Introduction to Spatial Computing CSE 555

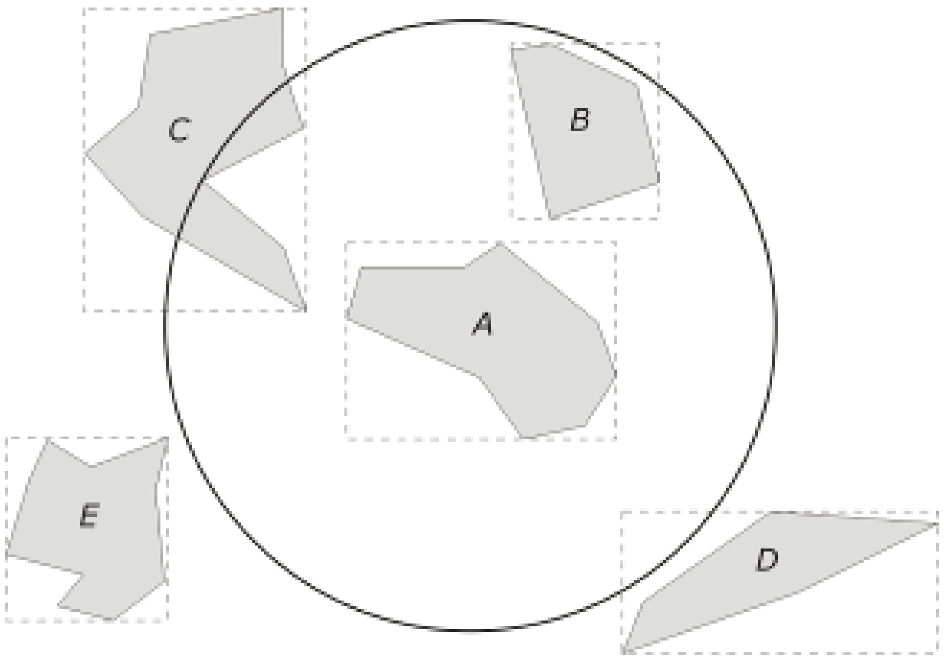
Spatial Indexing Techniques for Secondary Memory

Some slides adapted from Spatial Databases: *A Tour by Shashi Shekhar Prentice Hall (2003)*

R-Trees and its Variants

Rectangles and Minimum Bounding Boxes

- Minimum bounding box (MBB/MBR): the smallest rectangle bounding a shape with its axes parallel to the sides of the Cartesian frame
- Using MBB, some queries may be answered without retrieving the geometry of an object.



Some slides borrowed from "GIS a computational perspective: second edition" by M. Worboys CRC press 2004.

R-tree Properties and Invariants

- Balanced (similar to B+ tree)
- I is an n-dimensional rectangle of the form $(I_0, I_1, \ldots, I_{n-1})$ where I_i is a range $[a,b] \in [-\infty,\infty]$
- Leaf node index entries: (I, tuple_id)
- Non-leaf node entry: (I, child_ptr)
- *M* is maximum entries per node.
- $m \leq M/2$ is the minimum entries per node.

Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree Properties and Invariants

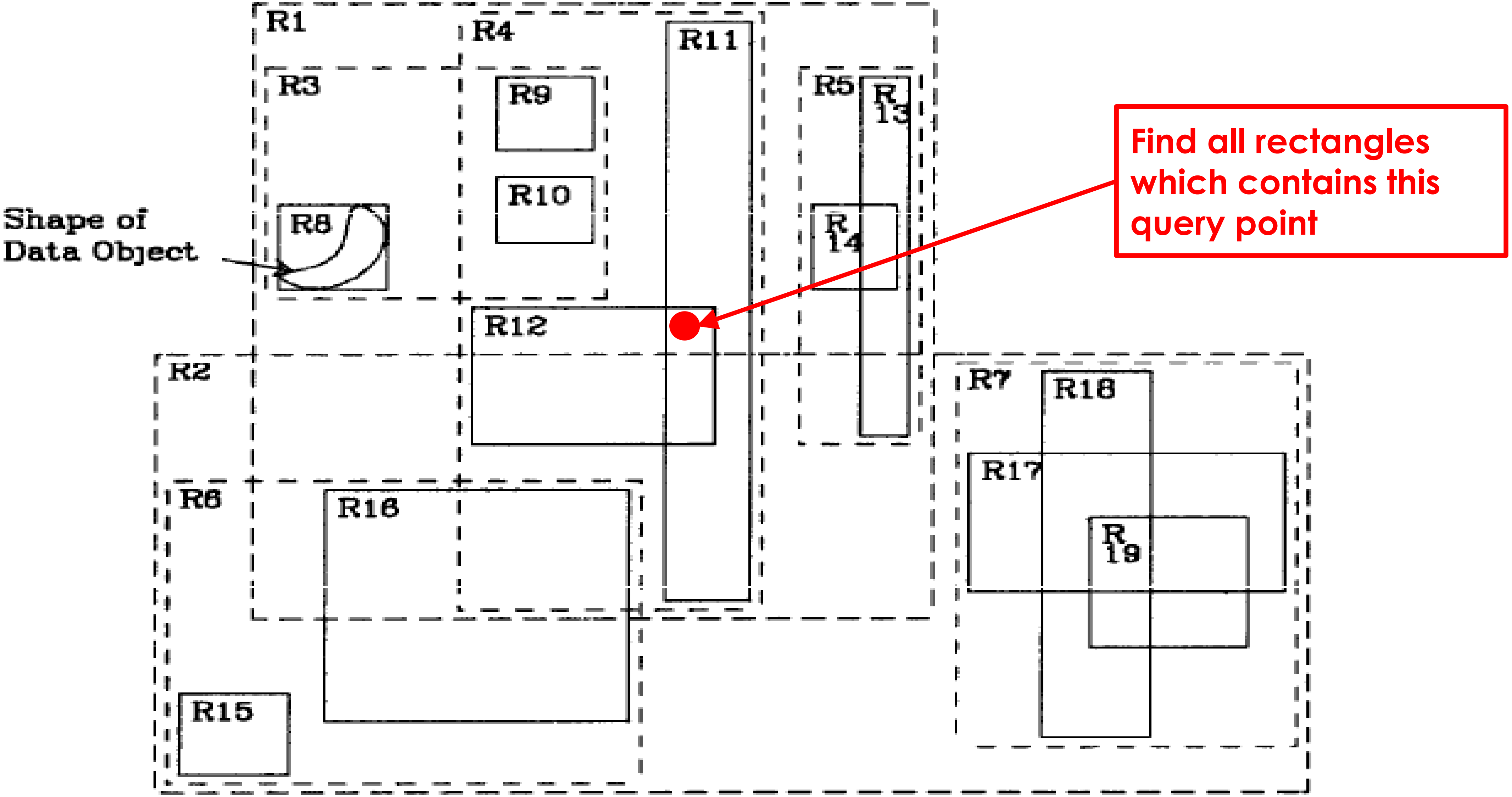
1. Every leaf (non-leaf) has between *m* and *M* records (children) except for the root.
2. Root has at least two children unless it is a leaf.
3. For each leaf (non-leaf) entry, I is the smallest rectangle that contains the data objects (children).
4. All leaves appear at the same level.

Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Searching Algorithm

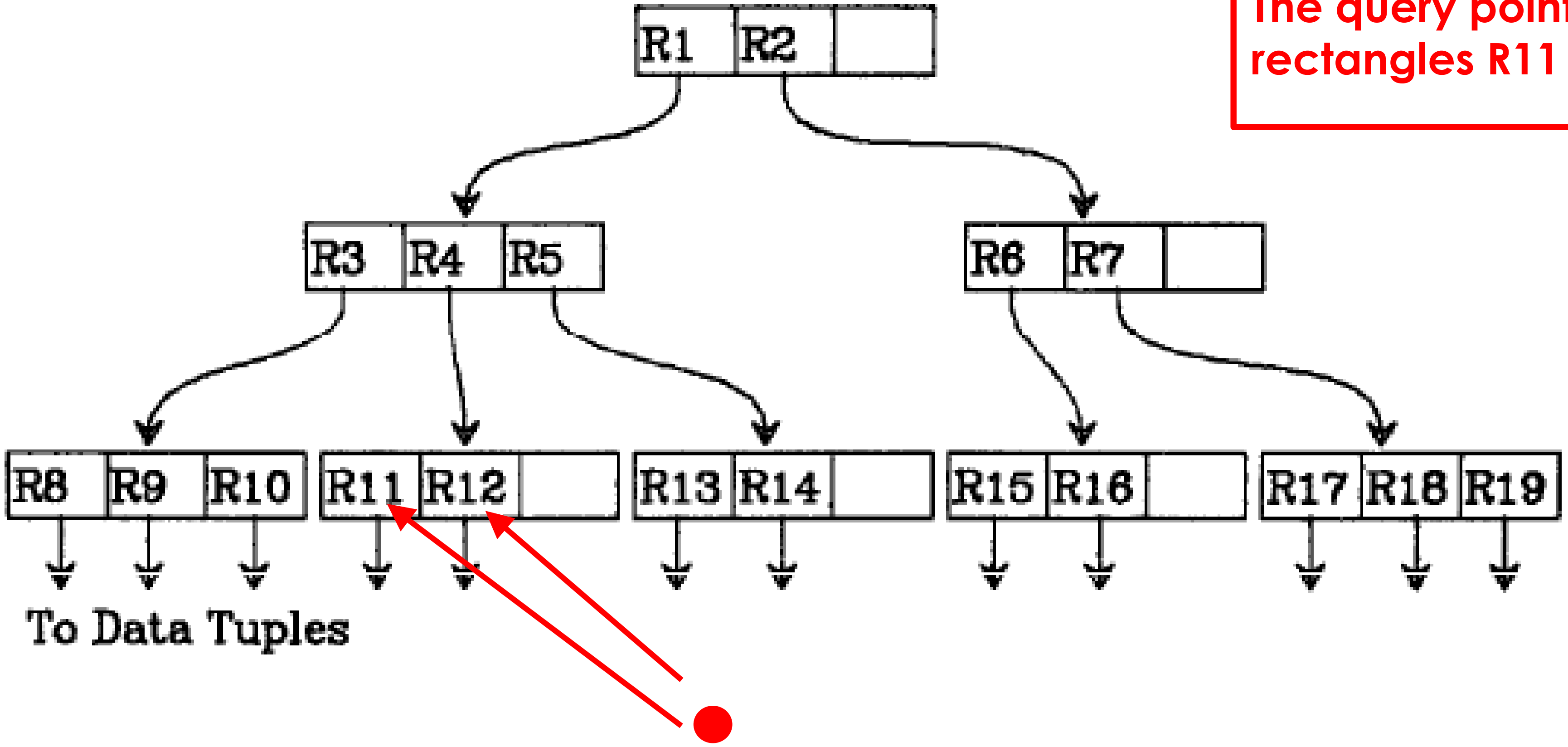
- Given a search rectangle S (or a geometry).
  1. Start at root and locate all child nodes whose rectangle I intersects S (via linear search).
  2. Search the subtrees of those child nodes.
  3. When you get to the leaves, return entries whose rectangles intersect S.
- Searches may require inspecting several paths.
- Worst case running time is not so good.

R-tree – Example (1/2)



Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Example (2/2)



The query point sits in rectangles R11 and R14

Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Insertion Algorithm (1/2)

- **Traverse the tree top down, starting from the root.**

  At each level:

  1. If there is a node whose directory rectangle contains the MBB to be inserted, then search the subtree.
  2. Else choose a node such that enlargement of its directory rectangle is minimal, then search the subtree.
  3. If more than one node satisfy this, then choose the one with the smallest area.

- **Repeat until a leaf node is reached.**

R-tree – Insertion Algorithm (2/2)

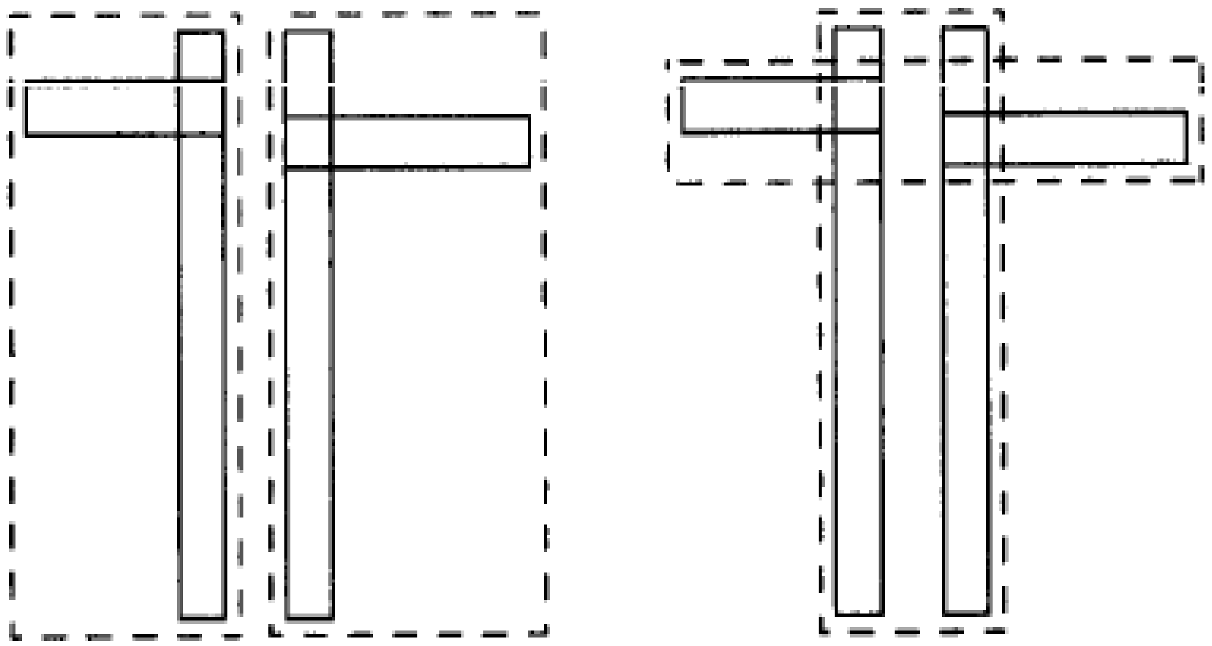
At leaf level:

