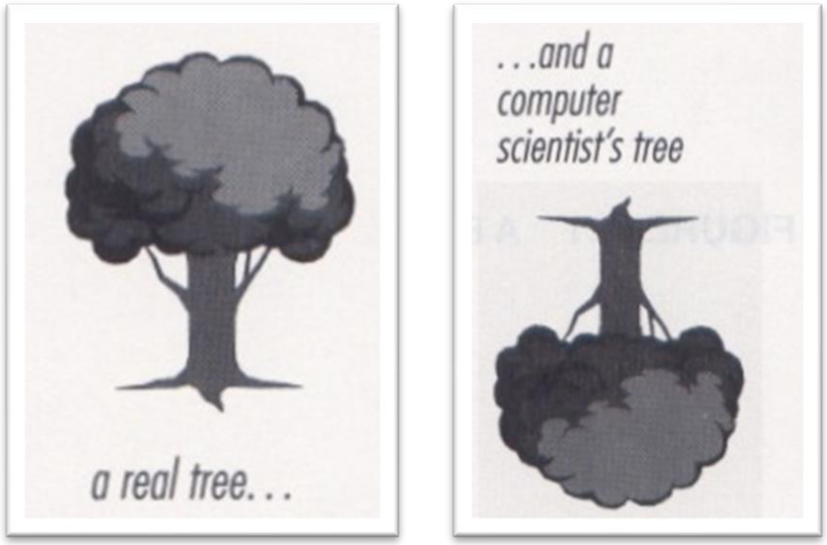
Trees

(in computer science)

Tree in Computer Science

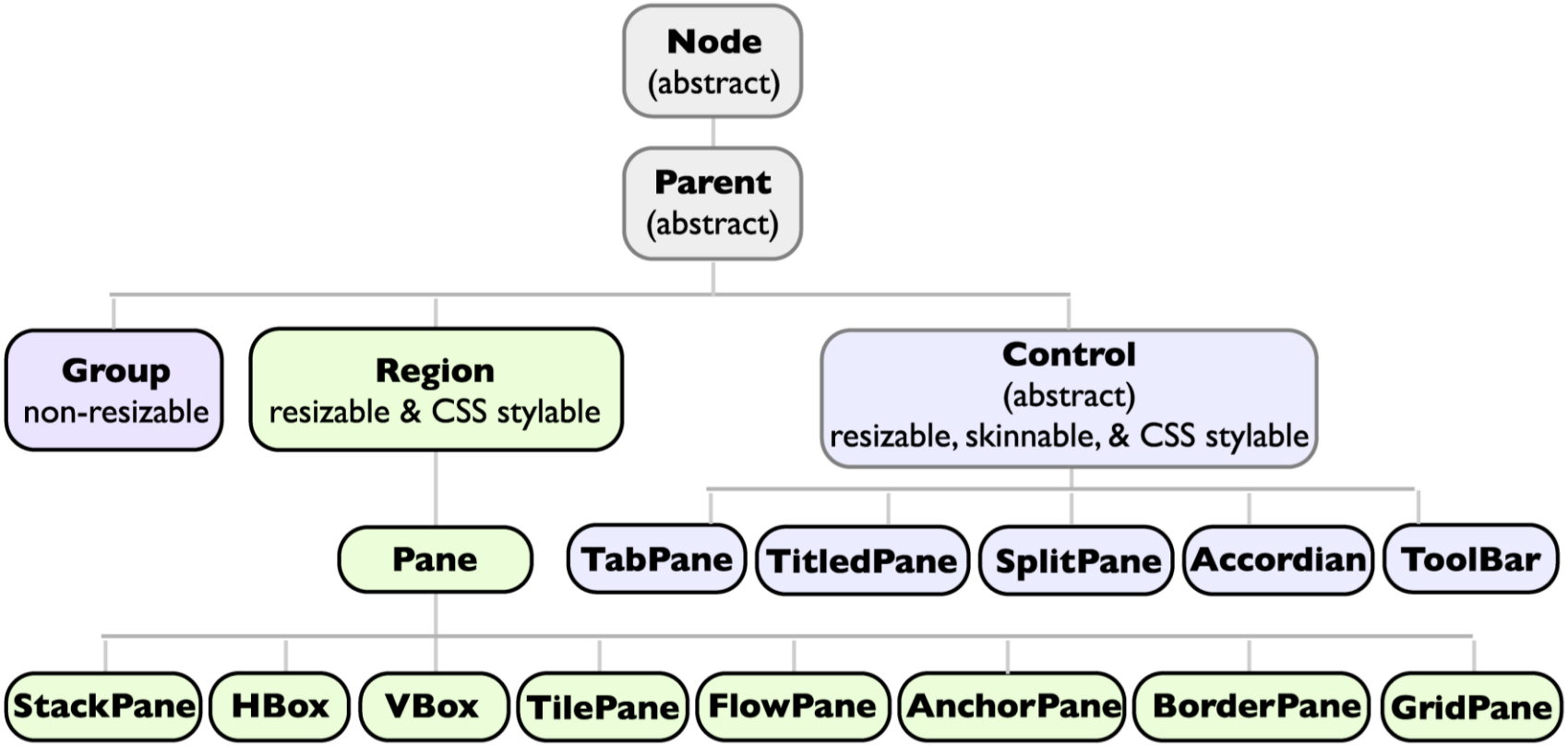
- A tree is a widely used data structure that simulates a hierarchical tree structure with a set of linked nodes



