PM2-Quadtrees

V.S. Subrahmanian

Spring 2013

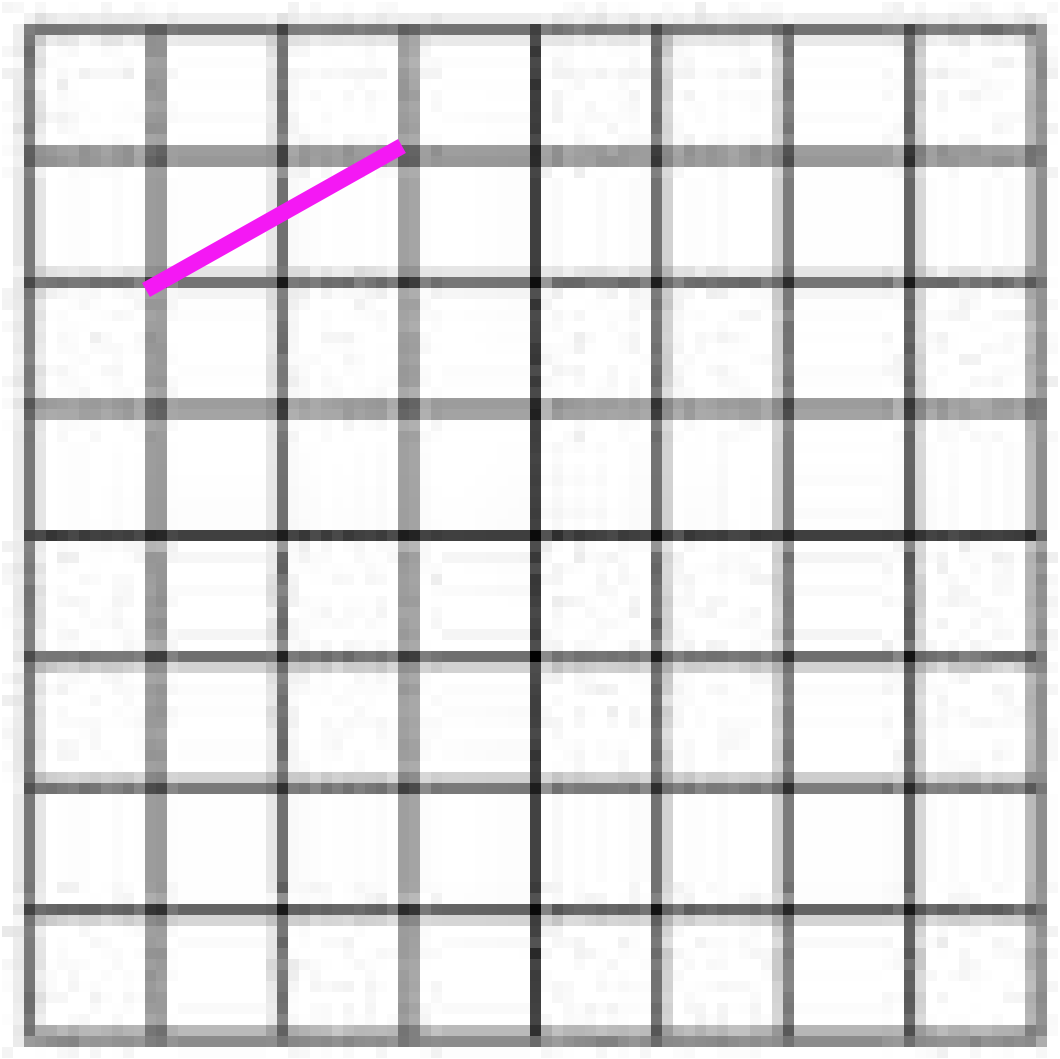
PM1-Quadtrees

- PM1-quadtrees are used to store *straight line segment* data.
- Each line segment has two *vertices* representing the end-points of the line segment.
- PM2-Quadtrees are just like PM1-Quadtrees. Nodes always split regions into equal-sized subregions.
- Like PR-quadtrees, PM1-quadtrees assume that
  - The overall region is a 2^n x 2^n region for some n
  - All vertices x,y are integers ranging from 0 up to (and including) $2^n - 1$.
- All data is stored in leaf nodes.
- But there is one exception.

PM2-Splitting Rules

- Each node must be split if
  - **(vertex split condition)** it contains 2 or more vertices or
  - **(line split condition)** It contains 2 or more line segments unless all of those line segments meet either
    - at a vertex within the region associated with a node.
    - at a common vertex outside the region.

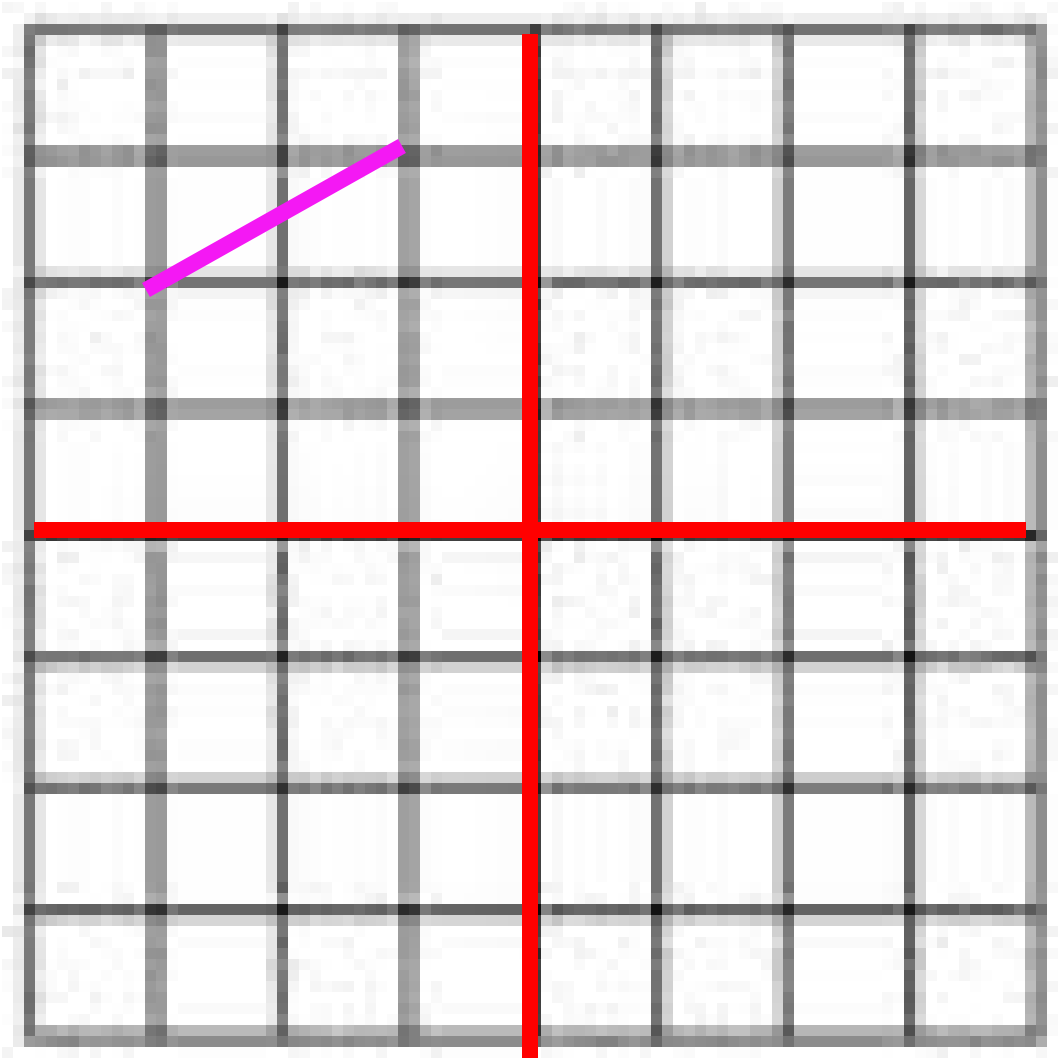
Example: Insert Line L1



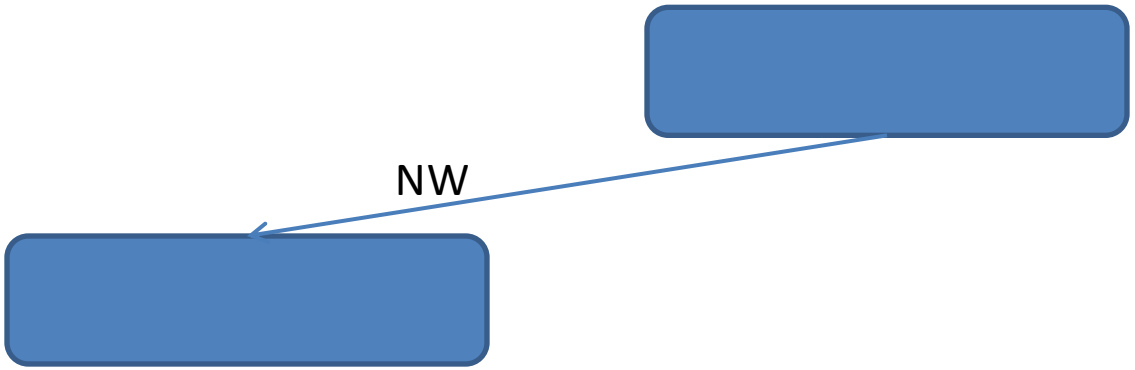
Insert L1

The region associated with the root only contains two vertices. So we must split the root.

Example: Insert Line L1

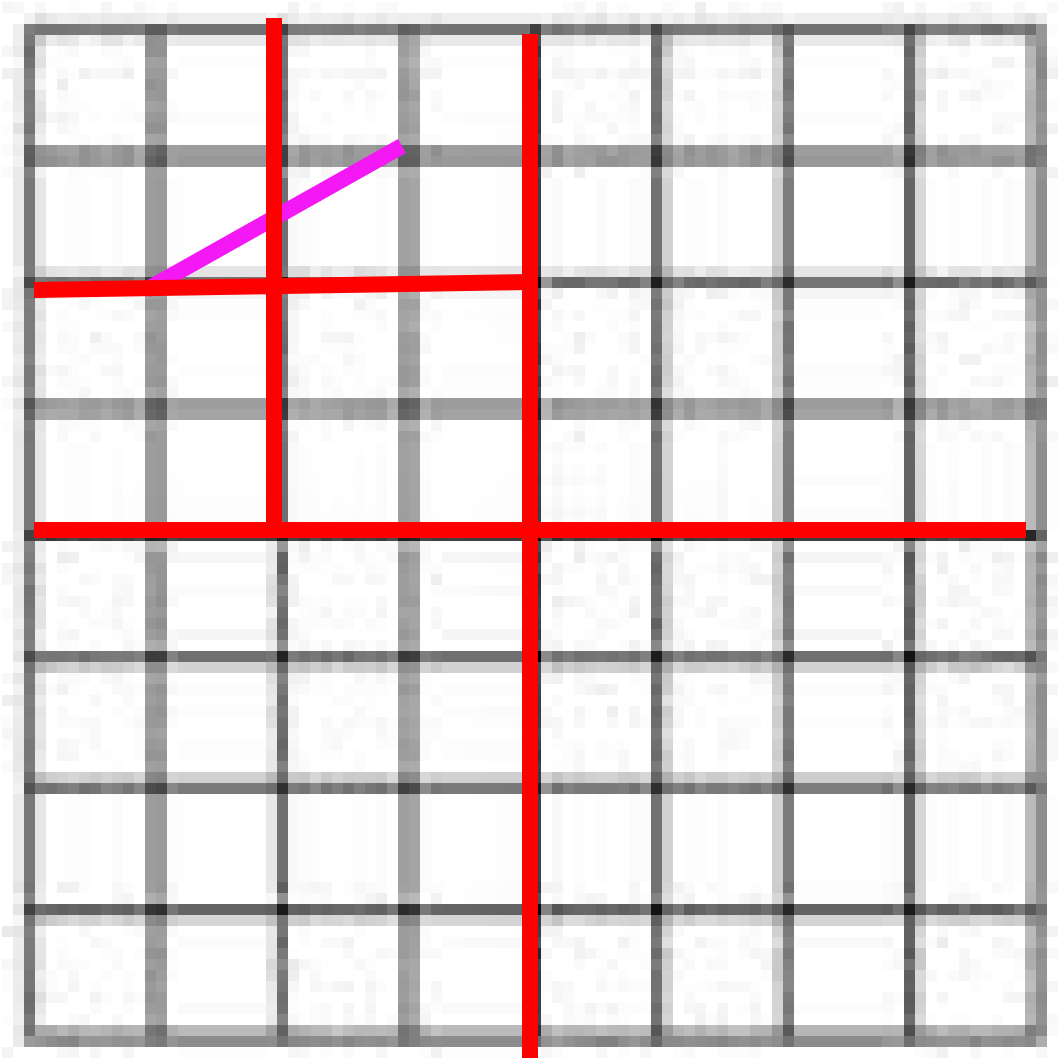


Insert L1

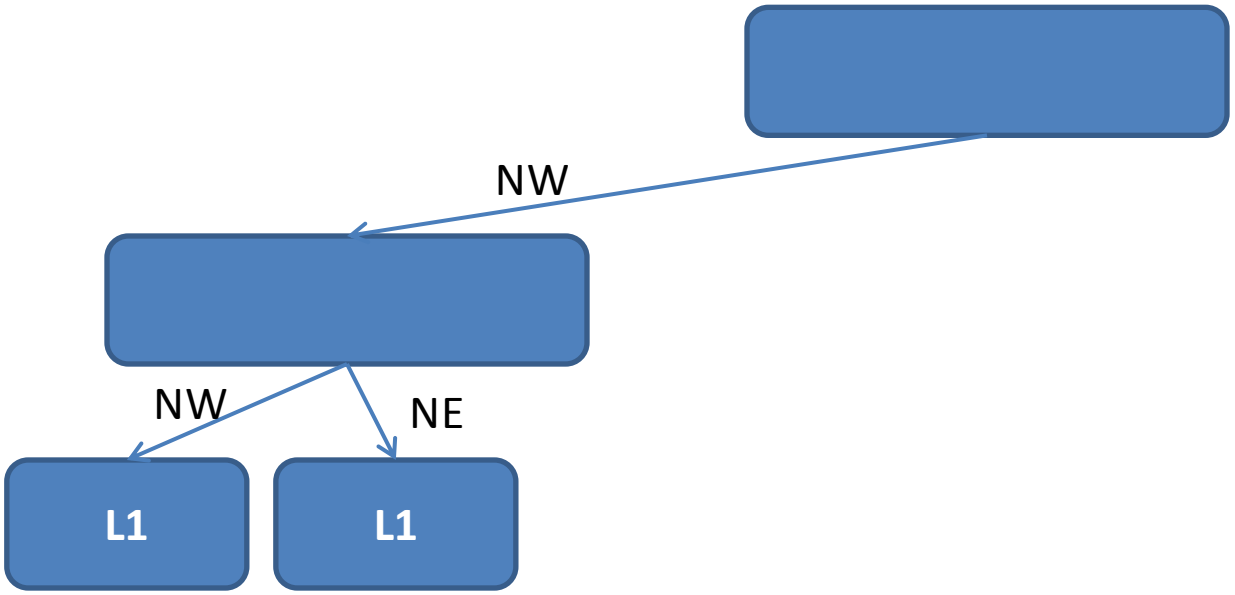


The entire line L1 lies within the NW quadrant of the root. So create an NW child. Must still split because the NW quadrant contains two vertices.

