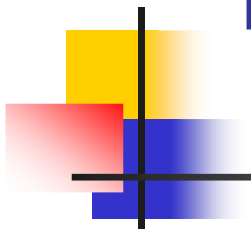
BST: Extension 01

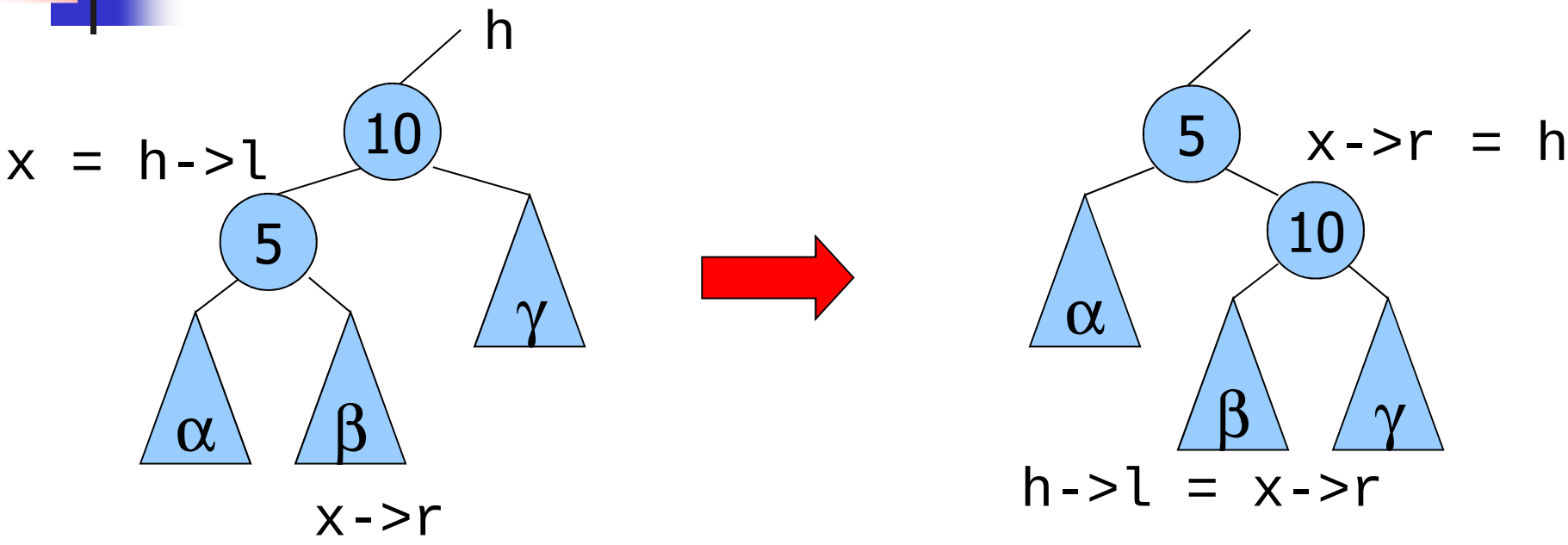
Gianpiero Cabodi and Paolo Camurati
Dip. Automatica e Informatica
Politecnico di Torino



