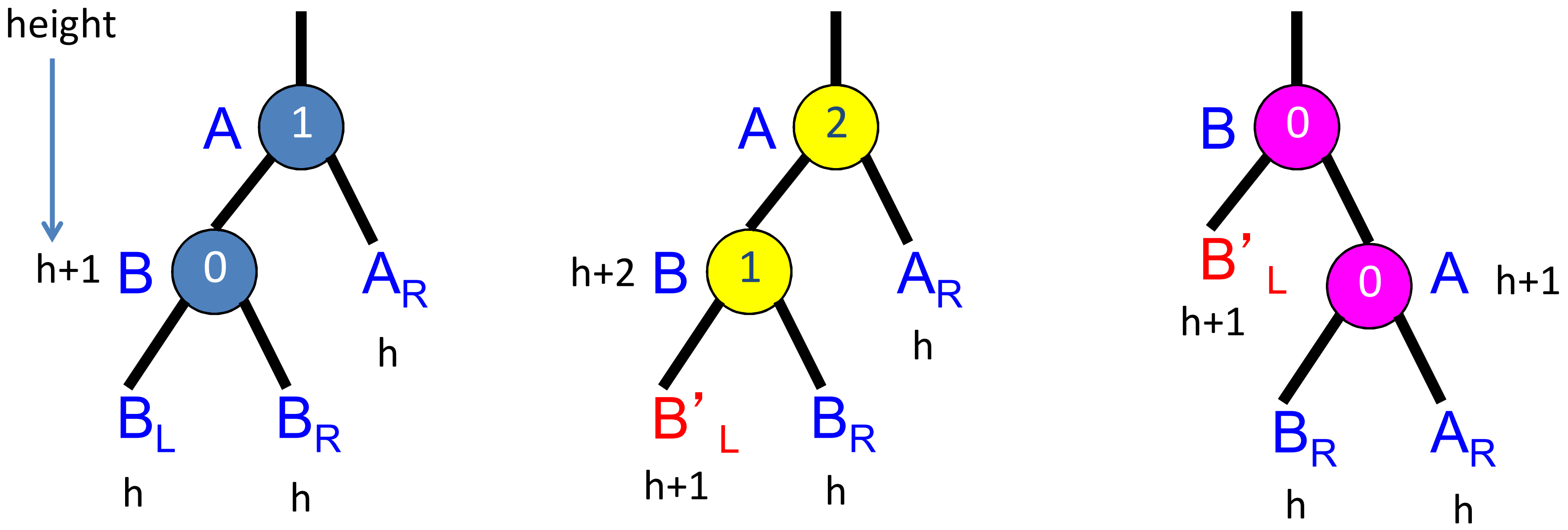
Imbalance Types

- RR … newly inserted node is in the right subtree of the right subtree of A
- LL … left subtree of left subtree of A
- RL… left subtree of right subtree of A
- LR… right subtree of left subtree of A

LL Rotation

- Right rotation on A



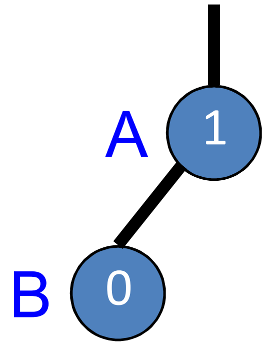
Before insertion After insertion After rotation

Red: subtree to which a new node is inserted

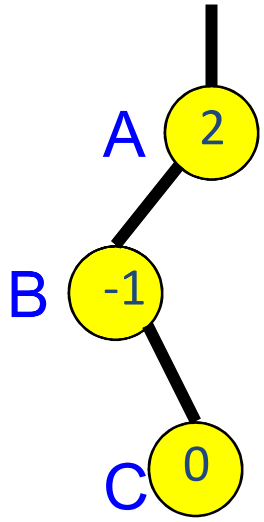
LR Rotation (case 1)

- Left – Right rotations
  - Left on B, right on A

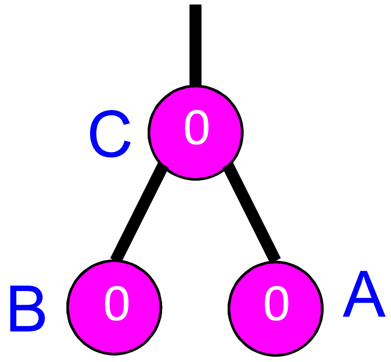
B was a leaf prior to insert!



