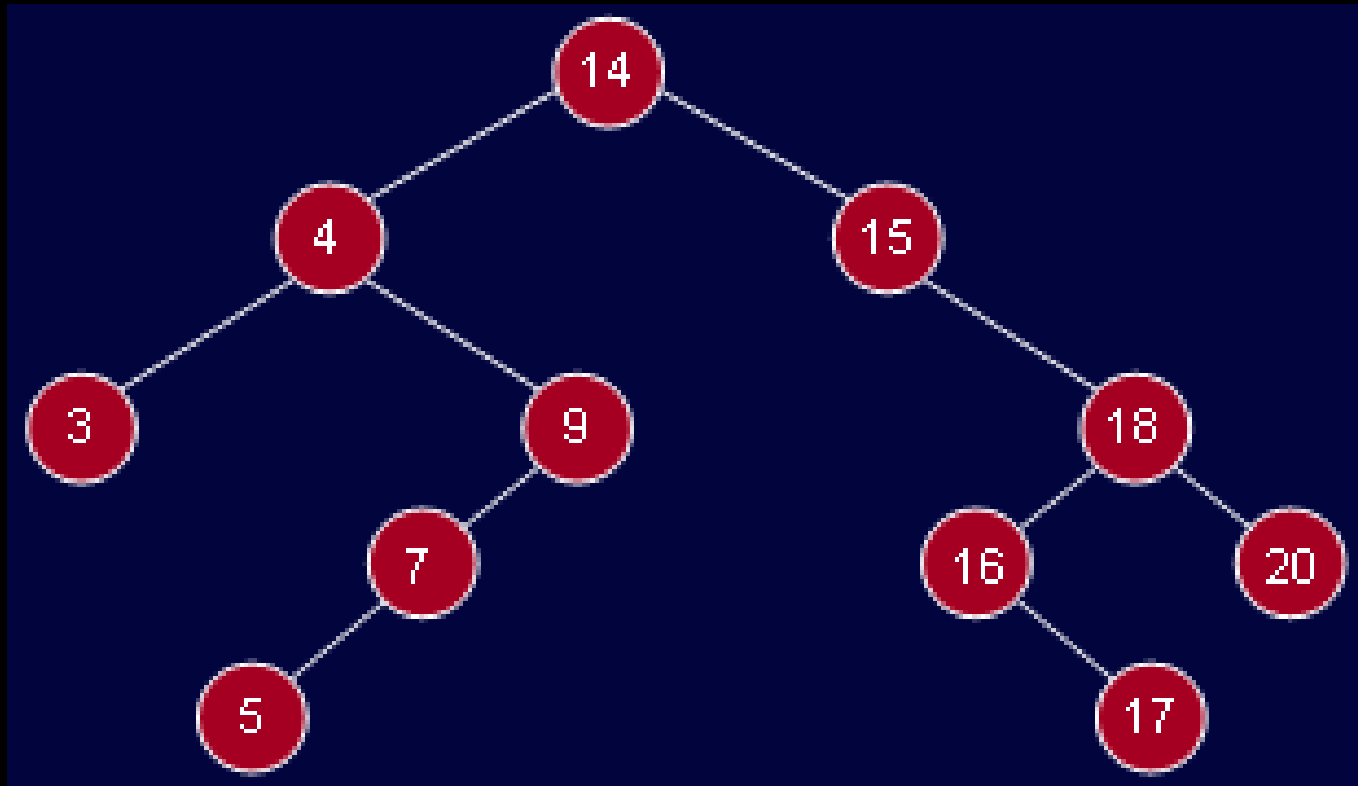


Lecture # 12

AVL Trees

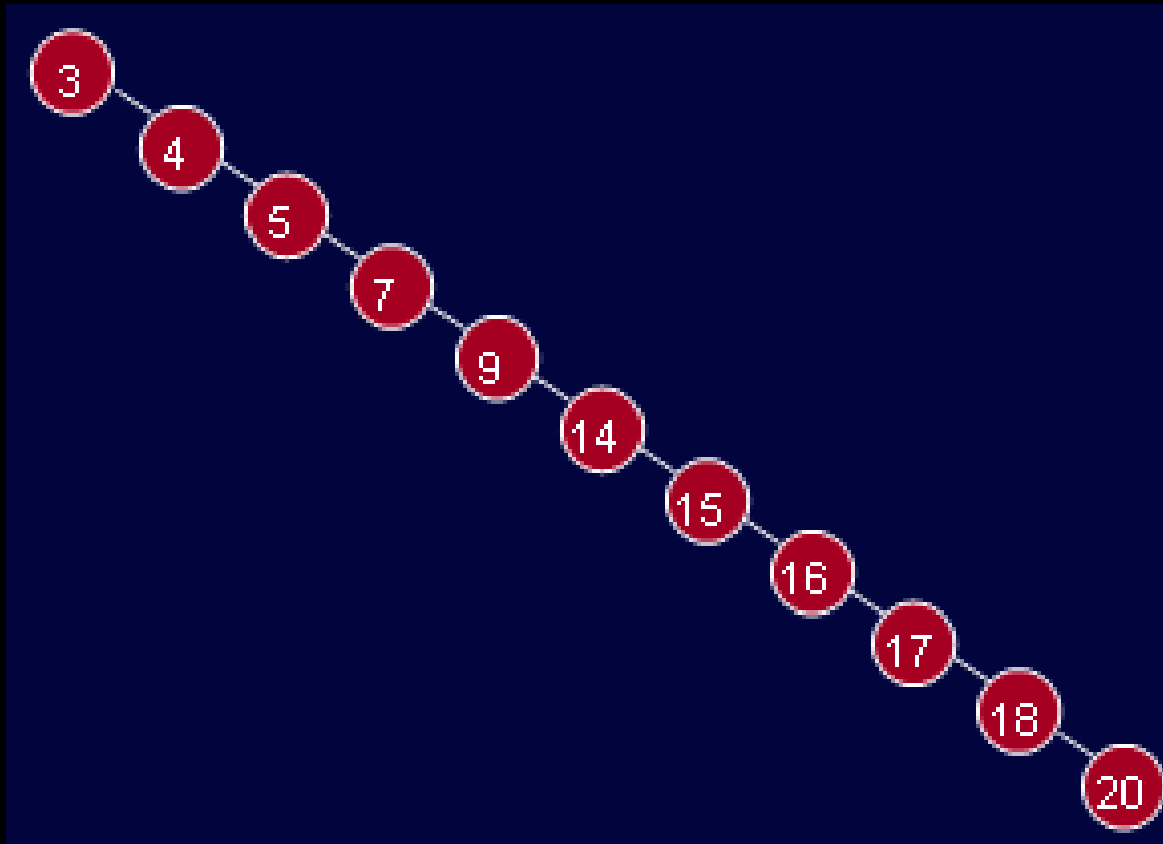
Binary Search Trees

- BST for 14, 15, 4, 9, 7, 18, 3, 5, 16, 20, 17



Binary Search Trees

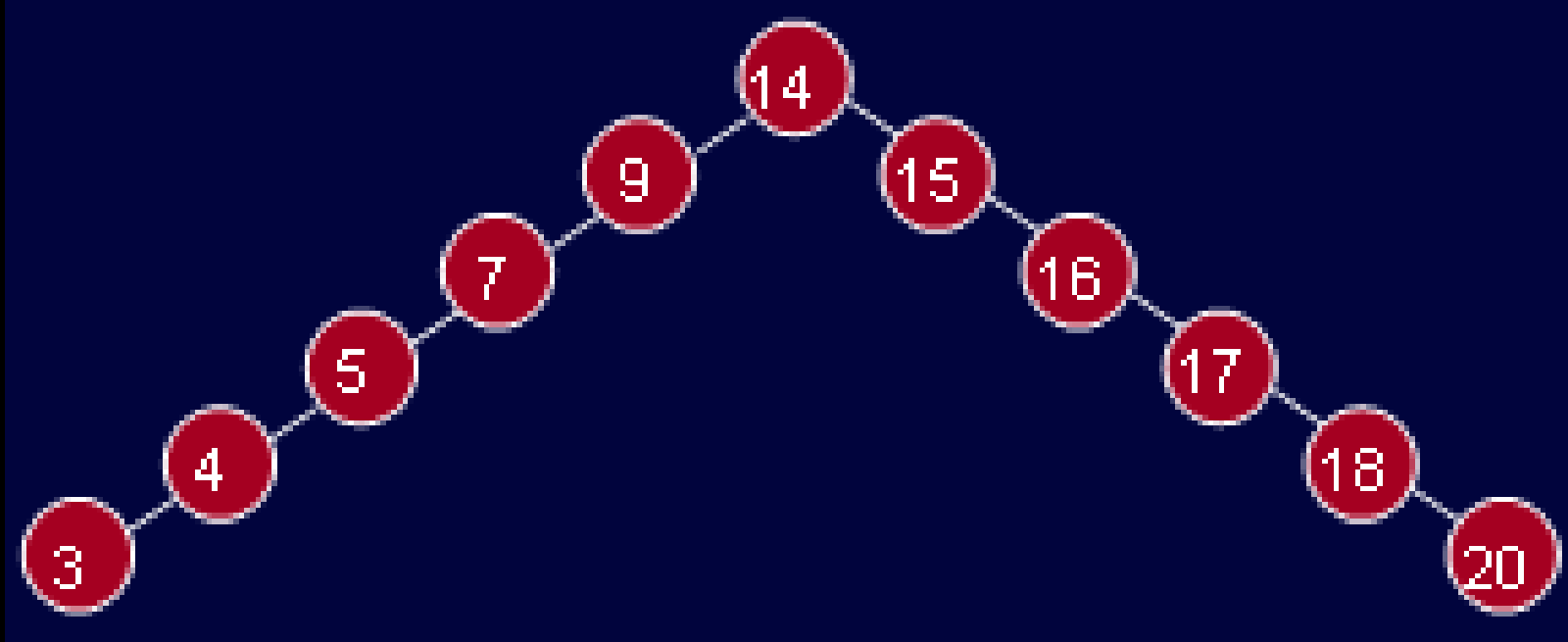
- BST for 3 4 5 7 9 14 15 16 17 18 20



Balanced BST

- We should keep the tree *balanced*.
- One idea would be to have the left and right subtrees have the same height

Balanced BST



Balanced BST

- We could insist that every node must have left and right subtrees of same height.
- But this requires that the tree be a complete binary tree
- To do this, there must have $(2^{d+1} - 1)$ data items, where d is the depth of the tree.
- This is too rigid a condition.