- **If the leaf node is not full** then an entry [MBB, object-id] is inserted.
- **Else** //the leaf node is full

1. **Split the leaf node.**
2. **Update the directory rectangles of the ancestor nodes if necessary.**

R-tree – Node Splitting

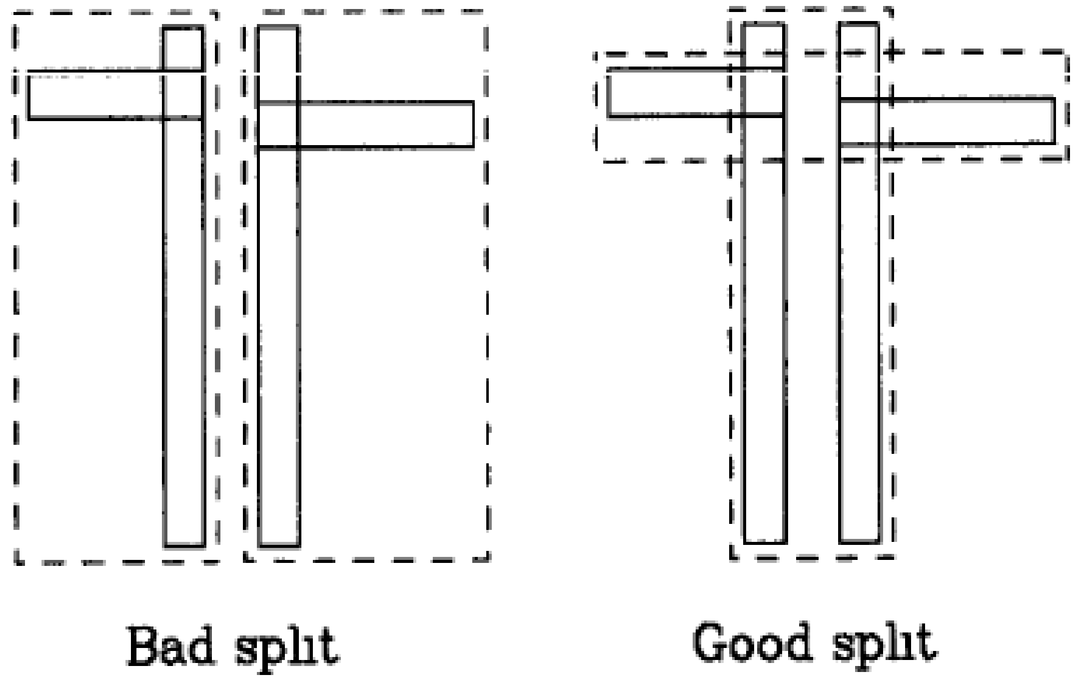
- **Problem:** Divide $M+1$ entries among two nodes so that it is unlikely that the nodes are needlessly examined during a search.
- **Objective:** Minimize total area of the covering rectangles for both nodes.
- Exponential algorithm.
- Quadratic algorithm.
- Linear time algorithm.



Bad split Good split

R-tree – Node Splitting: Exponential Algorithm

- **Problem:** Divide *M*+1 entries among two nodes so that it is unlikely that the nodes are needlessly examined during a search.
- **Solution:** Minimize total area of the covering rectangles for both nodes.
- **Exponential algorithm**
  - Try all possible combinations.
  - Optimal results!
  - Bad running time!



R-tree – Node Splitting: Quadratic Algorithm

- **Problem:** Divide *M*+1 entries among two nodes so that it is unlikely that the nodes are needlessly examined during a search.
- **Solution:** Minimize total area of the covering rectangles for both nodes.
- **Quadratic algorithm**

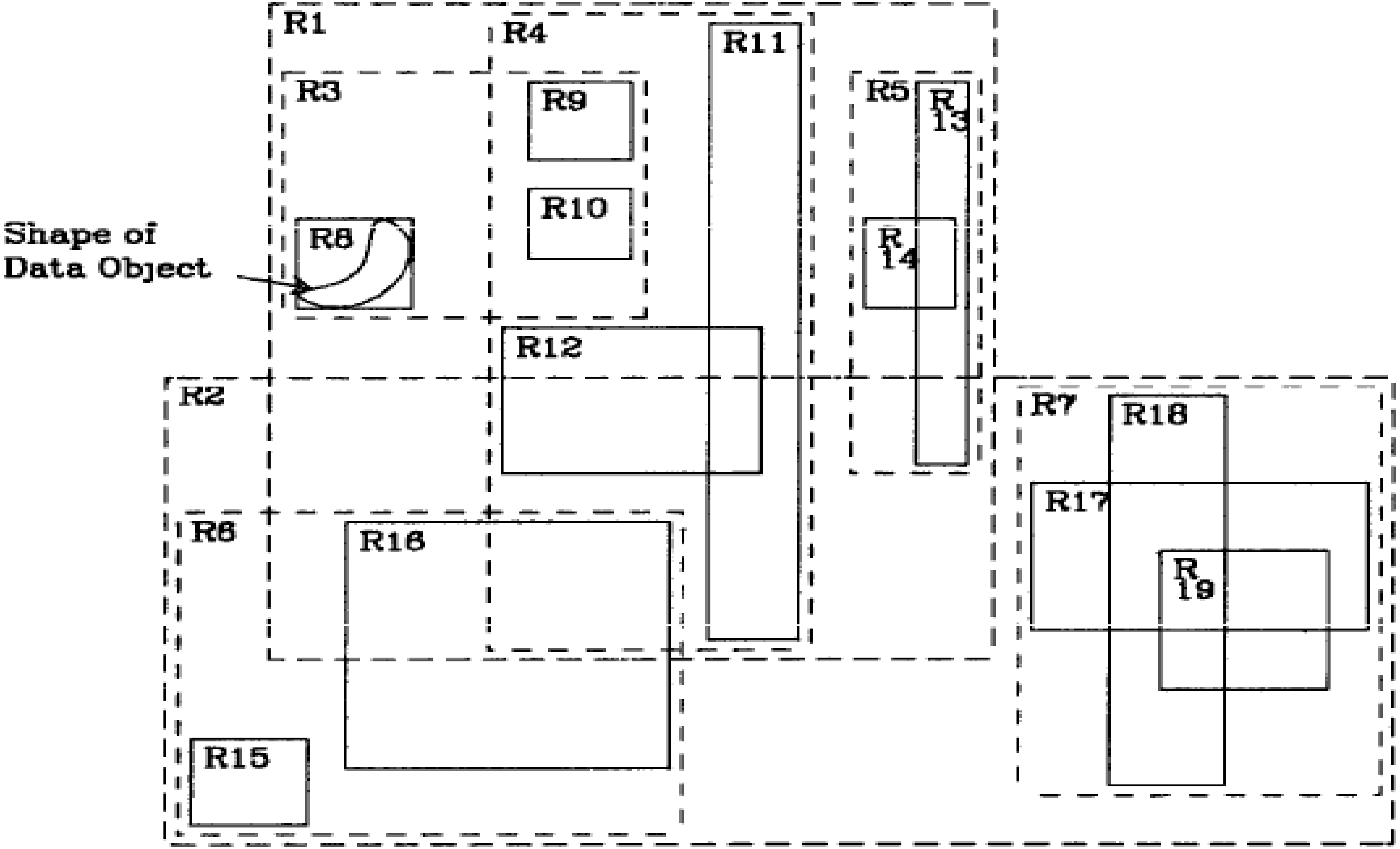
1. Find pair of entries E_1 and E_2 that maximizes *area*(J) - *area*(E_1) - *area*(E_2) where J is covering rectangle. J is the MBR containing only E1 and E2
2. Put E_1 in one group, E_2 in the other.
3. If one group has M-m+1 entries, put the remaining entries into the other group and stop. If all entries have been distributed then stop.
4. For each entry E, calculate d_1 and d_2 where d_i is the minimum area increase in covering rectangle of the group when E is added.
5. Find E with maximum $|d_1 - d_2|$ and add E to the group whose area will increase the least. If tied: (a) choose smaller area, (b) choose smaller group
6. Repeat starting with step 3.

R-tree – Tree Adjustment during overflow

1. N = leaf node. If there was a split, then NN is the other node.
2. If N is root, stop. Otherwise P = N's parent and E_N is its entry for N. Adjust the rectangle for E_N to tightly enclose new N.
3. If NN exists (i.e., N was split and NN is its second MBB from split), add entry E_{NN} (MBB corresponding to NN) to P. E_{NN} points to NN and its MBB rectangle tightly encloses NN.
4. If necessary, split P
5. Set N=P and go to step 2.

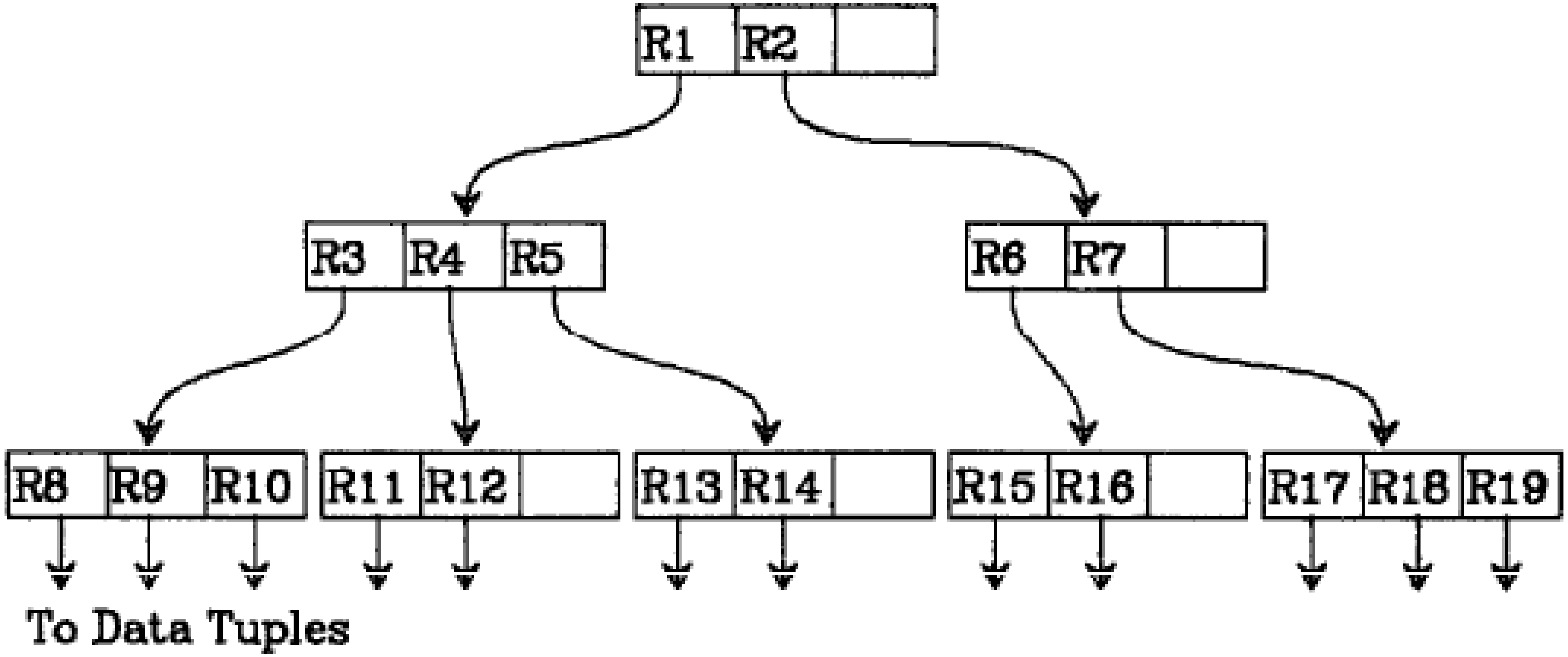
Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Example (1/2)