Before insertion



After insertion

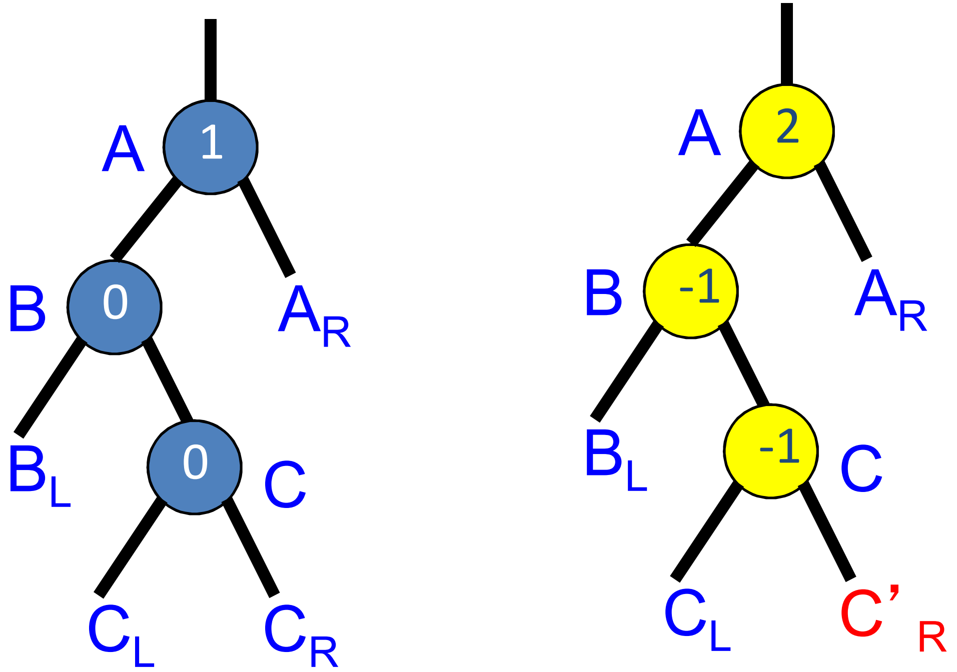


After rotation

LR Rotation (case 2)