Example: Insert Line L1

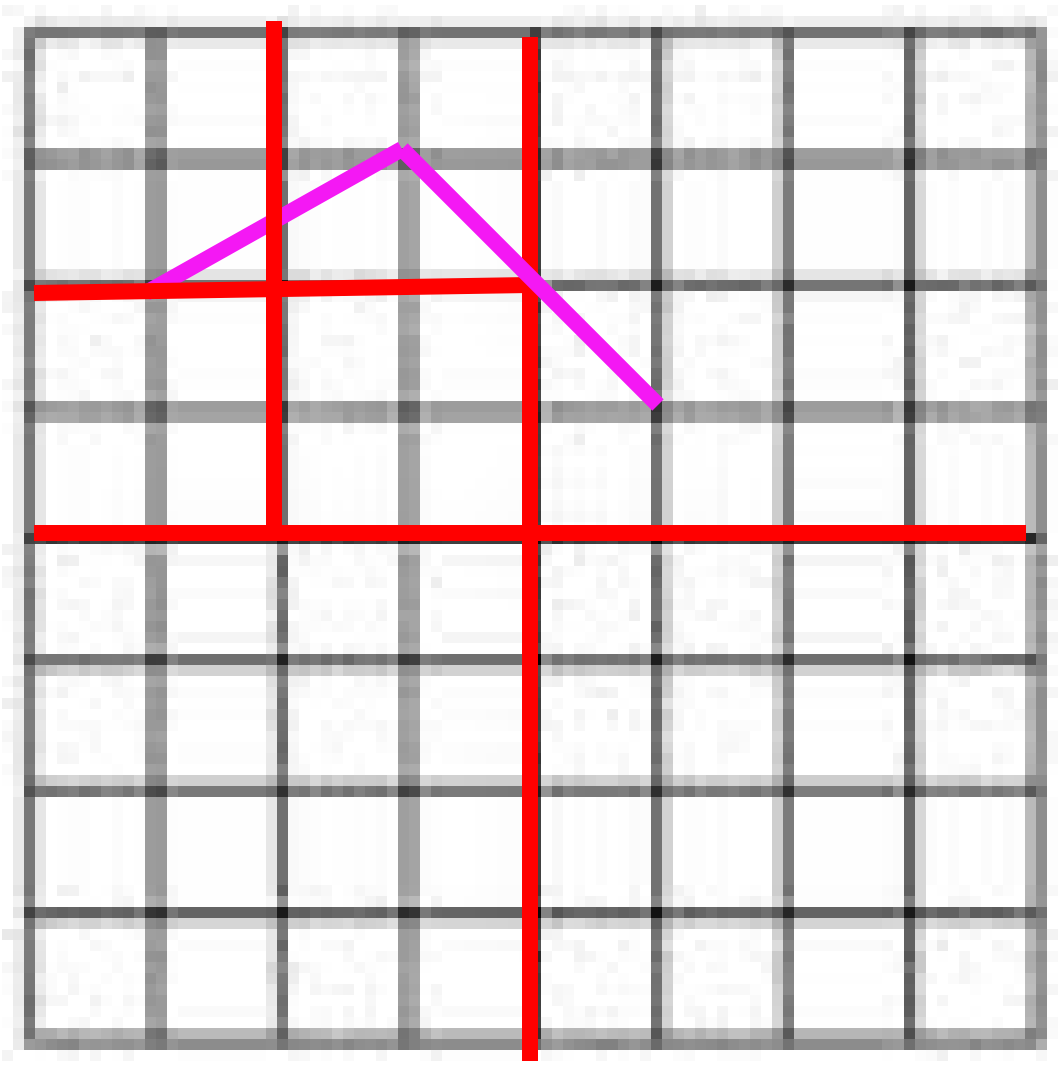


Insert L1

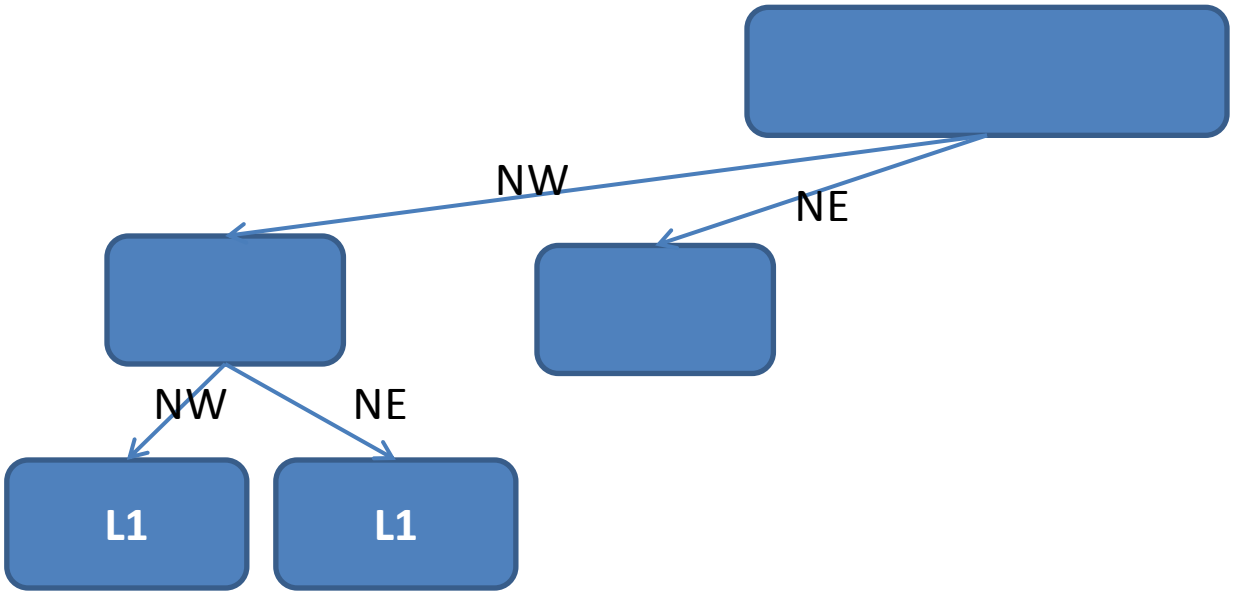


Line L1 intersects both the NE and NW subquadrants– so must create those nodes. No need to split this any further! Label both nodes with L1.

Example: Insert Line L2

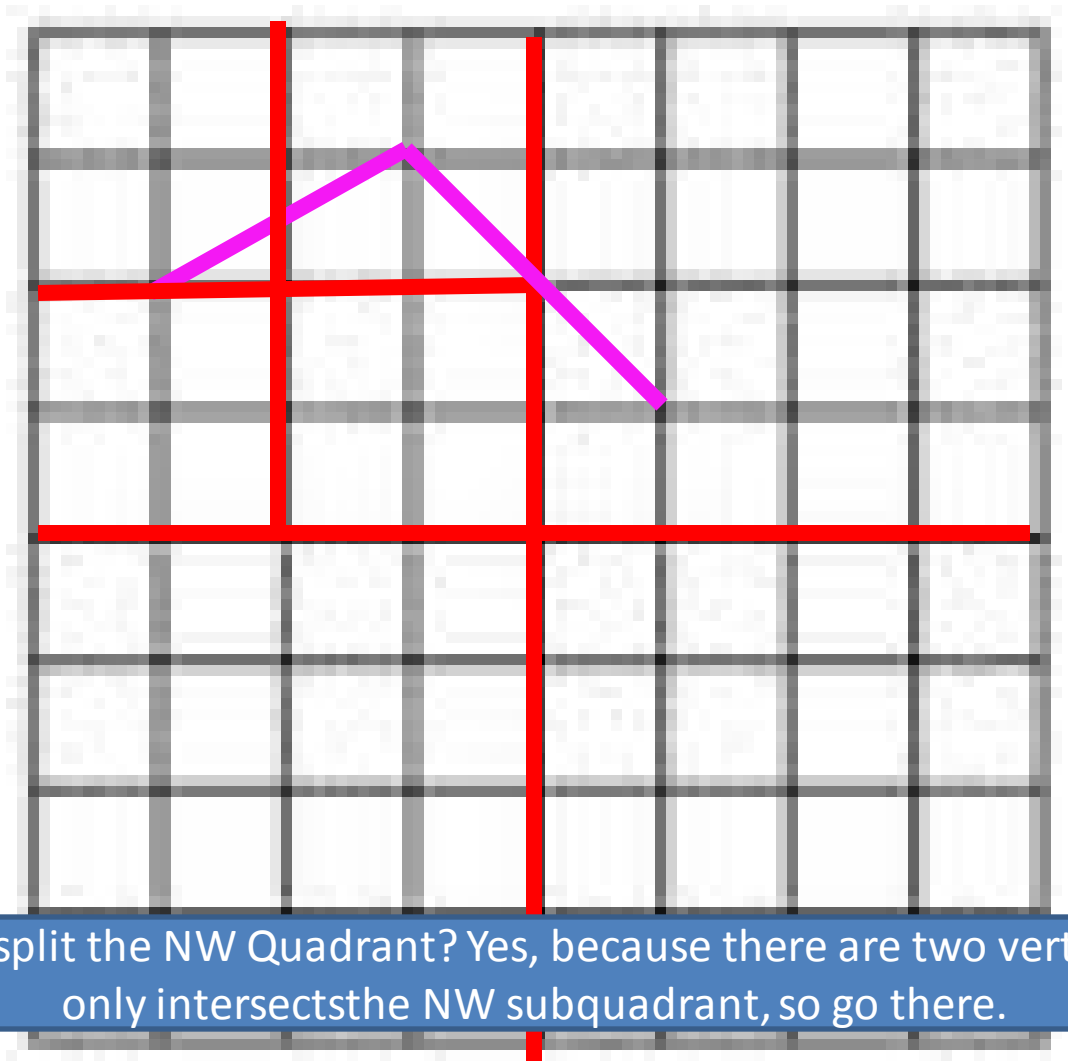


Insert L2



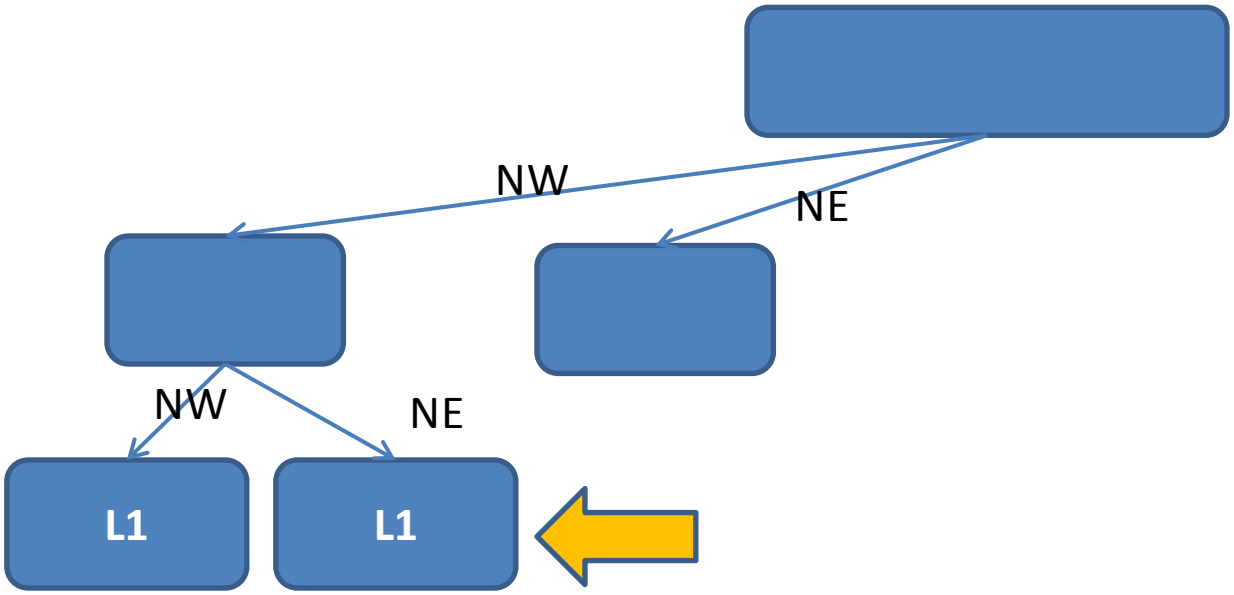
Line L2 intersects both the NE and NW quadrants of the root – so must create a NE child and continue inserting L2 into both. Let's look at the NW quadrant first.