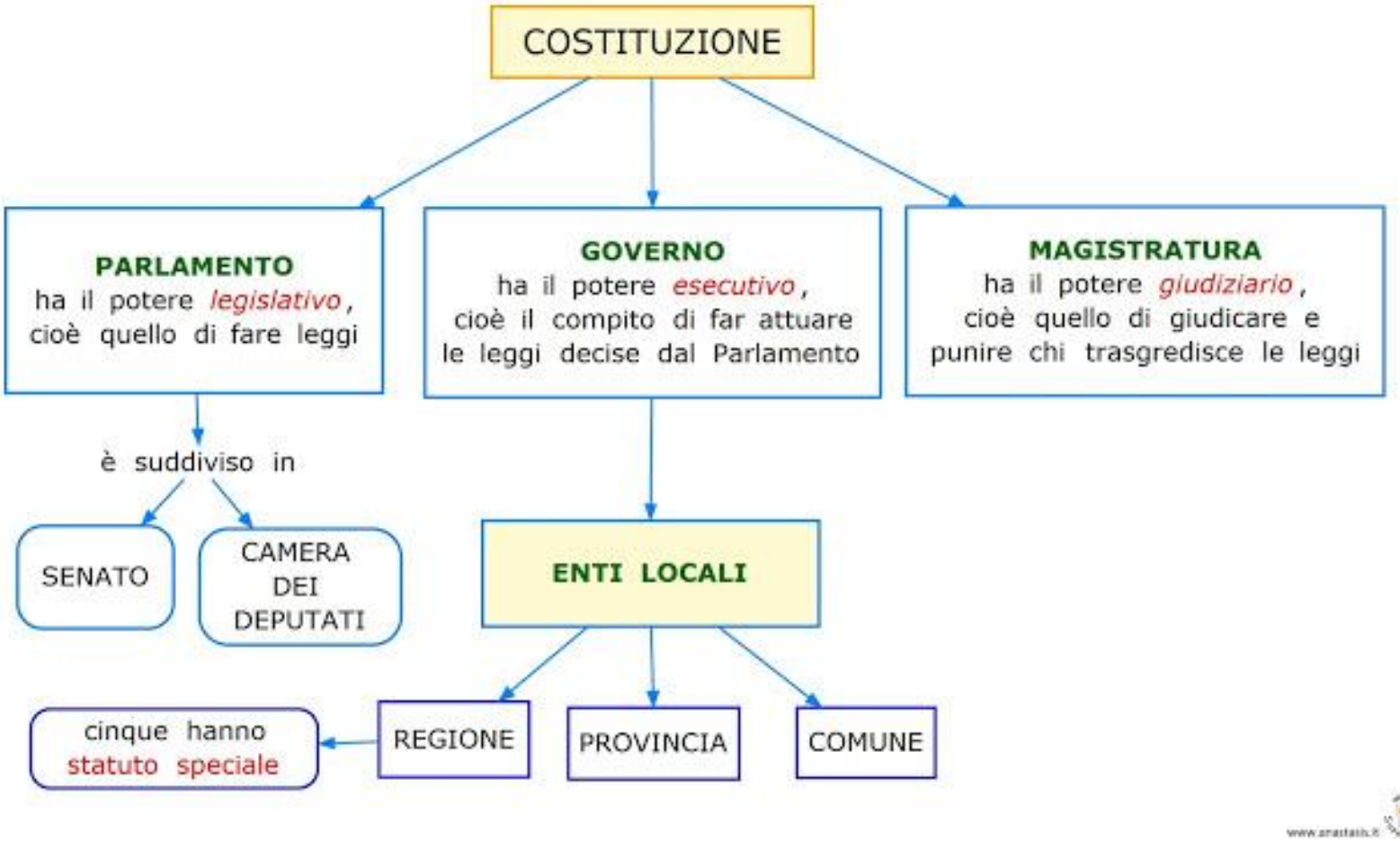
Tree in Computer Science

- **Fundamental** data storage structures used in programming
- Nonlinear structure
- Represents a *hierarchy*
- Items in a tree do not form a simple sequence
- Quite efficient for **retrieving** items (as arrays)
- Quite efficient for **inserting/deleting** items (as lists)

