Data Structures

Lecture 21: AVL Trees

Nopadon Juneam
Department of Computer Science
Kasetsart university

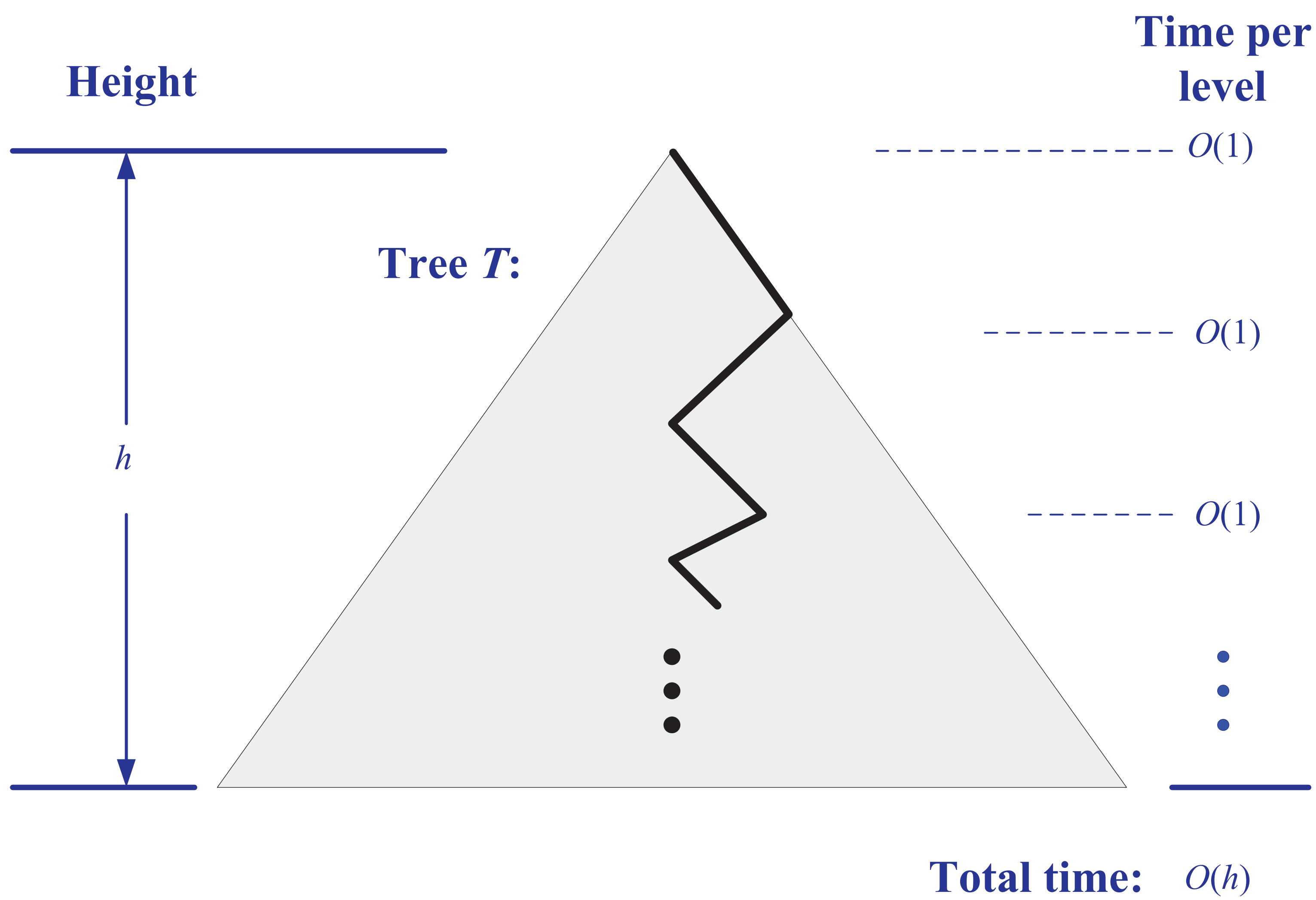
Outlines

- Balanced and unbalanced binary search trees
- Hight-balance property
- AVL trees

