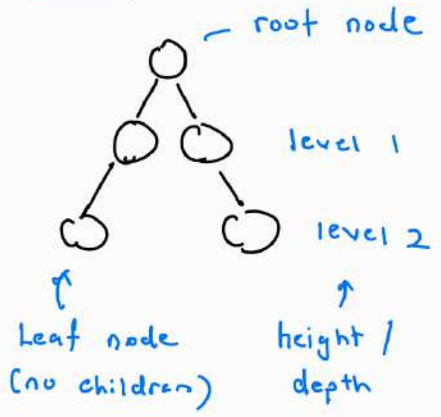
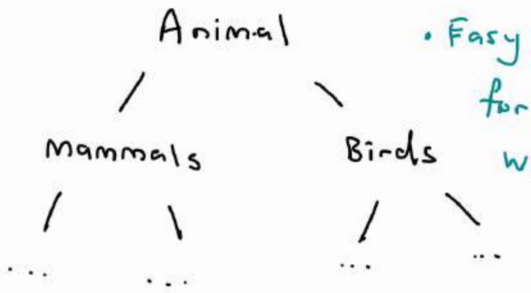
Trees



(hierarchical data structure)

- A way to store data, also using nodes
- Like Linked list but (instead of next & prev.)
  - left
  - parent
  - right

  } pointers.
- Access nodes using tree traversals (decisions)
- Good for organisation structure data

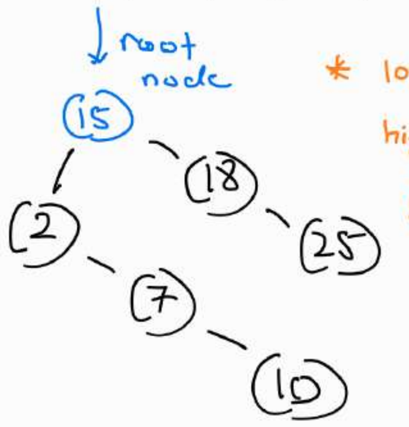


- Easy to just search for mammals / Birds without traversing all nodes

Binary Search Trees (BST)

- Also known as sorted Binary Tree.

Given 15, 2, 7, 10, 18, 25 O(log(n))



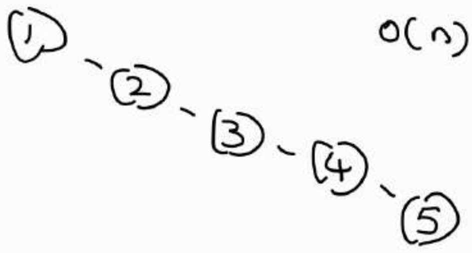
* lower value go left

higher value go right

At most only 2 nodes
- left
- right

- if Numbers already sorted, BST becomes Linked List

eg: 1, 2, 3, 4, 5



O(n) worst case

BST complexity

Search: Avg: O(log(n)) divide & conquer.

Worst: O(n) : Already sorted → Becomes Linked List

Insertion: & Deletion Avg: O(log(n)) : Traverse tree to find suitable location then insert.

Worst: O(n): Inserting into already sorted Tree, akin to traversing Linked List.

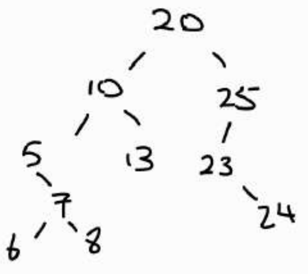
Tree Traversal

In order : Left, Root, Right 5, 6, 7, 8, 10, 13, 20, 23, 24, 25

Pre order : Root, Left, Right 20, 10, 5, 7, 6, 8, 13, 25, 23, 24

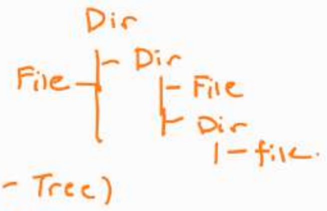
Post order : Left, Right, Root 6, 8, 7, 5, 13, 10, 24, 23, 25, 20

level order : By level (top-down default)



Real World

- File systems.
- DB indexing (B-Tree)
- Decision trees (ML)



Dir
File
Dir
File
Dir
file