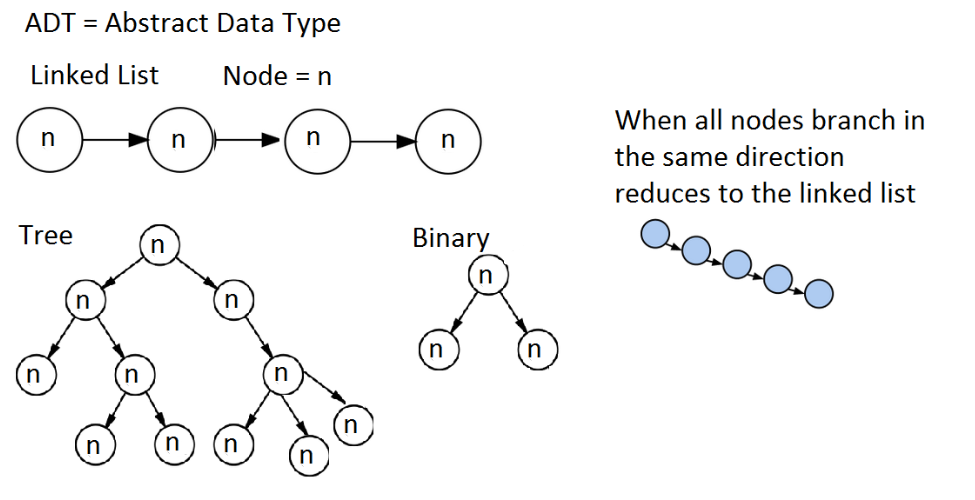
**Binary Trees**

**Document Id 35**

**1 Definitions**



**2 Binary trees**

* Binary trees -> used to construct more abstract data structures
* Nodes -> key, value and two sub-trees, left and right.
* No node can have more than 2 children

**3 Binary search trees**

* Sorted -> Nodes must have a comparison function (Comparator)
* Sorted -> lowest value on left branch
* Sorted -> Searching operations on average skip about half of the tree
* Sorted -> Searching on average O(log n) << O(n) large n
* Searching -> on average faster than linked list.

However can be as slow as on linked list O(n).

* Traversal of the tree is well suited to recursive code

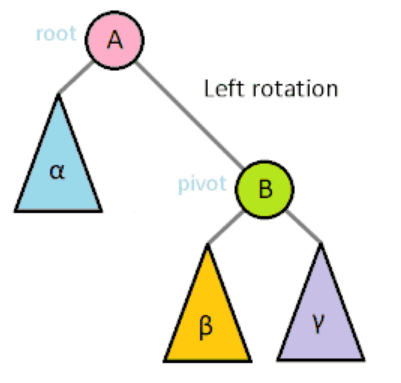
**4 AVL trees**

* the heights of the two child subtrees of any node differ by at most one
* if at any time they differ by more than one, rebalancing is done
* Rebalancing => rotations https://en.wikipedia.org/wiki/Tree\_rotation
* AVL trees are Binary search trees with additional attributes

for the height of the child node in the subtree

for balance

Left rotation start



Left rotation end