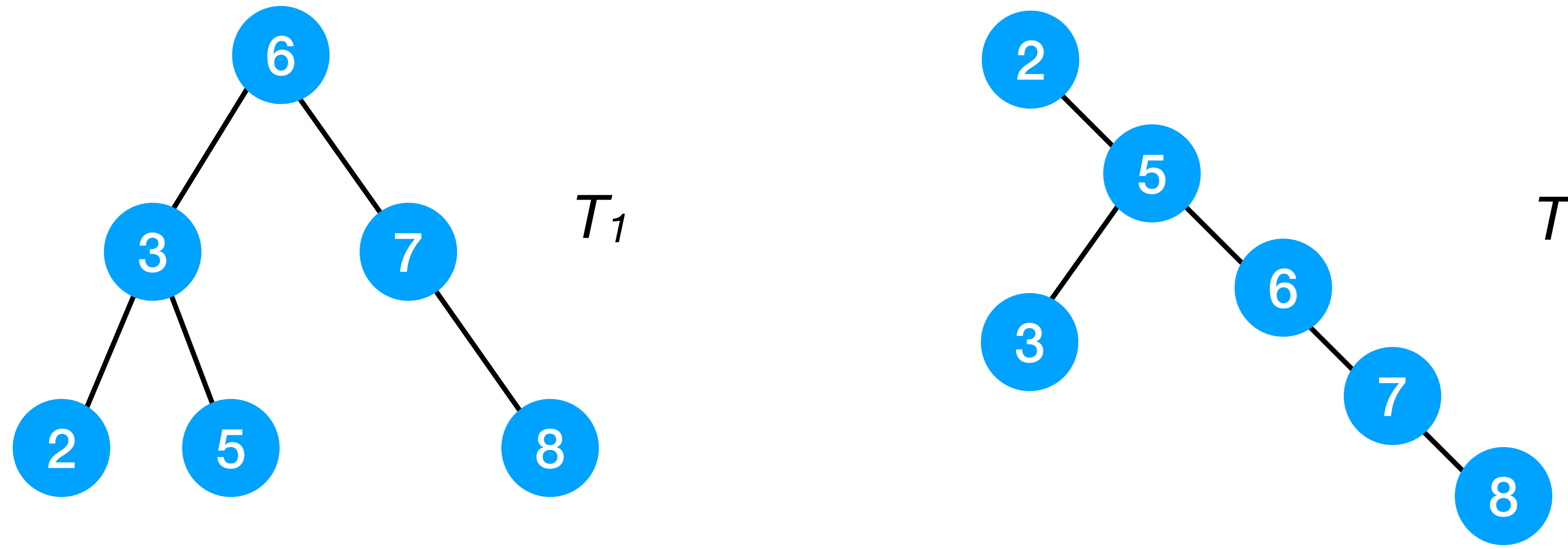
Complexity of Operations on Binary Search Trees (1)

Operations	Complexity
`search`	$O(h)$
`minimum`	$O(h)$
`maximum`	$O(h)$
`successor`	$O(h)$
`predecessor`	$O(h)$
`tree-insert`	$O(h)$
`Tree-delete`	$O(h)$
	Remark: h is the hight of a binary tree. - At worst, h can be n-1. - At best, h can be $\log(n+1)-1$.

Complexity of Operations on Binary Search Trees (2)



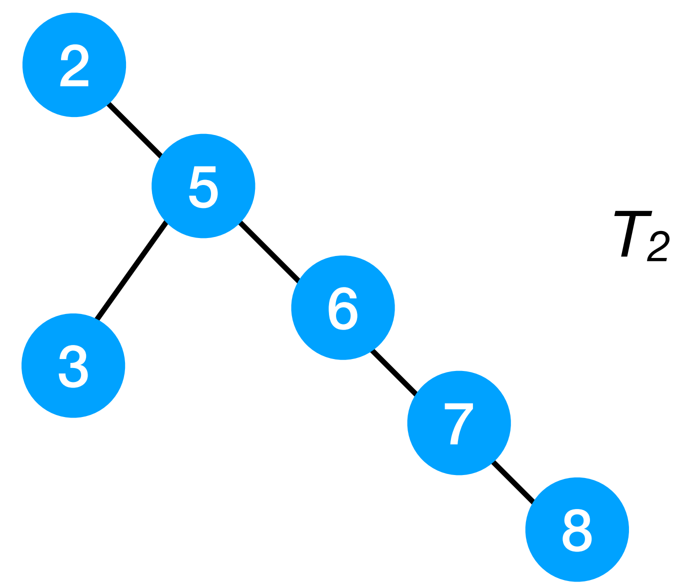
Binary Search Trees of Different Heights