First BST Extension: Root Operations

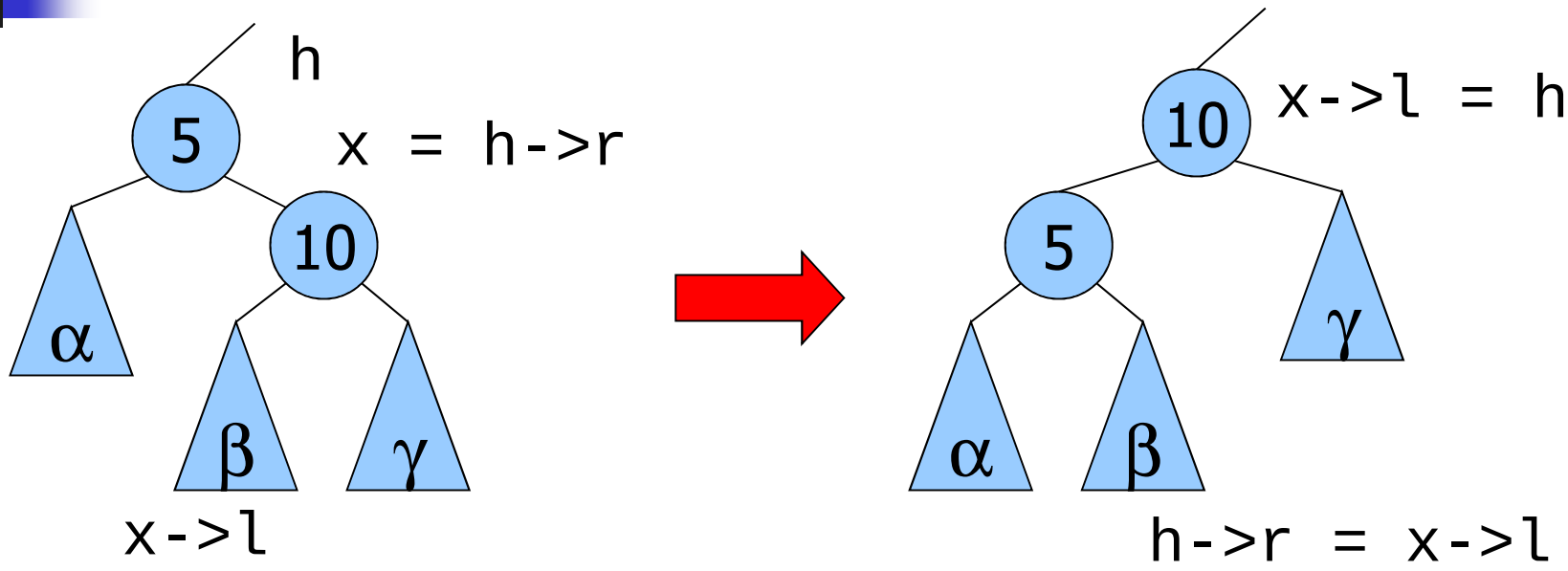
- Root insertions insert more recent nodes closer to the BST root
  - It is faster to search more recent keys
- The core idea is to
  - Insert a node onto a leaf
  - Then, eventually, move the node on the tree root
    - Moving a node to the tree root needs rotations

Right Rotation of a BST



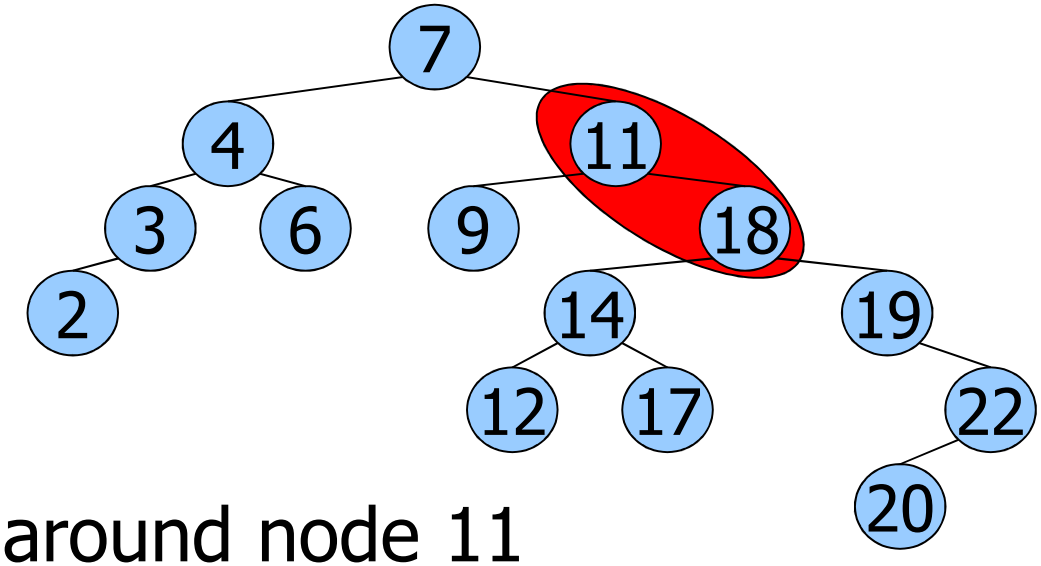

```
link rotR(link h) {
  link x = h->l;
  h->l = x->r;
  x->r = h;
  return x;
}
```