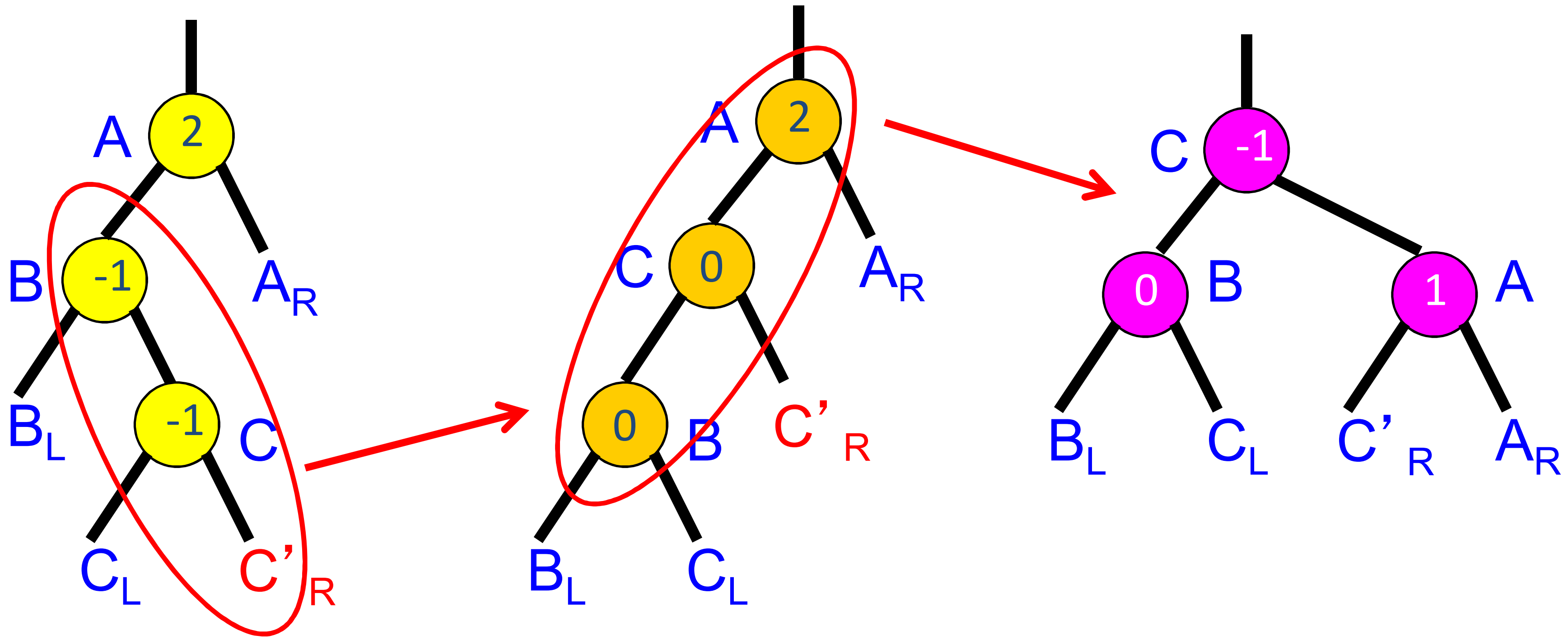
- Left – Right rotations
  - Left on B, right on A

B was not a leaf prior to insert!



Red: subtree to which a new node is inserted

LR Rotation (case 2)



