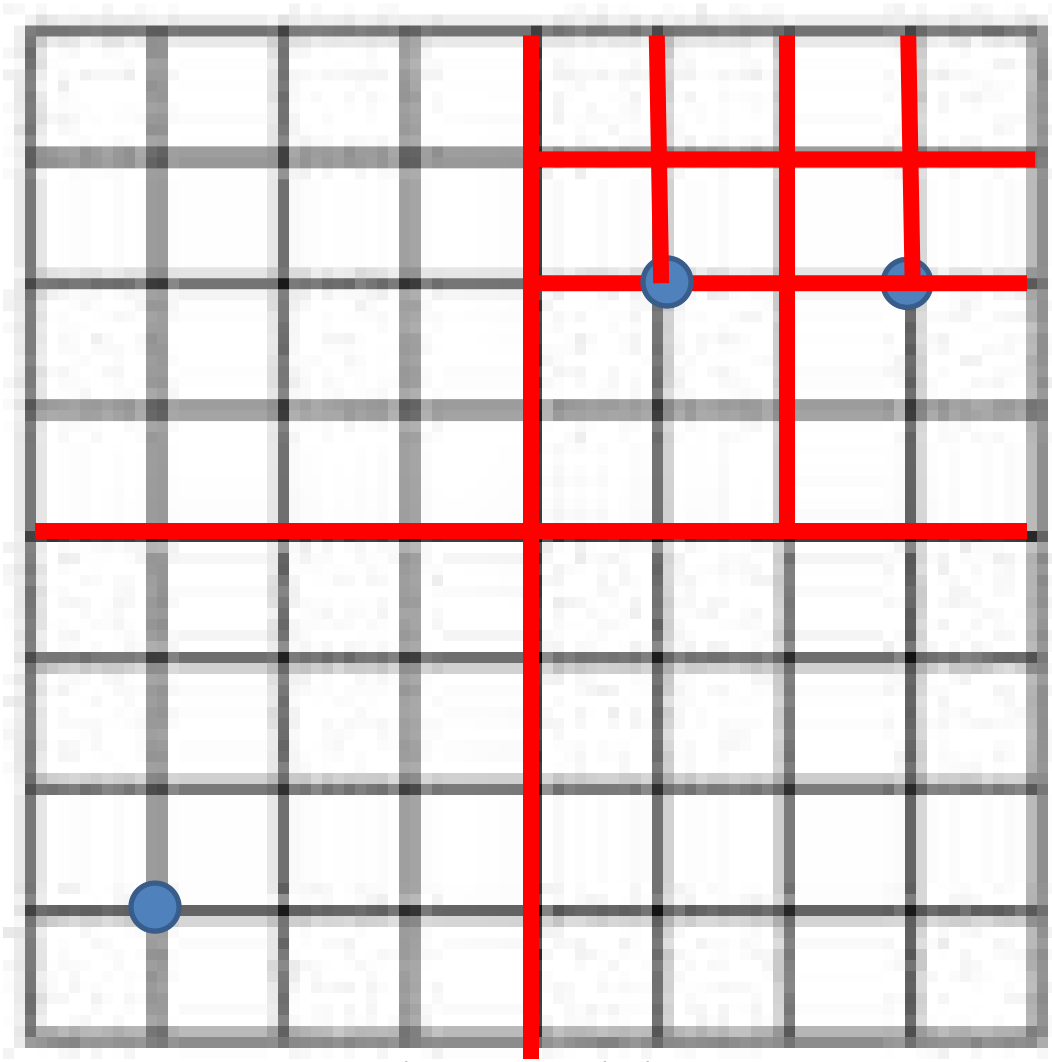


Insert (5,6)

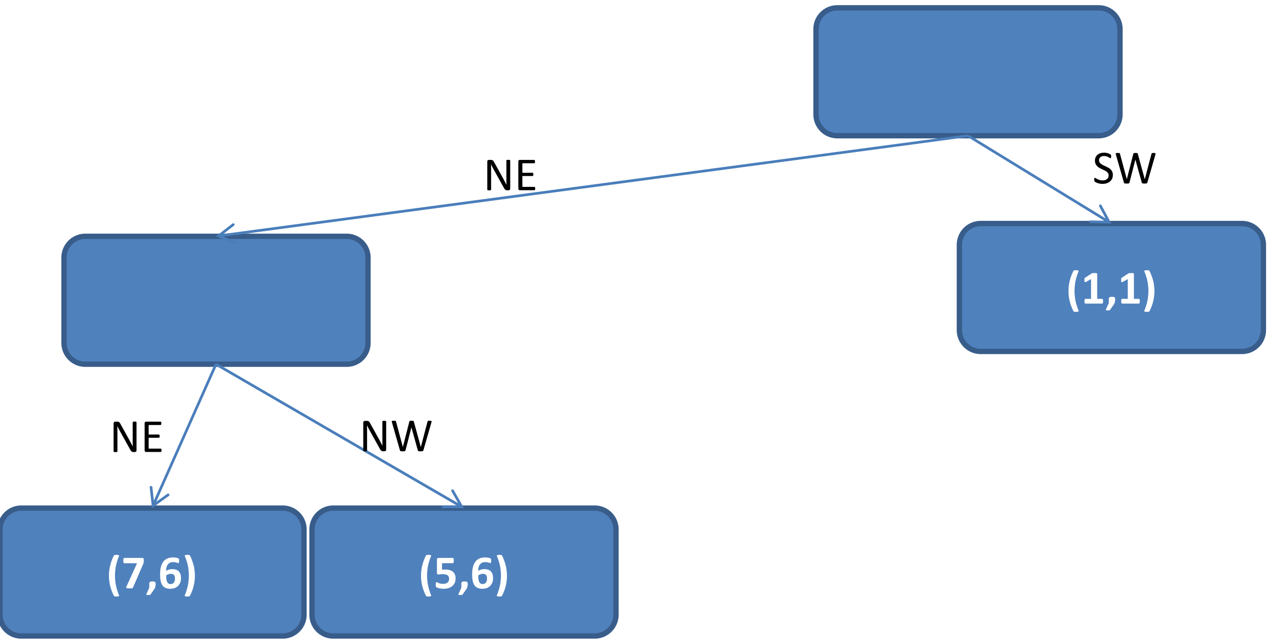


(7,6) is the NE quadrant, (5,6) is in the NW. We therefore get two children that have only one point each in their respective regions. So these nodes can be labeled with the points.

Example: Insert (1,1)



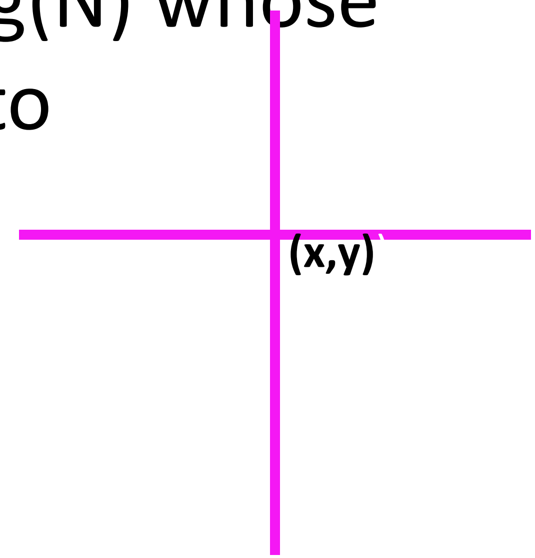
Insert (1,1)



(1,1) is in the SW quadrant. So create a SW child. This child node has only one point in it, so we are done.

Branching Convention

- If you are inserting point (x’,y’) and you are at a node representing region Reg(N) whose center is labeled (x,y), branch to
  - NE if $x' \geq x$ & $y' \geq y$.
  - NW if $x' < x$ & $y' \geq y$.
  - SE if $x' \geq x$ & $y' < y$
  - SW if $x' < x$ & $y' < y$.



Intuitively, quadrants are closed on the left and bottom, open on the right and top.