Left Rotation of a BST

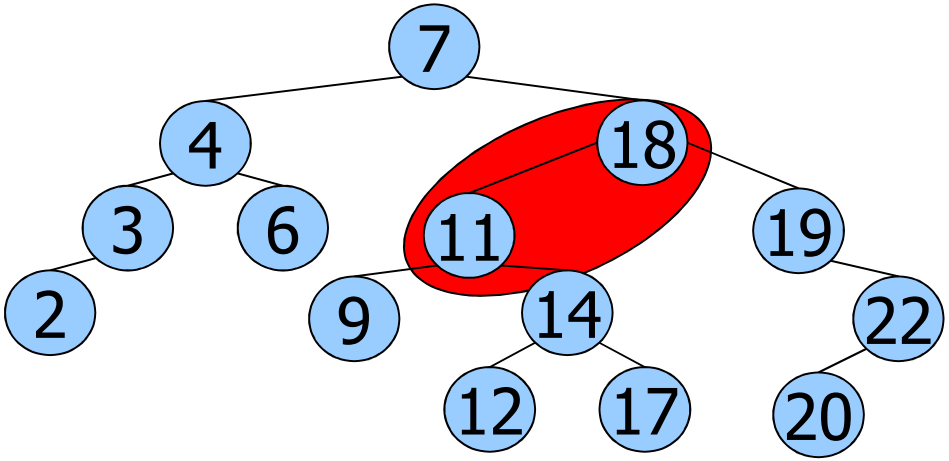


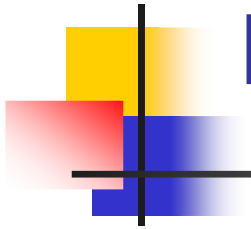

```
link rotL(link h) {
  link x = h->r;
  h->r = x->l;
  x->l = h;
  return x;
}
```