Unbalanced Binary Search Trees

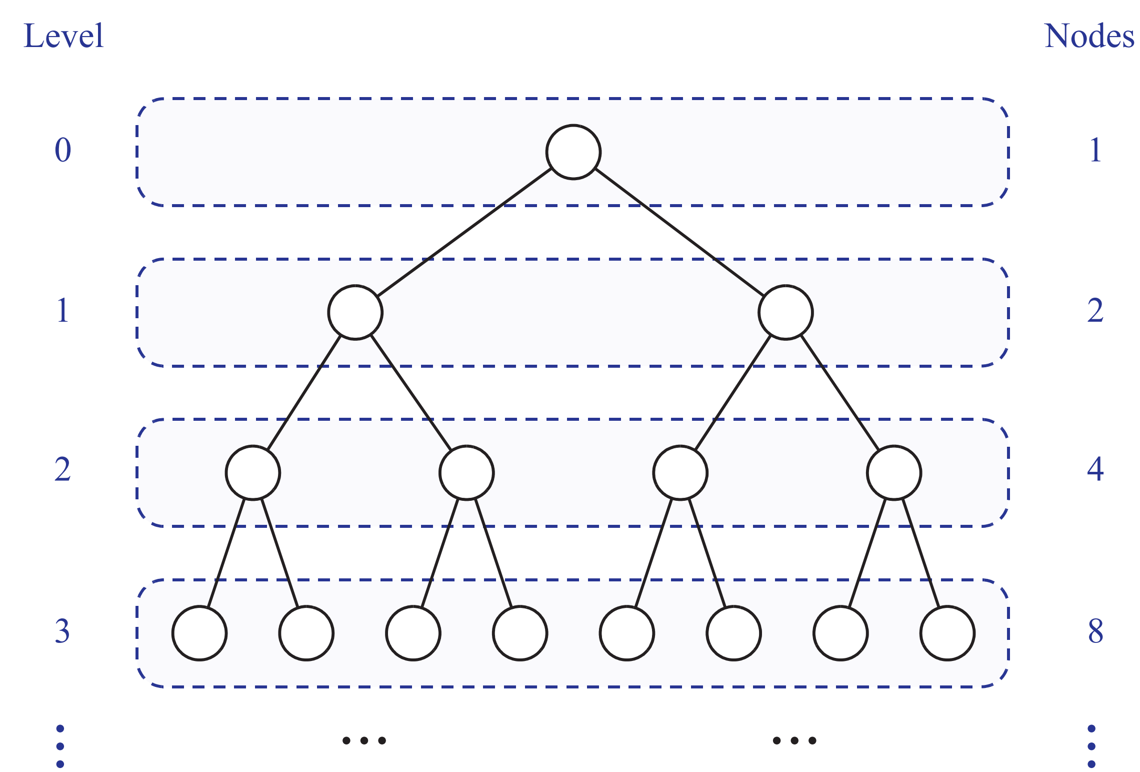
- **Unbalanced binary search tree**: The height of the search tree is approximately linear in the number of nodes

- If a binary search tree is unbalanced, the performance it achieves is no better than the linear data structures



Complete Binary Tree

- Ideally, we would like to have a binary search tree to be **complete**, that is, every level of completely filled
  - The hight of a full binary tree is $\log(n+1)-1$
- However, it is almost impossible to always maintain the search tree as a complete binary tree