Example: Insert Line L2



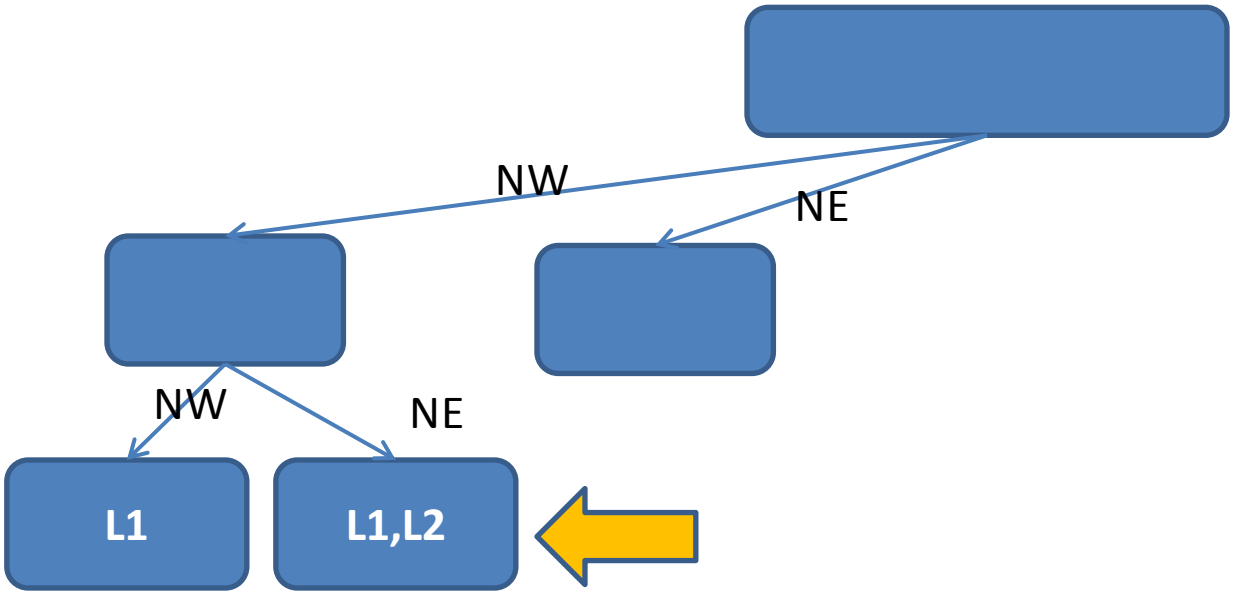
Do we need to split the NW Quadrant? Yes, because there are two vertices already. L2 only intersectsthe NW subquadrant, so go there.

Insert L2



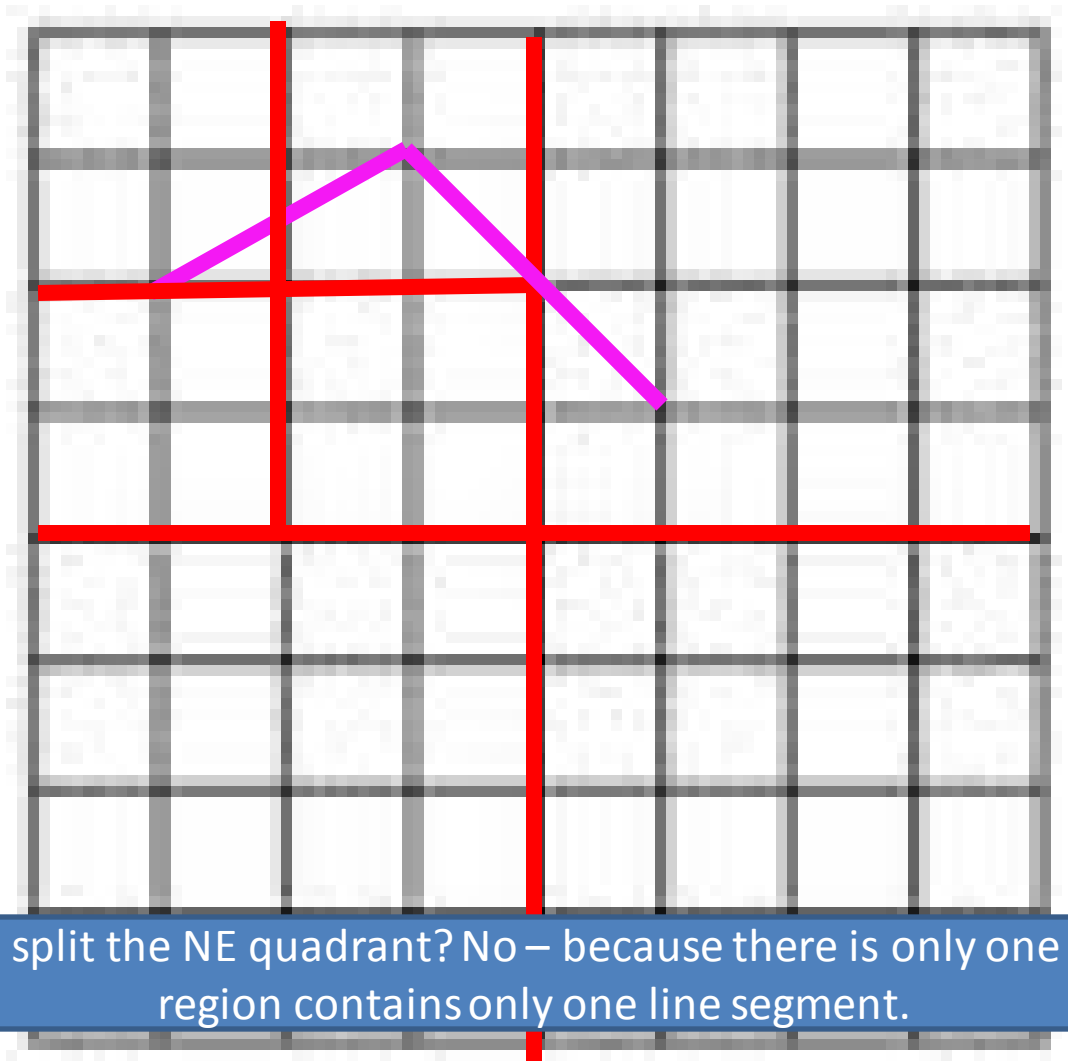
Do we need to split this node? No – because this region contains only one vertex. It has two lines which meet at a common point in this interval. So add L2 as the label to this node.

Insert L2



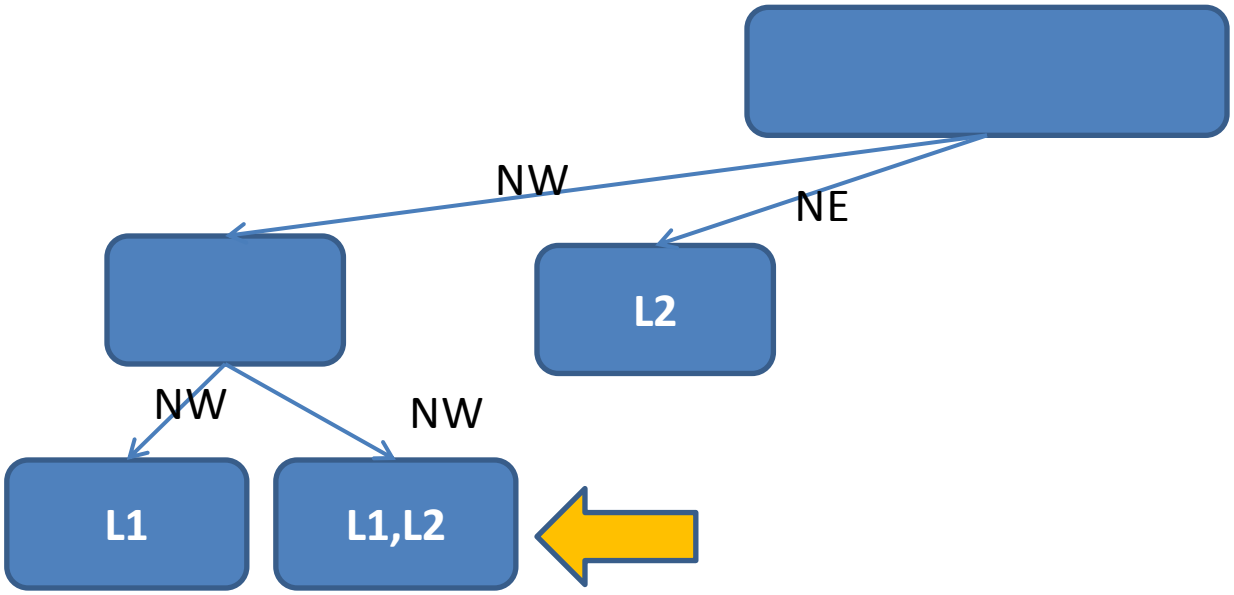
Now consider the NE quadrant of the root.

Example: Insert Line L2



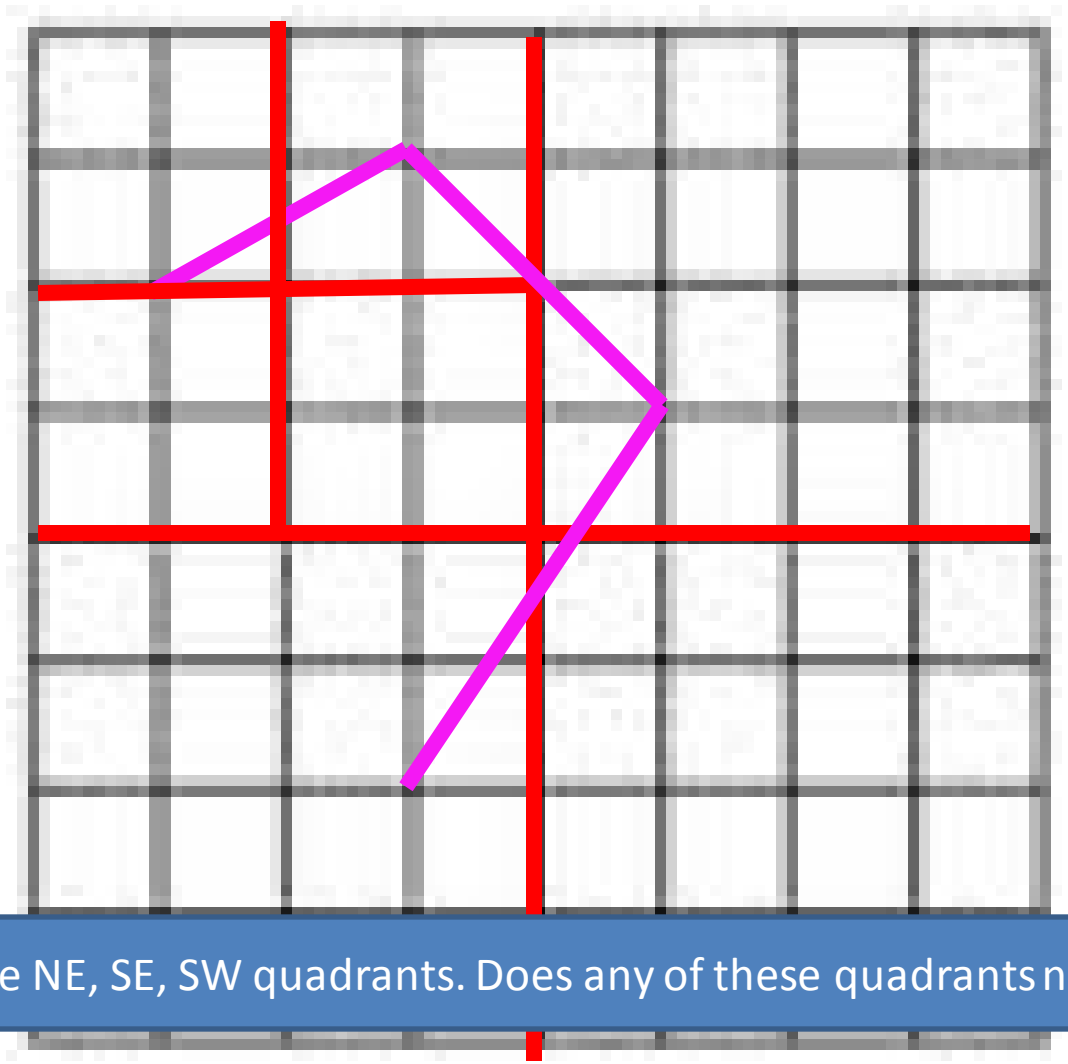
Do we need to split the NE quadrant? No – because there is only one vertex and the region contains only one line segment.

Insert L2



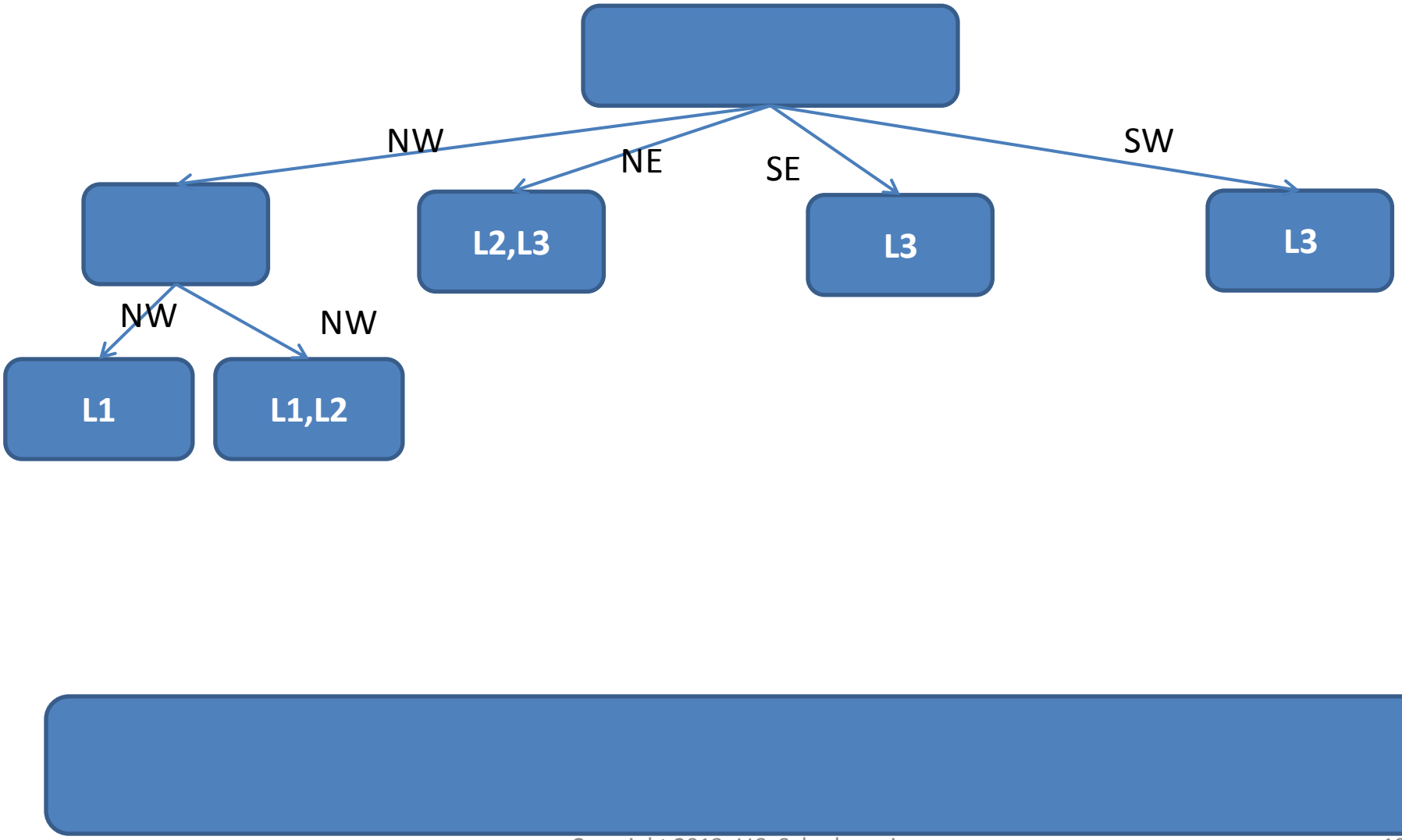
So label the NE child also with L2.