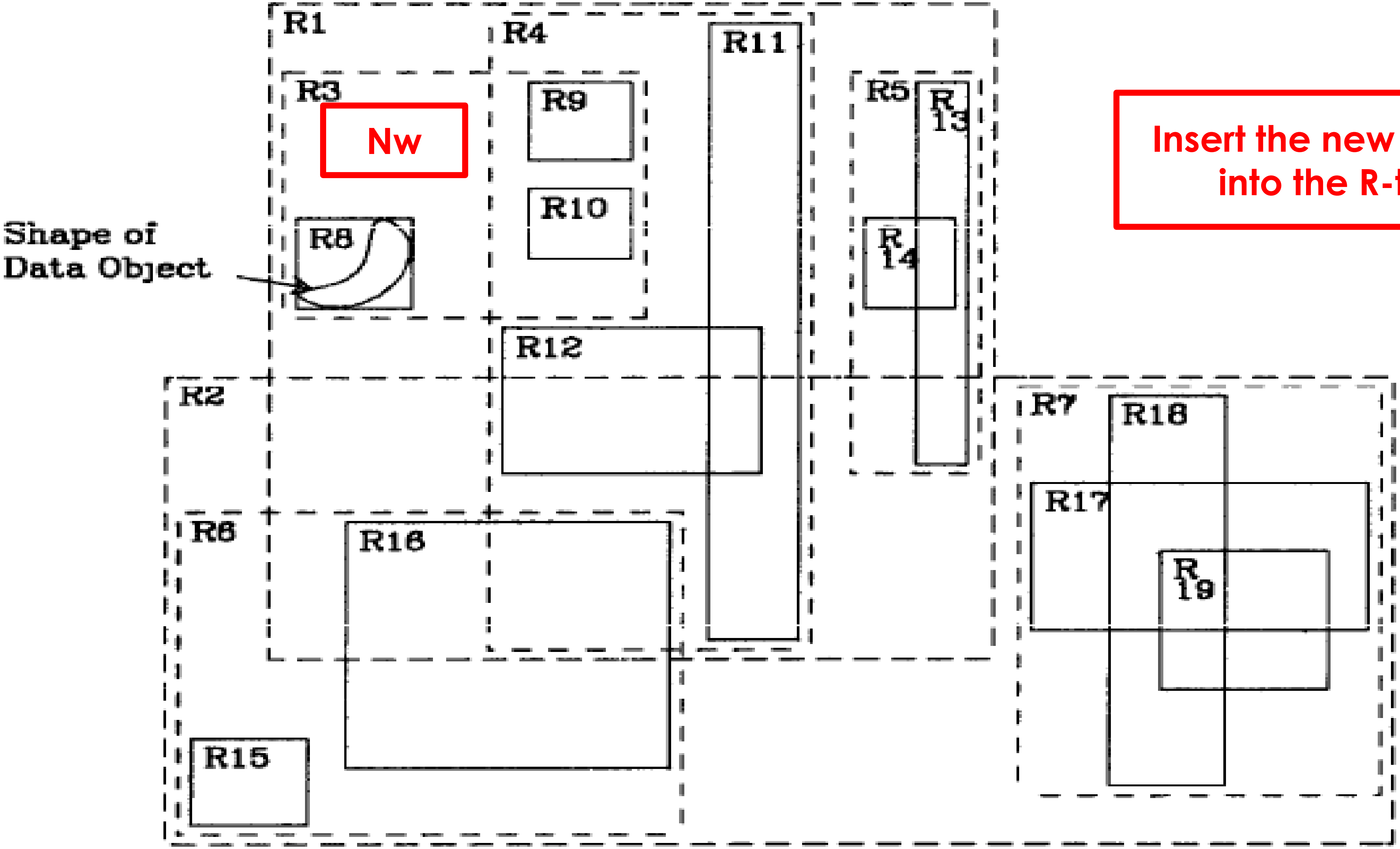
Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Example (2/2)



Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

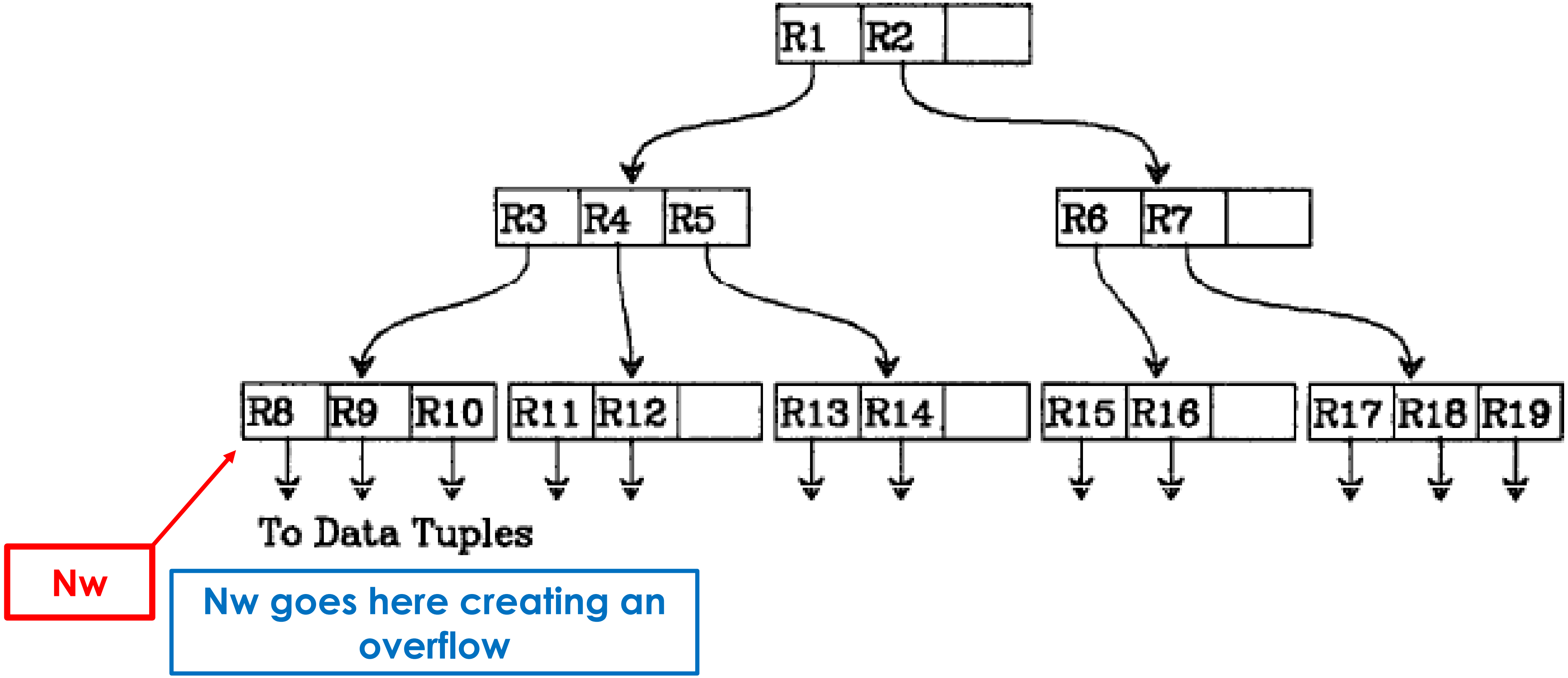
R-tree – Insertion Example



Insert the new data point Nw into the R-tree shown

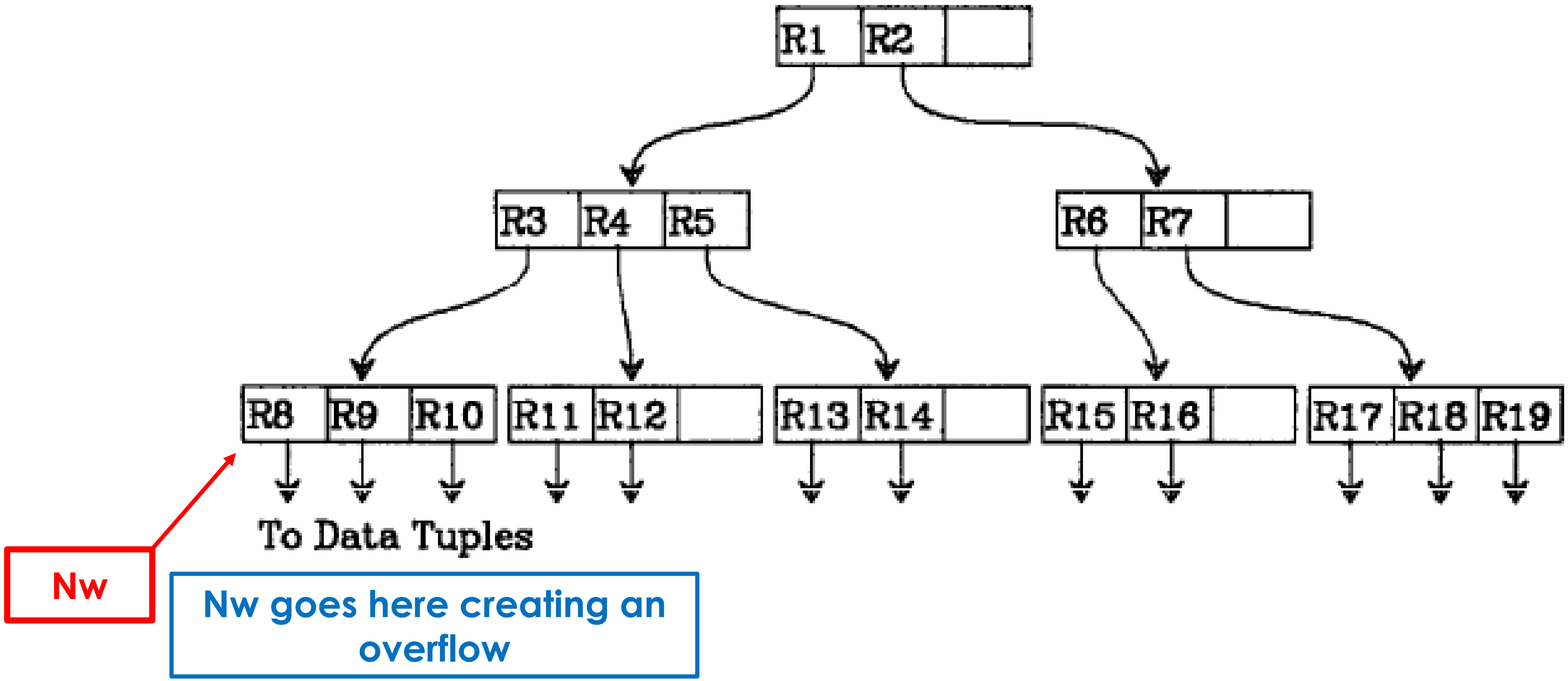
Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

R-tree – Insertion Example

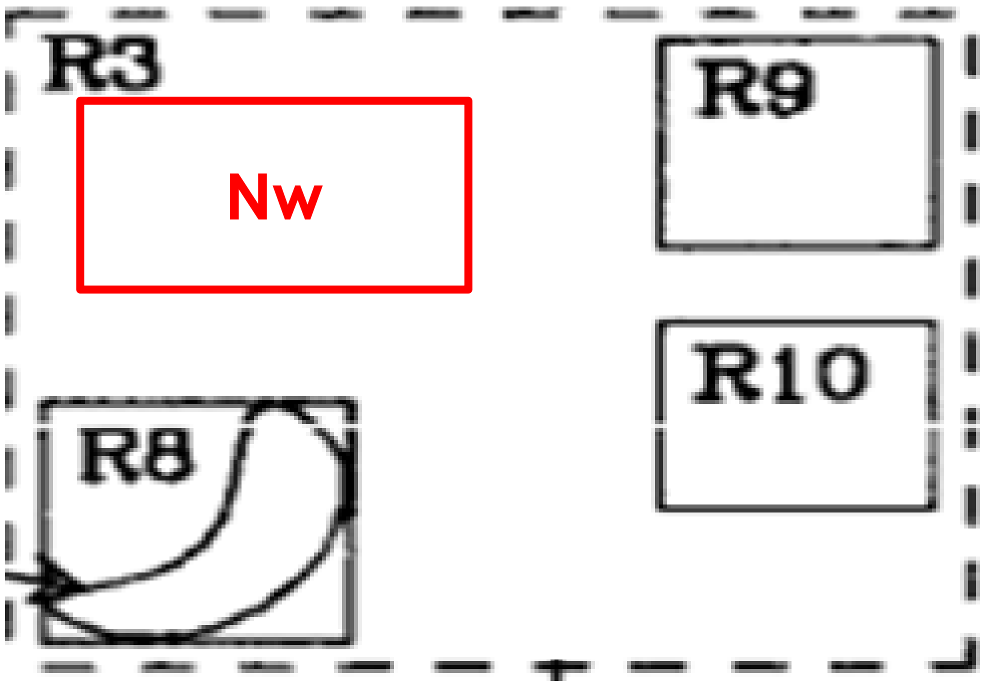


Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (SIGMOD '84)

