Adelson-Velskii and Landis

The AVL Balance Condition

Left and right subtrees of *every node* have *heights* **differing by at most 1**

Definition: **balance**(*node*) = height(*node*.left) – height(*node*.right)

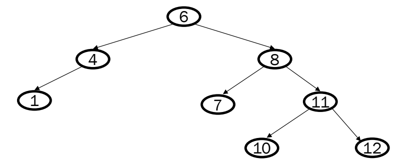
AVL *property*: **for every node *x*, –1 ≤ balance(*x*) ≤ 1**

- Ensures small depth
  - Will prove this by showing that an AVL tree of height *h* must have a number of nodes *exponential* in *h*
- Easy (well, efficient) to maintain
  - Using single and double rotations

3

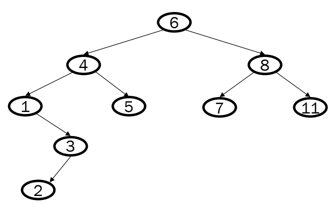
EX1: An AVL tree?

pollev.com/artliu



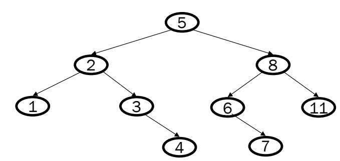
5

EX2: An AVL tree?



7

EX3: An AVL tree?

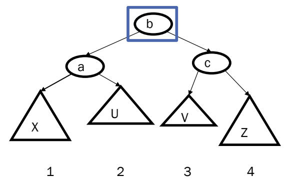


9

AVL tree insert

Let *b* be the node where an imbalance occurs.
Four cases to consider. The insertion is in the

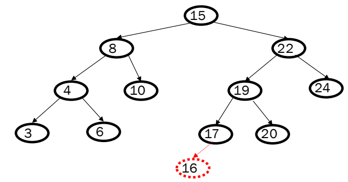
1. left subtree of the left child of *b*.
2. right subtree of the left child of *b*.
3. left subtree of the right child of *b*.
4. right subtree of the right child of *b*.



Idea: Cases 1 & 4 are solved by a single rotation
Cases 2 & 3 are solved by a double rotation

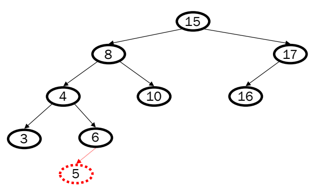
14

Another example: `insert(16)`



23

Insert 5



36

Insert into an AVL tree: a b e c d

40