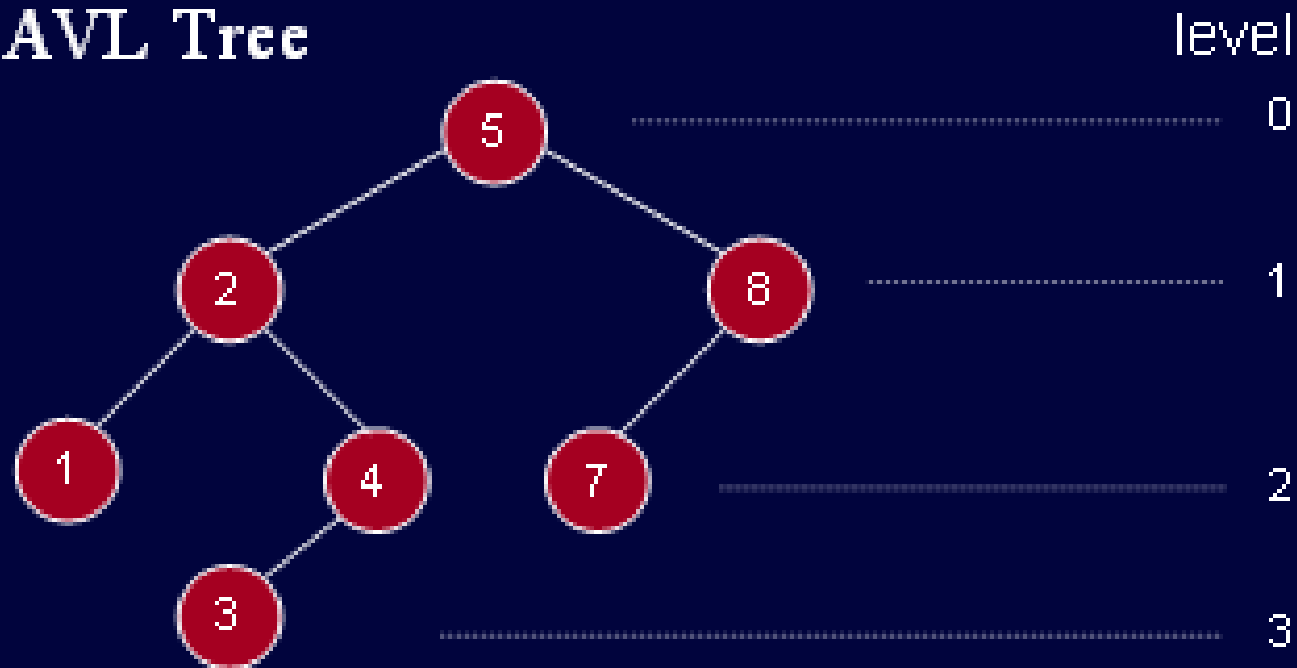
AVL Trees

AVL Trees

- AVL (Adelson-Velskii and Landis) tree.
 - An AVL tree is identical to a BST except
 - height of the left and right subtrees can differ by at most 1.
 - height of an empty tree is defined to be (-1) .