Example: Insert Line L3

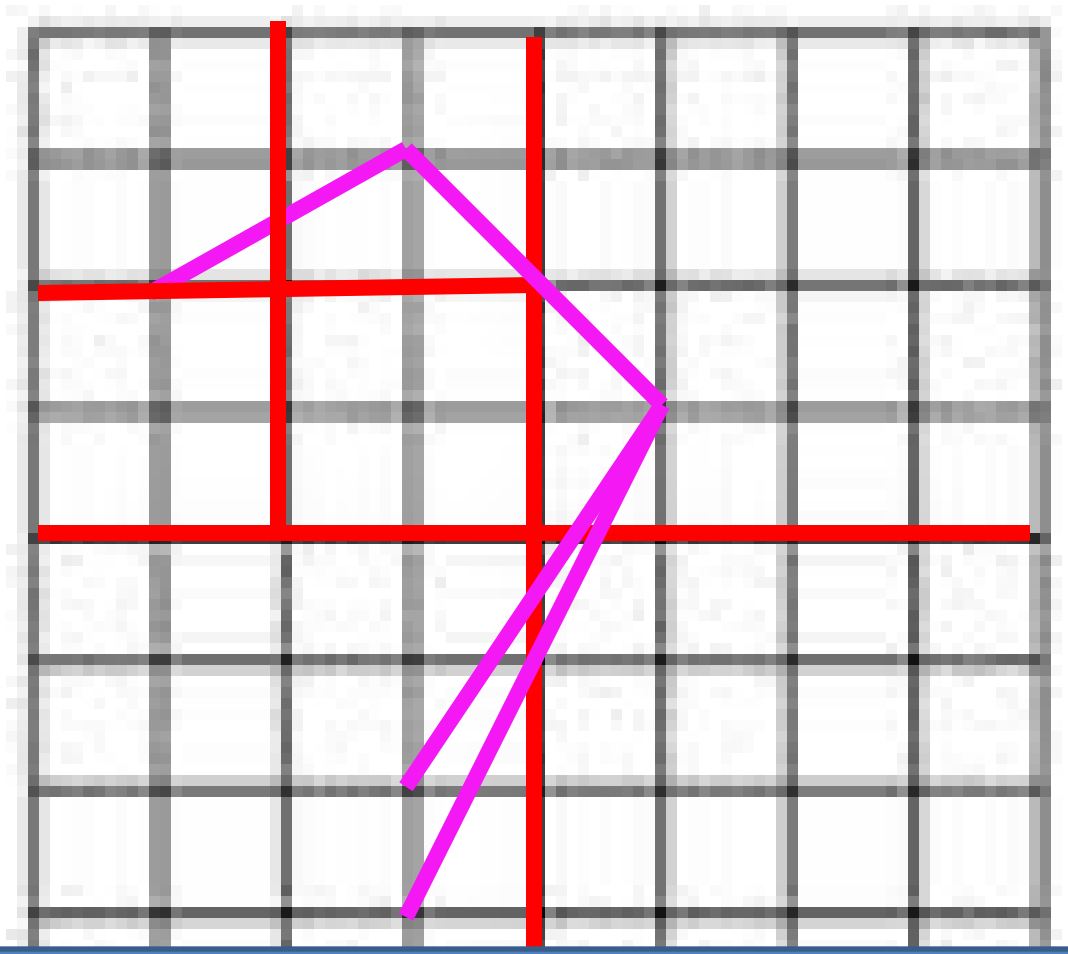


L3 intersects the NE, SE, SW quadrants. Does any of these quadrants need to be split?

Insert L3

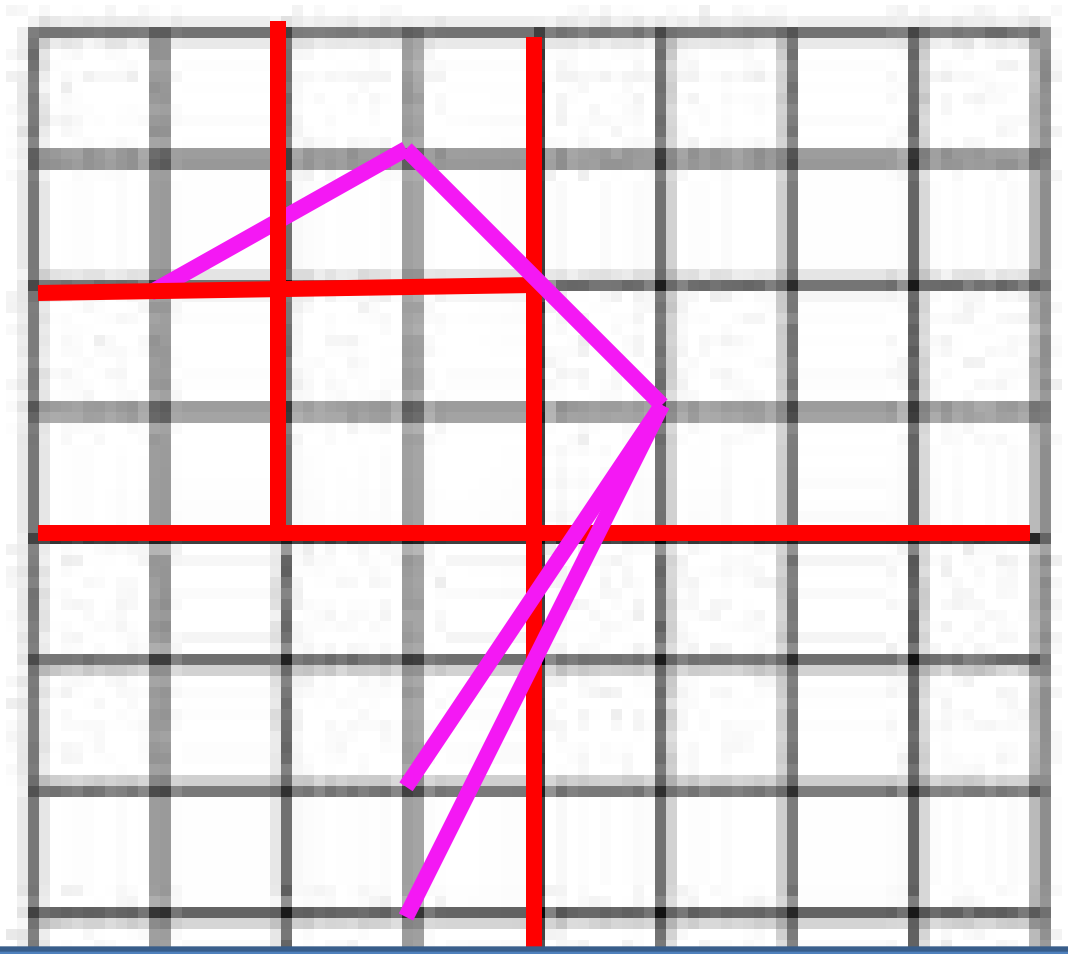


Example: Insert Line 4



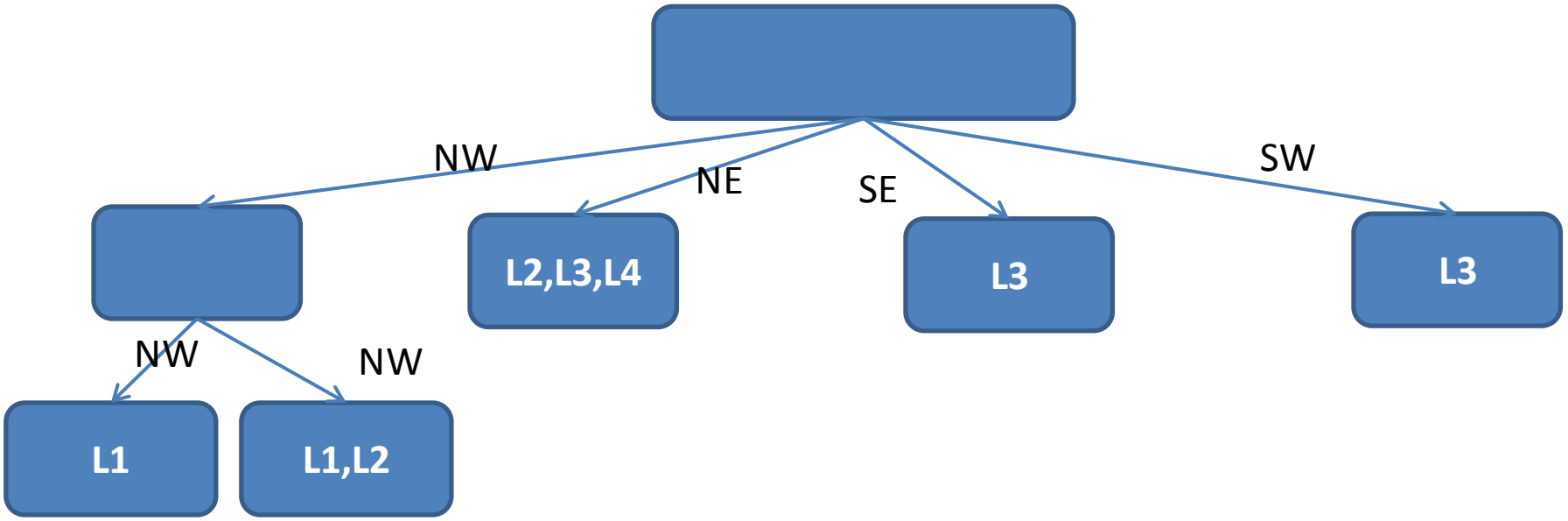
L4 intersects the NE, SE, SW quadrants. Does any of these quadrants need to be split?

Example: Insert Line 4



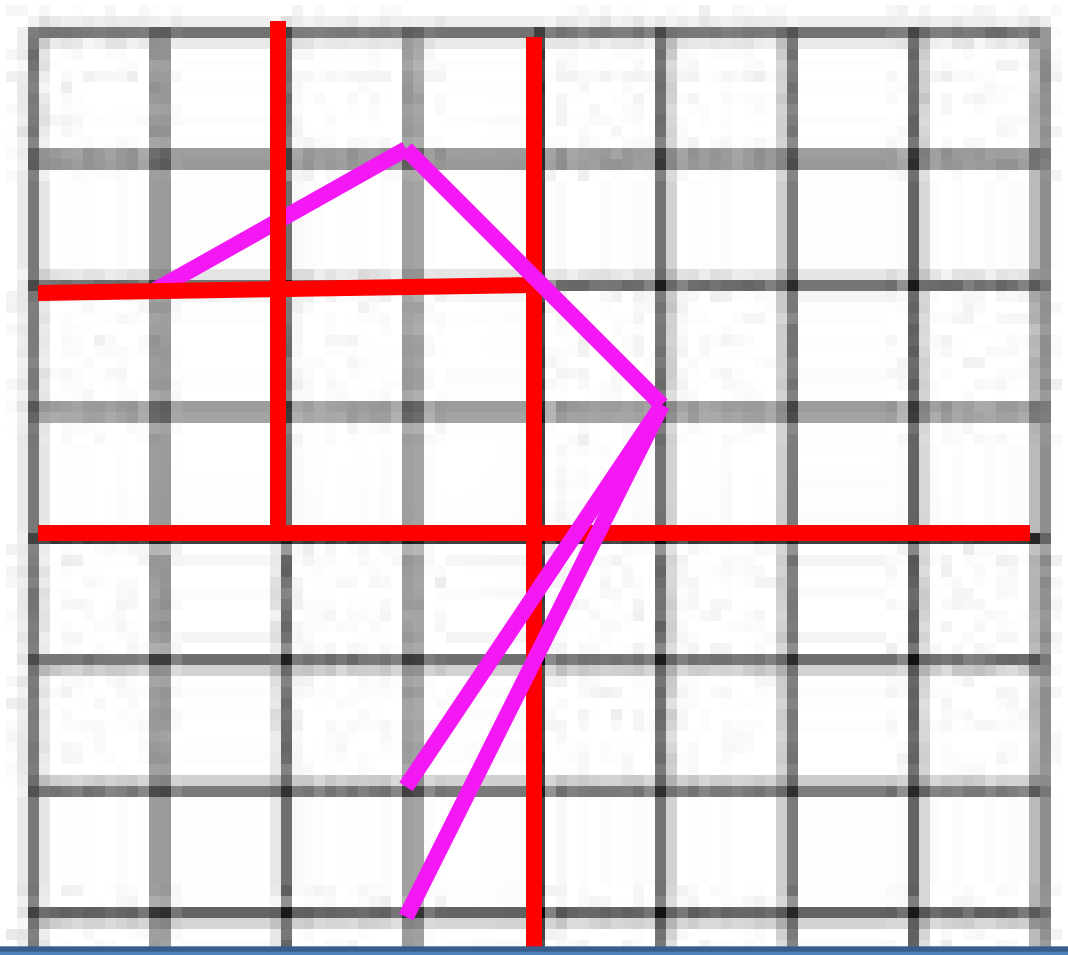
The NE quadrant need not be split.