JavaFX 2.0 Layout Classes



Ordinamento dello Stato Italiano

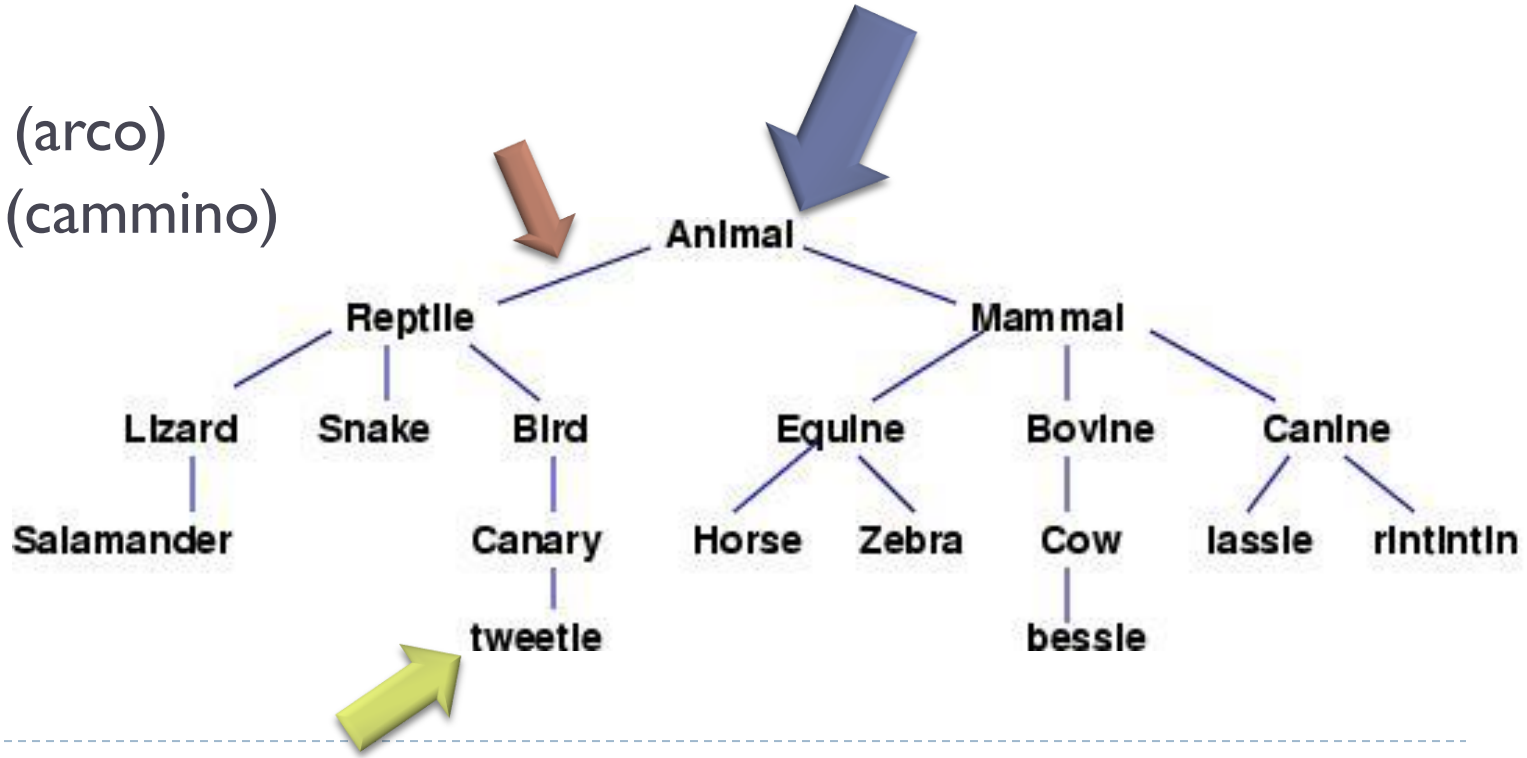


Tree basics

- Consists of nodes connected by edges
- Nodes often represent entities (complex objects)
- Edges between the nodes represent the way the nodes are related
- The only way to get from node to node is to follow a path along the edges

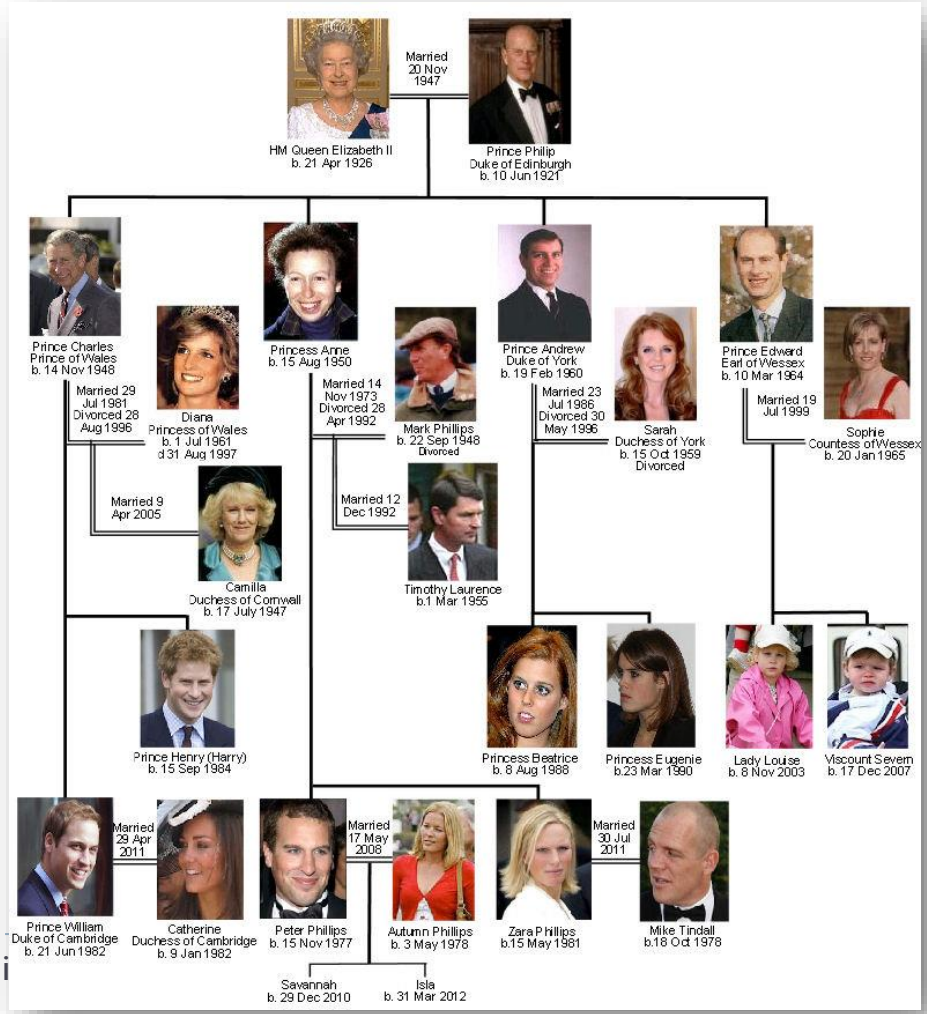
Tree Basics

- Node
  - Root (radice)
  - Leaf (foglia)
  - Interior node/branch (nodo interno)
- Links
  - Edge (arco)
  - Path (cammino)

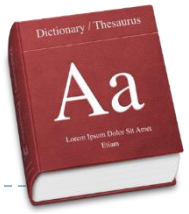


Tree Basics

- Relationship
  - Parent (padre)
  - Child nodes (nodi figli)
  - Sibling (fratelli)
  - Descendant (discendente, successore)
  - Ancestor (antenato, predecessore)