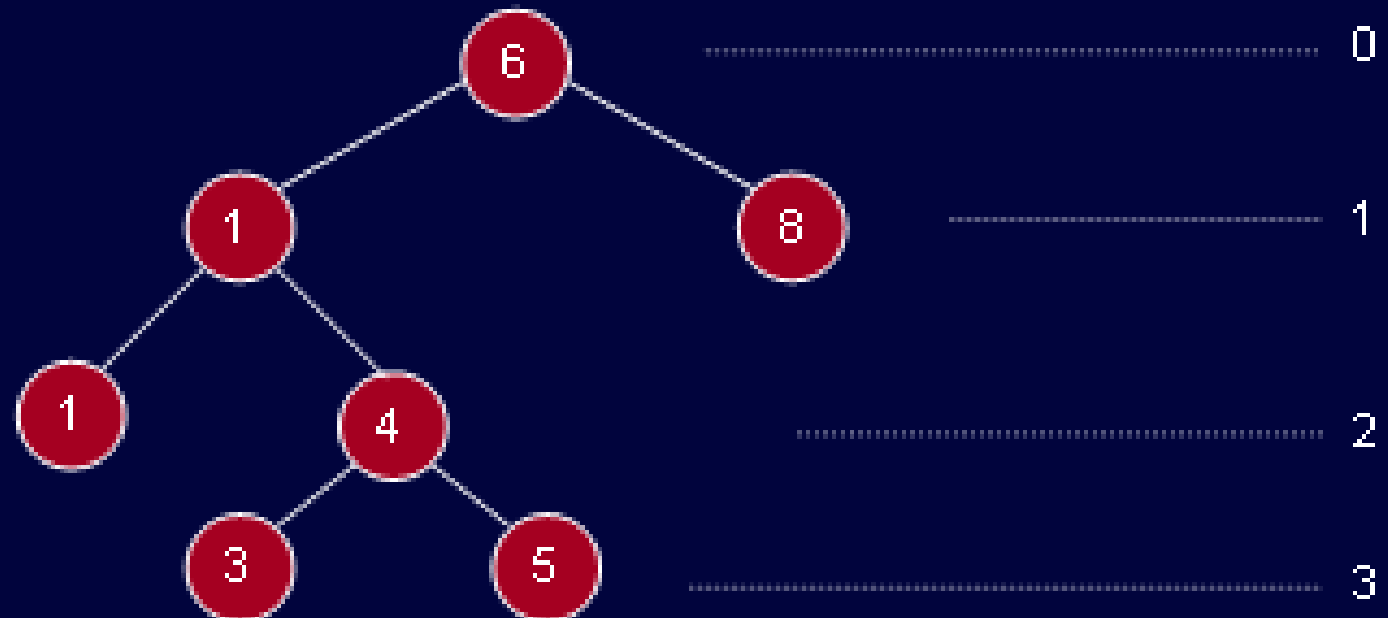
AVL Trees

- **An AVL Tree**



AVL Trees

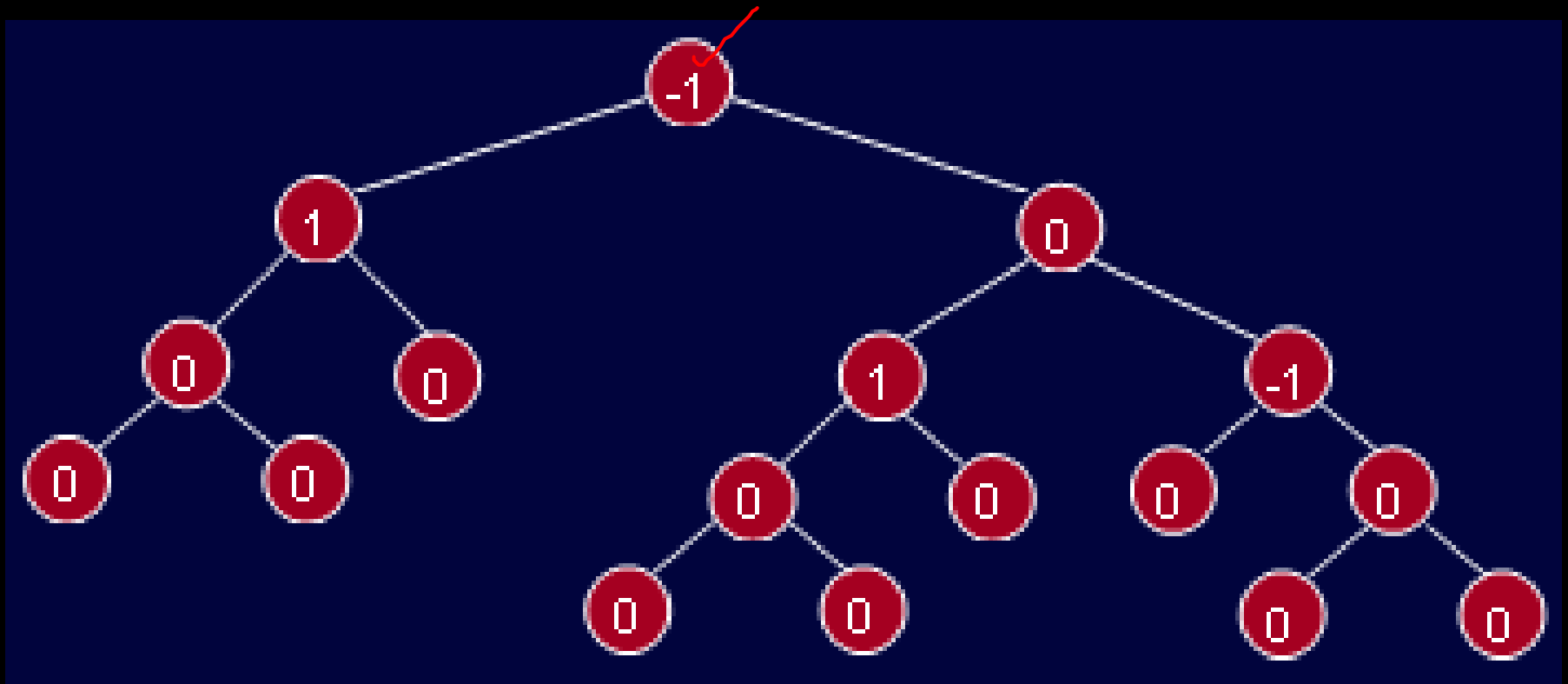
- Not an AVL tree



Balanced Binary Tree

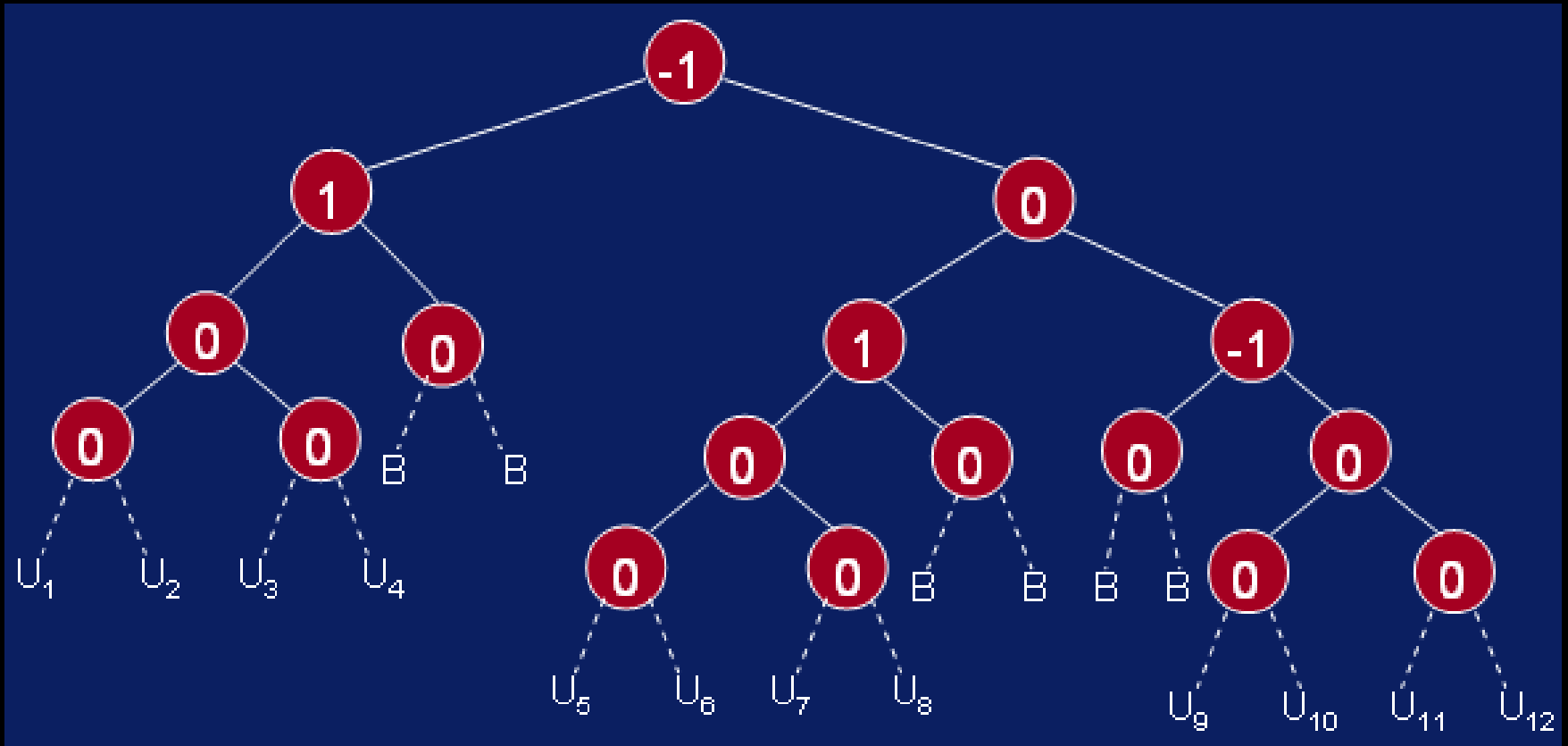
- The *height* of a binary tree is the maximum level of its leaves (also called the depth).
- The **balance of a node** in a binary tree is defined as the height of its left subtree **minus** height of its right subtree.
- Here, for example, is a balanced tree. Each node has an indicated balance of 1, 0, or -1 .