In-class Exercise

- Insert (1,3)
- Insert (6,7)
- Insert (3,1)
- Insert (2,2)
- Insert (1,8)

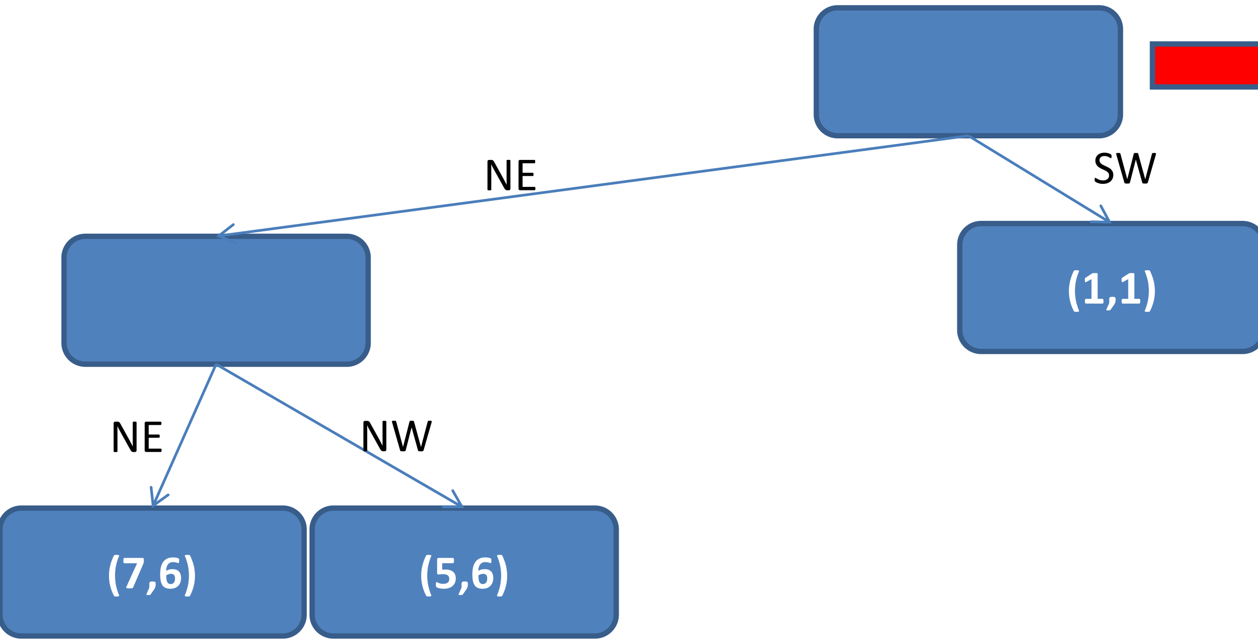
Points for Discussion

- Is the “shape” of a PR-quadtree affected by the order in which nodes are inserted?
- What is the worst-case complexity of searches for a given point in an PR-quadtree?
- How does
  – Insertion time compare for a PR- vs. MX-quadtree?
  – Search time compare for a PR- vs. MX-quadtree?

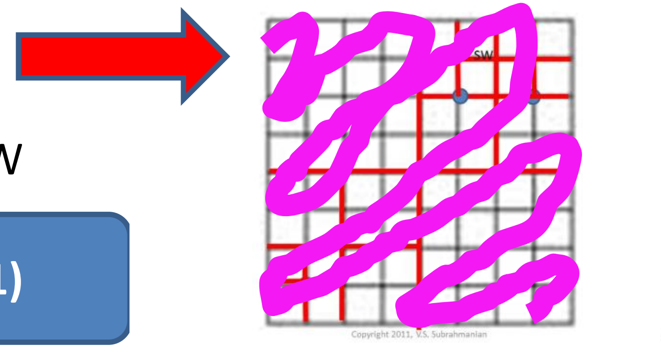
Regions associated with nodes

- Every node N in a PR-Quadtree is implicitly associated with a region, Reg(N).
- The root represents the entire $2^n \times 2^n$ region.
- Regions are <u>always</u> split by drawing a vertical line and a horizontal line through the center point of a region.

Regions associated with nodes

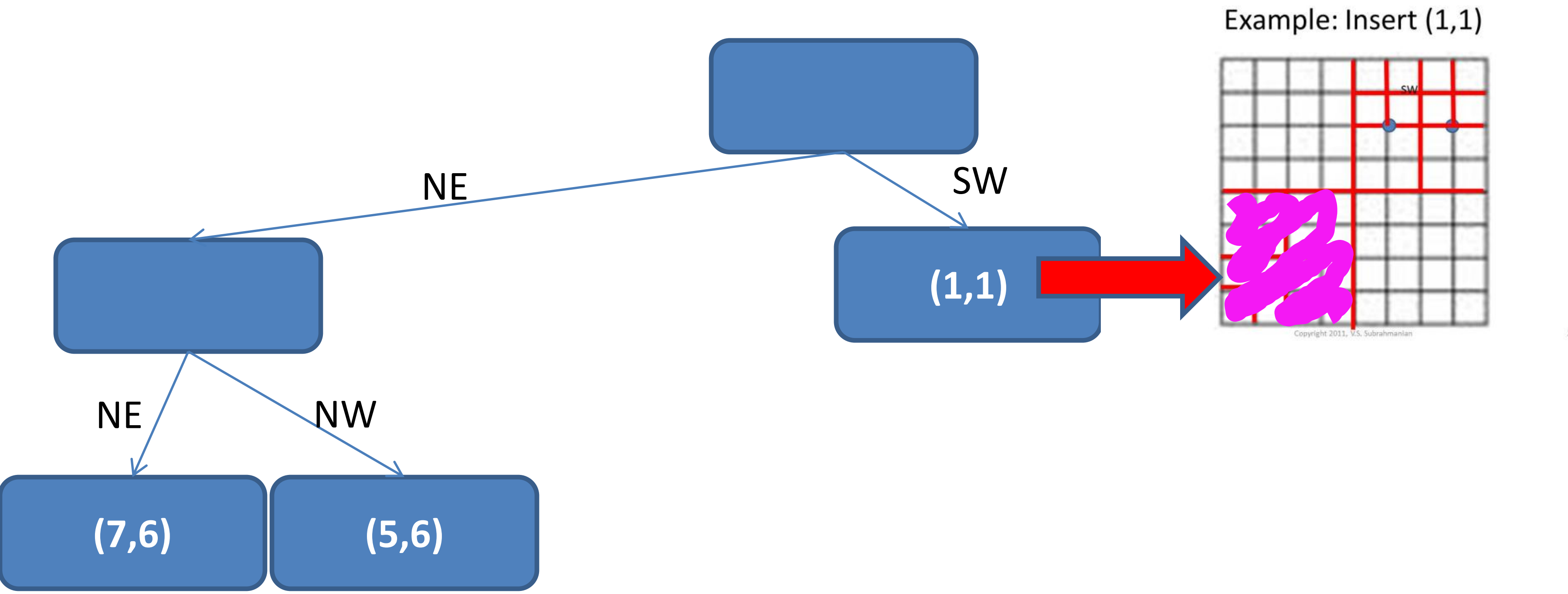


Example: Insert (1,1)



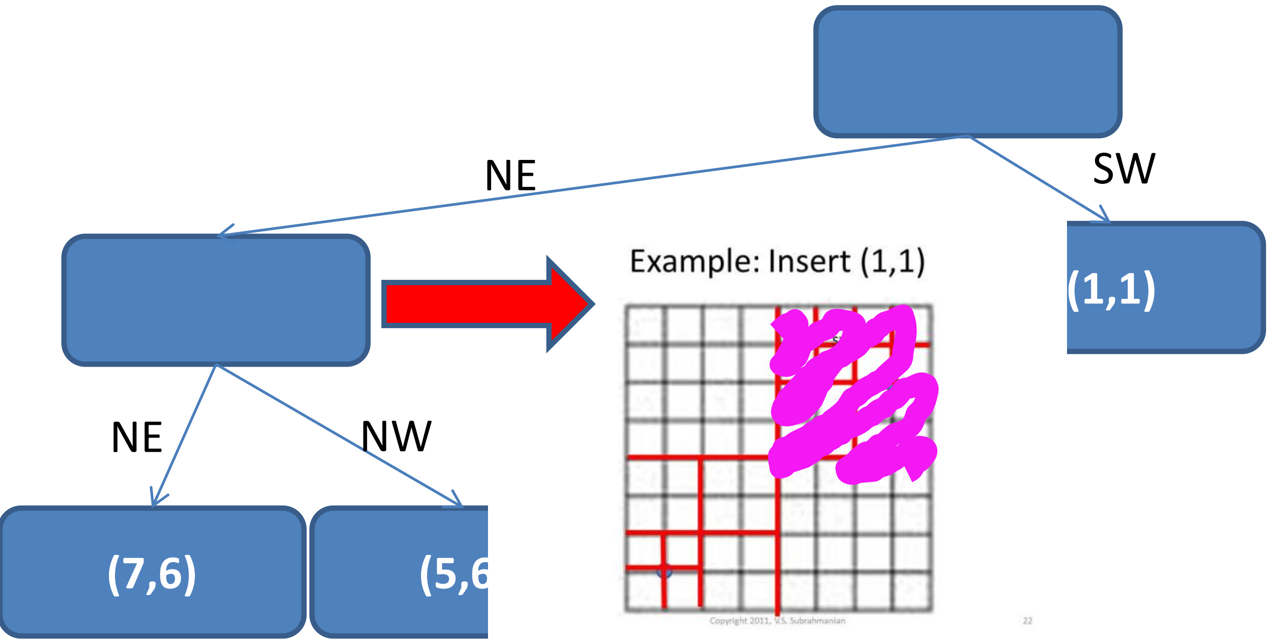
The root represents the whole region.