Example



Left Rotation around node 11

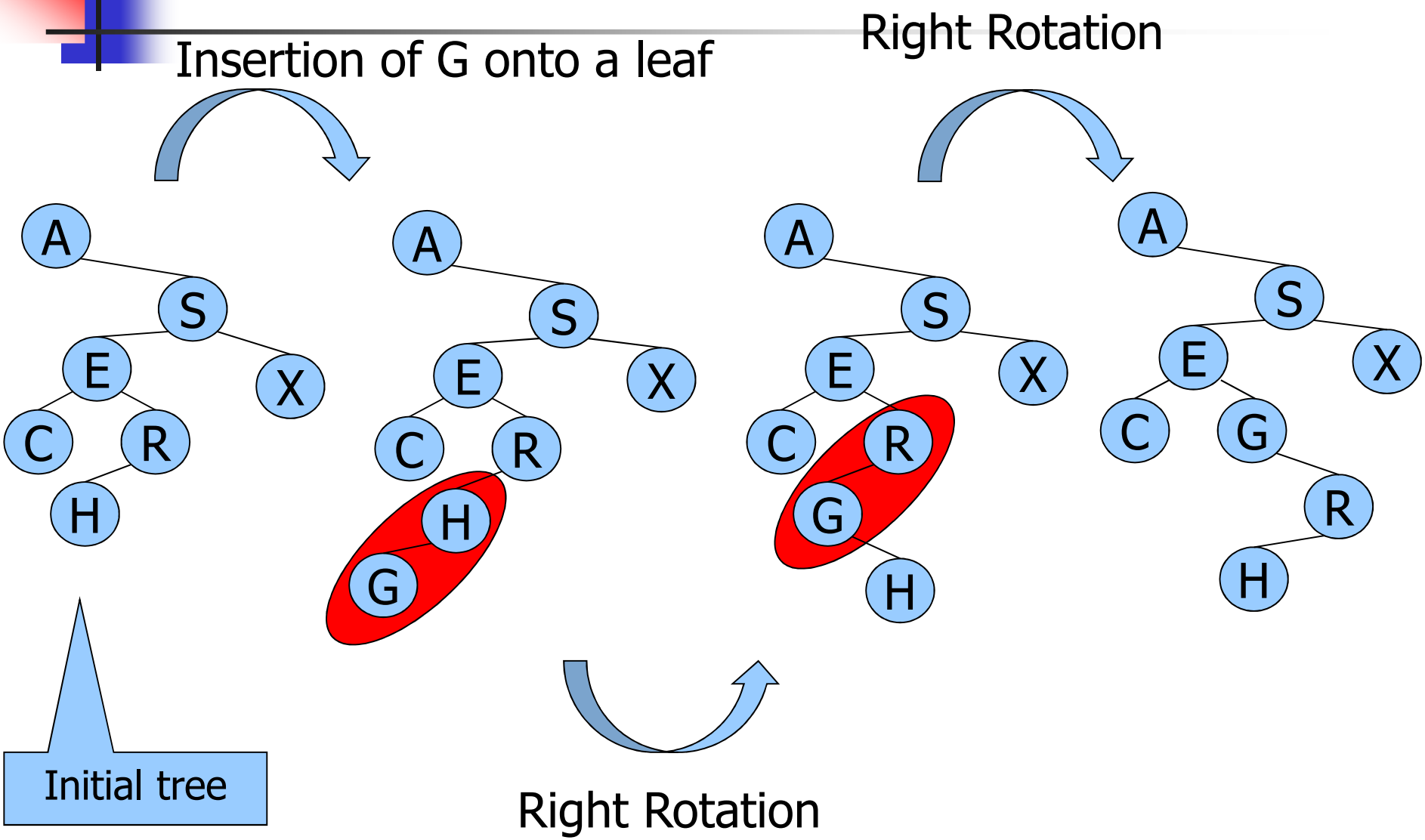




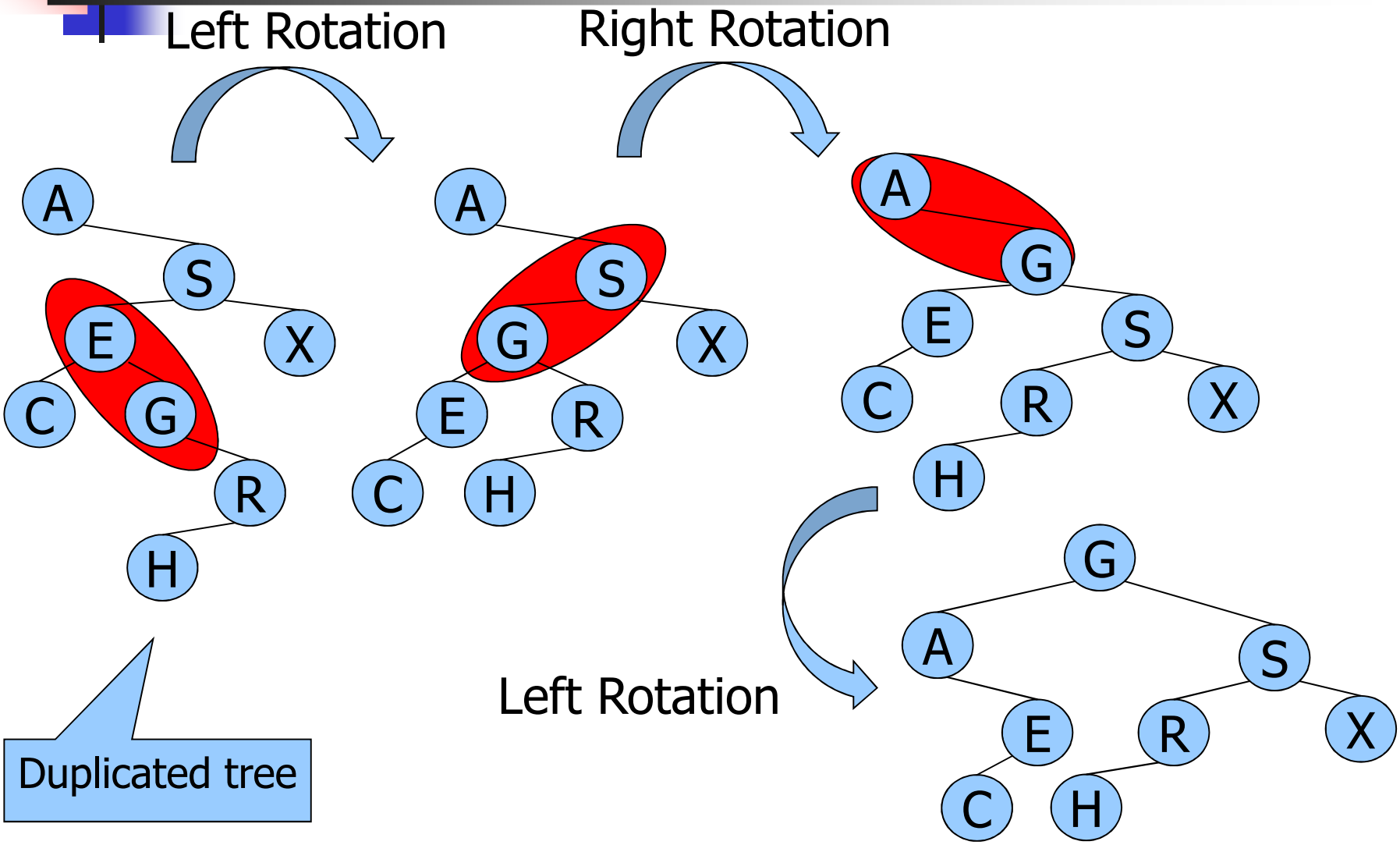
Root Insertion

- To insert a new node on the root of a BST it is possible to use the following recursive procedure
  - Insert the new node into the appropriate sub-tree following the leaf insertion procedure
  - Rotate the node the force it onto the tree root

Example


Insertion of G onto a leaf
Right Rotation
A
S
E
X
C
R
H
G
Initial tree
Right Rotation

Example


Left Rotation
Right Rotation
A
S
E
X
C
G
R
H
A
S
G
X
E
R
C
H
A
G
E
S
C
R
X
H
Left Rotation
G
A
S
E
R
X
C
H
Duplicated tree

Implementation

```
link insert_root_r (link root, Item x, link z) {
  if (root == z)
    return NEW (x, z, z);

  if (ITEMless(x, root->item)) {
    root->l = insert_root_r (root->l, x, z);
    root = rotR (root);
  } else  {
    root->r = insert_root_r (root->r, x, z);
    root = rotL (root);
  }

  return root;
}
```

Recur left
→ Rotate right

Recur right
→ Rotate left