Insert L4



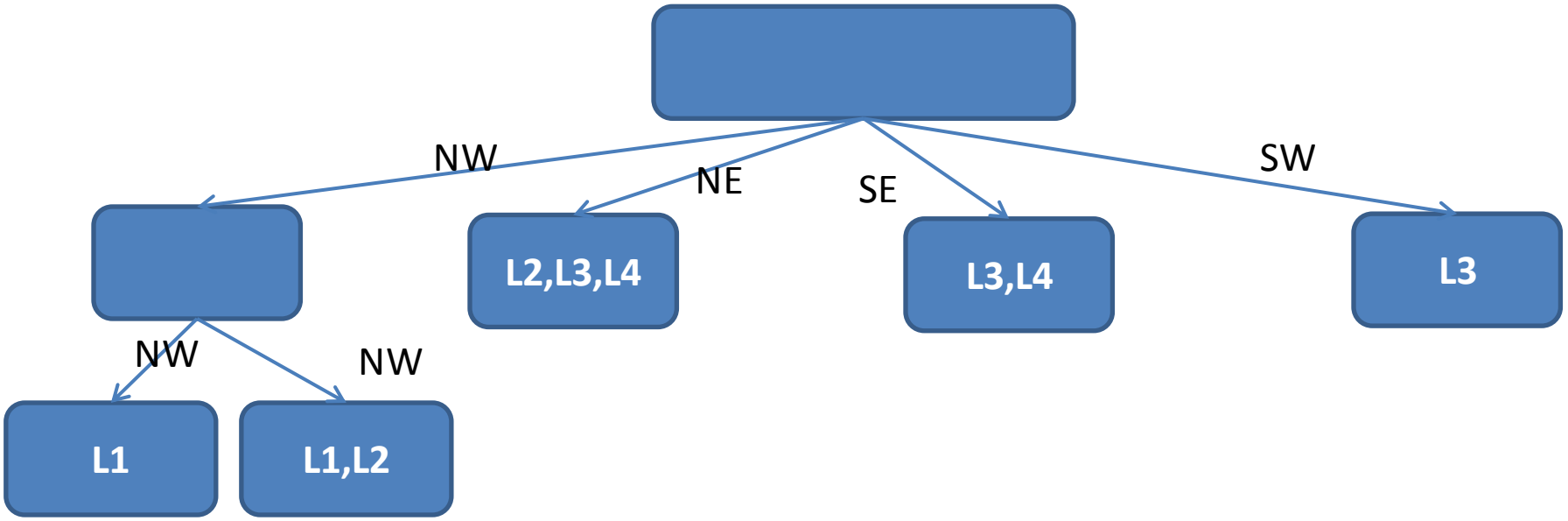
Label the NE quadrant with L4

Example: Insert Line 4



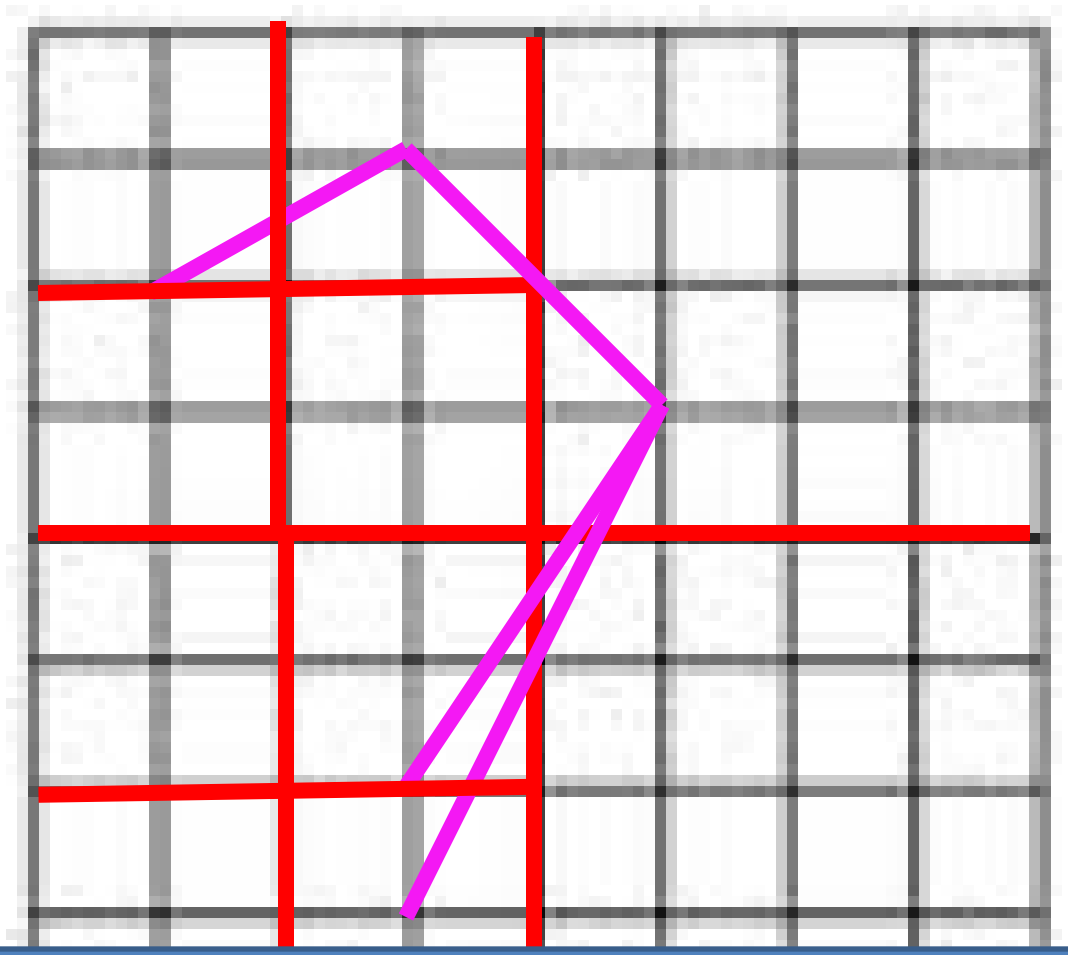
The SE quadrant need not be split because the two line segments in this region meet at a common vertex (outside the region). What would happen with a PM1-quadtree?

Insert L4



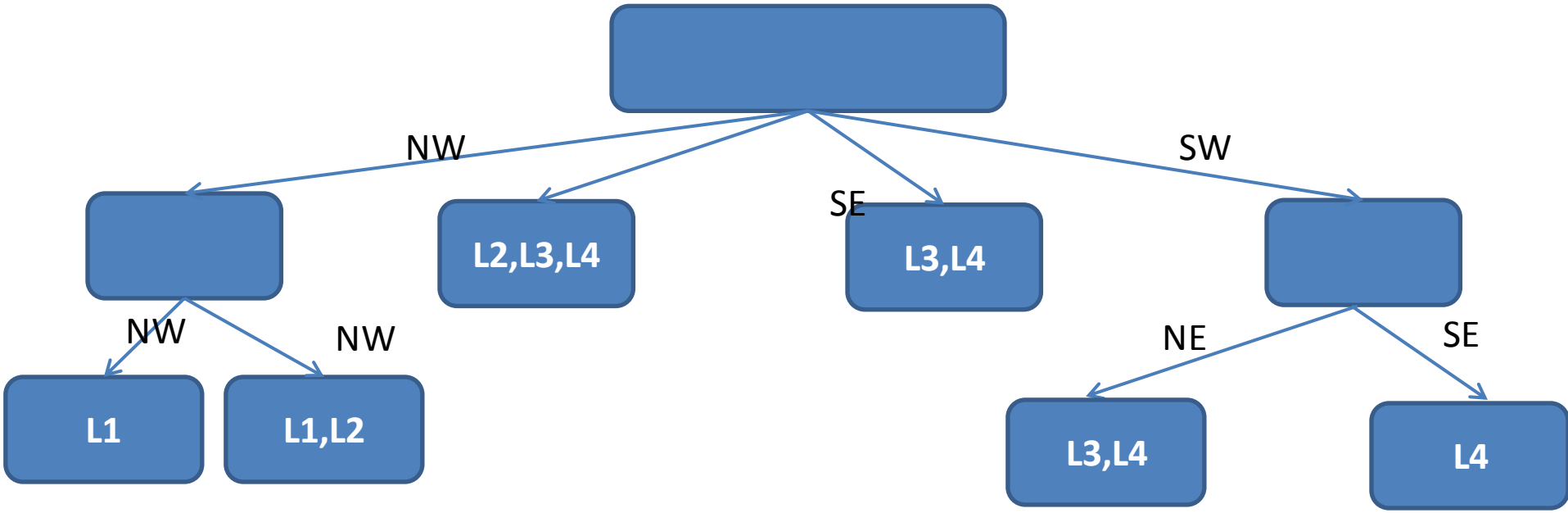
Label the NE quadrant with L4

Example: Insert Line 4



The SW quadrant must be split because it has two vertices in it – Line L3 goes to the NE quadrant and L4 to the SE.

Insert L4

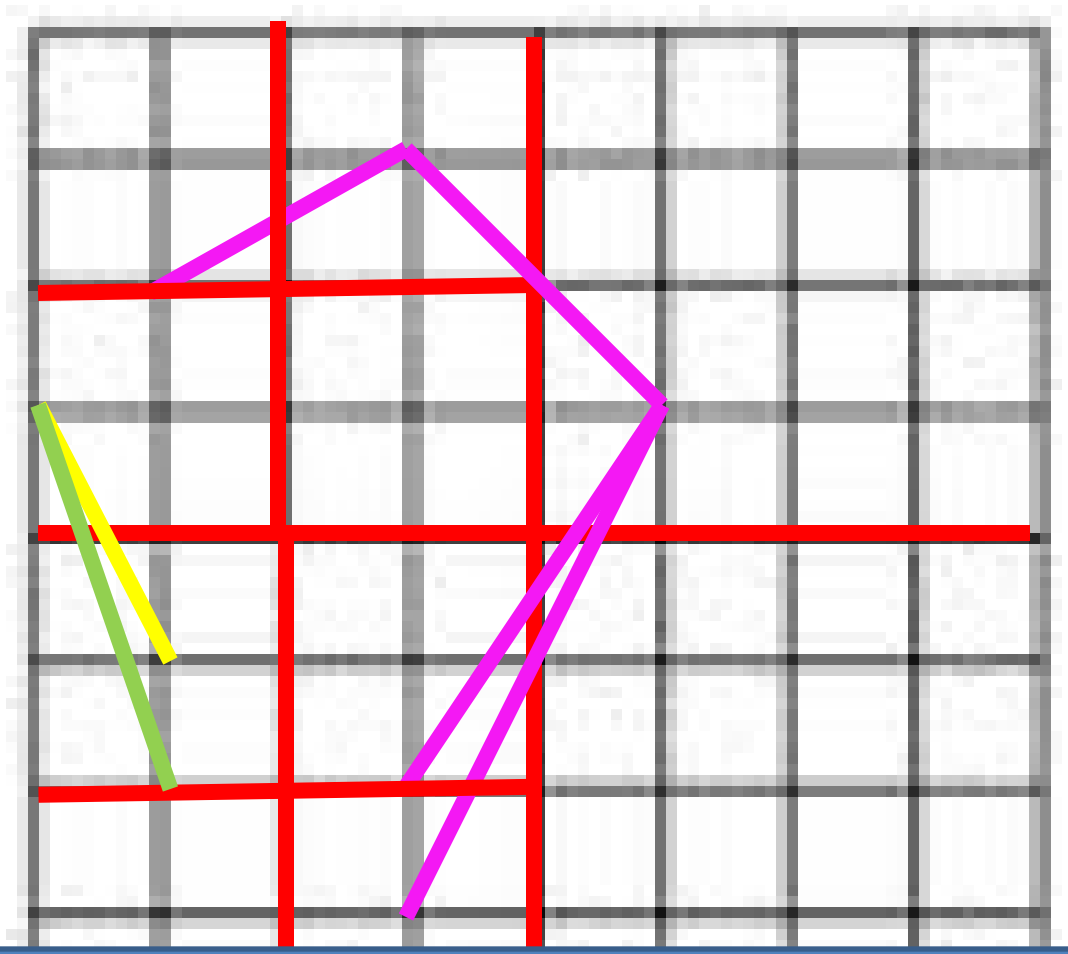


Label the NE quadrant with L4

In-Class Exercise

- Add the green and yellow lines shown to the PM1-quadtree.

