CSE 3010 – Data Structures & Algorithms

Lecture #32

What will be covered today

- Running time of sorting and searching methods
- Introduction to hierarchical data structures

Sorting - Comparison of running time

Sorting Technique	Best Case	Average Case	Worst Case	Additional Space
Bubble sort	O(n)	$O(n^2)$	$O(n^2)$	None
Selection sort	O(n)	$O(n^2)$	$O(n^2)$	None
Insertion sort	O(n)	$O(n^2)$	$O(n^2)$	None
Quick sort	O(nlogn)	O(nlogn)	$O(n^2)$	No
Merge sort	O(n)	O(nlogn)	O(nlogn)	Yes

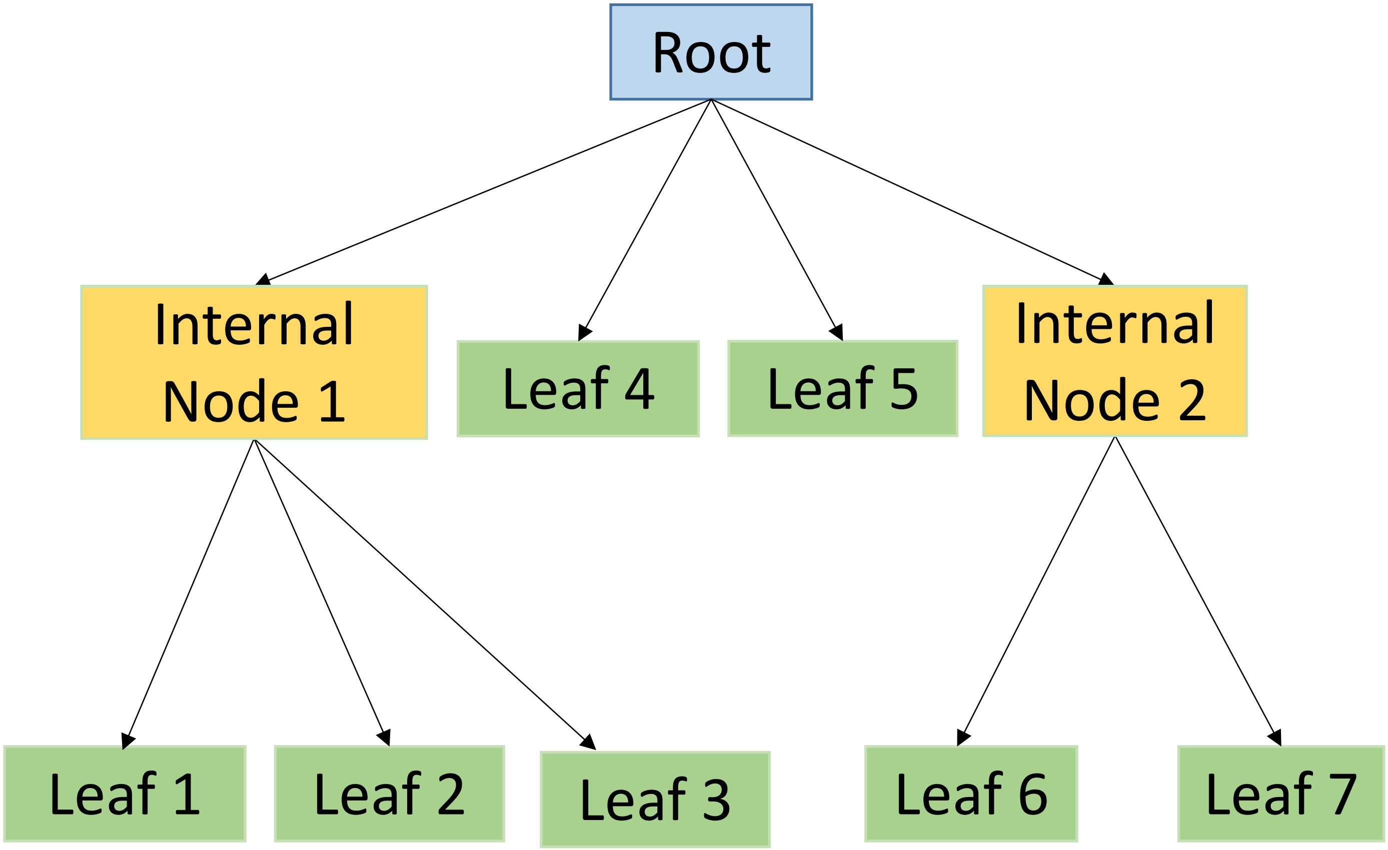
Searching – Comparison of running time

Sorting Technique	Best Case	Average Case	Worst Case
Linear search	O(1)	O(n)	O(n)
Binary search	O(1)	O(logn)	O(logn)
Jump search	O(1)	O(√n)	O(√n)
Hashing - Open addressing	O(1)	O(n)	O(n)
Hashing - Chaining	O(1)	O(n)	O(n)