Regions associated with nodes



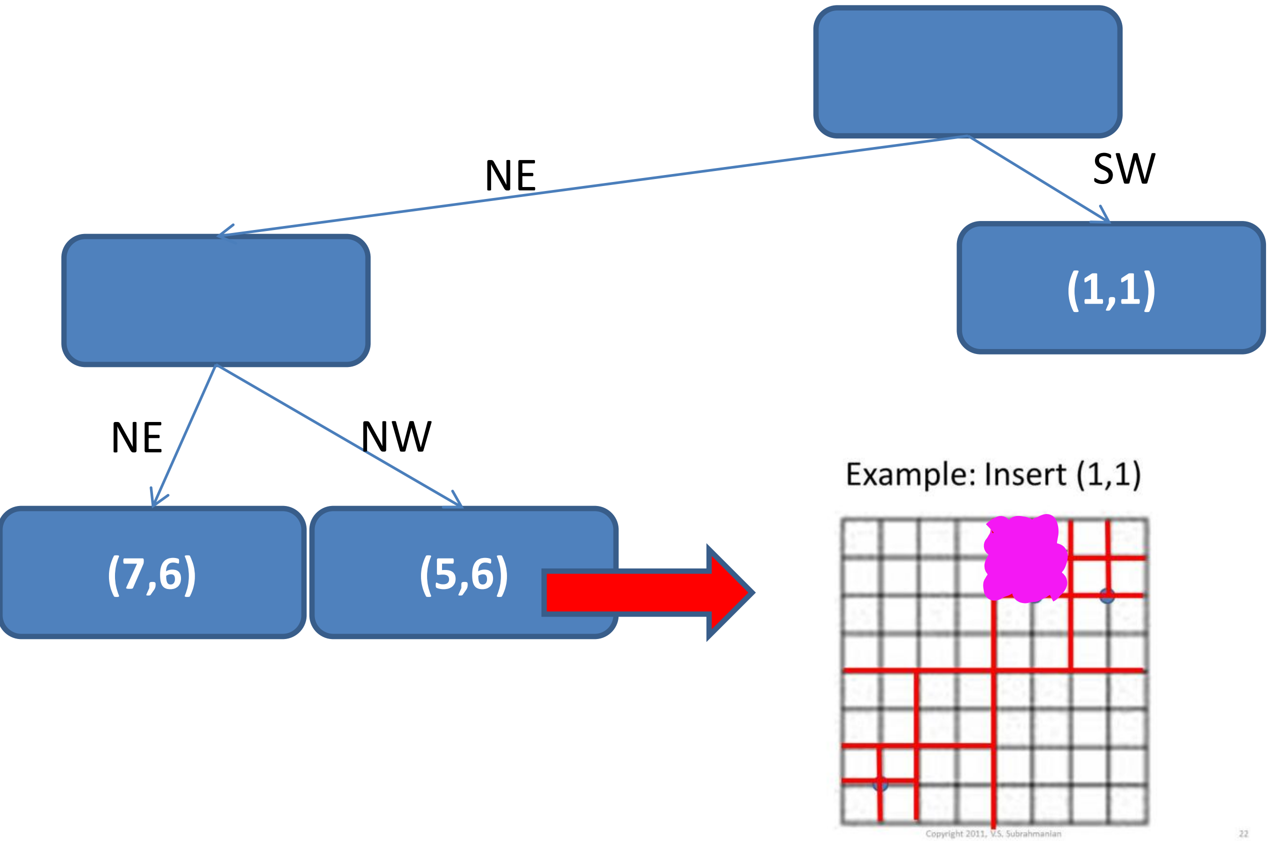
The SW child of the root represents the entire SW quadrant

Regions associated with nodes



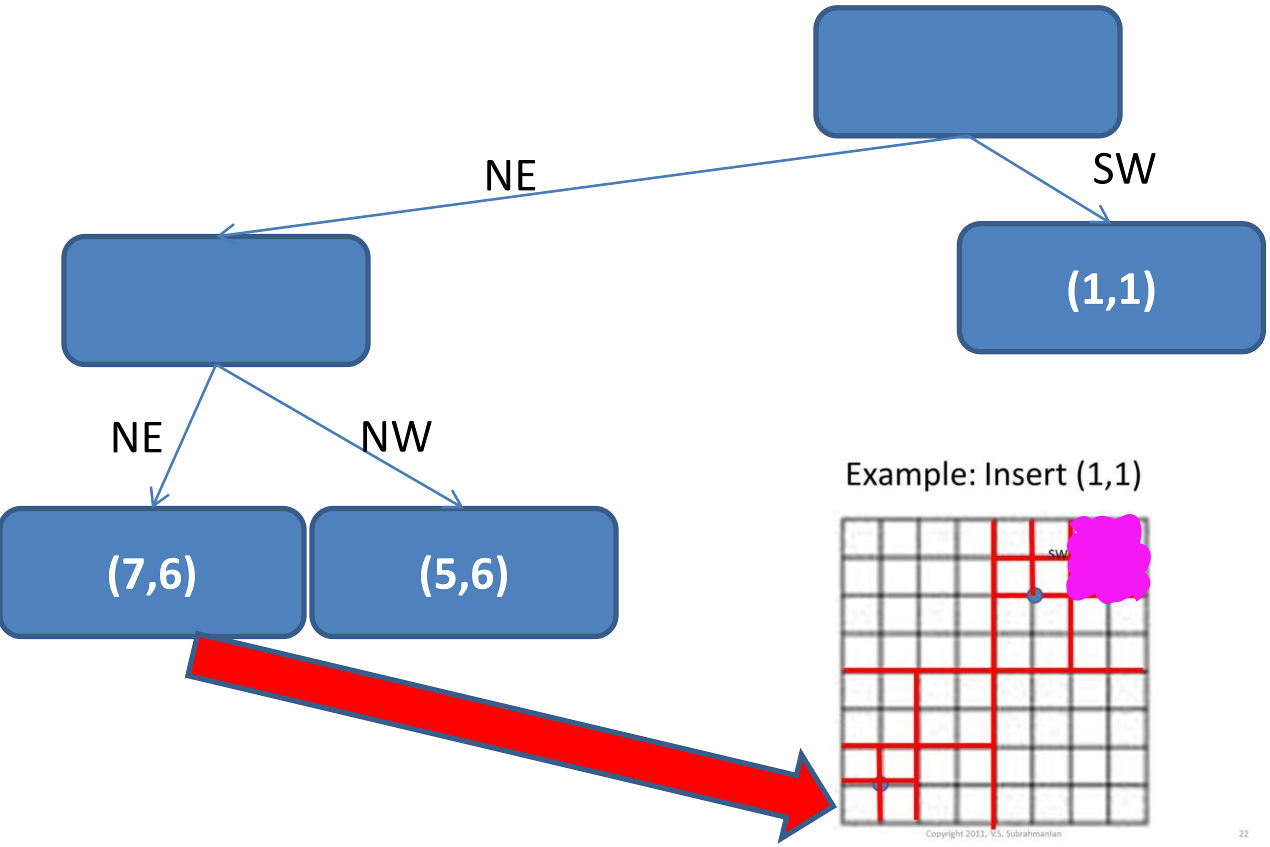
The NE child of the root represents the entire NE quadrant

Regions associated with nodes



The NW child of the root's NE child represents the entire NW subquadrant of the NE quadrant of the root.

Regions associated with nodes



The NE child of the root’s NE child represents the entire NE subquadrant of the NE quadrant of the root.

Range Searches

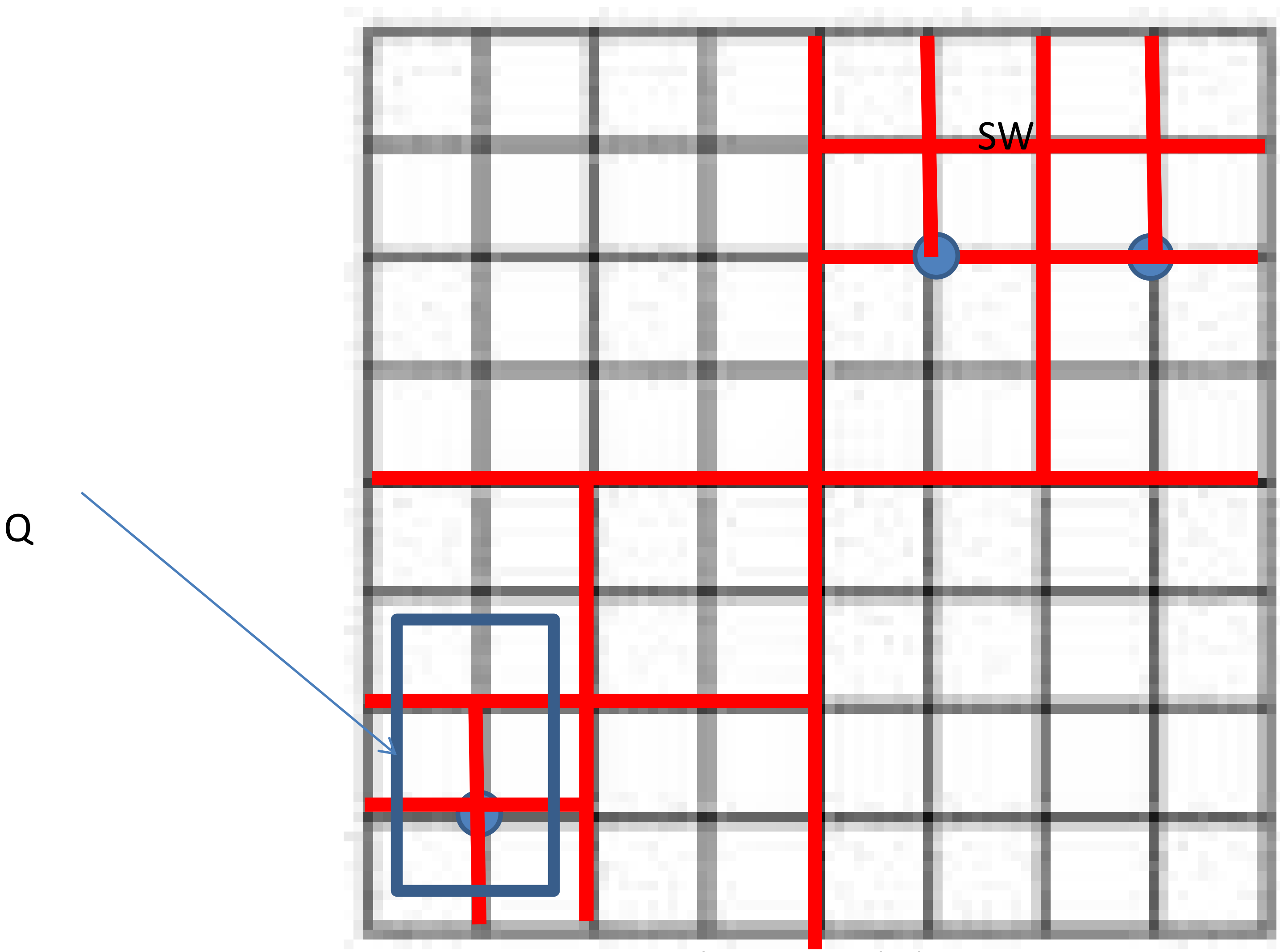
- INPUTS:
  - pointer T to the root of an MX-Quadtree
  - Region Q.
- Need to find all points in T that are in region Q