Example: Insert green and yellow lines



Range Query

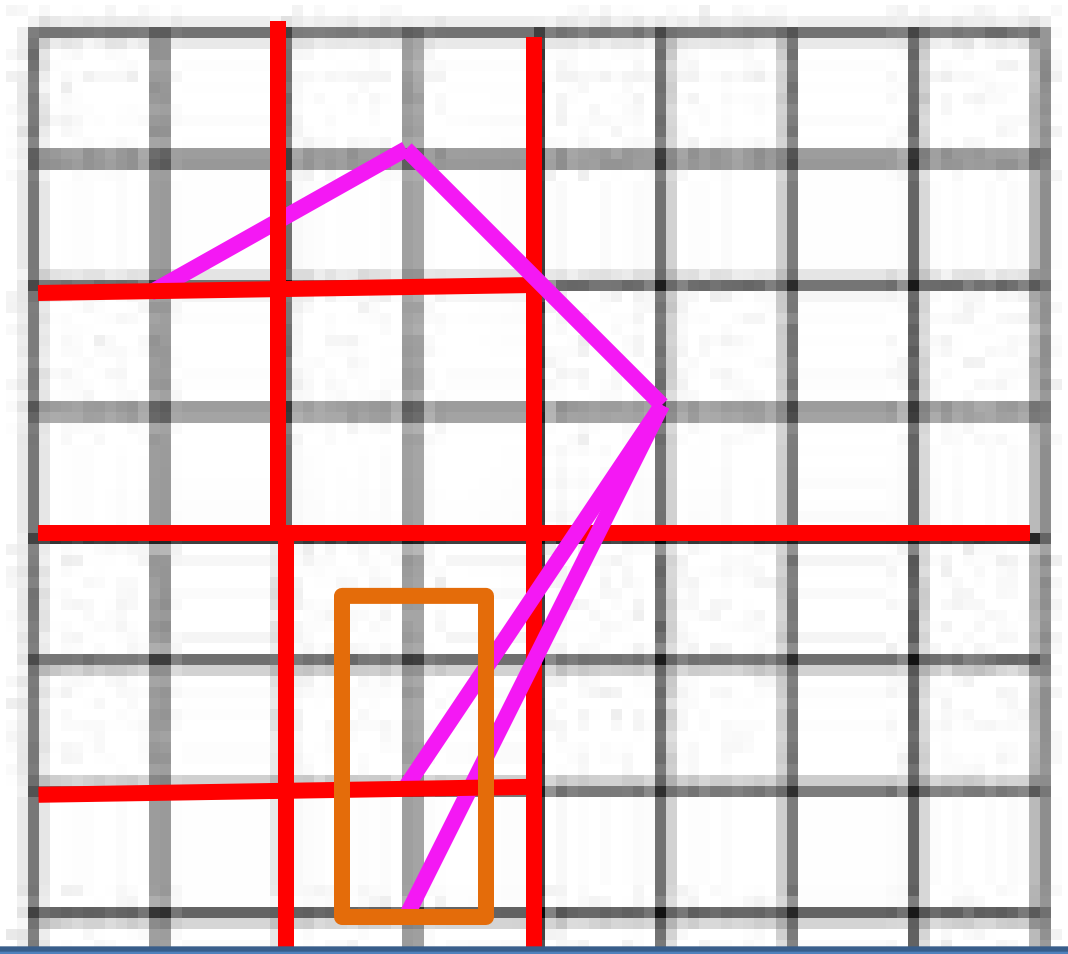
- Input
  - A rectangular region Q and a pointer to the root of a PM2-quadtree.
- Output
  - Find all lines that intersect Q.

- The algorithm is identical to the case of a PM1-quadtree.

Range Query Algorithm Sketch

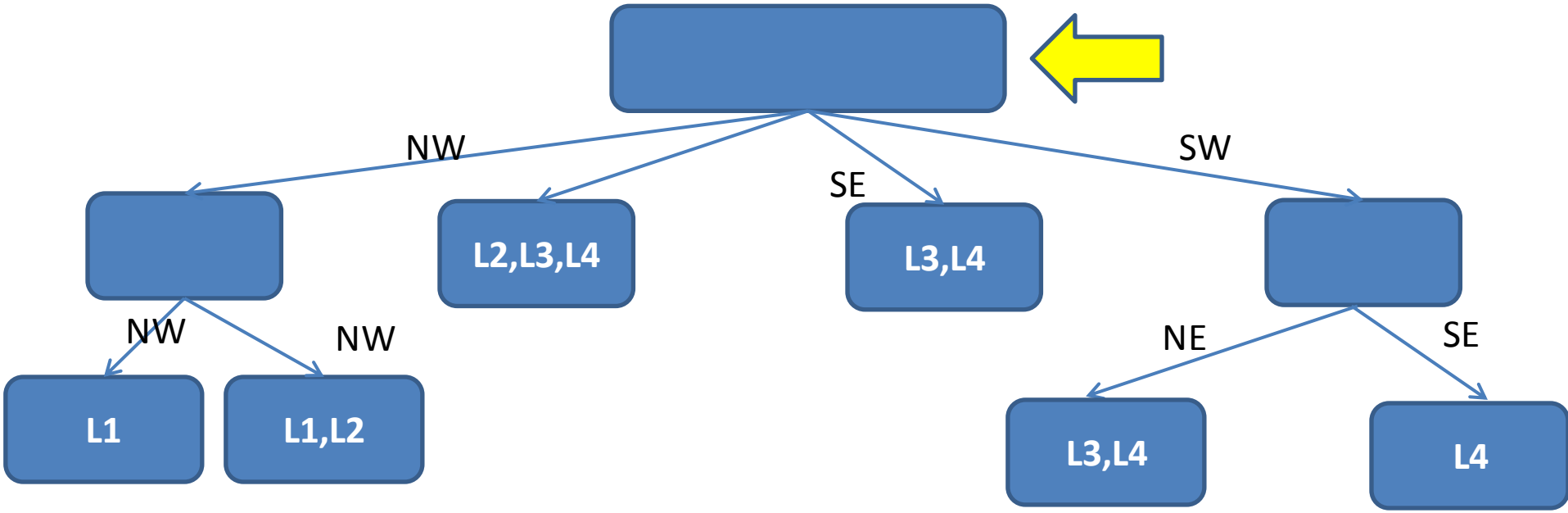
- When Visiting a Node N
  - If Reg(N) does not intersect Q, then prune.
  - Otherwise,
    - If N is a leaf node, check if each line labeling the leaf intersects Q. Add those that do to the answer.
    - If N is not a leaf, visit all of its children recursively.

Example: Range Query



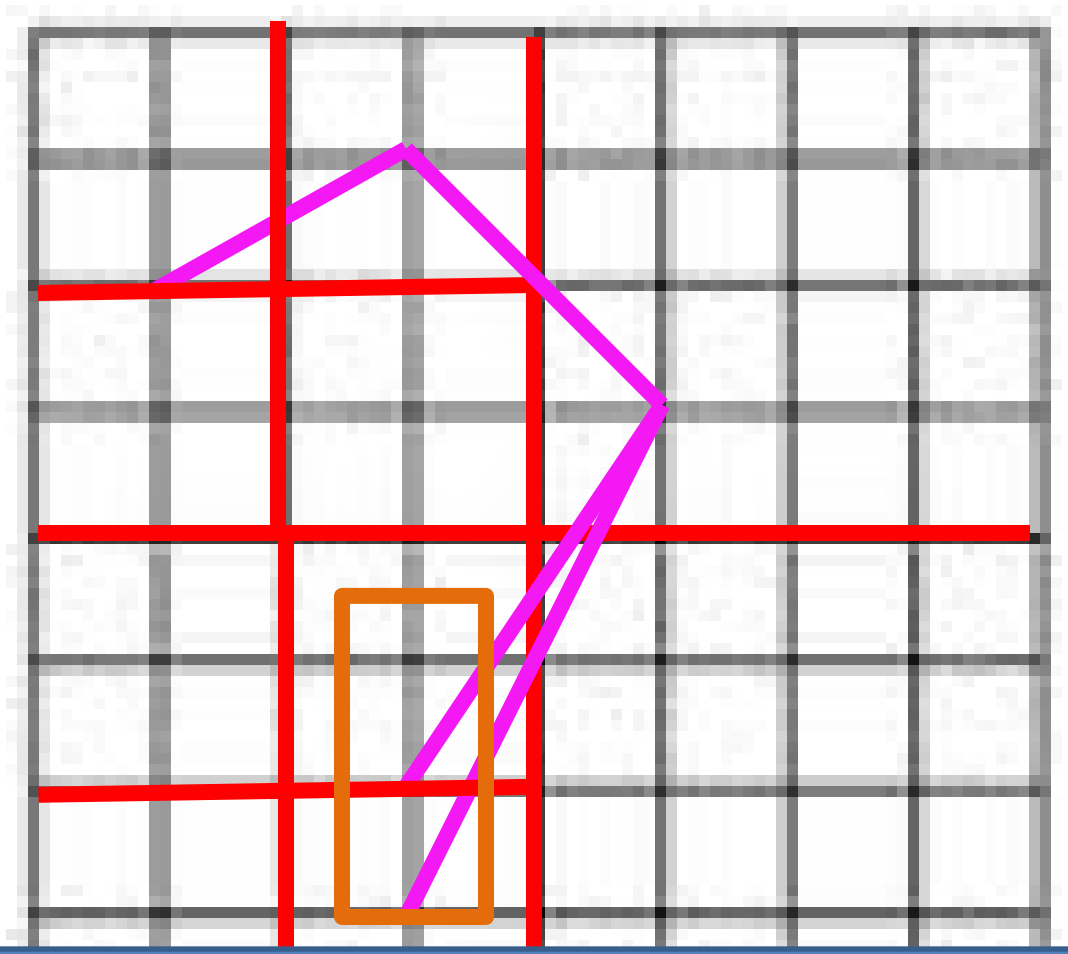
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



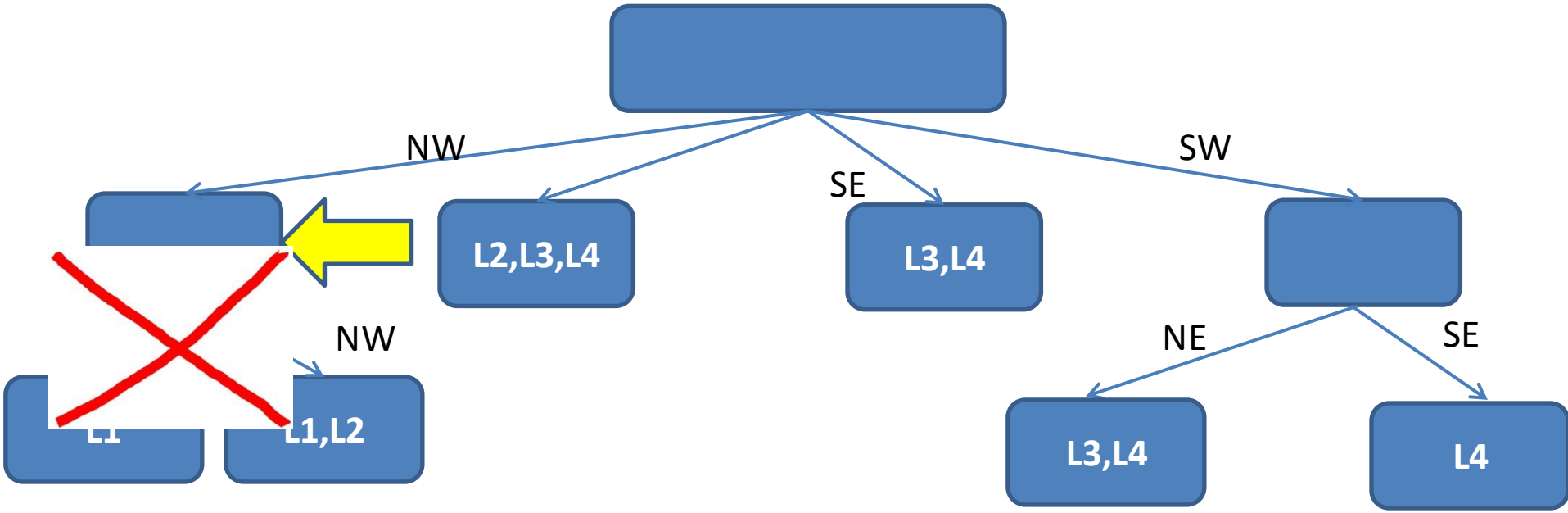
Q intersects Reg(N). N is not a leaf, so explore all its children

Example: Range Query



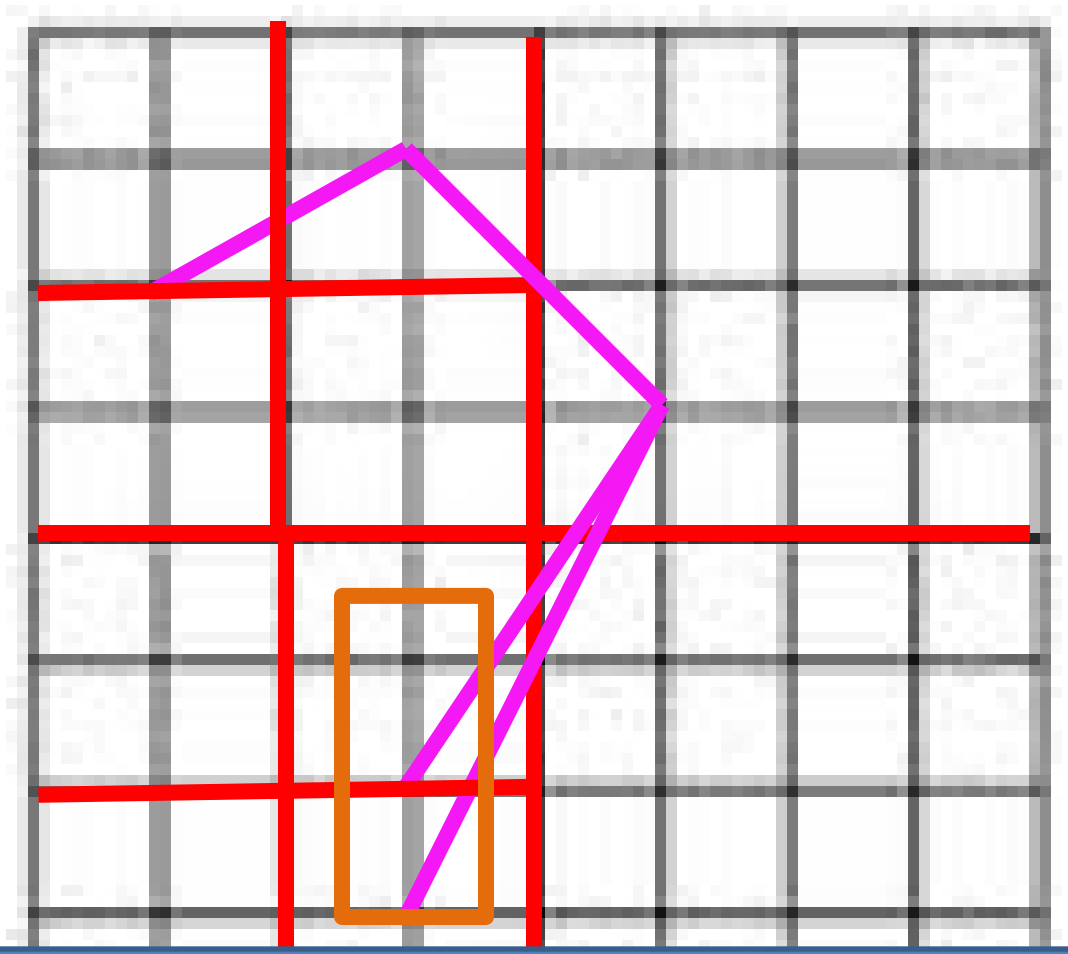
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