Balanced Binary Tree



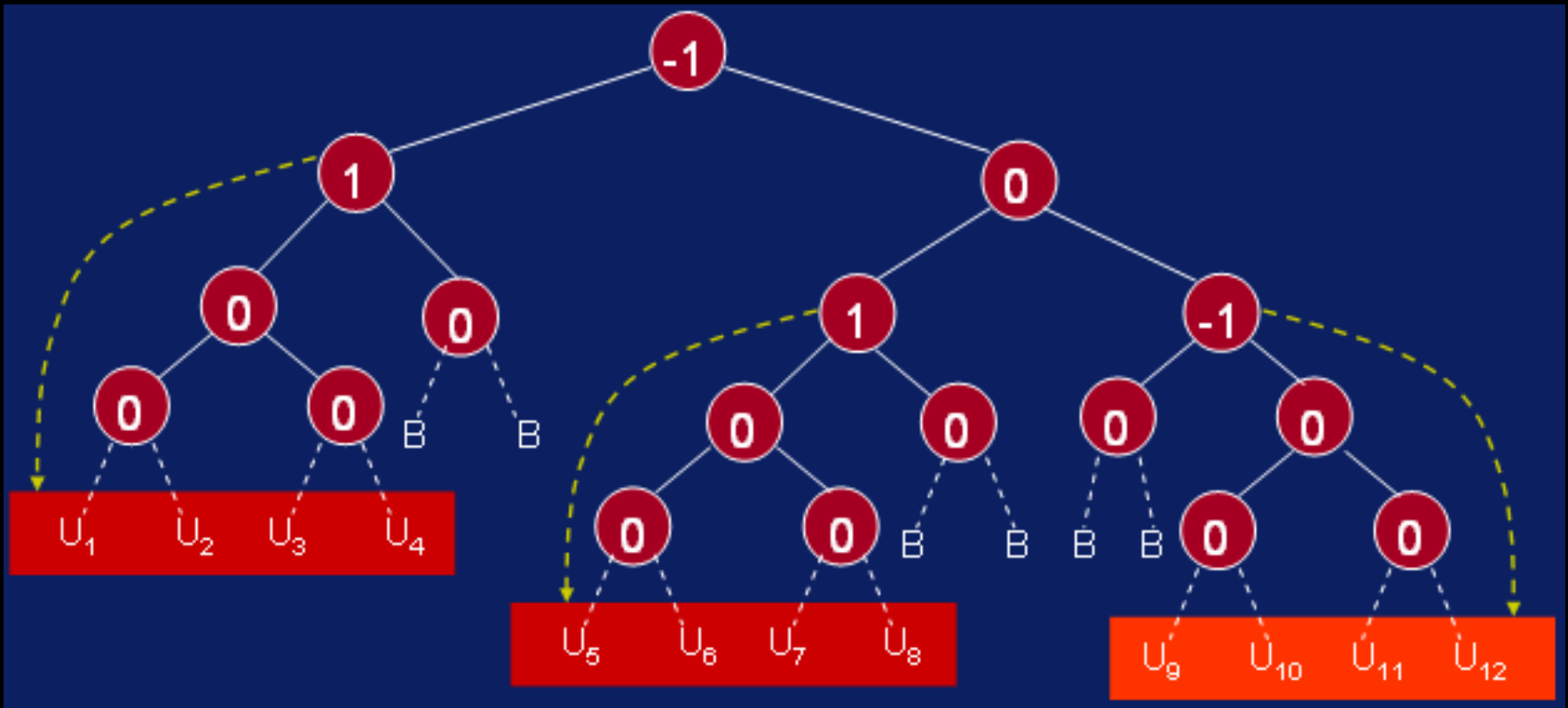
Balanced Binary Tree

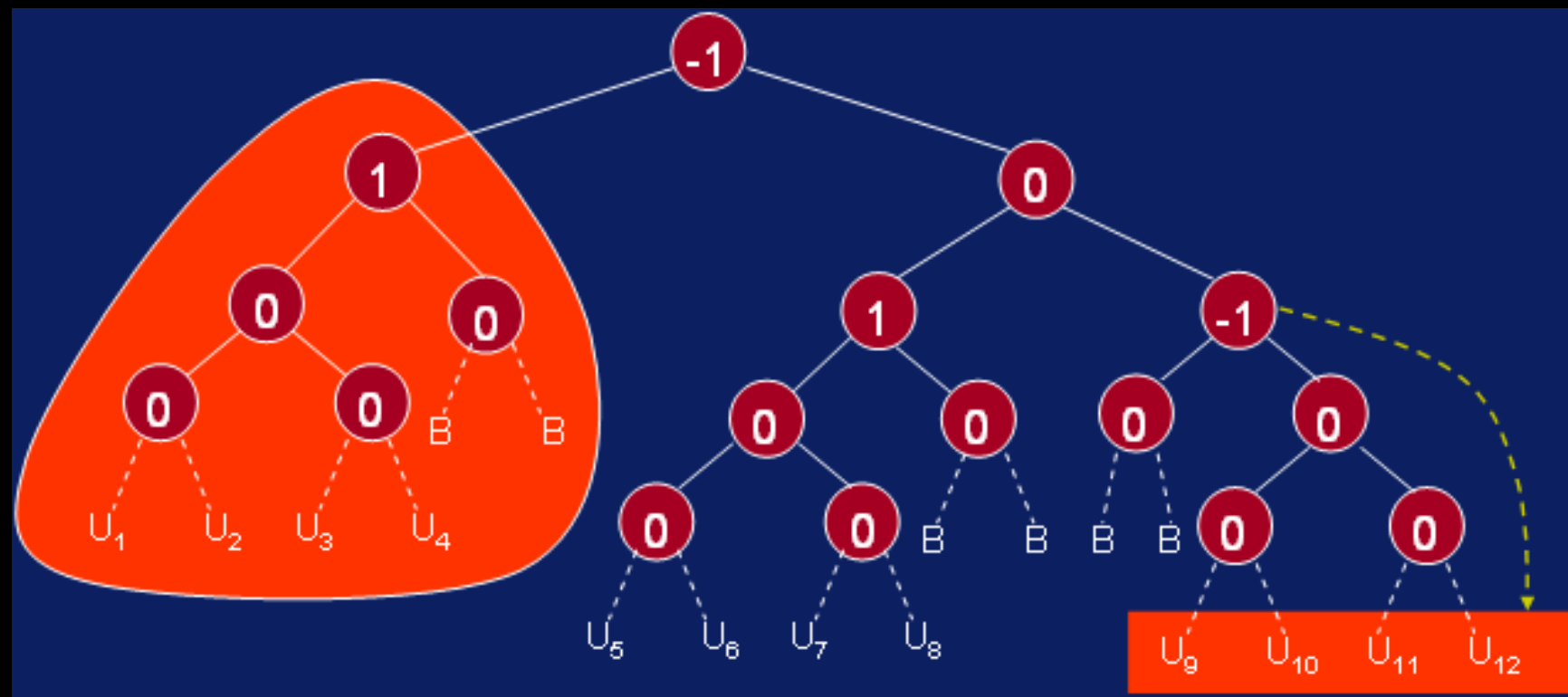
- Insertions and effect on balance