Visit(N)

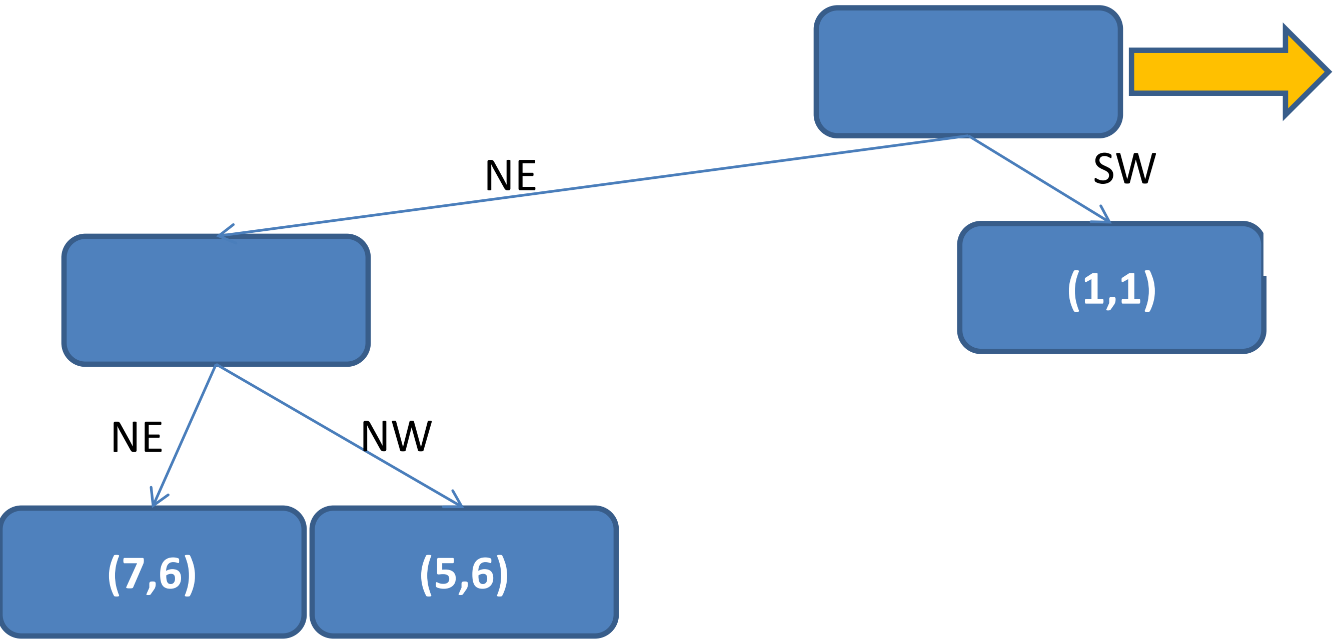
- If N is a non-leaf node
  - If Reg(N) intersects Q, then recursively visit all of N's children.
  - Otherwise PRUNE !
- Else (N is a leaf) check if N.Point in Q. If yes, insert N.Point into SOL.

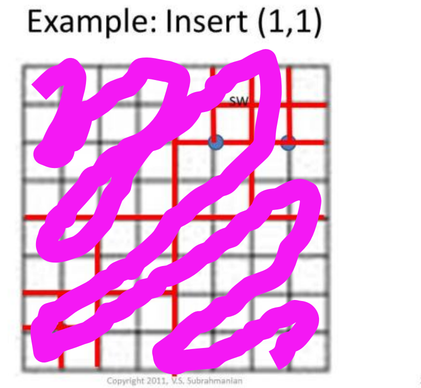
Algorithm is identical to that for MX-quadtrees.