After insertion

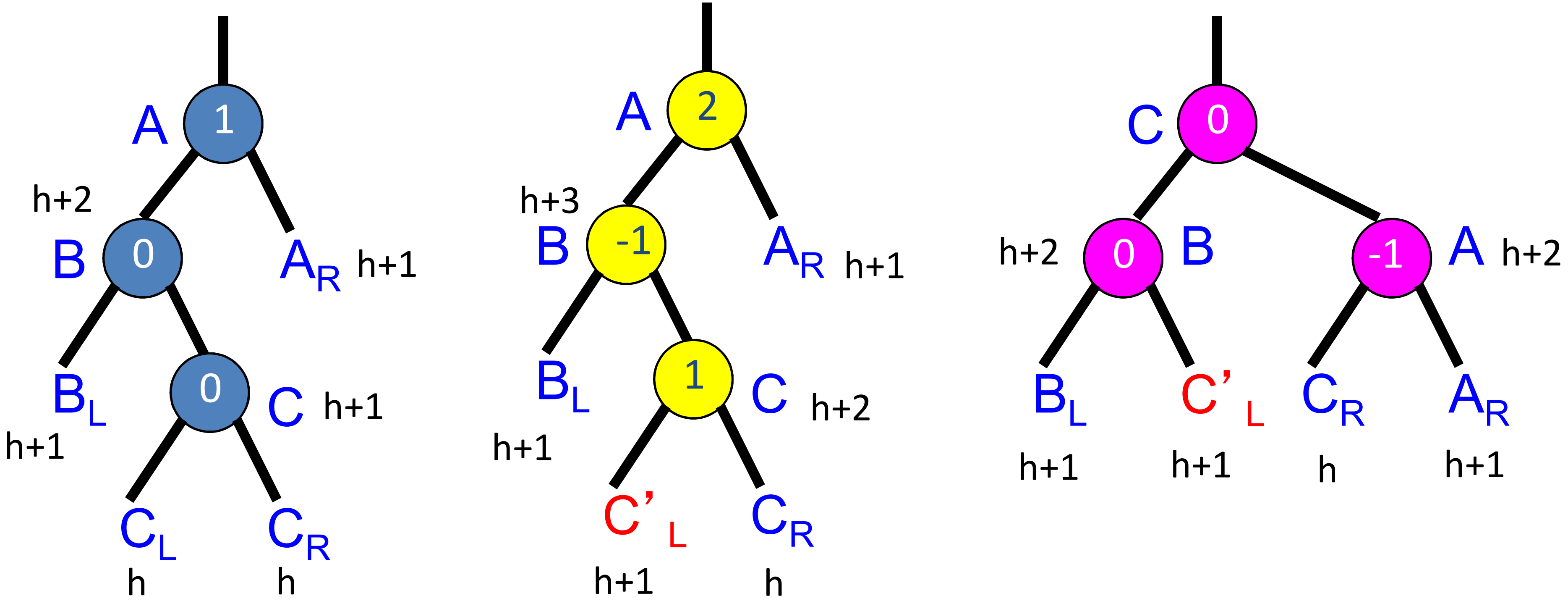
After RR rotation

After LL rotation

Red: subtree to which a new node is inserted

LR Rotation (case 3)