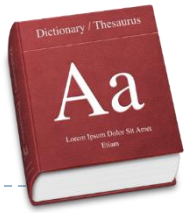


Terminology



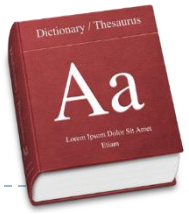
- Visiting
  - A node is visited when program control arrives at the node, usually for processing
- Traversing
  - To traverse a tree means to visit all the nodes in some specified order

Terminology



- Levels
  - The level of a particular node refers to how many generations the node is from the root
  - Root is assumed to be level 0

Terminology



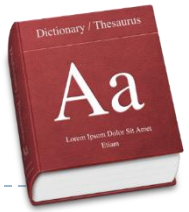
- Height
  - The height of a node is the length of the path to its farthest descendant (i.e. farthest leaf node)
  - The height of a tree is the height of the root
  - A tree with only root node has height 0

Binary Trees

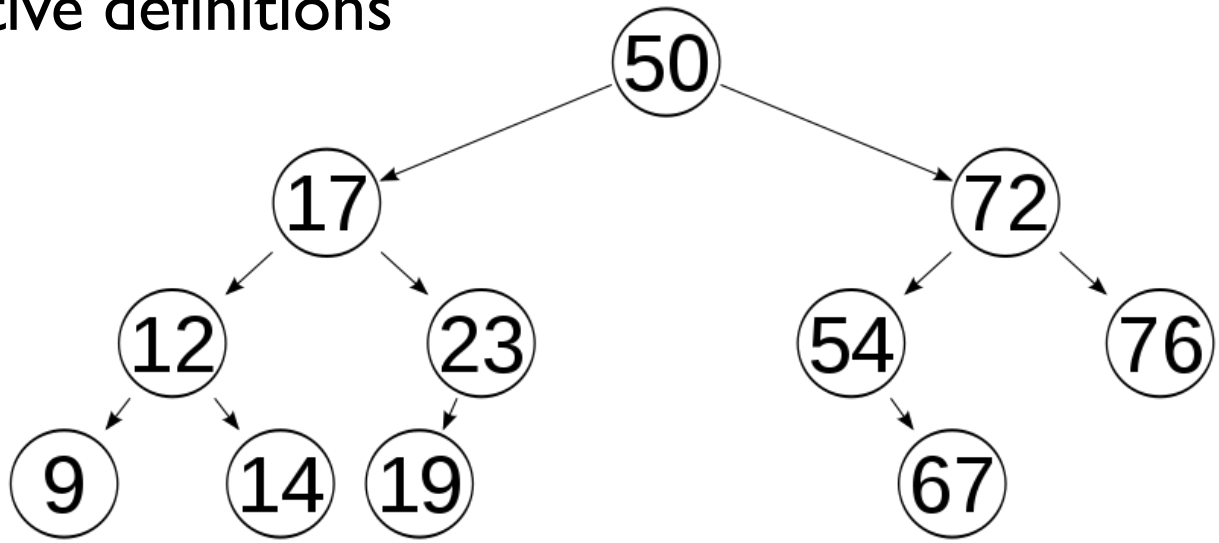
Binary Tree

- A binary tree is a tree where each node has at most two children
- The two children are ordered ("left", "right")
  - Right sub-tree vs. Left sub-tree

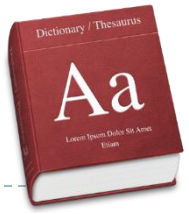
Balanced trees



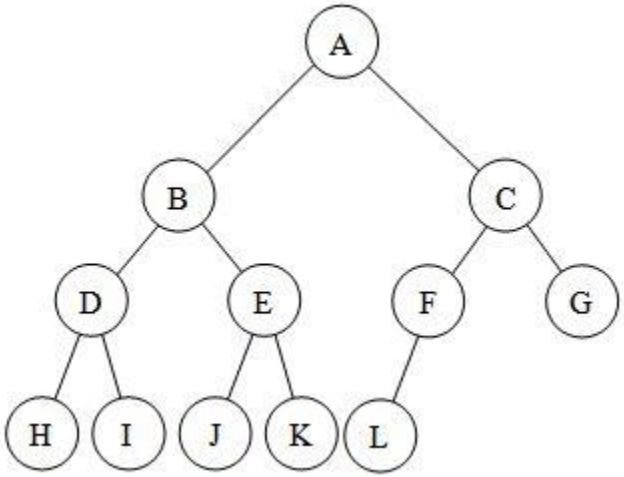
- (Height-)balanced trees
  - The left and right sub-trees' heights differ by at most one
  - The two sub-trees are (height-)balanced
- Perfectly balanced
  - $2^h - 1$ nodes
- Several alternative definitions