Example Range Search



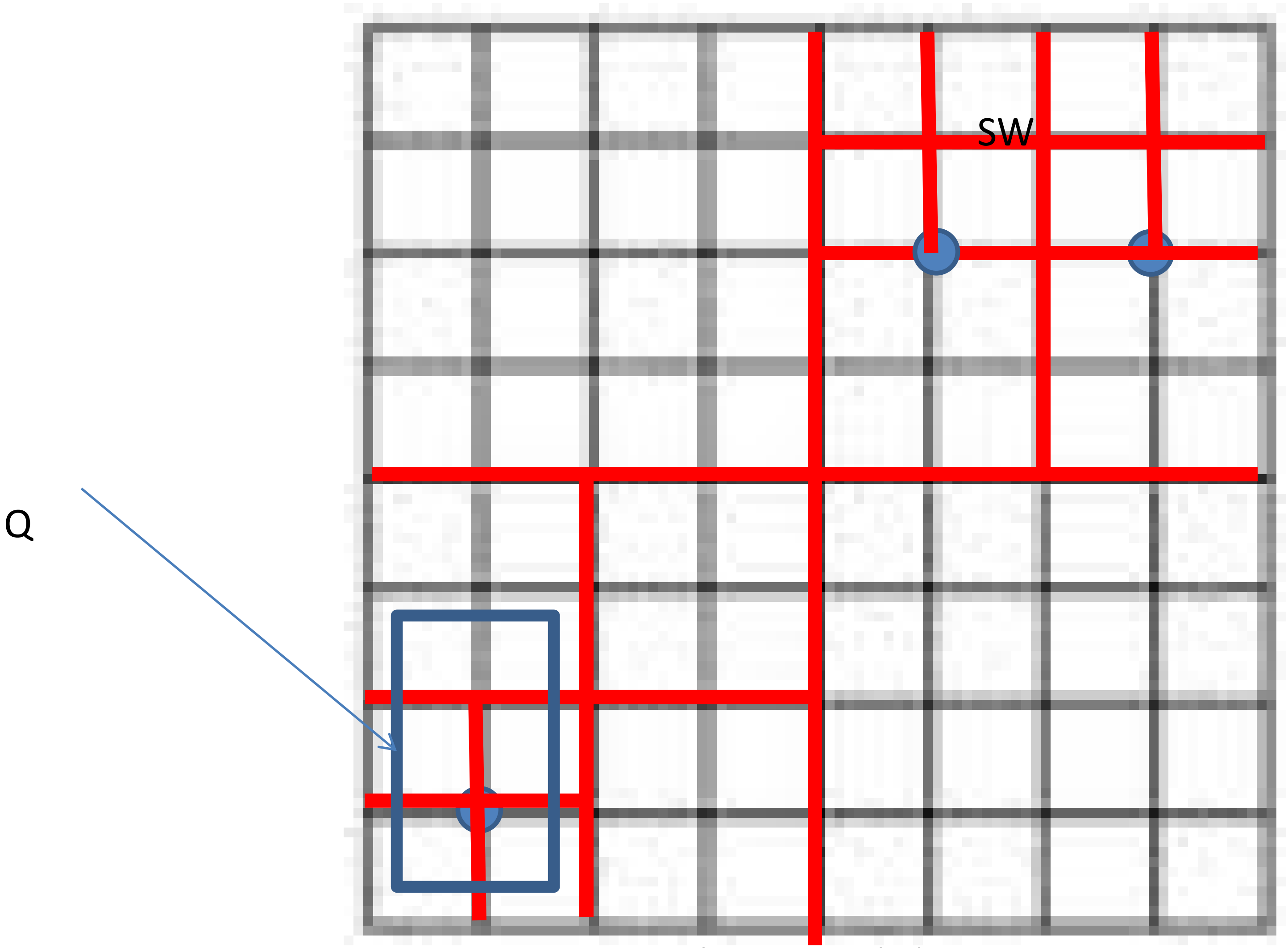
Example range search



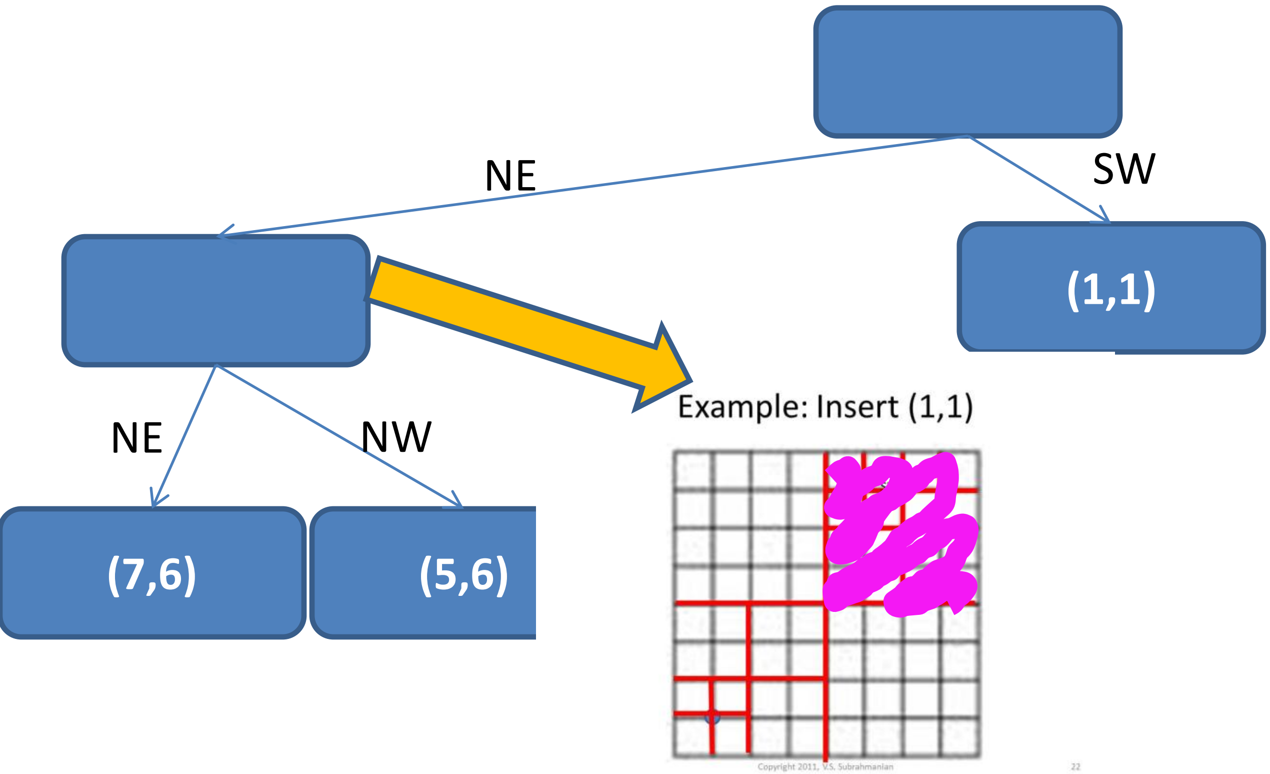


Q intersects the root – which is not a leaf. So must recursively search children.

Example Range Search

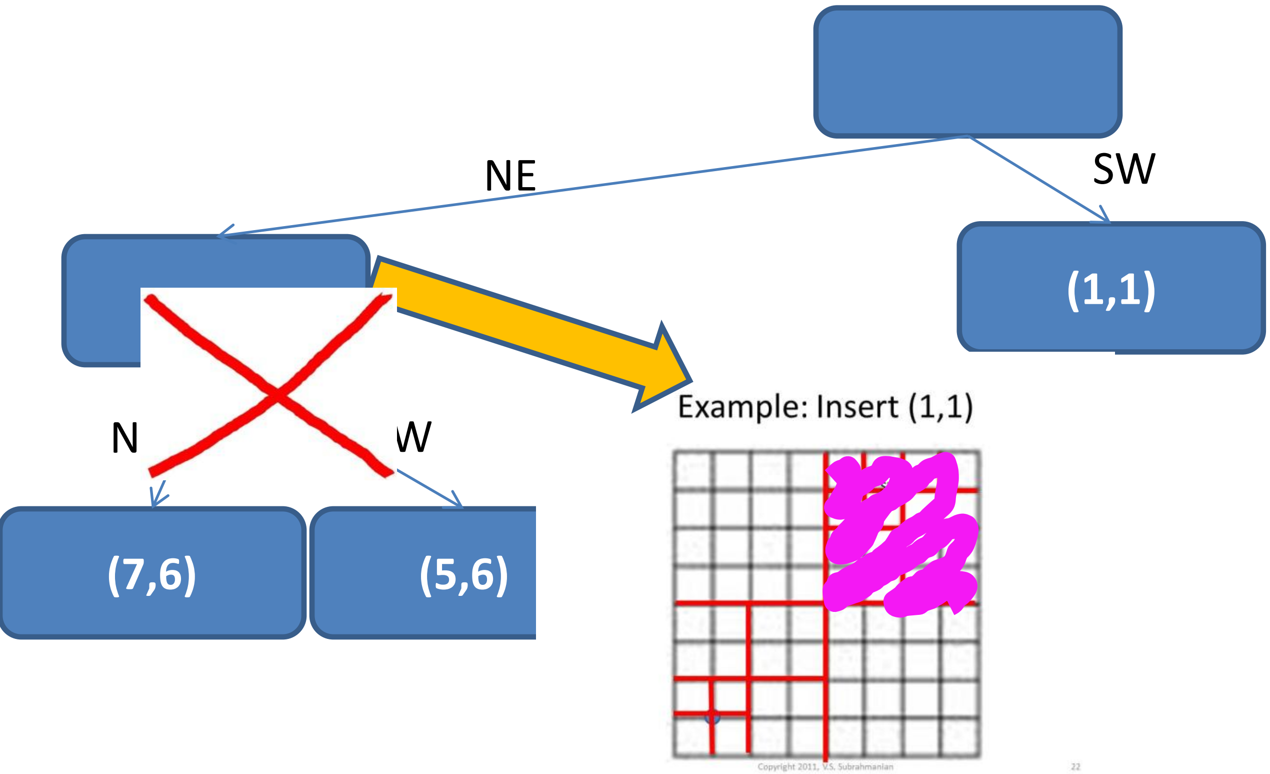


Example range search



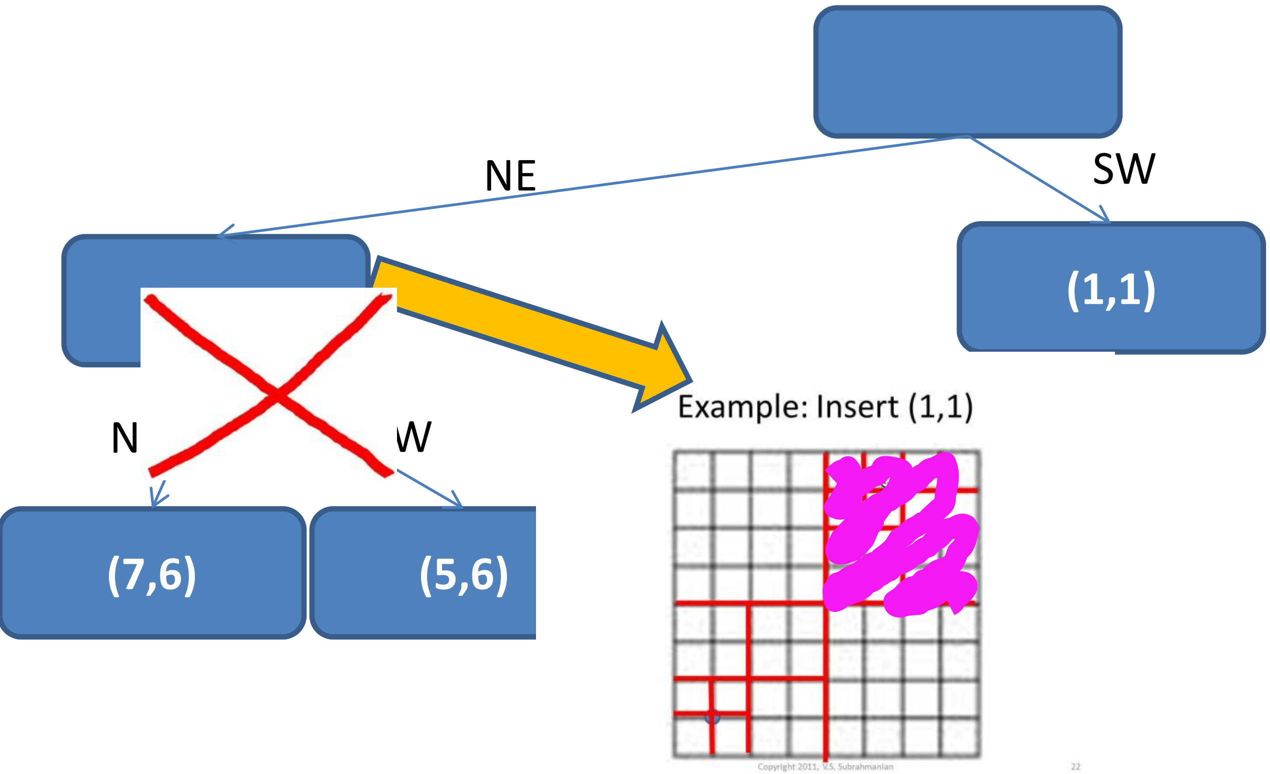
Q does NOT intersect the region associated with the NE quadrant of the root. So prune.