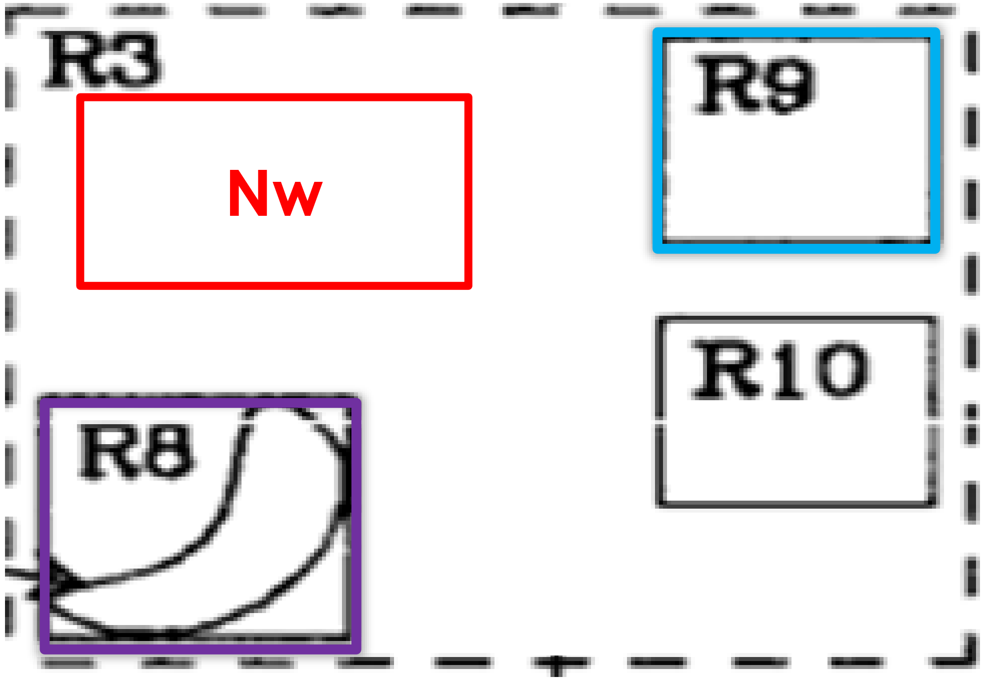
R-tree – Insertion Example



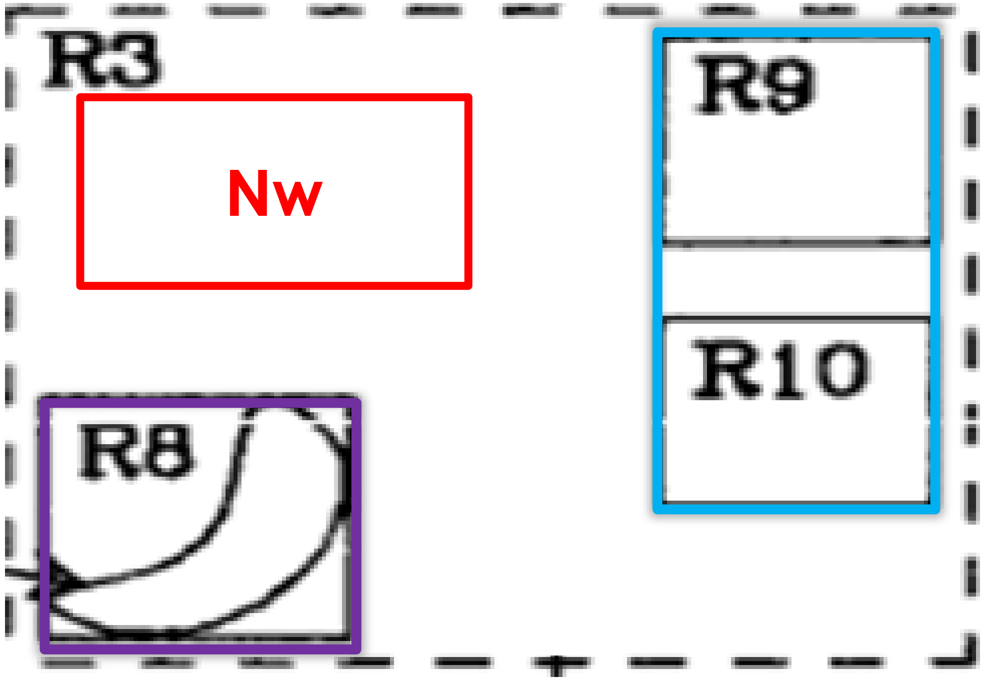
R-tree – Example Splitting R3



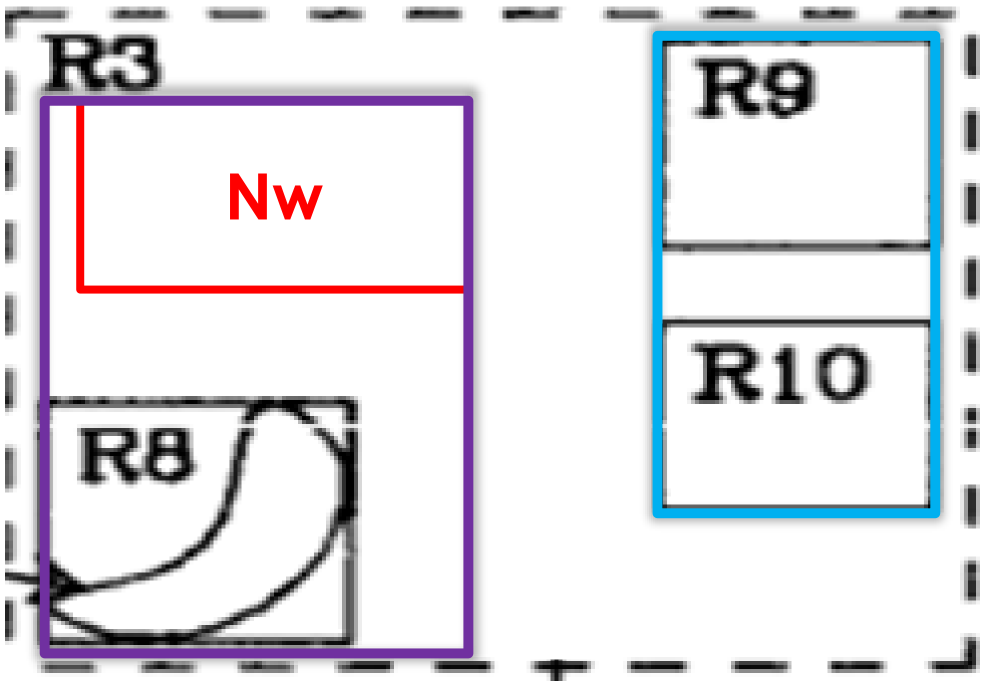
R-tree – Example Splitting R3: Step 1



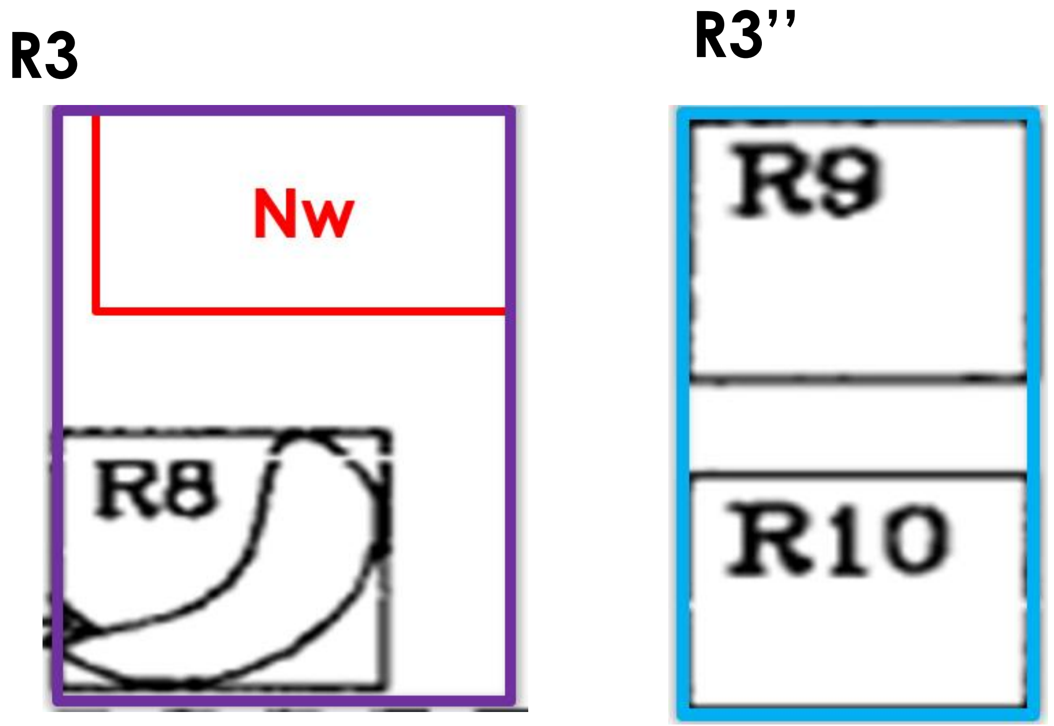
R-tree – Example Splitting R3: Step 2



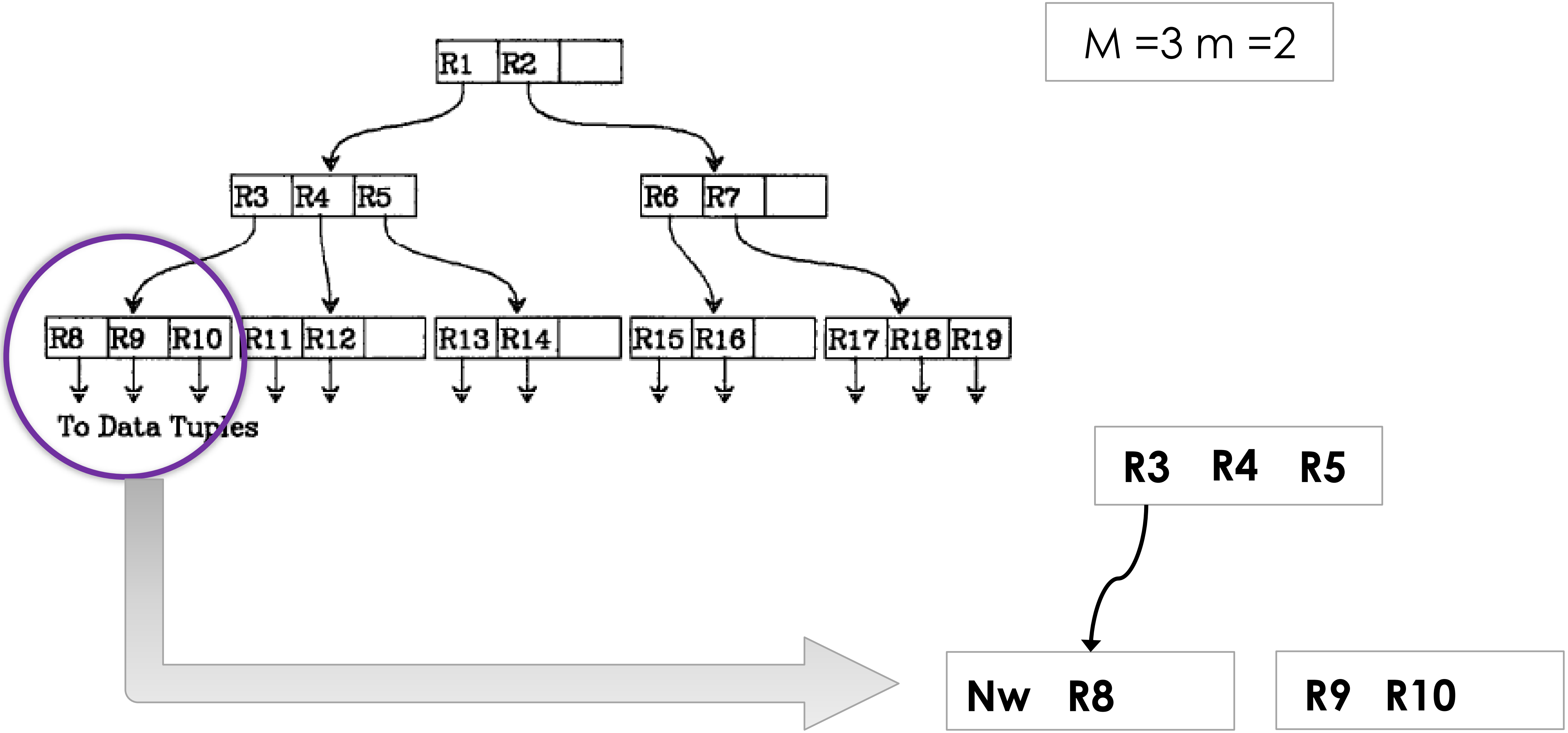
R-tree – Example Splitting R3: Step 3