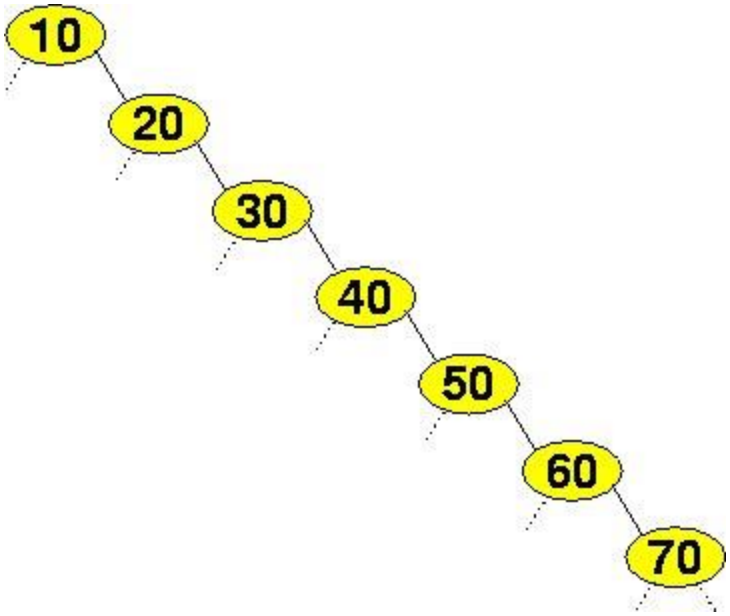
Complete trees



- Complete binary tree
  - Every level, except possibly the last, is completely filled, and all nodes are as far left as possible

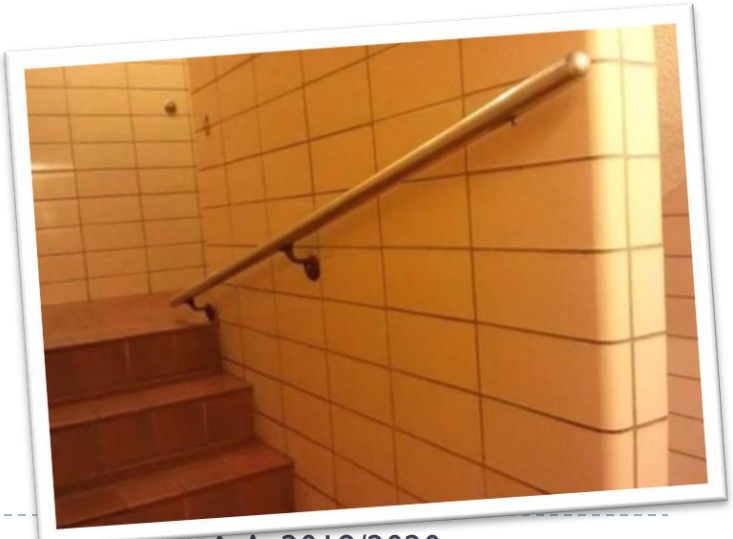


Degenerate trees



Traversal in binary trees

- Pre-order
  - process root node, then its left/right sub-trees
- In-order
  - process left sub-tree, then root node, then right
- Post-order
  - process left/right sub-trees, then root node