Example range search



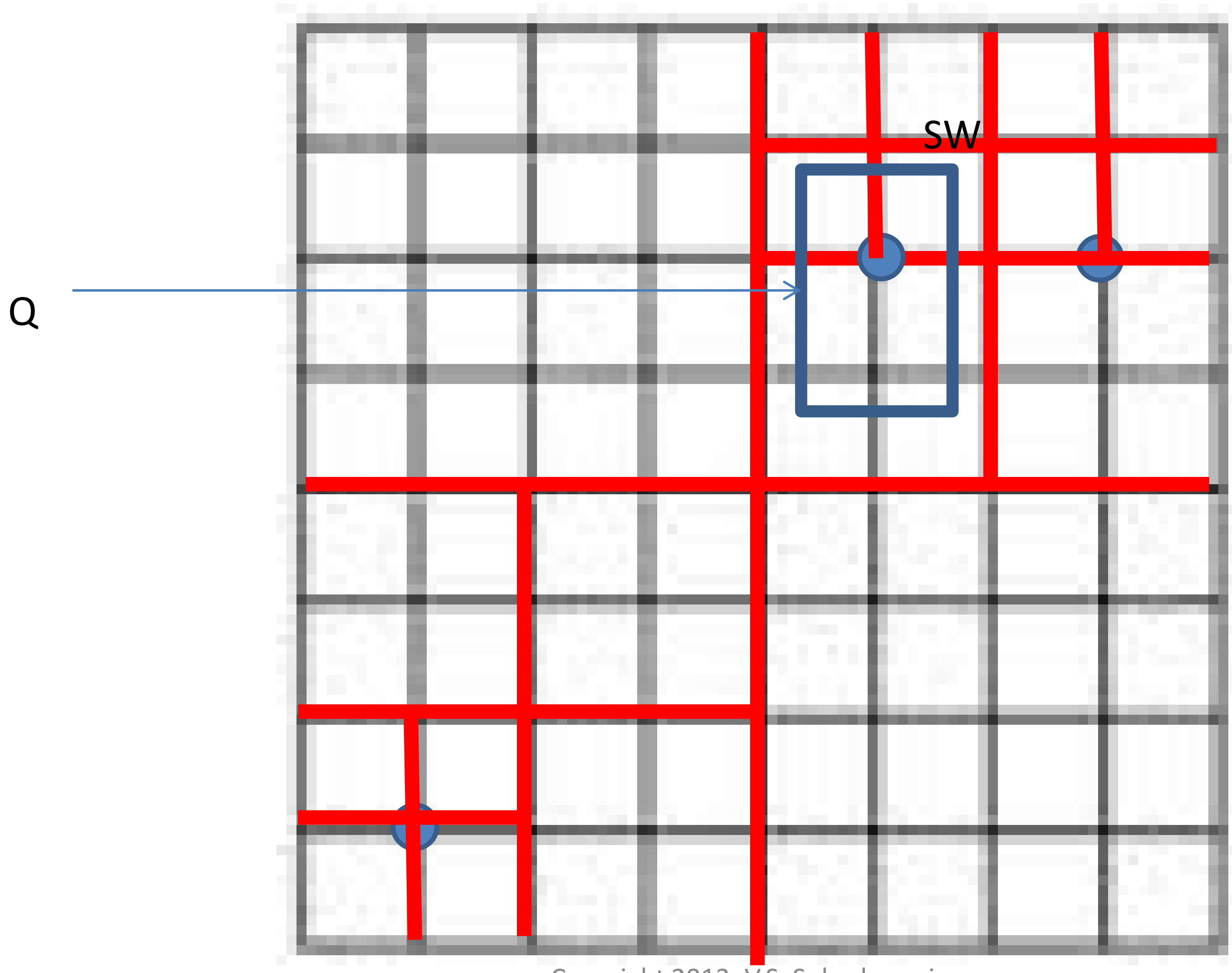
Q does NOT intersect the region associated with the NE quadrant of the root. So prune.

Example range search



Now consider the SW child of the root. Q intersects this region. As the SW child is a leaf, we check if the point labeling it (1,1) is in the region. Yes ! So return it.

In Class Exercise: Example Range Search



Nearest Neighbor Searches in PR-Quadtrees

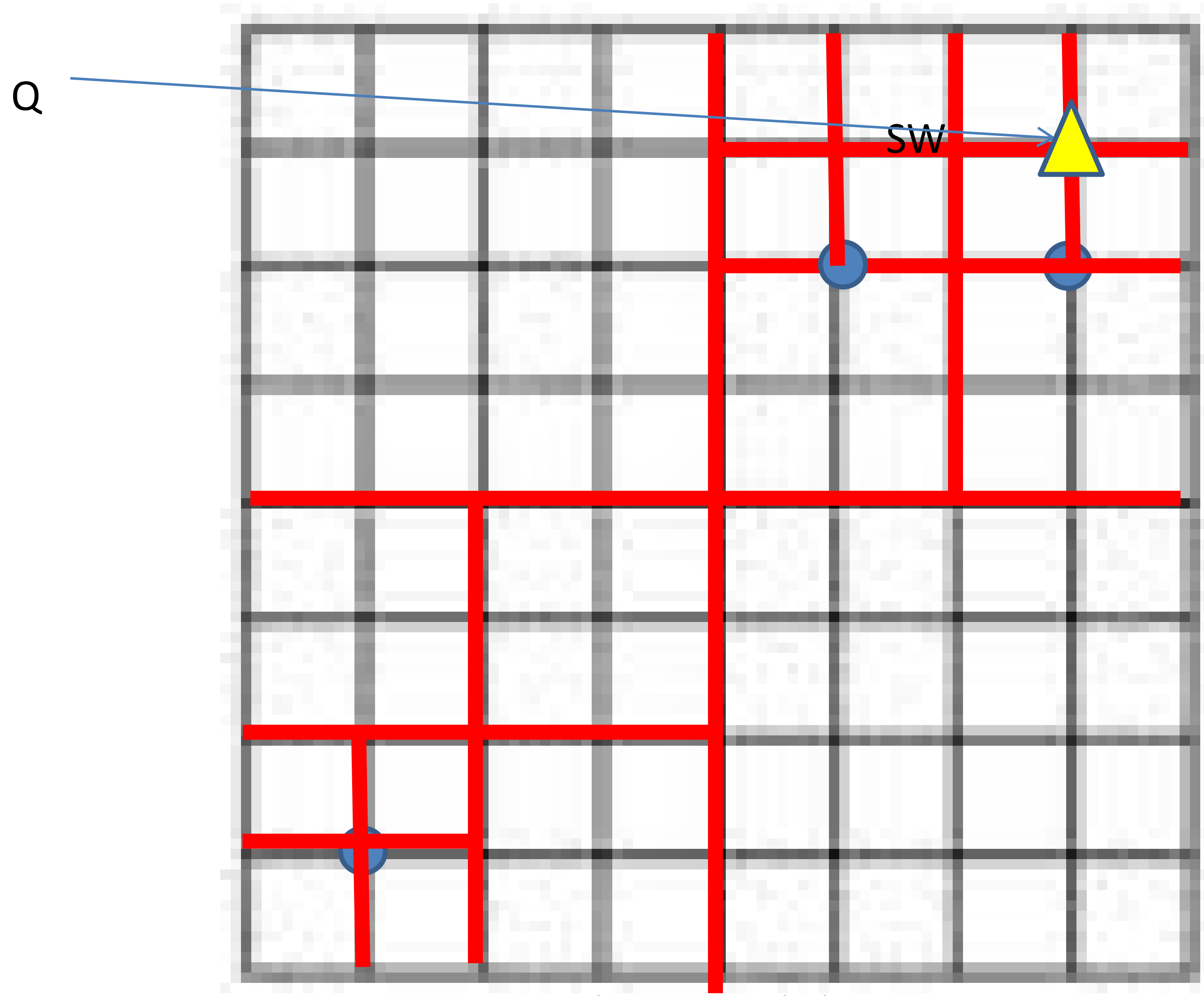
- INPUT:
  - Pointer T to the root of an MX-quadtree.
  - Point Q (not necessarily with integer coordinates).
- OUTPUT:
  - Any nearest neighbor of Q.