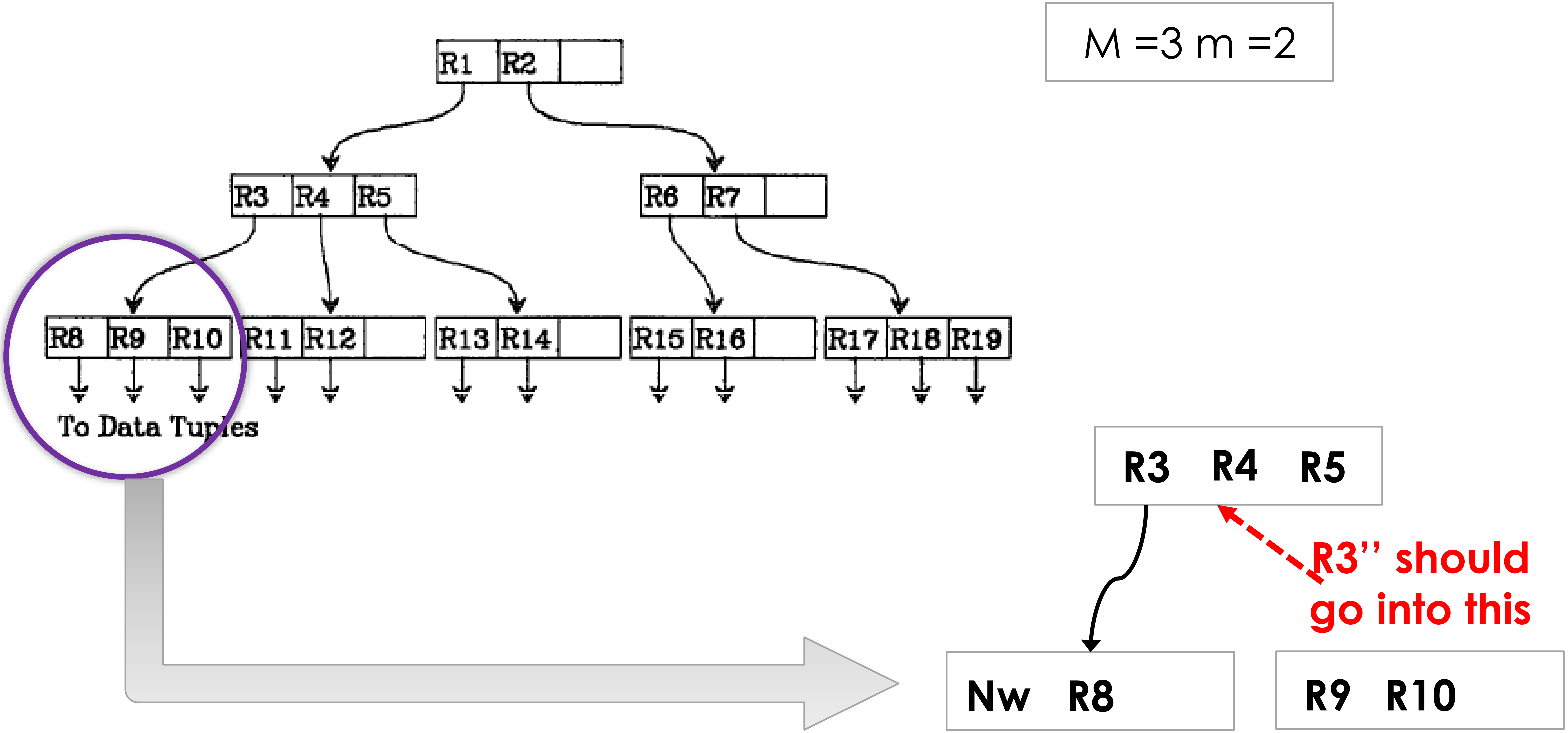
R-tree – Example Splitting R3



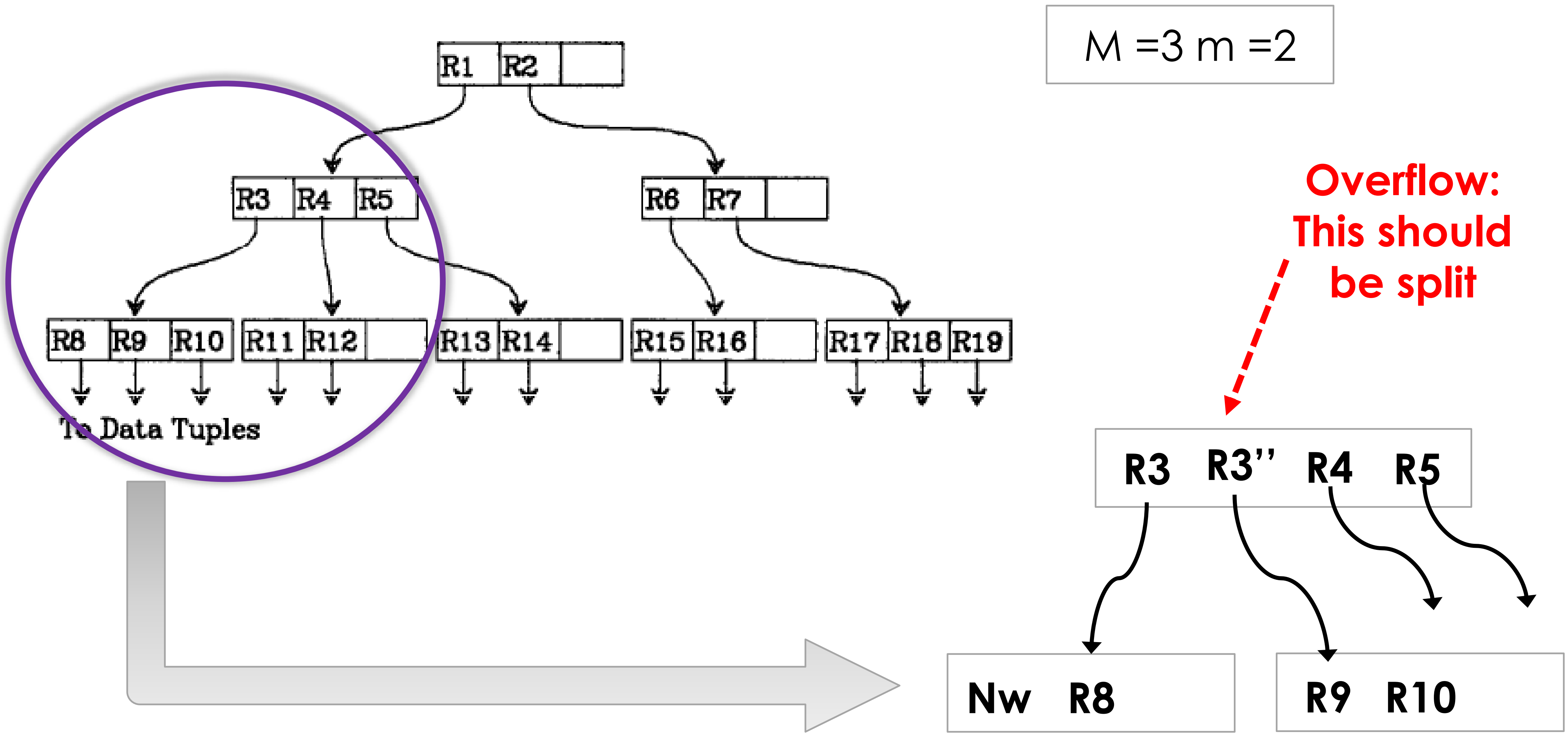
R-tree – Example Adjusting the tree



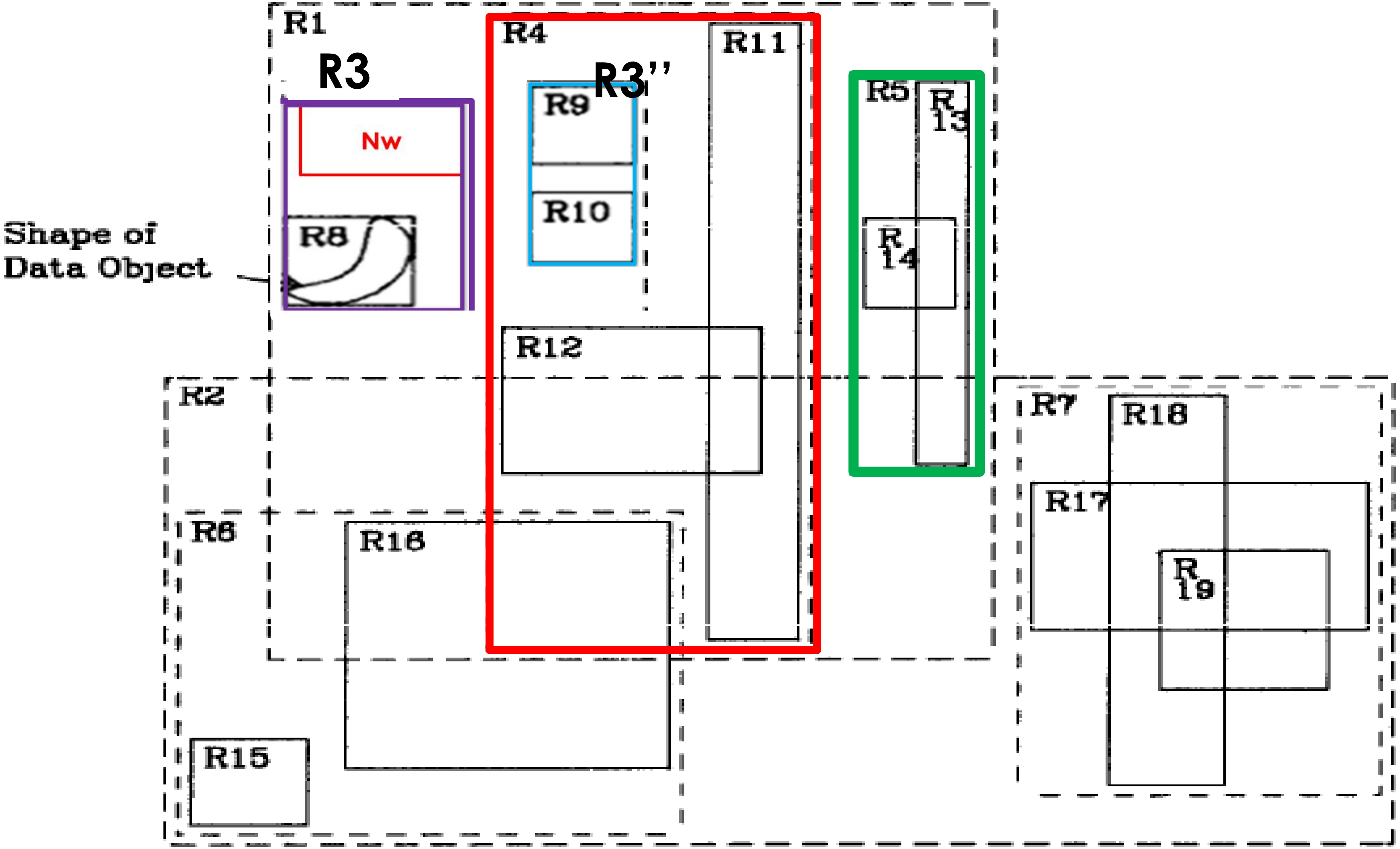
R-tree – Example Adjusting the tree



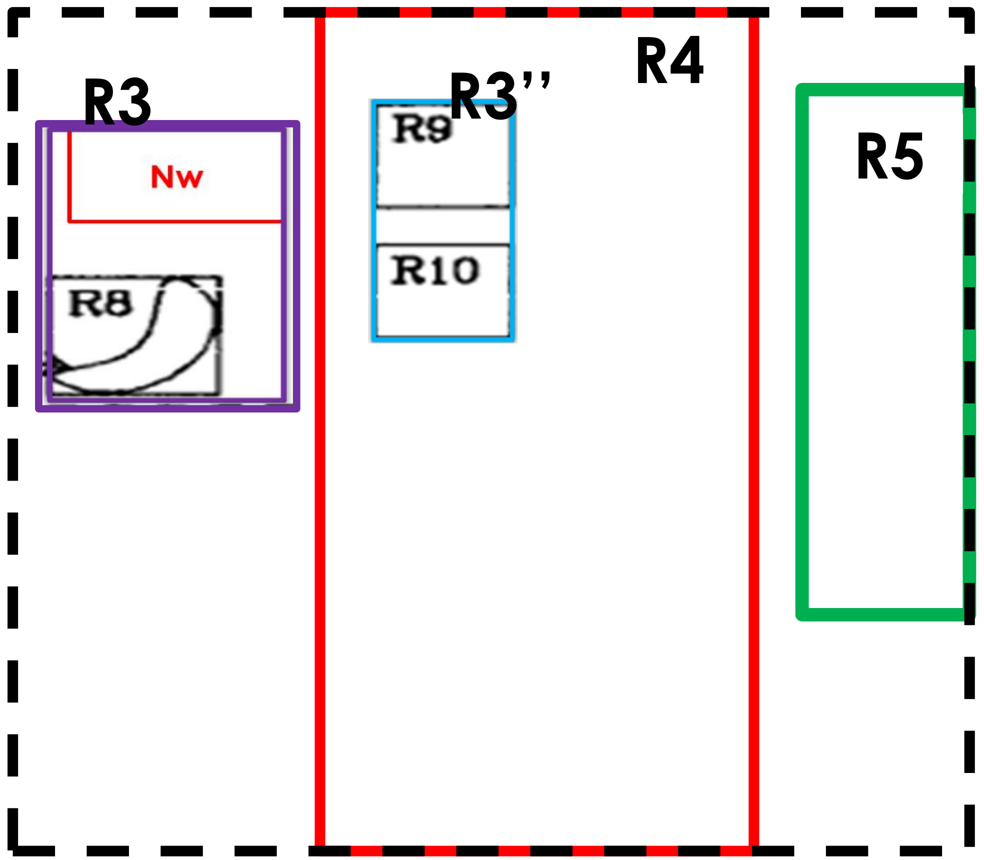
R-tree – Example Adjusting the tree



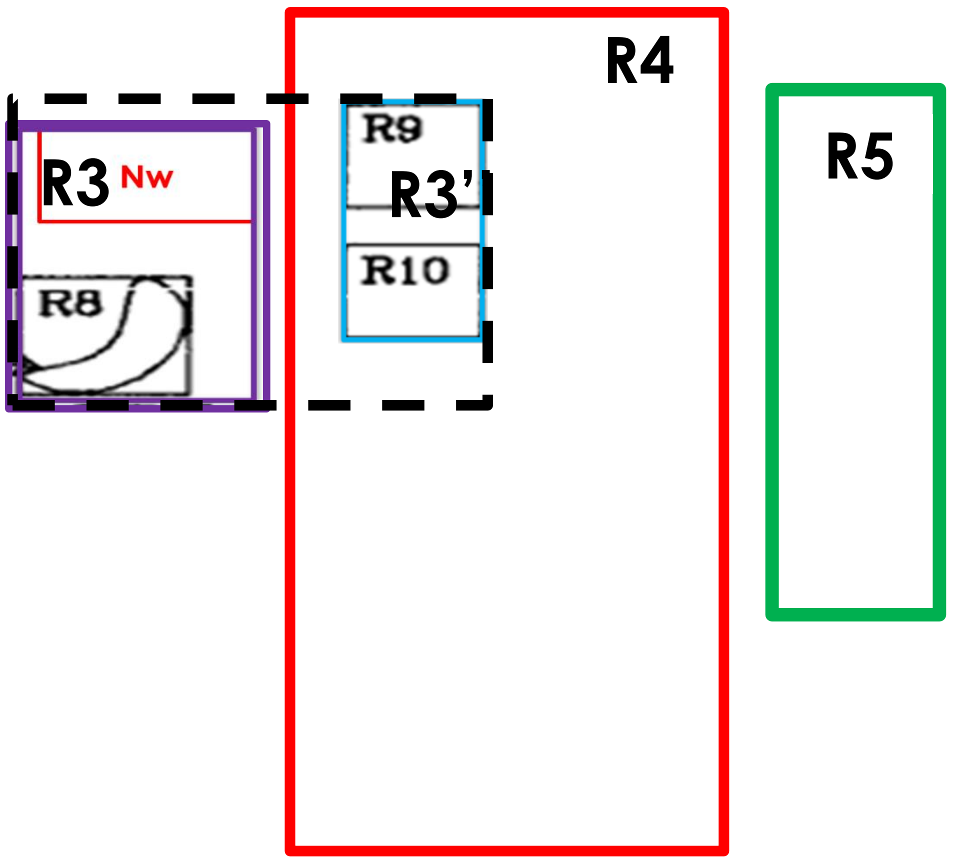
R-tree – Example Splitting R3 R3'' R4 and R5