- Left – Right rotations
  - Left on B, right on A

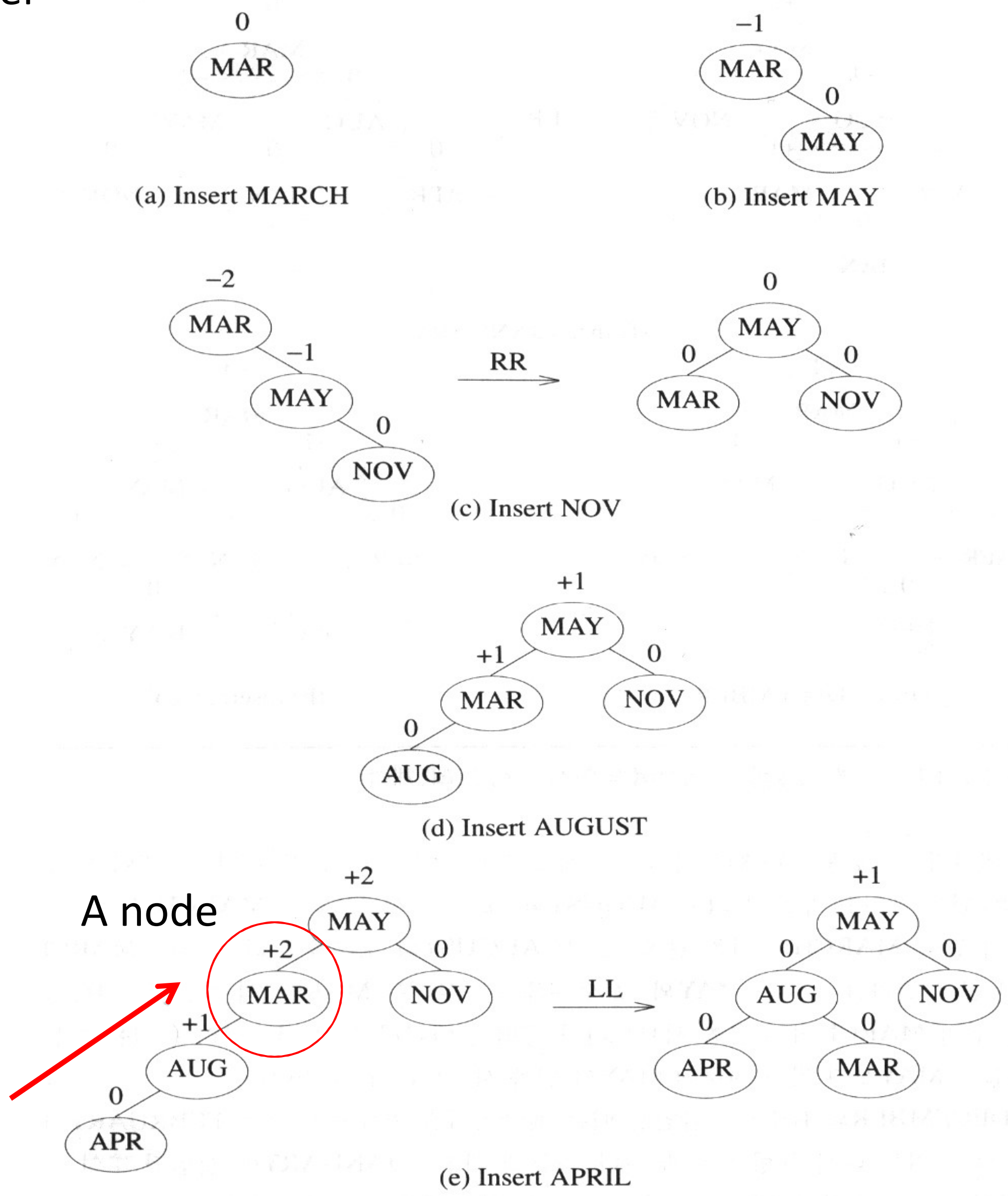


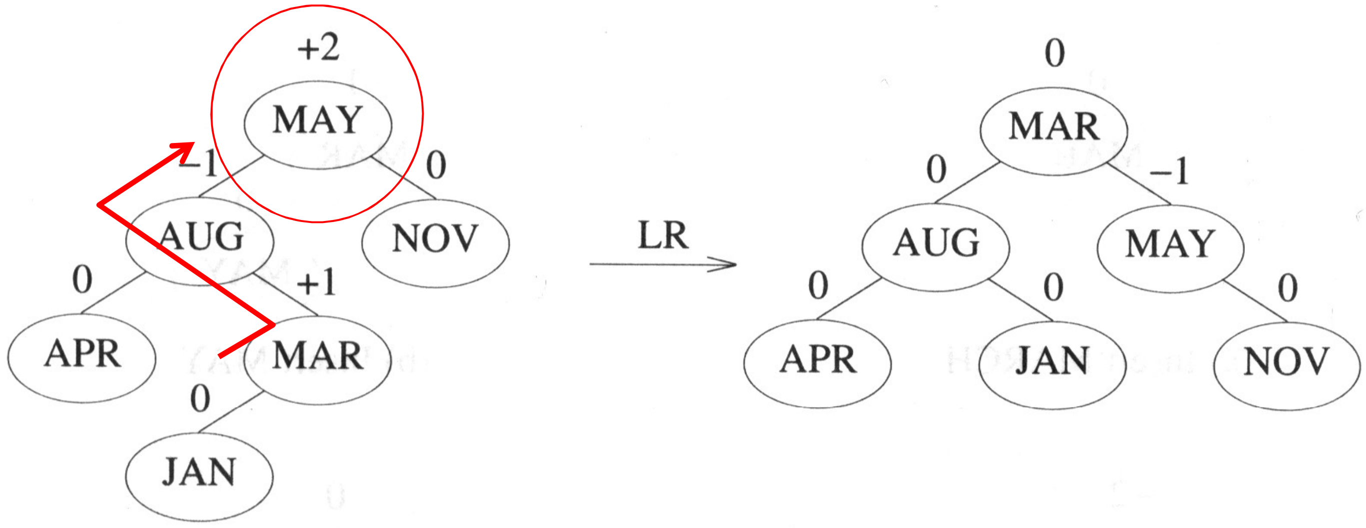
Red: subtree to which a new node is inserted