Nearest Neighbor Searches in PR-Quadtrees

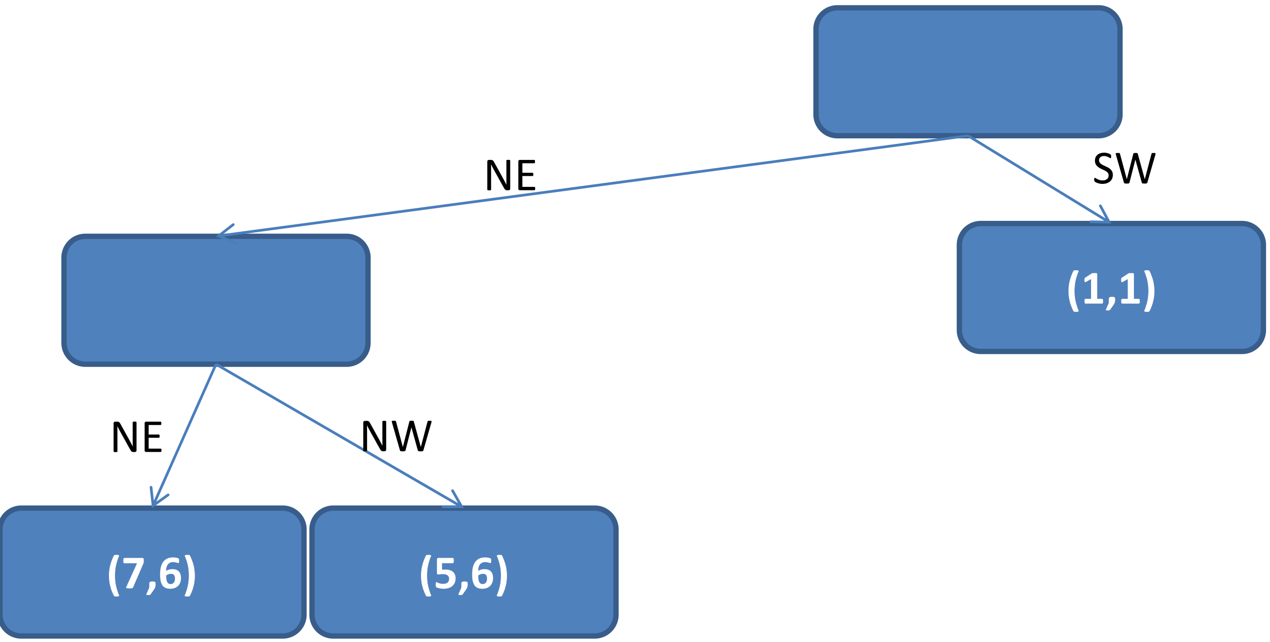
- BestDist = infty, BestSOL = NIL.
- VISIT(N)
  - If N is not a leaf
    - If d(Reg(N),Q) < BestDist then
      - Visit all of N's children
    - Else prune
  - Else
    - If d(N.Point,Q) < BestDist then
      - BestSol = N.Point, BestDist = d(N.Point,Q).
  - Return BestSOL.

Algorithm is the same as for MX-Quadtrees.

Nearest Neighbor Search Example



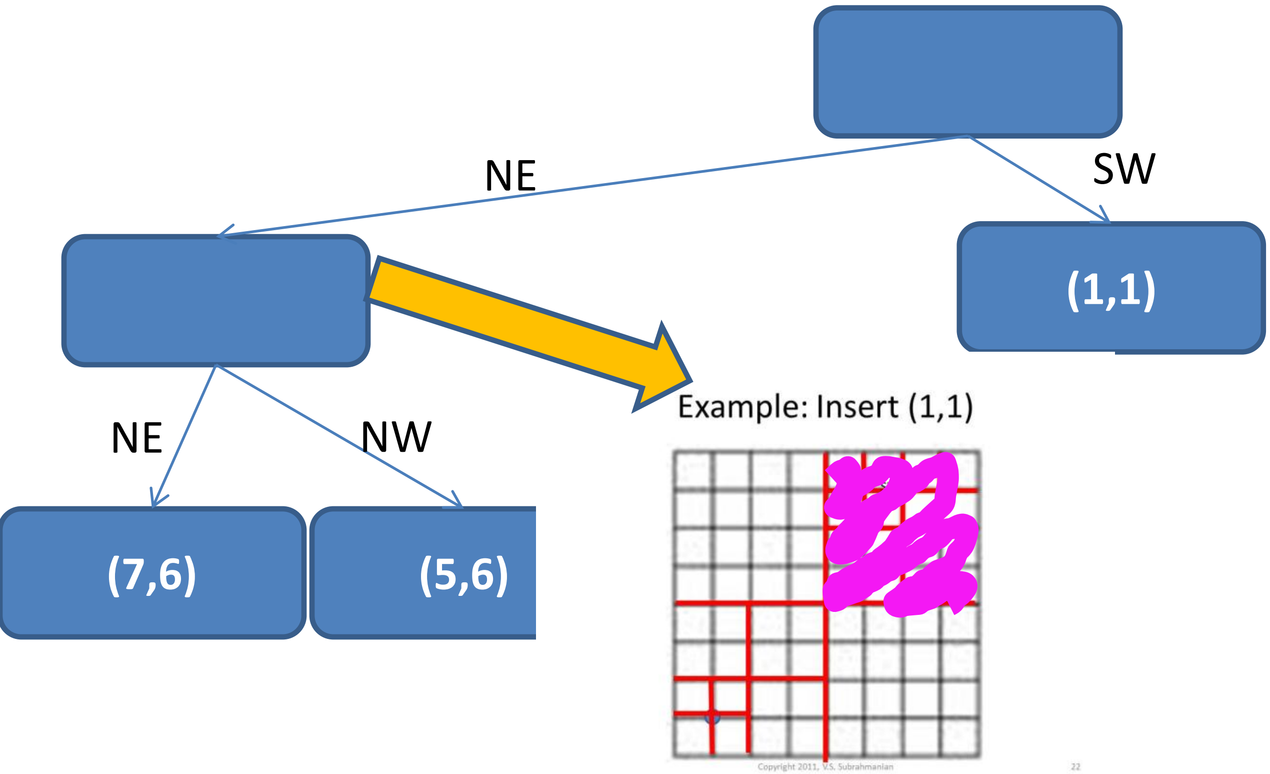
Example NN search Q=(7,7)



BestDist = infty
BestSOL = NIL

Is d(Q,Reg(Root)) < BestDist? Yes, it is 0 – so we must look at both children!

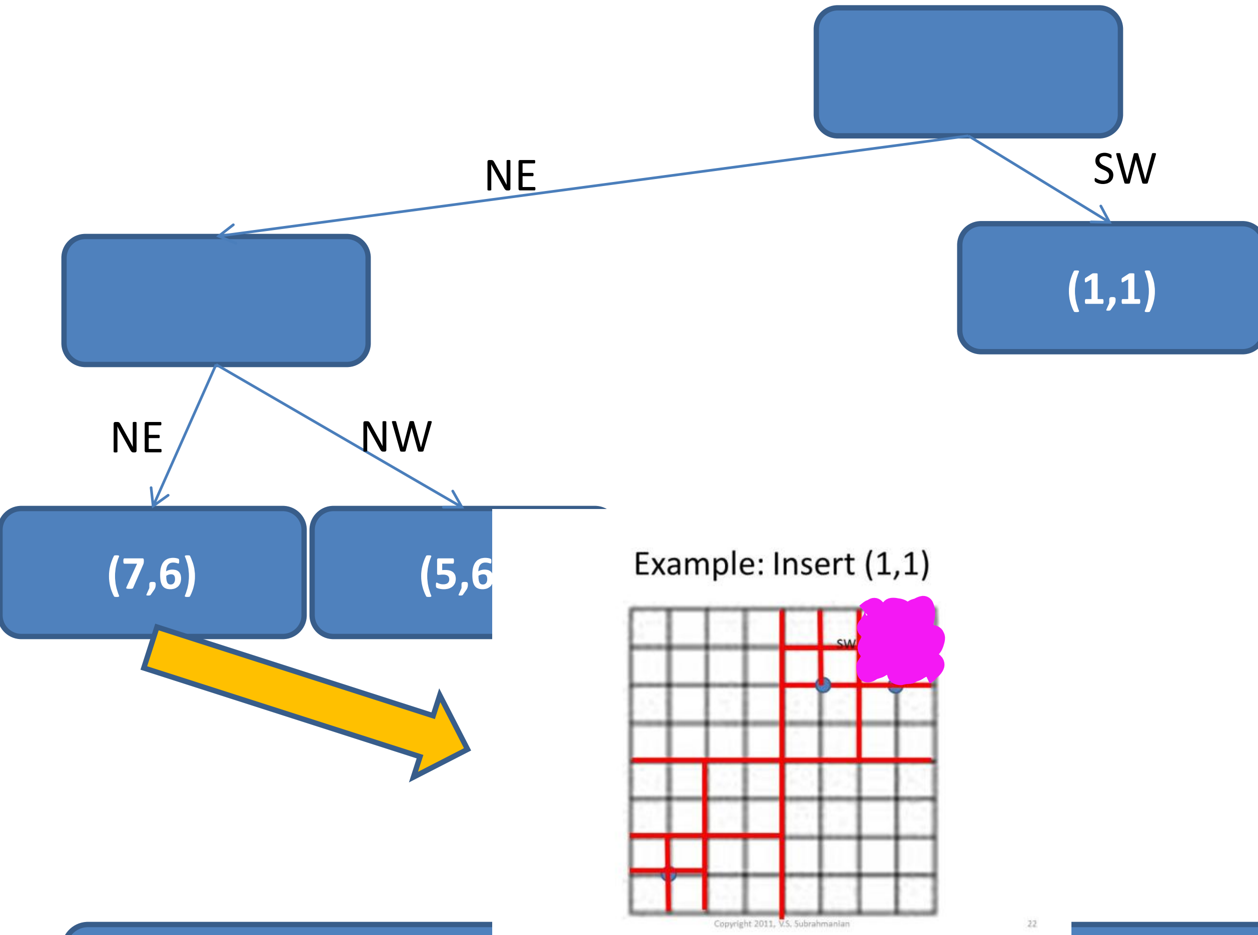
Example NN search Q=(7,7)



BestDist = infty
BestSOL = NIL

Consider the NE child. Is d(Q,Reg(N)) < infty? Yes, it is 0. So must consider both its children.