Tree abstract data type

- Non-linear data structure
- Represents a hierarchy
- Collection of nodes
  - Collection may be empty
    - Empty tree
  - Collection may have one node
    - Root of the tree
  - Collection may have more than one node
    - Internal nodes and leaf node

Understanding nodes



An empty tree has ZERO nodes

Terminologies used in tree data structure

Tree Terminology	Meaning
Root	Node with no parent
Internal node	Node that is not a root or a leaf
Leaf	Last node in one branch of tree with no children
Parent	Father/mother of a node
Child	Daughter/son of a node
Sibling	Nodes with the same parent
Ancestors	Node itself, parent, parent of parent and so on

Terminologies used in tree data structure

Tree Terminology	Meaning
Descendants	Node itself, child, child of child and so on
Subtree	Tree consisting of child (if any) and its descendants
Edge	Connection between one node and another
Path	Set of edges from root to a node
Path length	Number of edges in the path
Height of tree	Number of nodes on the longest path from root to a leaf

Types of trees

- General tree
  - Any number of sub trees for every node
  - Different number of sub trees possible for each node
- N-ary tree
  - Every node has at most N sub trees
  - Special case is binary tree – when N = 2

Some sample operations on a tree

Operation	Description
`root()`	Get the root node of the tree
`size()`	Get the number of nodes in the tree including the root
`isEmpty()`	Find if the tree is empty or not
`parent(node)`	Get the parent of a given node

Some sample operations on a tree … contd.

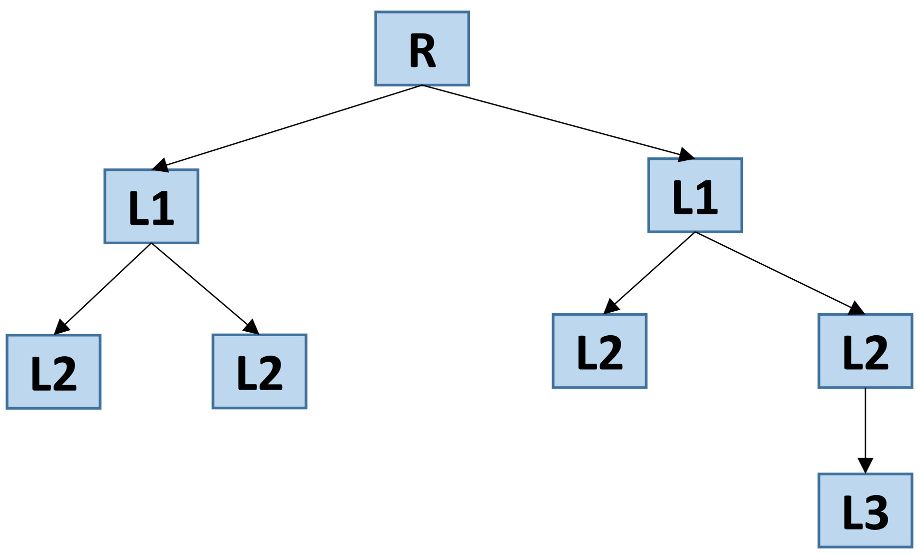
Operation	Description
`children(node)`	Get the children of a given node NULL if no children exists
`isInternal(node)`	Find if the given node is an internal node
`isLeaf(node)`	Find if the given node is a leaf node
`isRoot()`	Find if the given node is the root of the tree

Some sample operations on a tree … contd.

Operation	Description
`traverse(root)`	Visit every node of the tree [Many techniques available to traverse a tree]
`addSubTree(node)`	Add a subtree to a given node in the tree
`removeSubTree(node)`	Remove a subtree from a given node in the tree
`height(root)`	Get the height of the tree

Binary tree

- Special kind of a n-ary tree
- Has at most two children
  - Left child
  - Right child



Properties of a binary tree

- Level 0 has <= 1 node
- Level 1 has <= 2 nodes
- Level 2 has <= 4 nodes

...

- Level i has <= 2^i nodes

Applications of binary trees

1. Expression tree
2. Binary search tree
3. Binary Space Partition
4. Binary Tries
5. Hash Trees
6. Heaps
7. Huffman coding tree
8. GGM Trees
9. Syntax Tree
10. Treap
11. T-tree