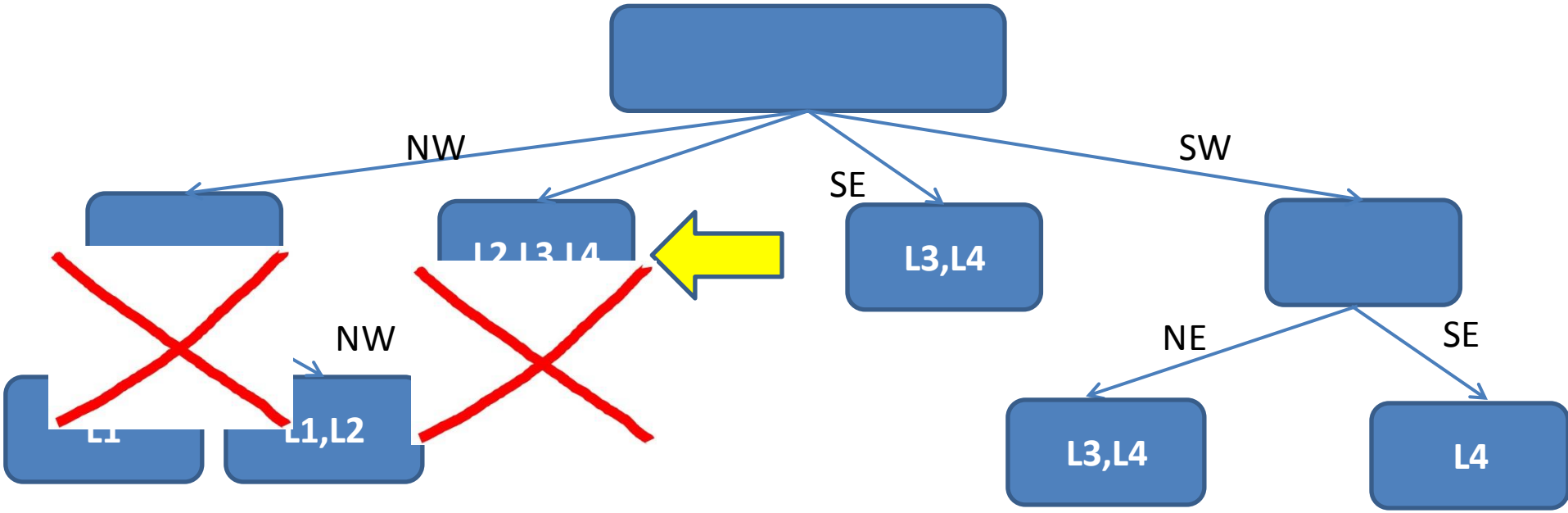
Q does NOT intersect Reg(N). So can prune.

Example: Range Query



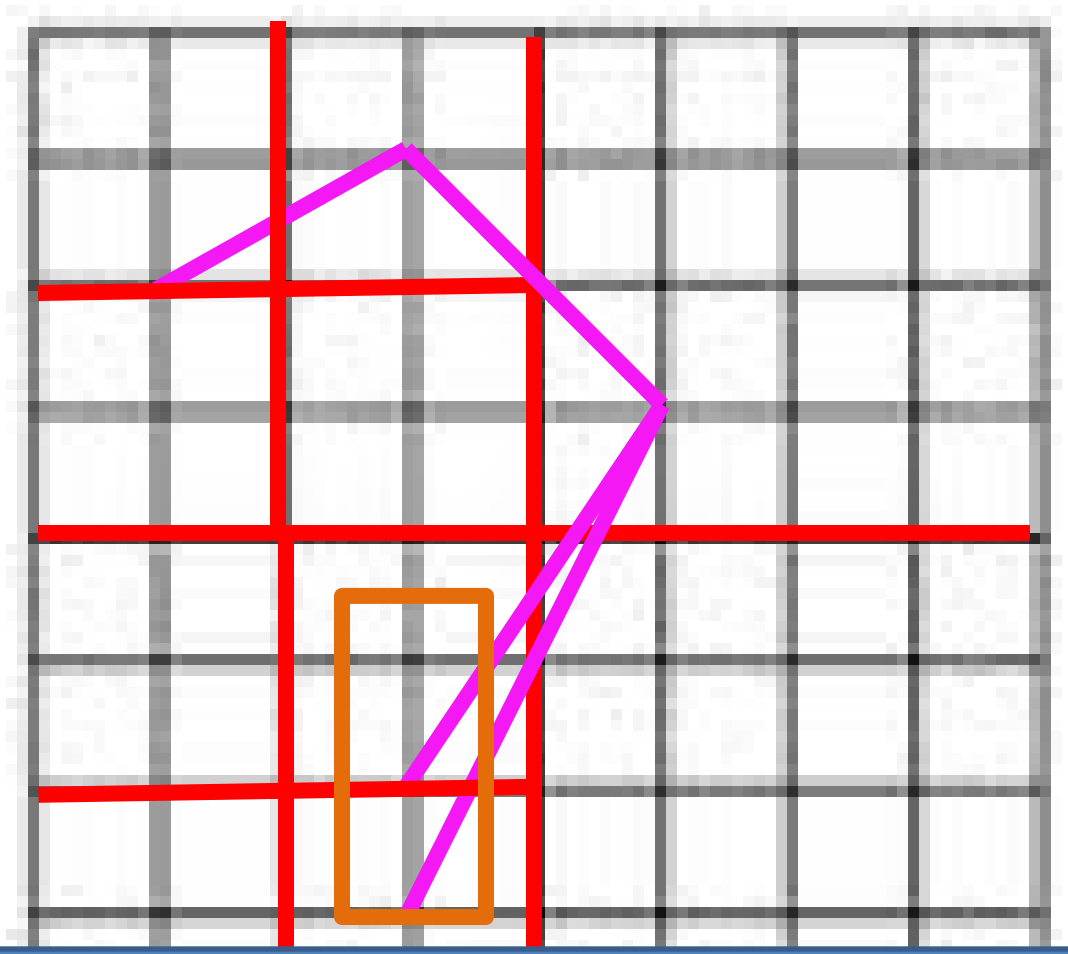
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



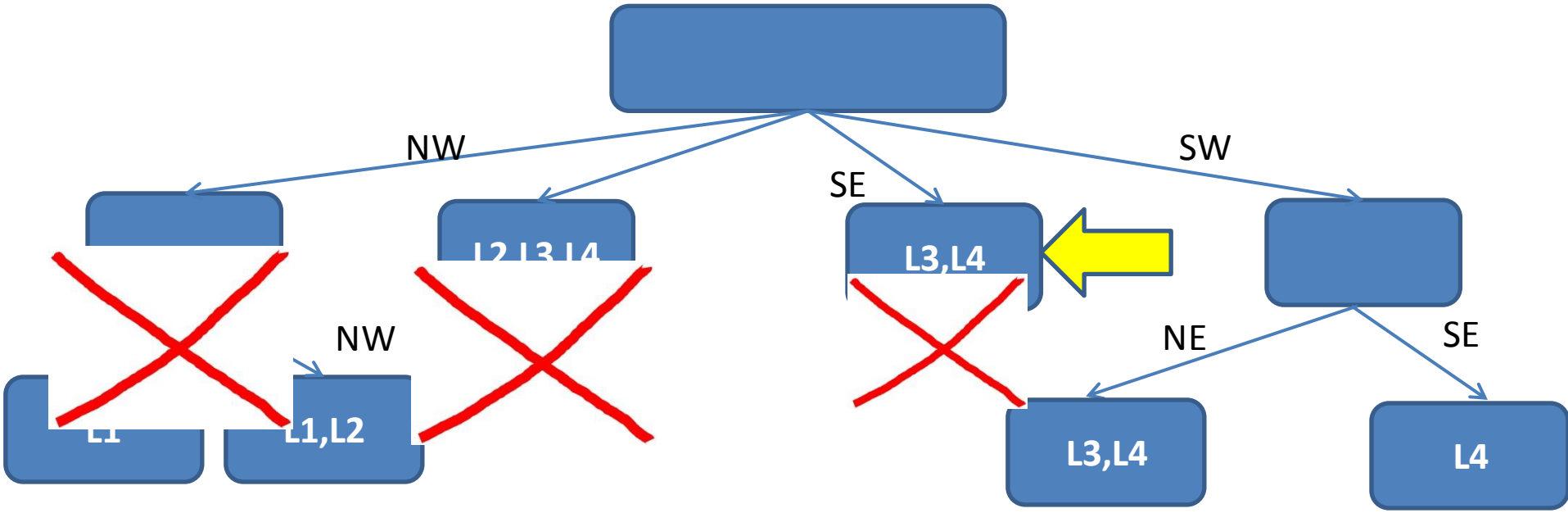
Q does NOT intersect Reg(N). So no need to check if the lines labeling this node intersect Q. Can just prune.

Example: Range Query



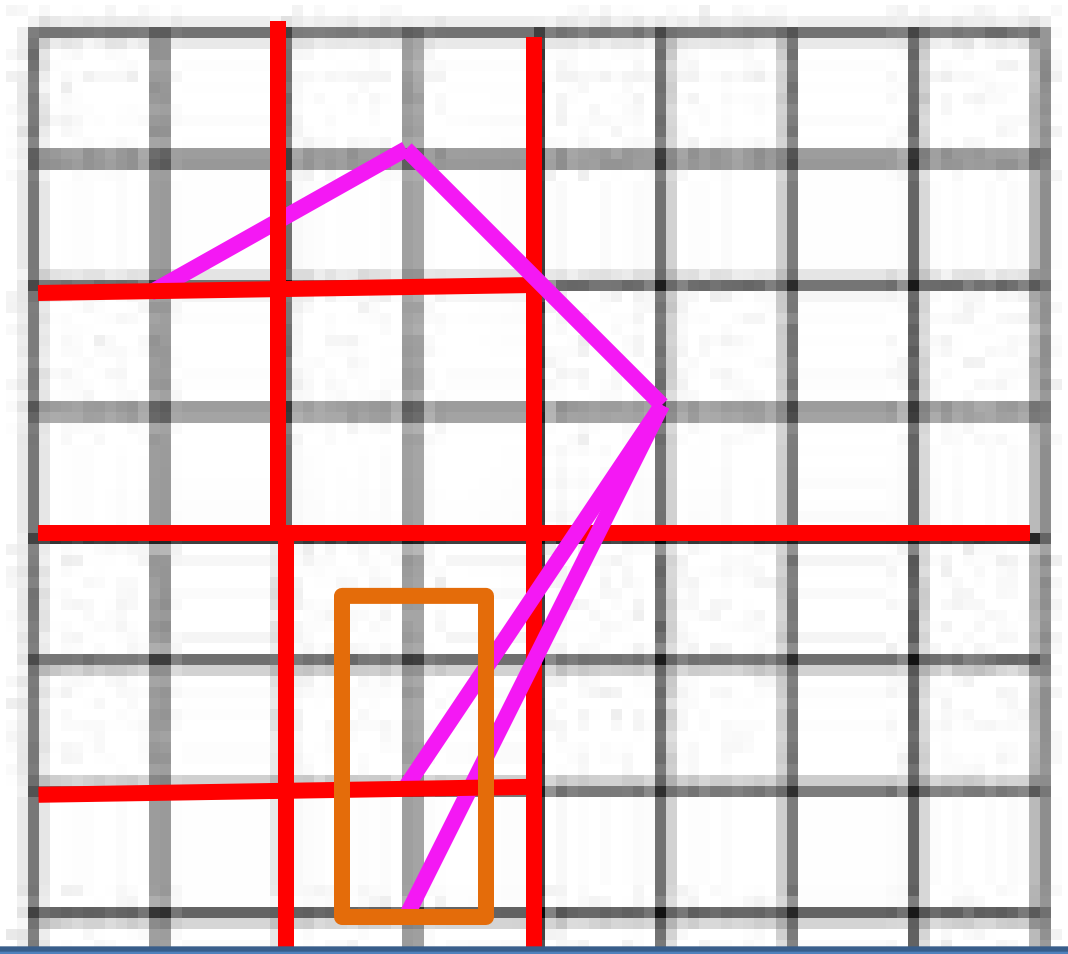
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



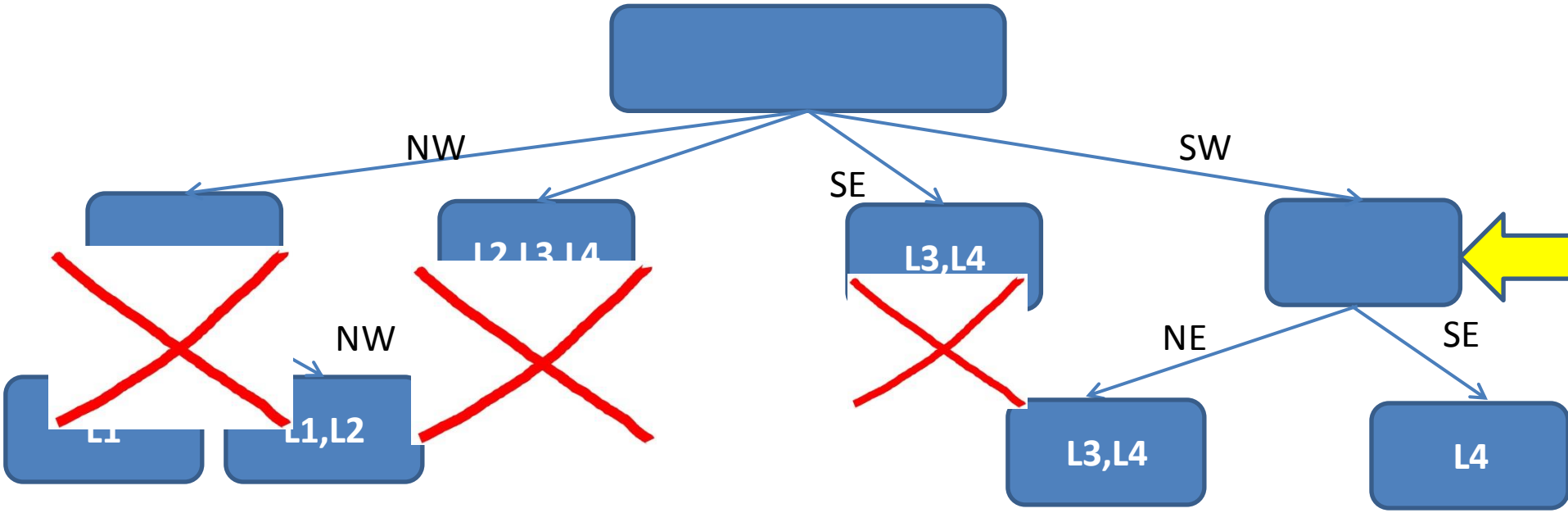
Q does NOT intersect Reg(N). So no need to check if the lines labeling this node intersect Q. Can just prune.