Single & Double Rotations

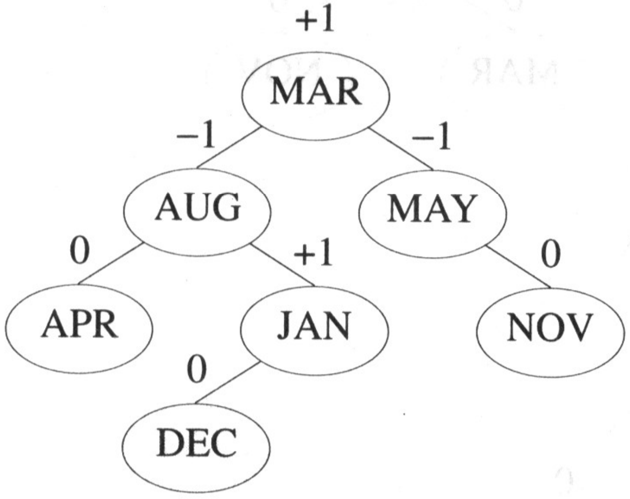
- Single rotation
  - –LL and RR
- Double rotation
  - –LR and RL
  - –LR can be viewed as RR followed by LL
  - –RL can be viewed as LL followed by RR

Alphabetical order

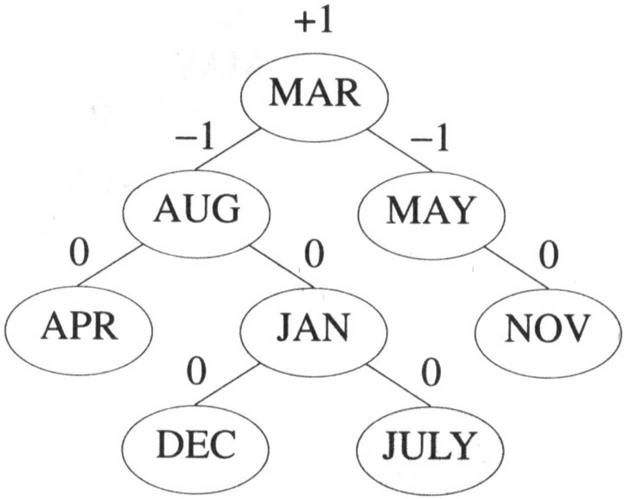




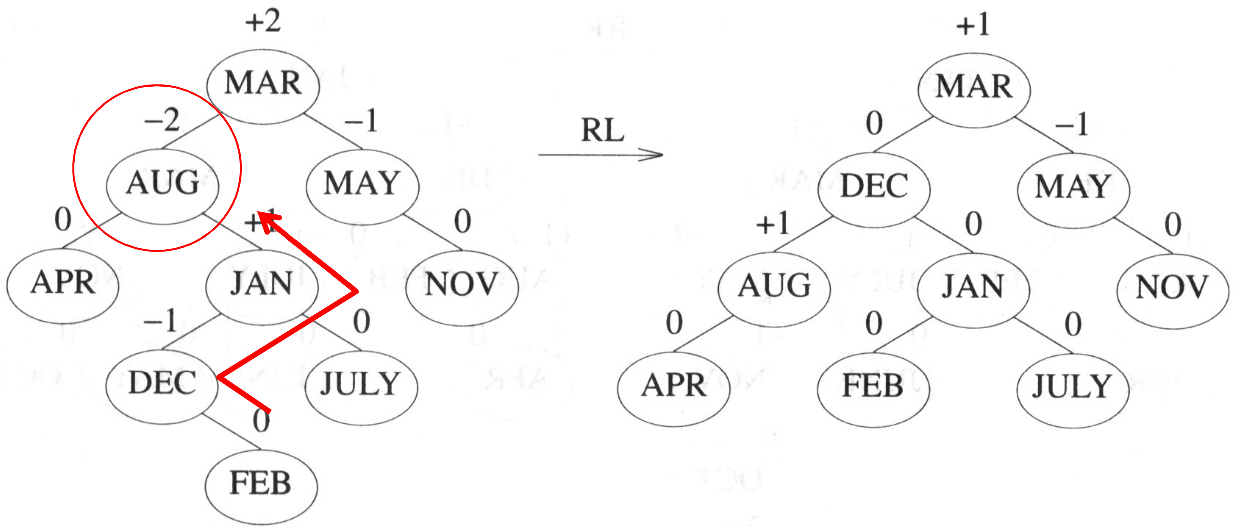
(f) Insert JANUARY