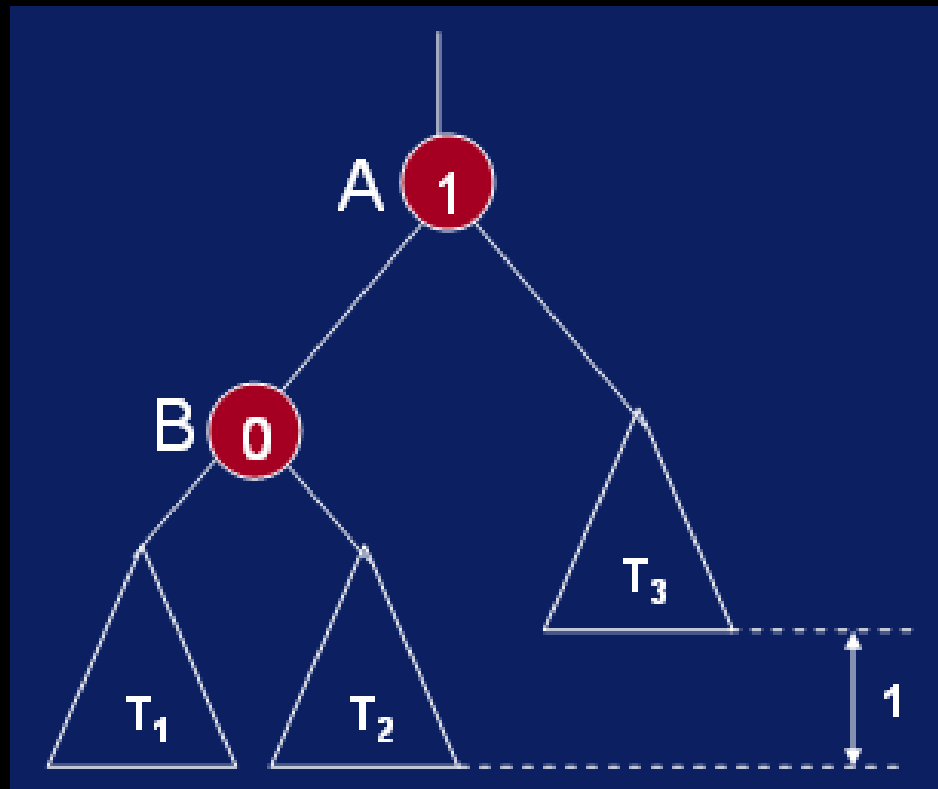
Balanced Binary Tree

- Insertions and effect on balance

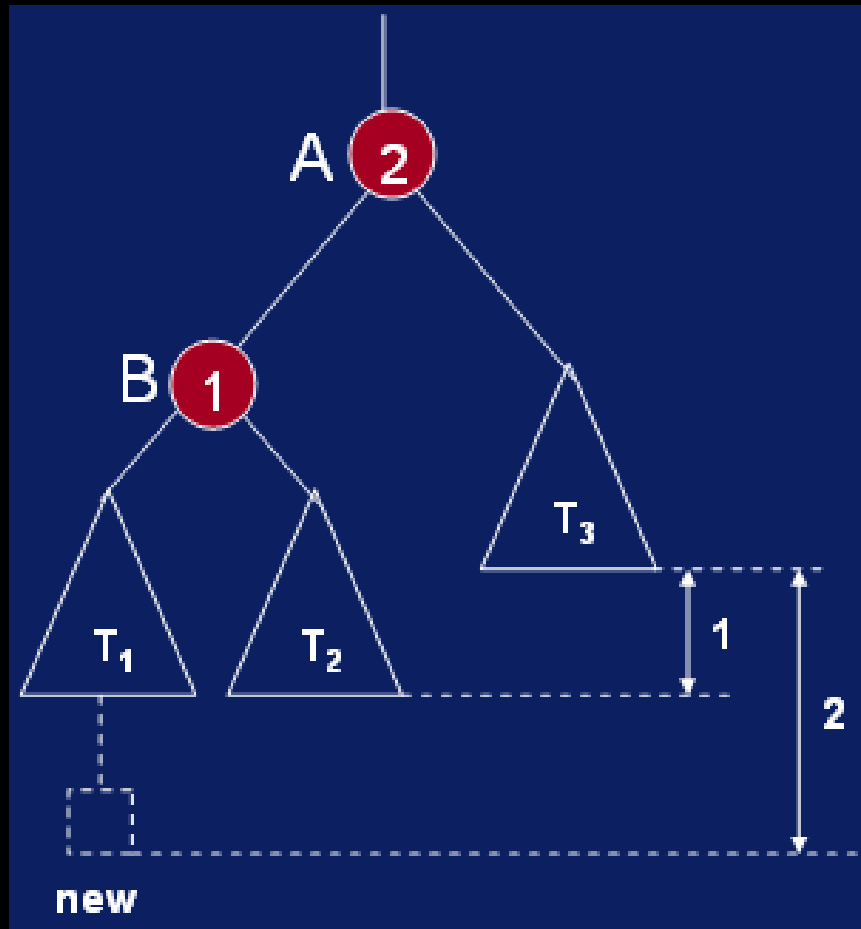