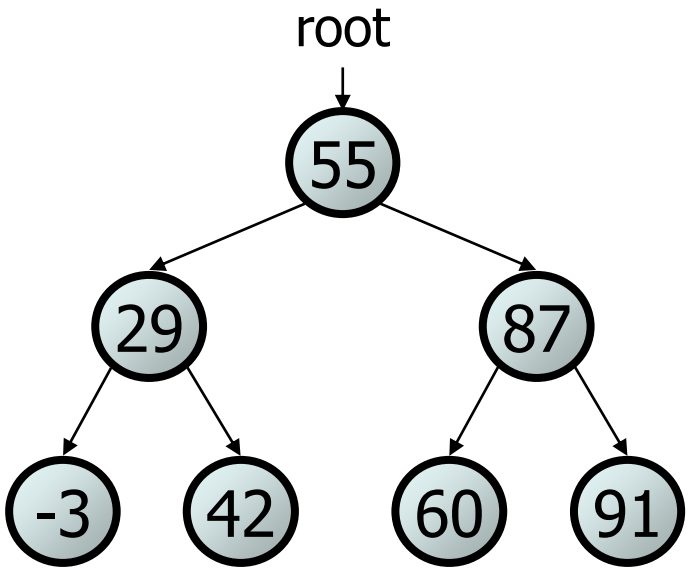


BST

Binary Search Tree

Binary search trees

- A binary tree where each non-empty node R has the following properties:
  - Elements of R's left sub-tree contain data "less than" R's data
  - Elements of R's right sub-tree contain data "greater than" R's
  - R's left and right sub-trees are also binary search trees

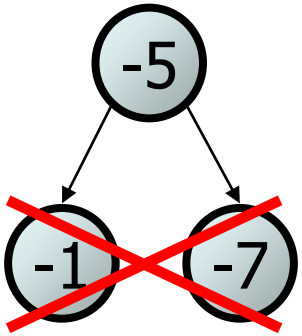


Binary search trees

- BSTs store their elements in sorted order, which is helpful for searching/sorting tasks

Exercise

- Is it a legal binary search tree?

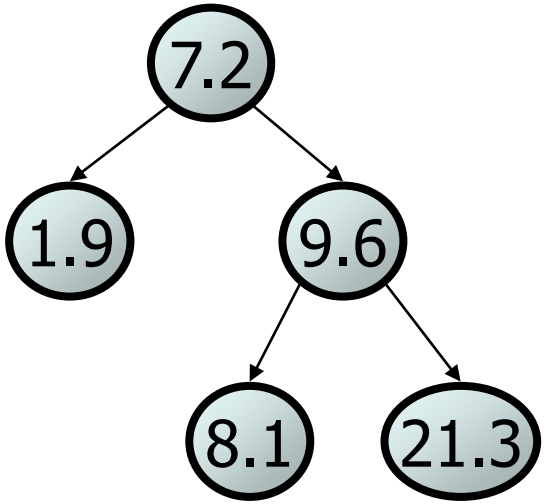


Exercise

- Is it a legal binary search tree?

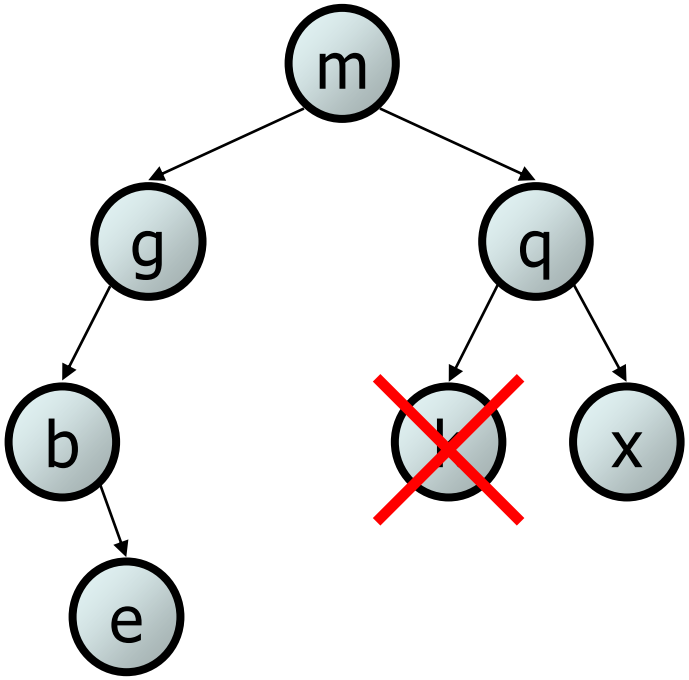
Exercise

- Is it a legal binary search tree?