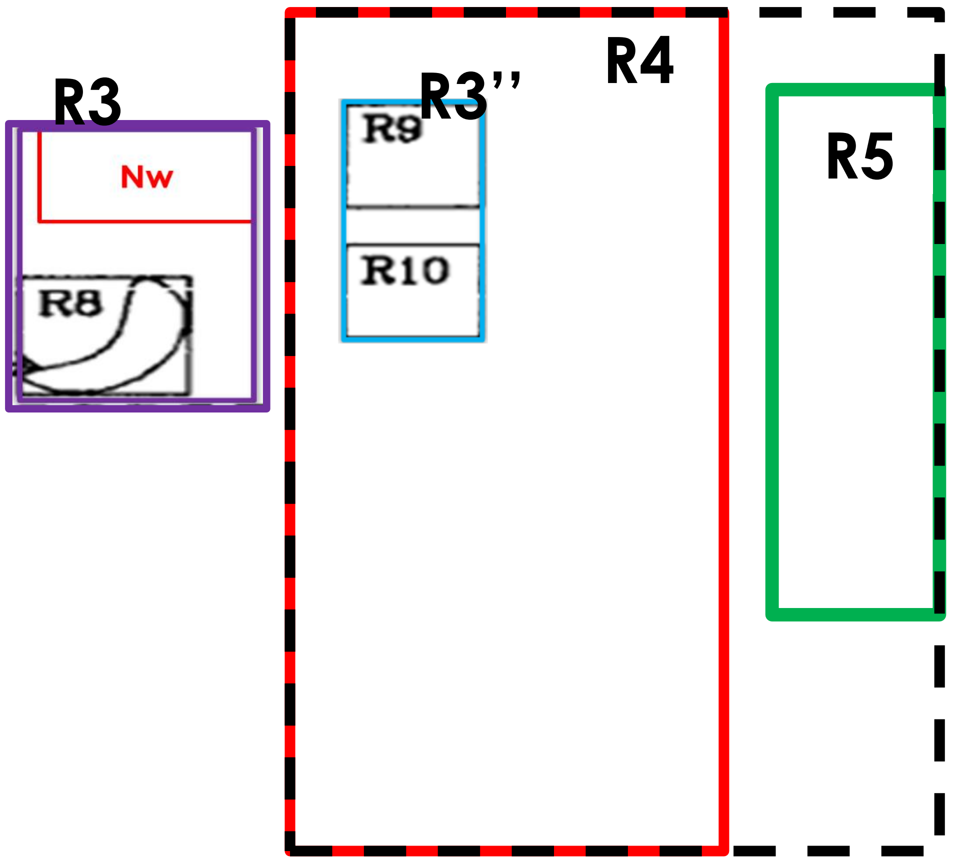
R-tree – Example Splitting R3 R3'' R4 and R5



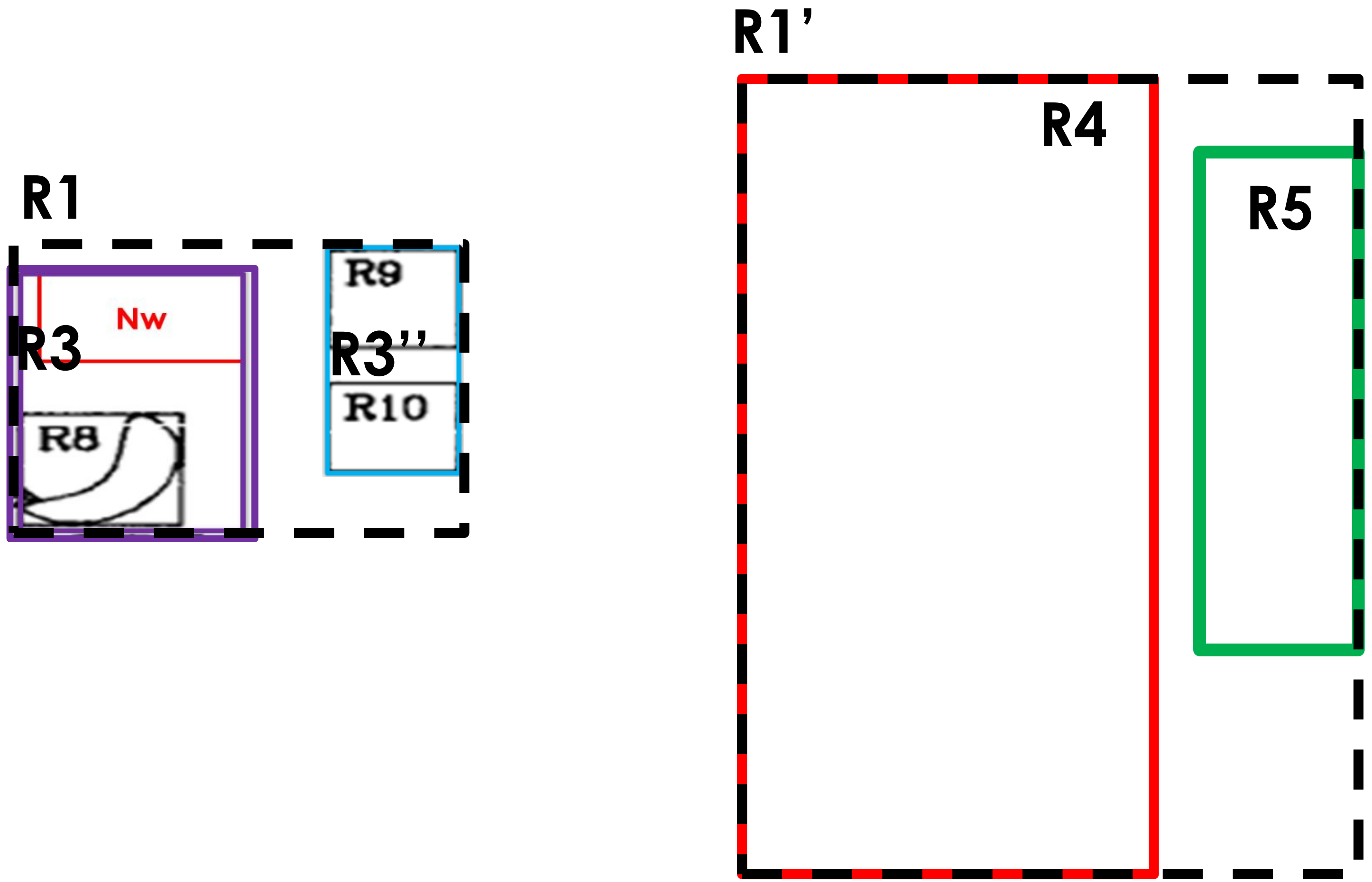
R-tree – Example Splitting R3 R3'' R4 and R5



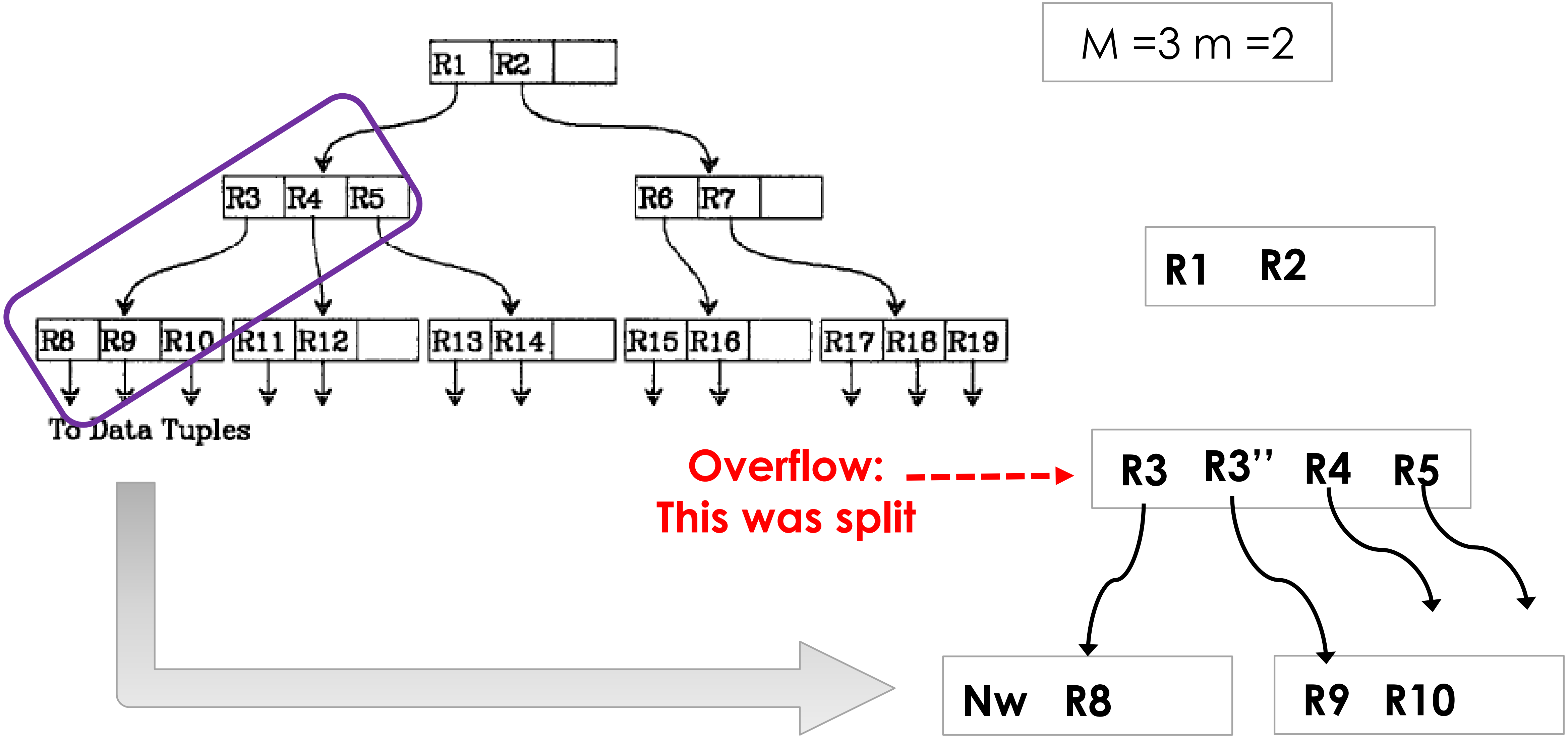
R-tree – Example Splitting R3 R3'' R4 and R5