Example: Range Query



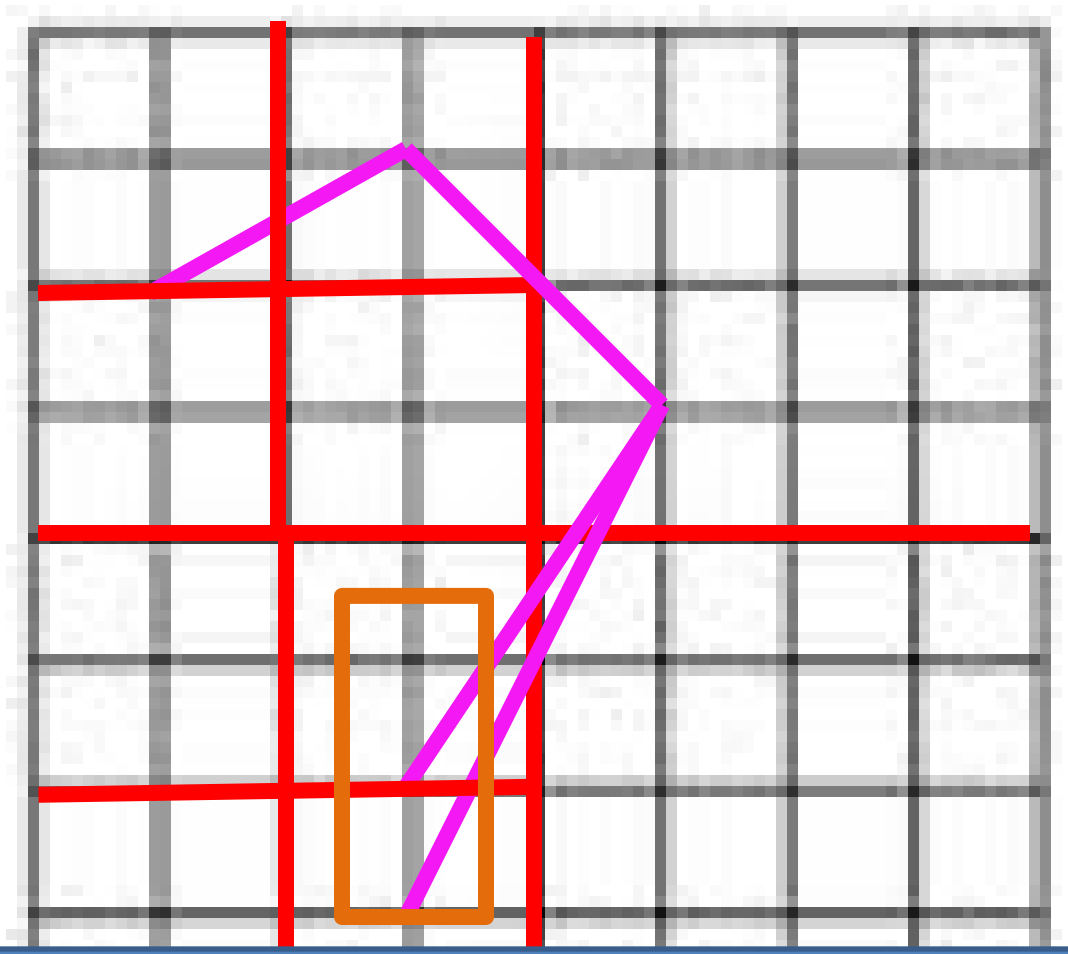
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



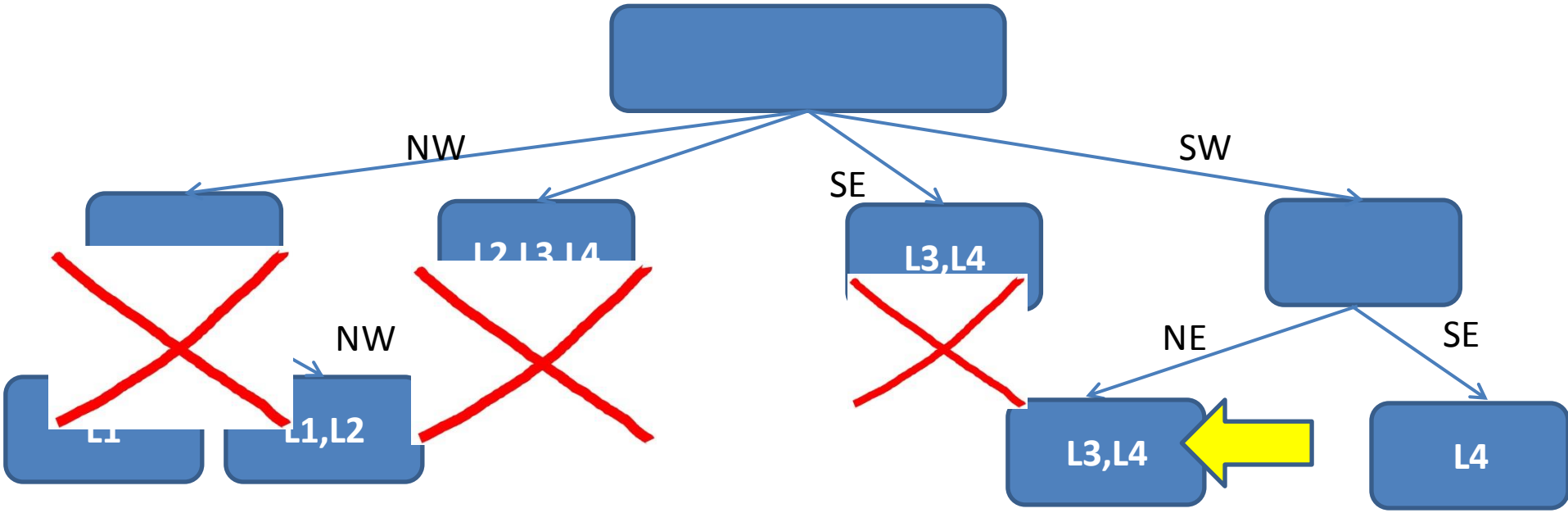
Q does intersect Reg(N). As this node is not a leaf, we must explore both its children.

Example: Range Query



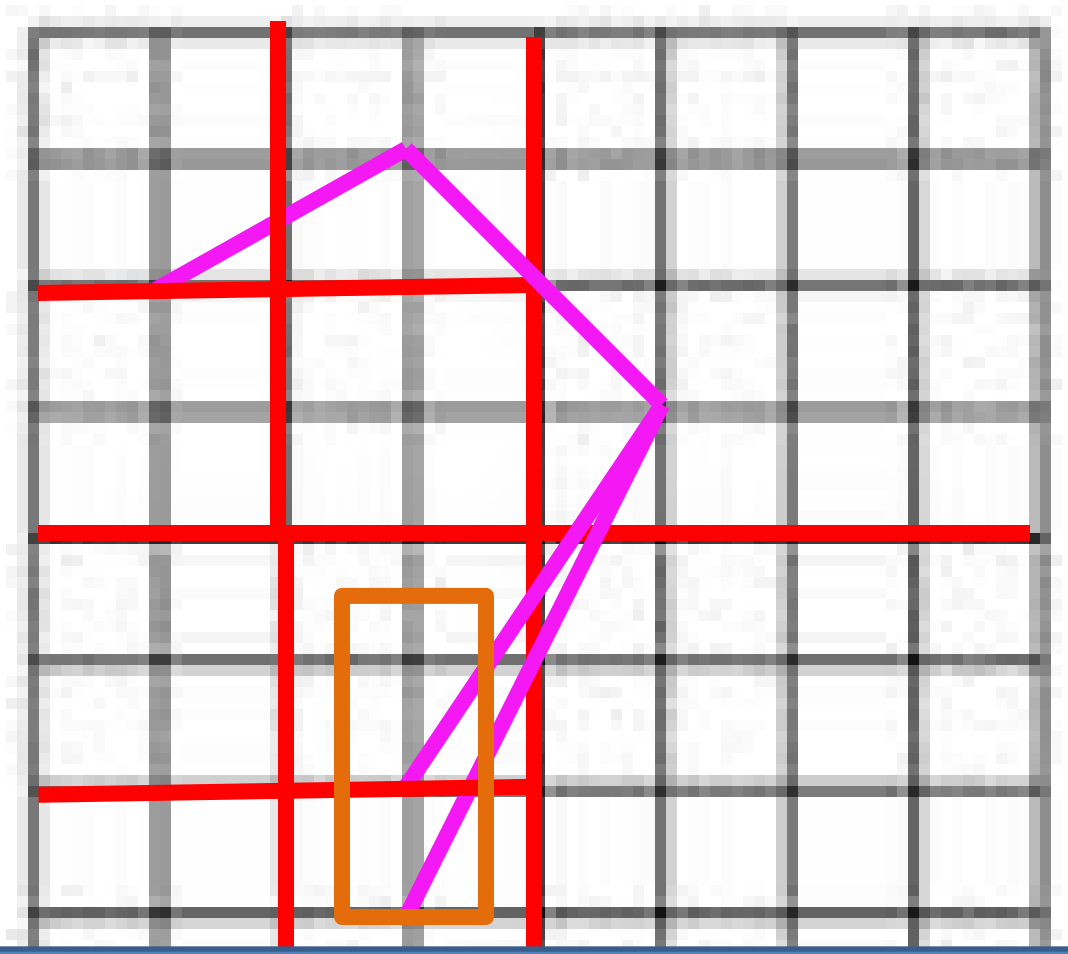
Find all lines that intersect the query rectangle Q that is shown.

Insert L4



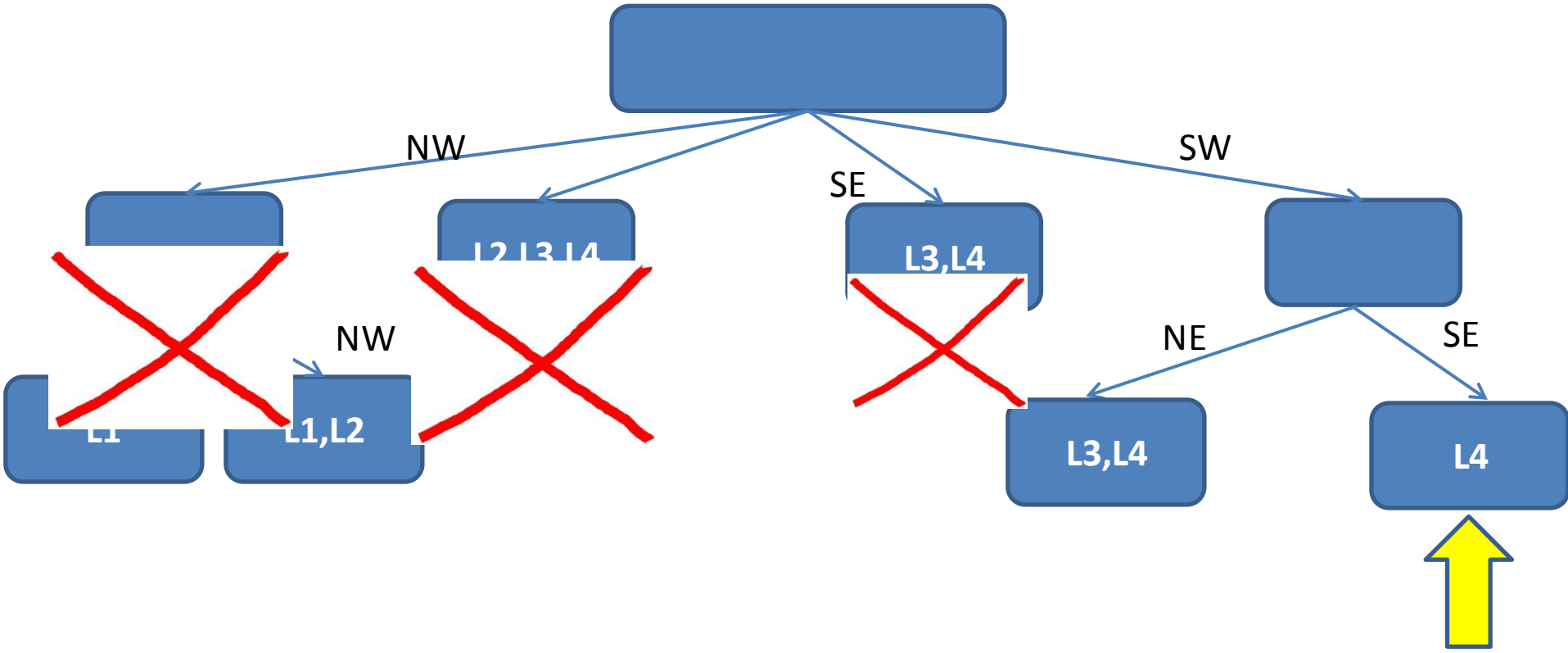
Q does intersect Reg(N). As this node is a leaf, we must check if the lines labeling it intersect Q. Yes, they do – so set SOL = {L3,L4}. Must still consider the SE quadrant

Example: Range Query



Find all lines that intersect the query rectangle Q that is shown.

Insert L4



Q does intersect Reg(N). As this node is a leaf, we must check if the lines labeling it intersect Q. Yes, they do – so set SOL = {L3,L4}. Done!