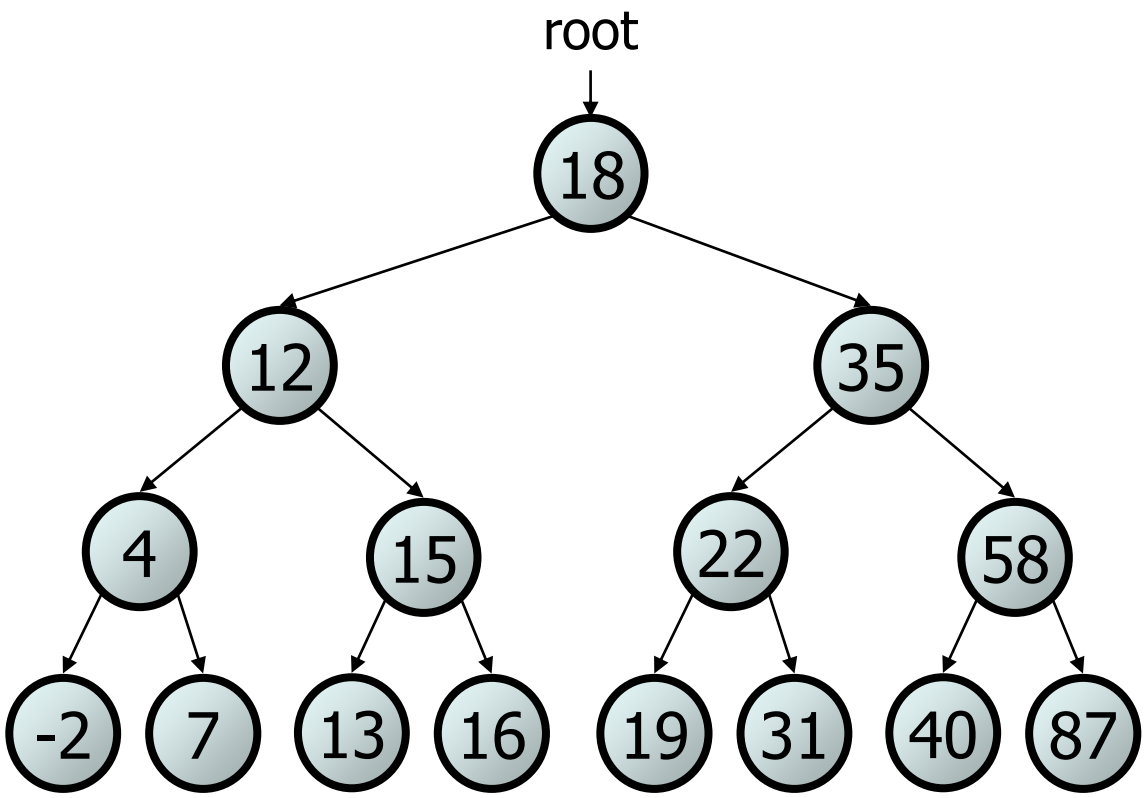
Exercise

- Is it a legal binary search tree?



Searching in a BST

- Describe an algorithm for searching a binary search tree (try searching for 31, then 6)



Searching in a BST

- Searching in a BST is $O(h)$

  If the tree is balanced, then $h \cong \log_2 N$

  $\Rightarrow$ Searching for an element is $O(\ln N)$

Showdown

	Array	List	Hash	BST
add(element)	O(1)	O(1)	O(1)	O(ln n)
remove(object)	O(n) + O(n)	O(n) + O(1)	O(1)	O(ln n)
get(index)	O(1)	O(n)	n.a.	n.a.
set(index, element)	O(1)	O(n) + O(1)	n.a.	n.a.
add(index, element)	O(1) + O(n)	O(n) + O(1)	n.a.	n.a.
remove(index)	O(n)	O(n) + O(1)	n.a.	n.a.
contains(object)	O(n)	O(n)	O(1)	O(ln n)
indexOf(object)	O(n)	O(n)	n.a.	n.a.

Licenza d'uso

- Queste diapositive sono distribuite con licenza Creative Commons "Attribuzione - Non commerciale - Condividi allo stesso modo (CC BY-NC-SA)"
- Sei libero:
  - di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
  - di modificare quest'opera

- Alle seguenti condizioni:
  - **Attribuzione** — Devi attribuire la paternità dell'opera agli autori originali e in modo tale da non suggerire che essi avallino te o il modo i cui tu usi l'opera.

  - **Non commerciale** — Non puoi usare quest'opera per fini commerciali.

  - **Condividi allo stesso modo** — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con un licenza identica o equivalente a questa.

- http://creativecommons.org/licenses/by-nc-sa/3.0/