R-tree – Example Splitting R3 R3'' R4 and R5

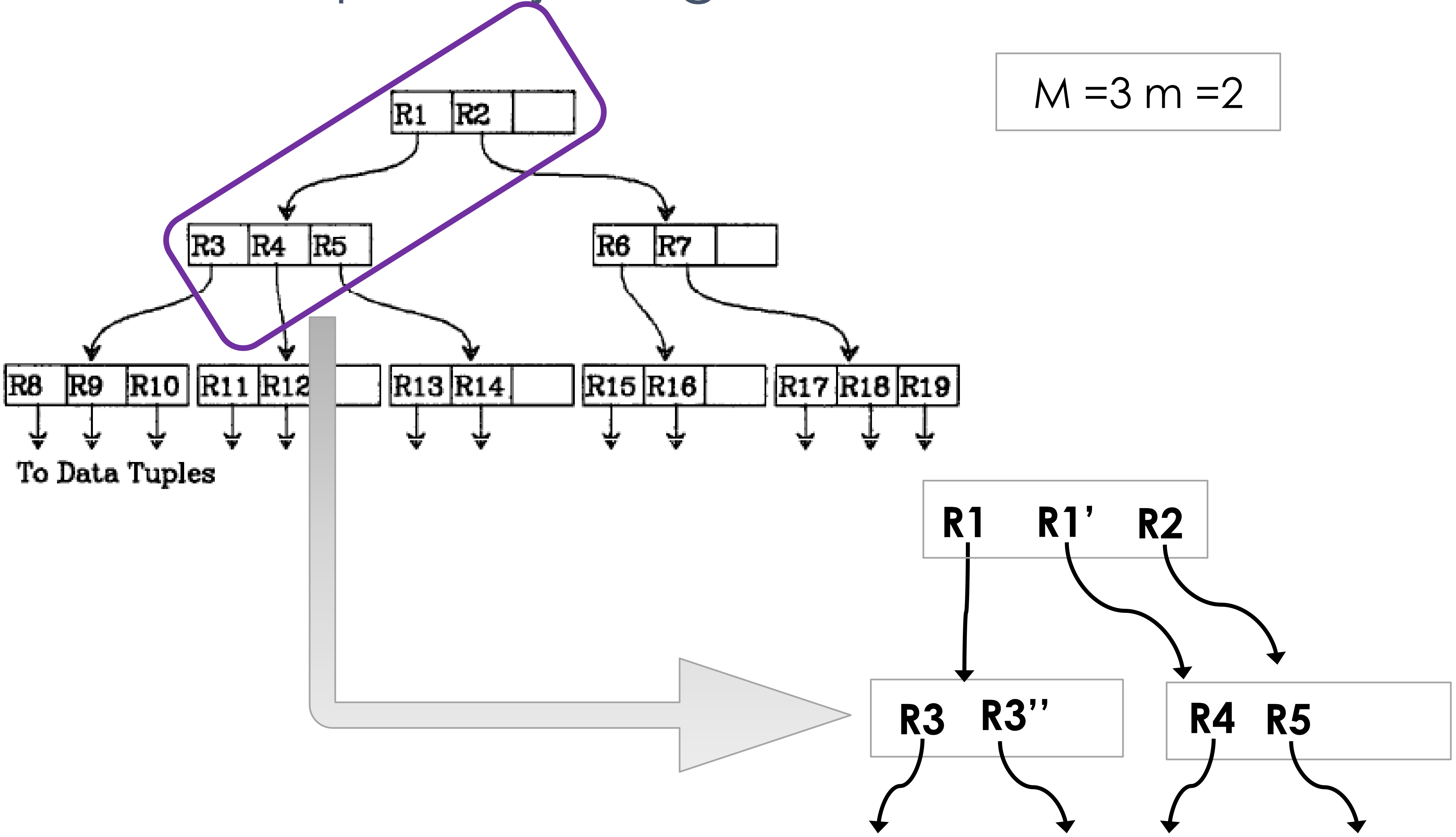


R-tree – Example Adjusting the tree



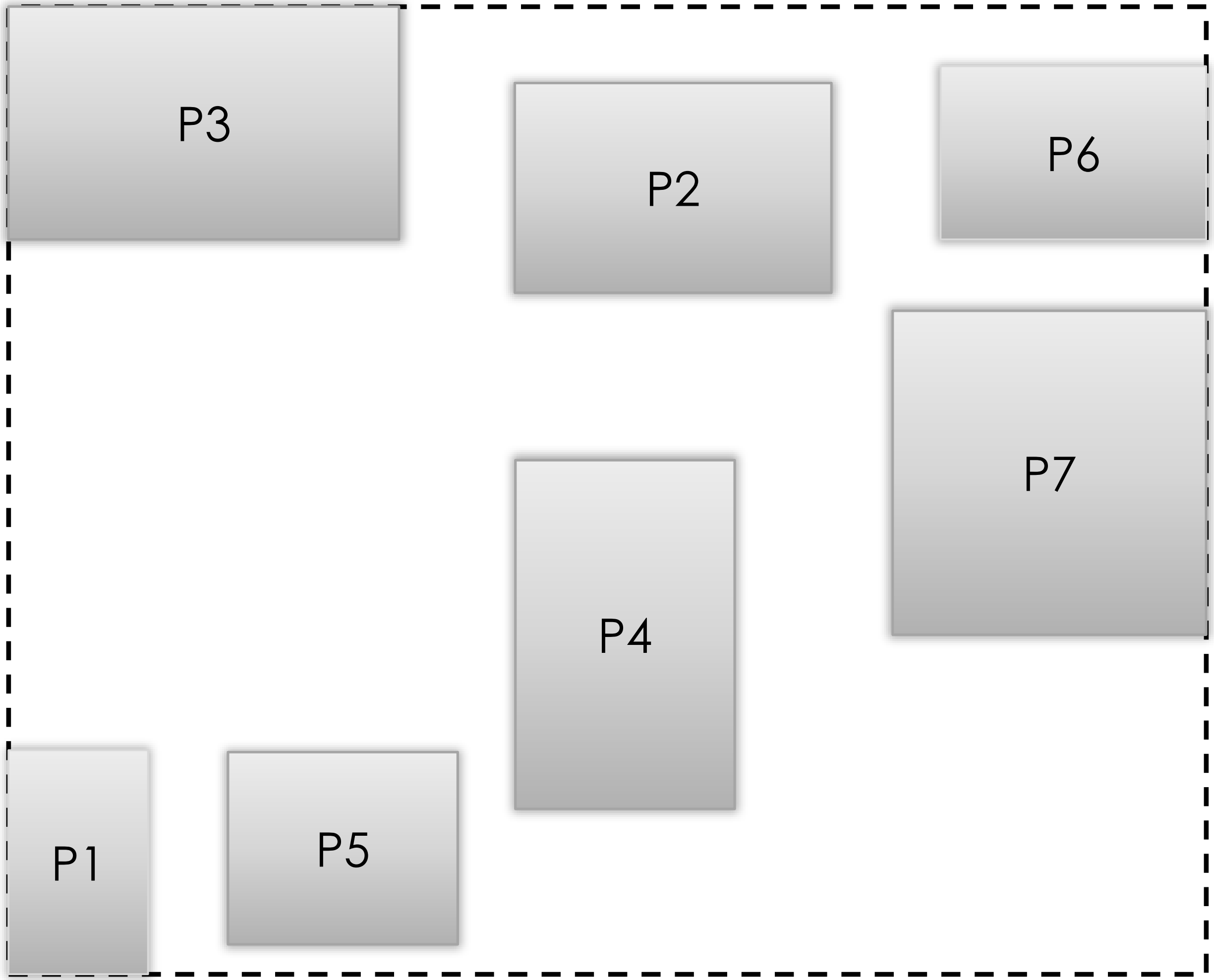
R-tree – Example Adjusting the tree

M =3 m =2



R-tree – Node Splitting: Quadratic Algorithm Example

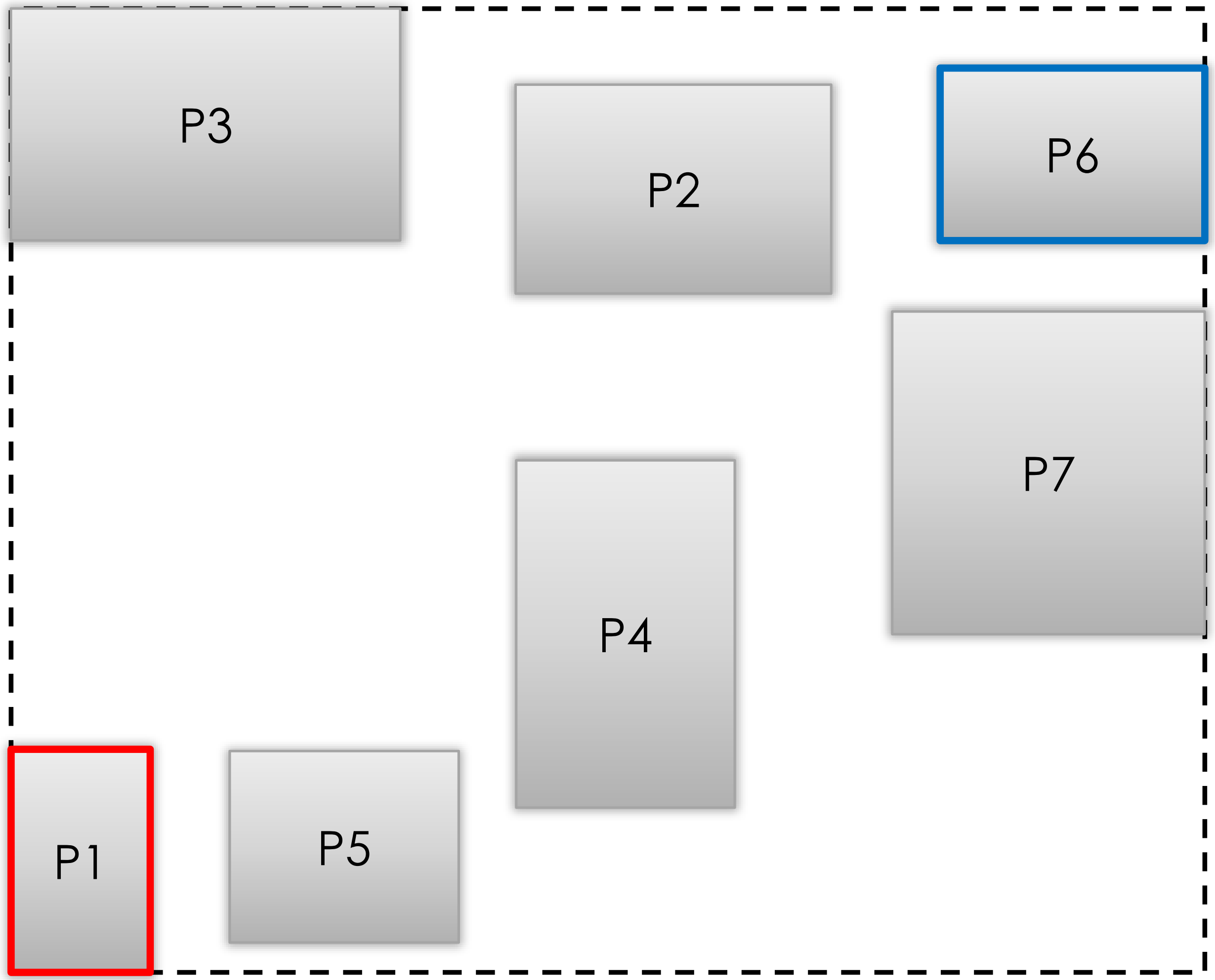
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 1

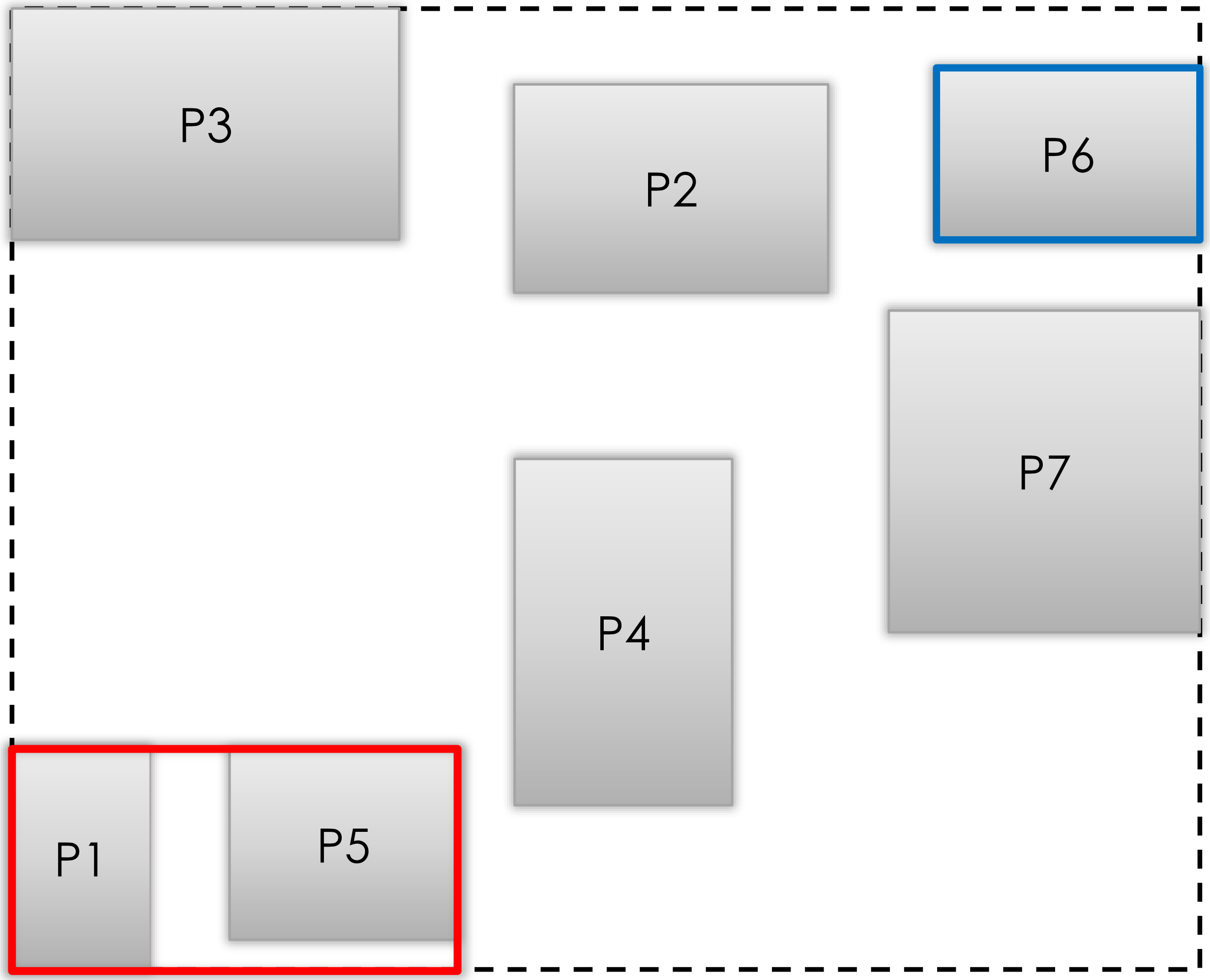
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 2