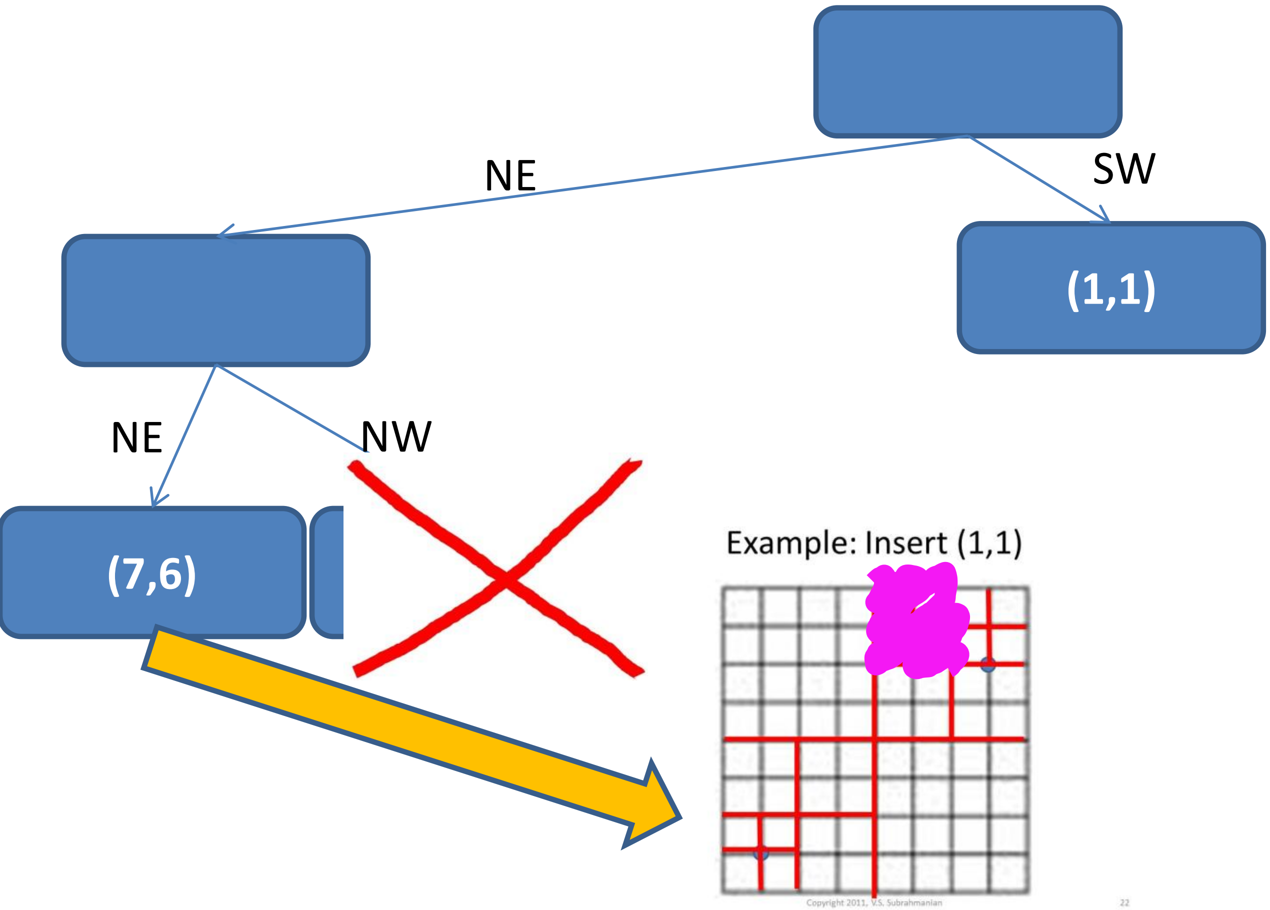
Example NN search Q=(7,7)



BestDist = 1
BestSOL = (7,6)

Consider the NE child. Is d(Q,Reg(N)) < infty? Yes, it is 0. Moreover, this node is a leaf. So compute d((7,7),(7,6)) = 1. Update BestDist and BestSOL.

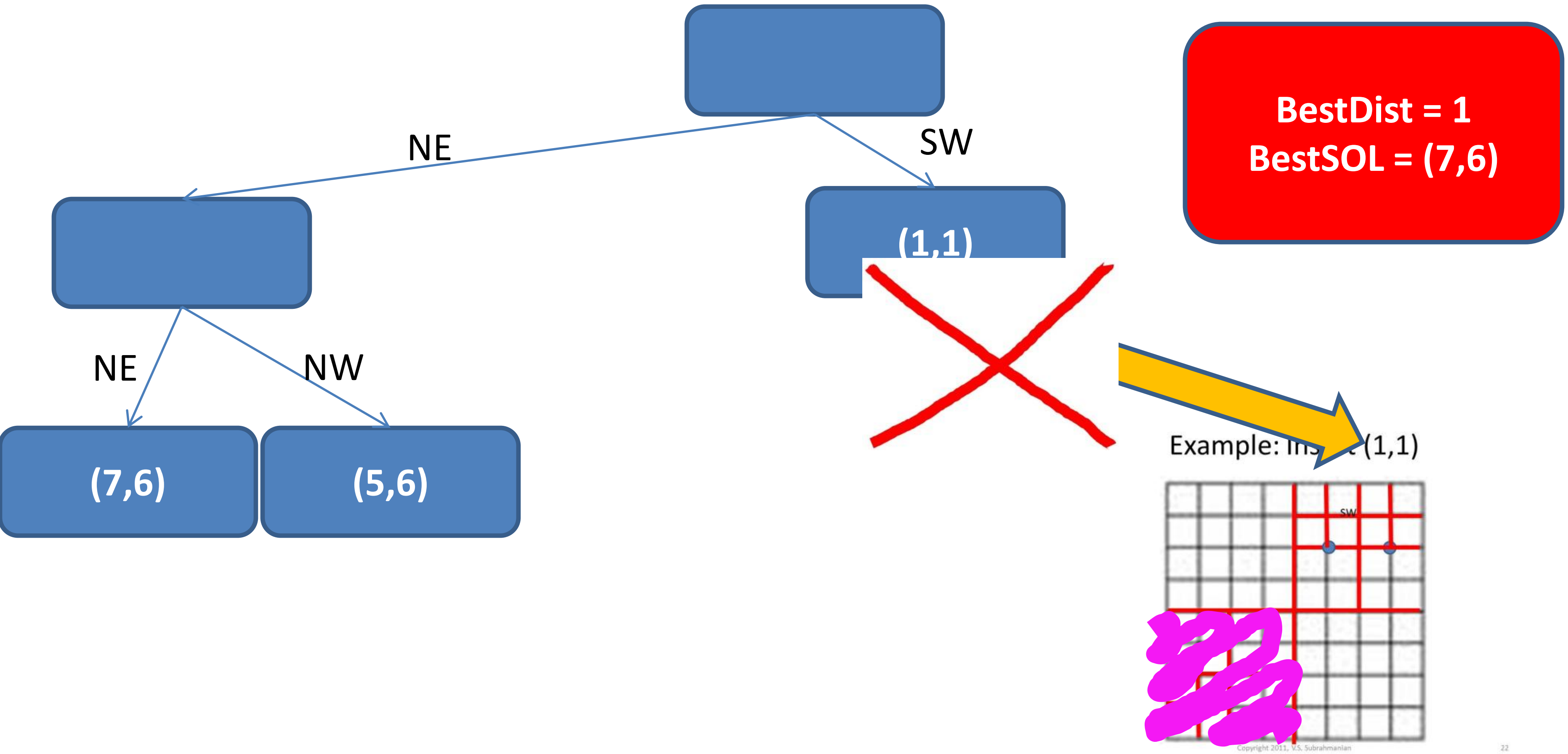
Example NN search Q=(7,7)



BestDist = 1
BestSOL = (7,6)

Consider the NW child. Is d(Q,Reg(N)) < 1? It is just over 1. So prune.

Example NN search Q=(7,7)



Consider the SW child. Is d(Q,Reg(N)) < 1? No. So prune..

What about Deletion?

- Really easy.
- All points are stored at leaves.
- Deletion algorithm sketch:
  - Search for point
  - If point is found
    - Set appropriate link of its parent P to NIL
    - If all 4 of P’s child links are NIL, then set the appropriate link of the parent’s parent to NIL. (“Collapsing step”).
    - If 3 of P’s child links are NIL and the one other child is a leaf node, then copy the entry in that leaf into P and delete the leaf.
- **Deletion algorithm is exactly identical to the one for MX-Quadtrees.**

Discussion

- Compare the efficiency of MX vs. PR-quadtrees w.r.t.
  - Range searches
  - NN searches
  - Deletion.
- Which is better? And why?