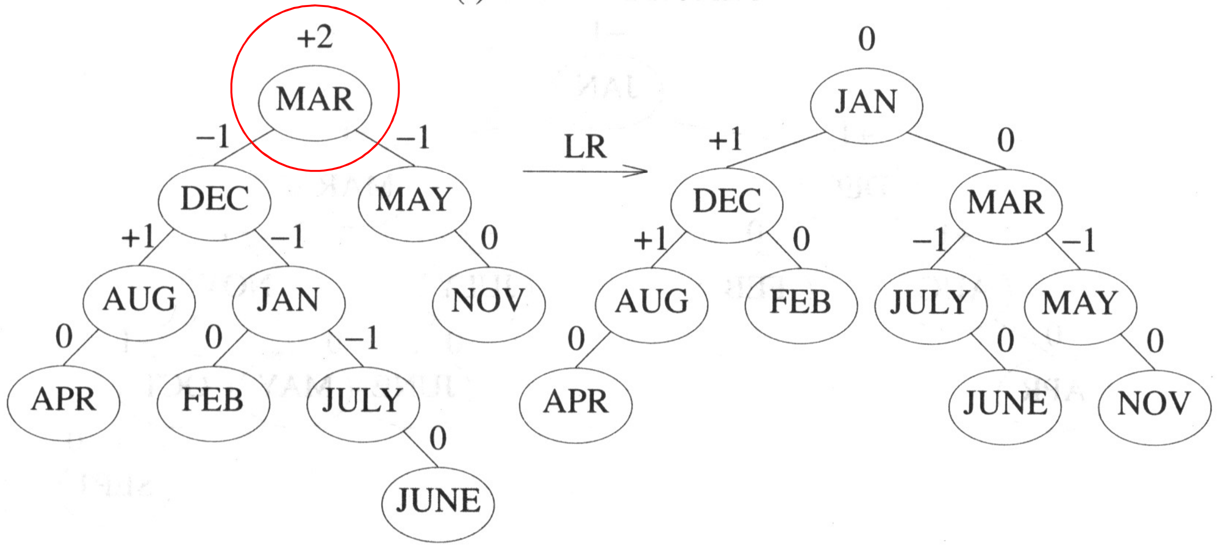
(g) Insert DECEMBER

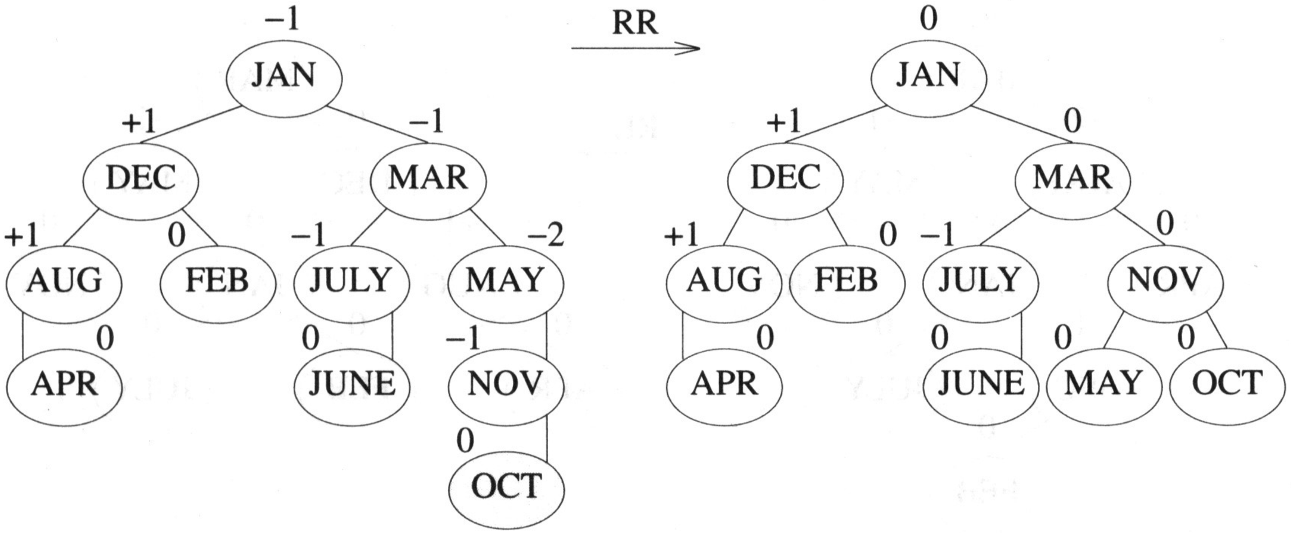


(h) Insert JULY

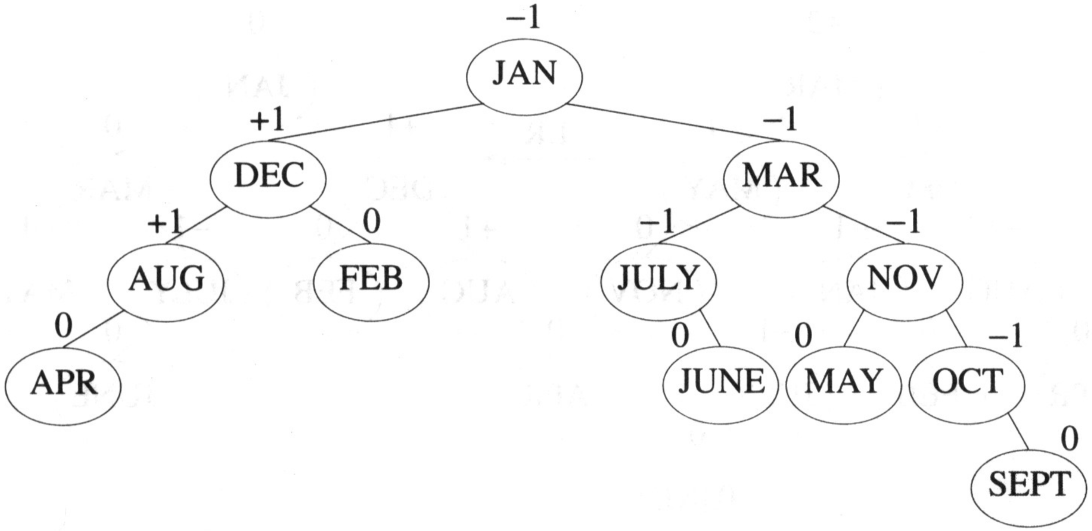


(i) Insert FEBRUARY





(k) Insert OCTOBER



(l) Insert SEPTEMBER

Number of Rebalancing Rotations

- Insert : at most 2 rotations
- Delete : at most O(log n) rotations
- Rotation frequency when insert random numbers
  - No rotation … 53.4% (approx)
  - LL/RR … 23.3% (approx)
  - LR/RL … 23.2% (approx)

Discussion

- AVL trees manage strict height-balanced structure
  - Height is bounded by O(log n)
    - Search, insertion, deletion are O(log n)
  - Fast for lookup intensive problems
  - Insert, delete can be slower than lookup

Questions?