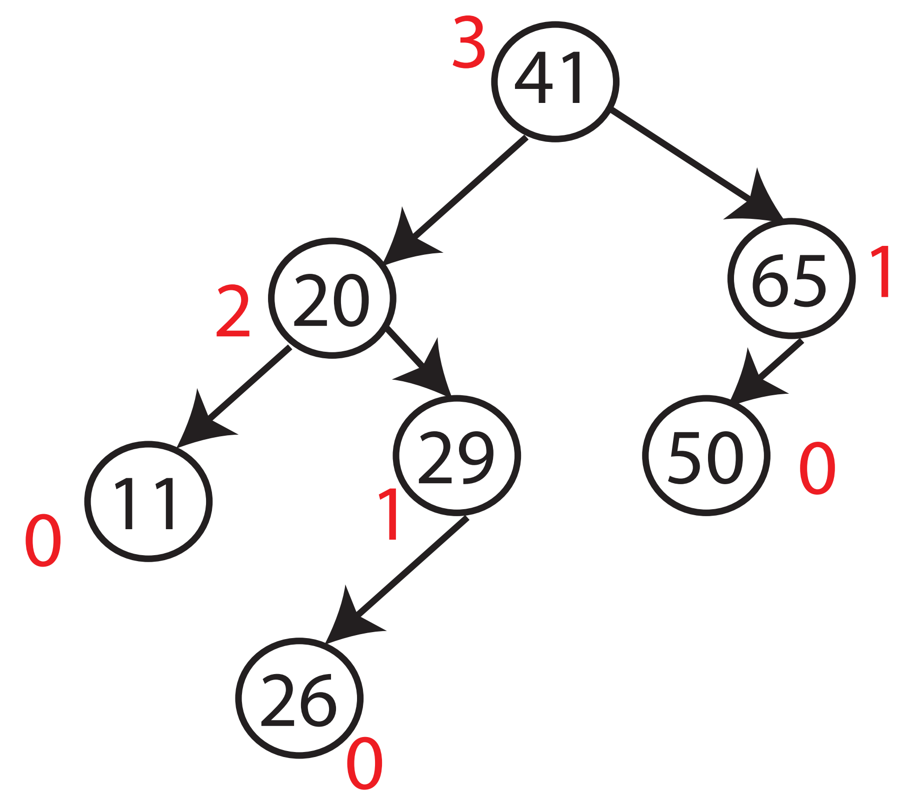
AVL Property

- **AVL property:** For every internal node *v,* the height of the subtrees rooted at children of *v* differs by at most 1

- **Node's height (subtree's height):** The height of a node *u* in a tree is recursively defined by:

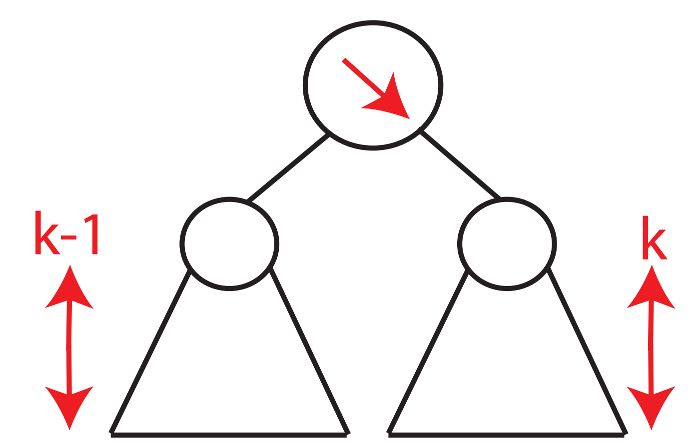
  - If *u* is external, then the height of *u* is zero

  - Otherwise, the height of *u* is one plus the maximum height of a child of *u*



AVL Trees

- Any binary search tree that satisfies the AVL property is said to be an **AVL tree**
  - An immediate consequence of the AVL property is that a subtree of an AVL tree itself is an AVL tree
  - In other words, the difference between the heights of left and right subtrees cannot be more than one for all nodes
- AVL Trees are named after the initials of its inventors (Adelson, Velskii, and Landis 1962)



AVL Tree's Height Property

- **Proposition 1**: Let *T* be a binary tree that exhibits the AVL property. The height of the binary tree *T* is O(log *n*).

Complexity of Operations on AVL Trees

Operations	Complexity
`search`	$O(\log n)$
`minimum`	$O(\log n)$
`maximum`	$O(\log n)$
`successor`	$O(\log n)$
`predecessor`	$O(\log n)$