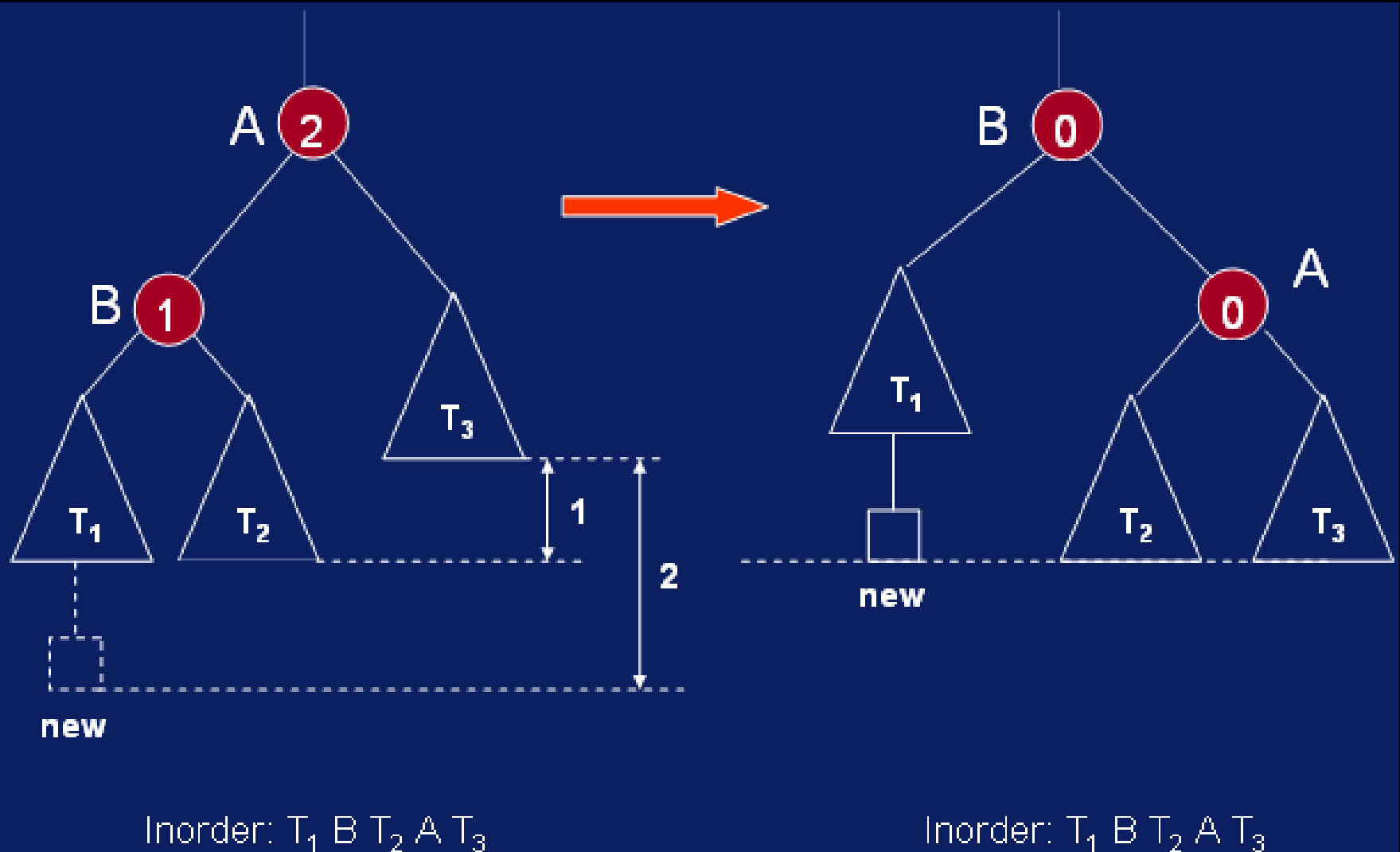
Inserting New Node in AVL Tree



Inserting New Node in AVL Tree



Inserting New Node in AVL Tree



Thanks ...