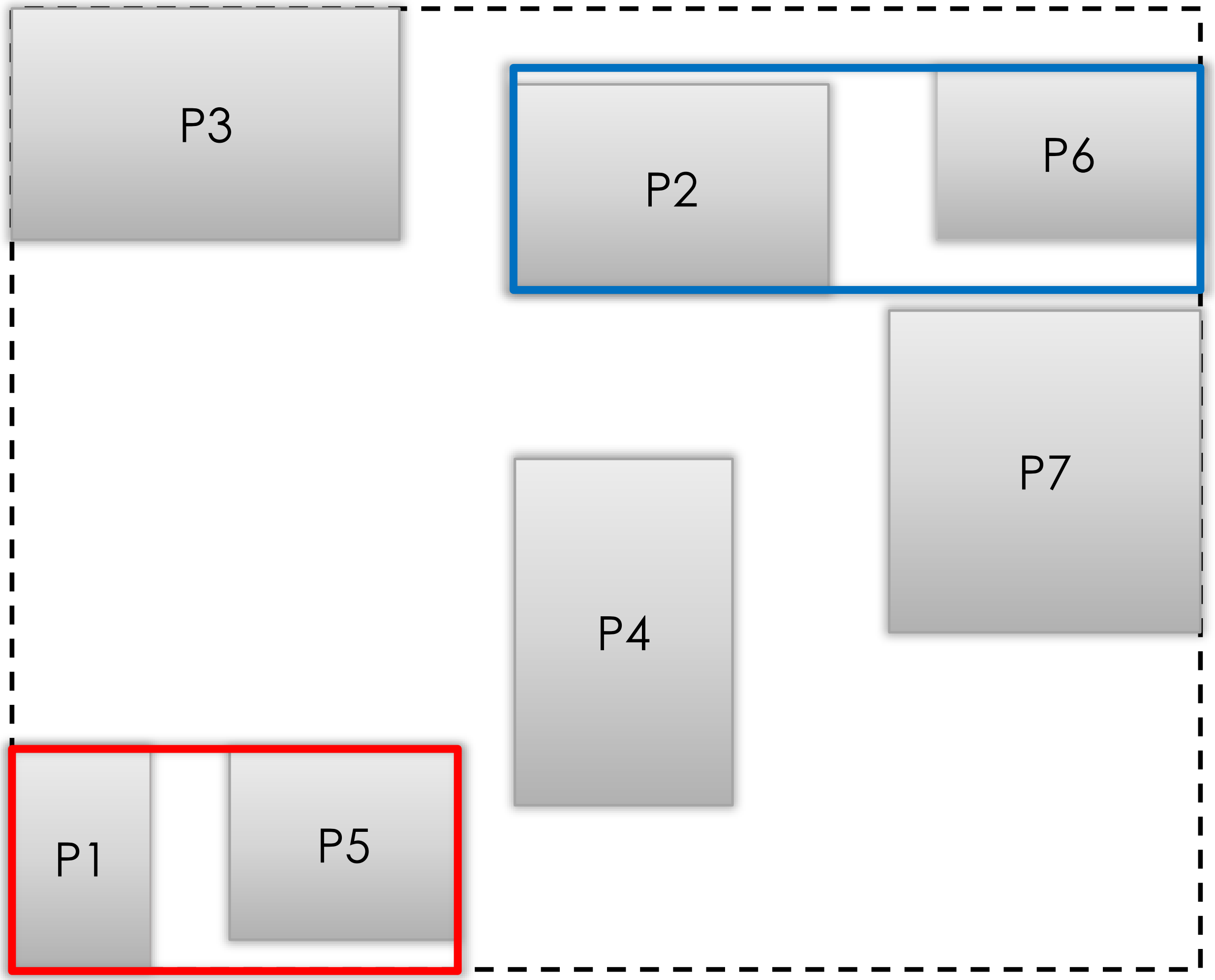
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 3

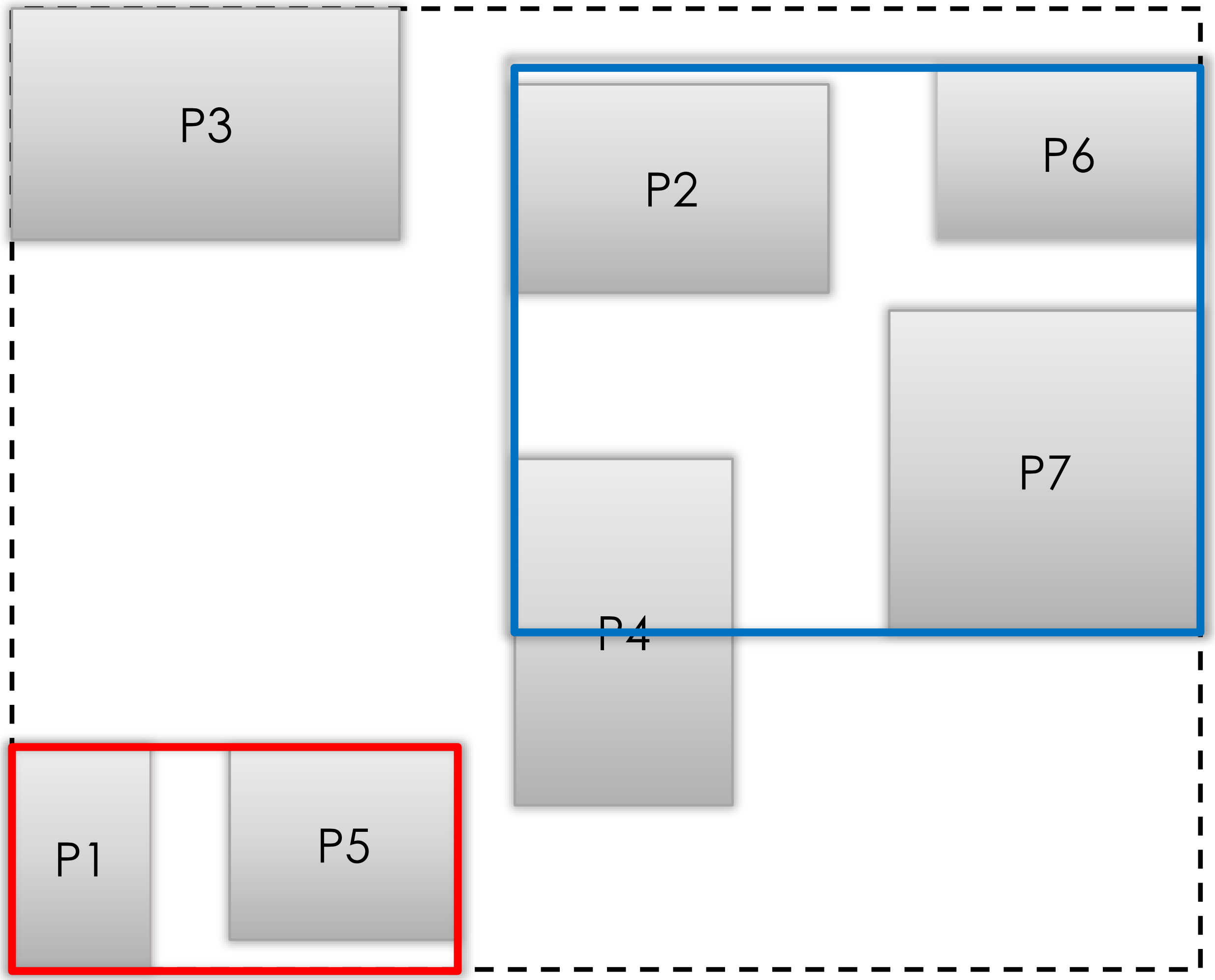
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 4

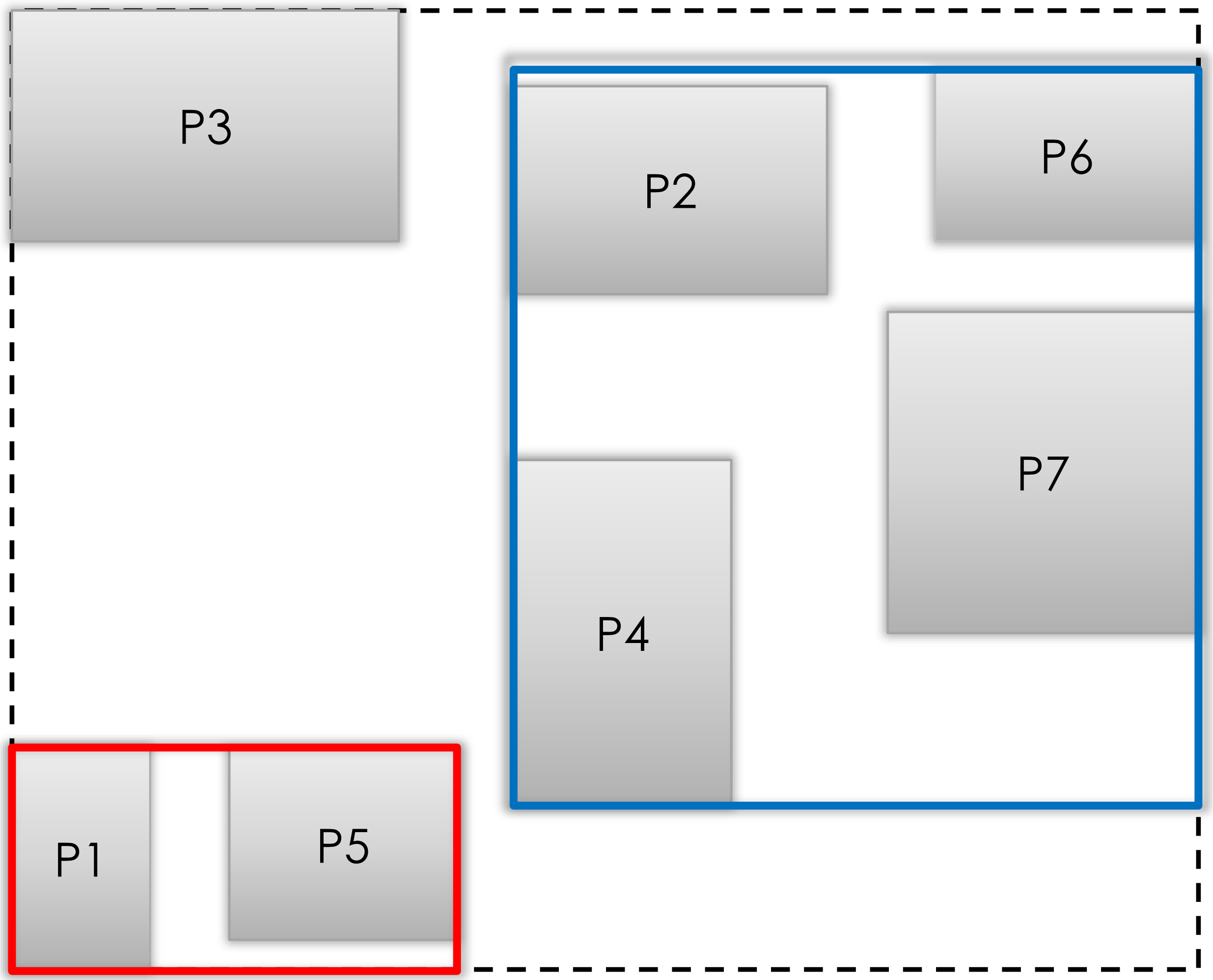
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 5

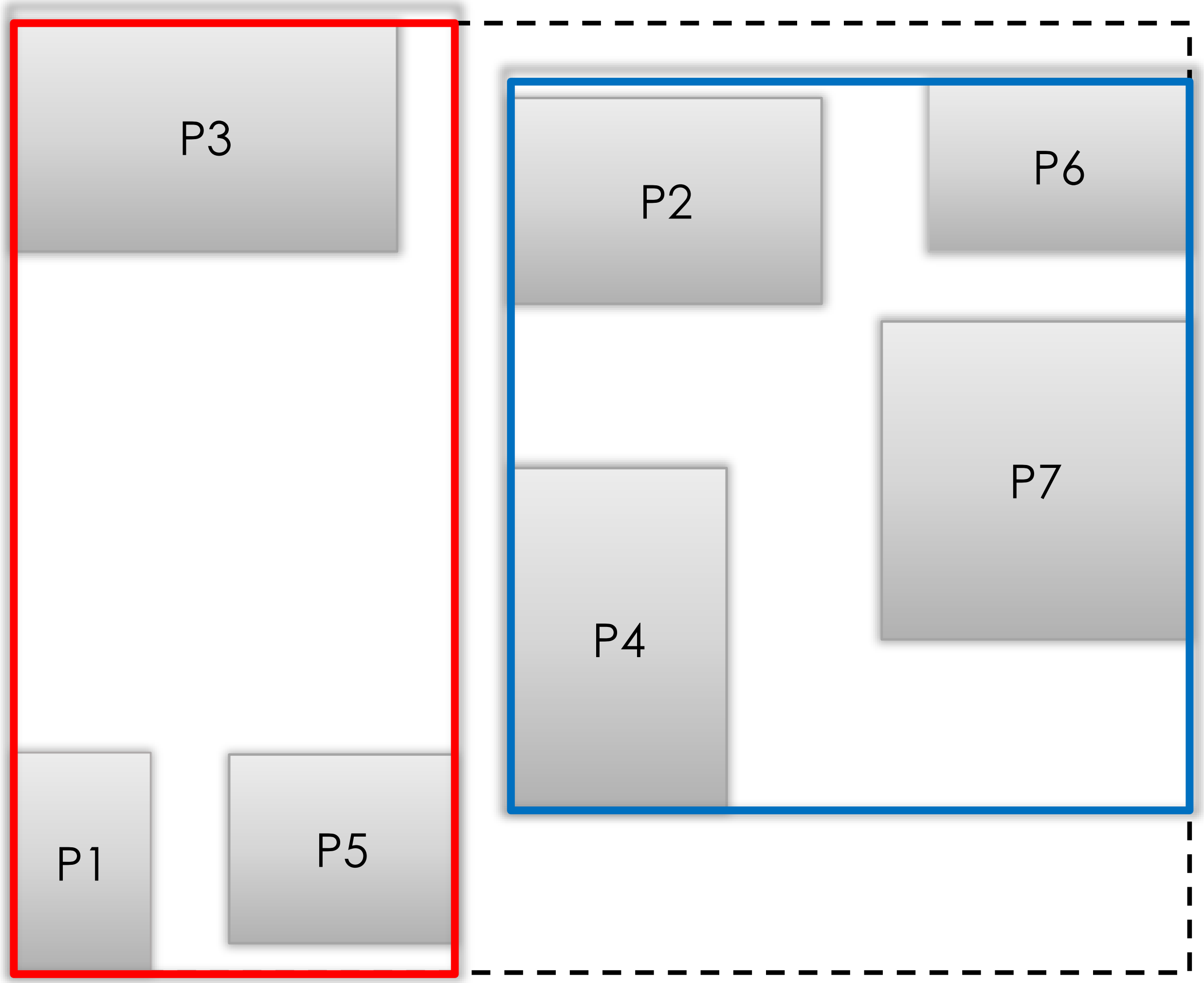
M = 6
m = 3



Node to be Split

R-tree – Node Splitting: Quadratic Algorithm Step 6

M = 6
m = 3



Node to be Split