Trees

N. Lutz
MIT Media Lab
Object Based Media

Algorithms Are For Everyone

Who am I?

Grad student here at OBM.

Have TAed and taken various courses in Computer Science, specifically in computational for not computer science and algorithms

MIT gave me a B.Sc in Computer Science.

Recap

An **algorithm** is a process of instructions for a computer and can be written in **pseudocode.**

They run in time and space. We denote this with big O notation.

O(1)
O(lgn)
O(n)
O(nlgn)
O(n^2)
O(2^n)
O(n!)

Recap 2

A sorting algorithm puts elements of a data structure in an order based on a comparator.

Lower bound of nlgn based on comparison operators.

Trees

A **tree** is an **abstract data type (ADT)**.

They can be implemented in many ways and are studied in many branches of computer science and math.

Including **set, type,** and **graph theory.**

Like any ADT they must have a representation that is consistent and operations.

Use Cases of Trees

- Hierarchal data storage and structures
- Learning and syntax structures and grammars
- Searching
- Sorting
- Simulation

Terminology Groups

“family” - Parent, Child, Sibling, Descendant and Ancestor

“nature” - Root, Leaf, Branch

“graph” - Edge, Node, Path

Terms

Leaf - no children, end of branch
Degree - number of children of a node
Edge - connection between two nodes
Level - 1 + number of edges from root
Depth - number of edges from root
Height - longest path from nodes
Forest - a collection of trees

Types of Trees

Binary Search Trees
AVL
Red Black
van Emde Boas
Splay Trees
Minimum Spanning Trees (MST)

and….many many more!

Tree Operations

Enumeration

Search

Adding a new item to a certain position

Deleting an item

Removing or adding a subsection

Finding root from any node

Finding lowest common ancestor

N-ary Tree

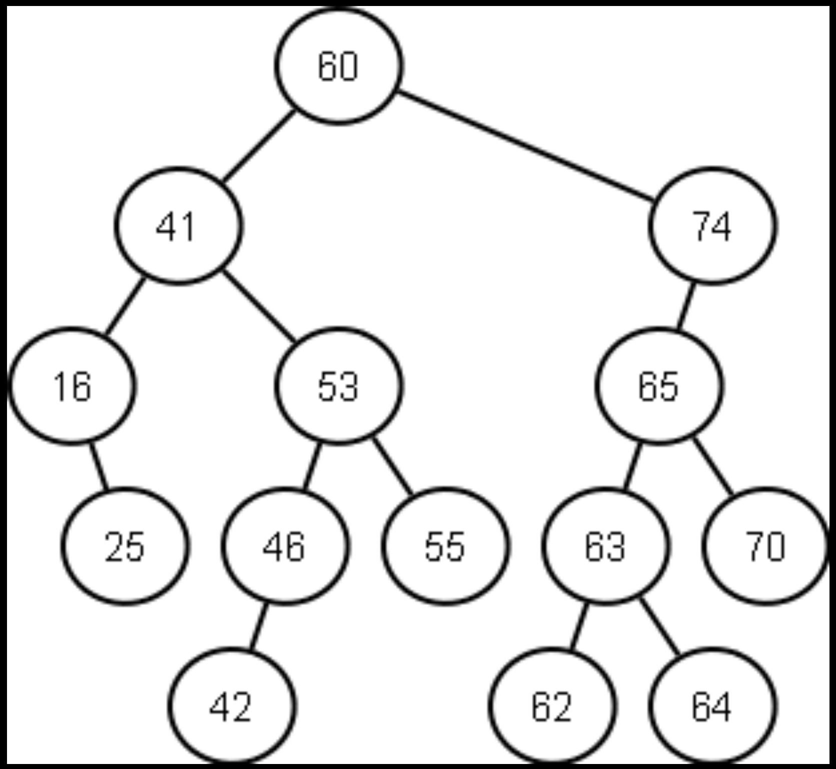
A rooted tree such that each node has no more than N children.

If it is complete — such that each is completely filled — it is maximally space efficient.

Often used for validation schemes. Can be converted to binary tree.

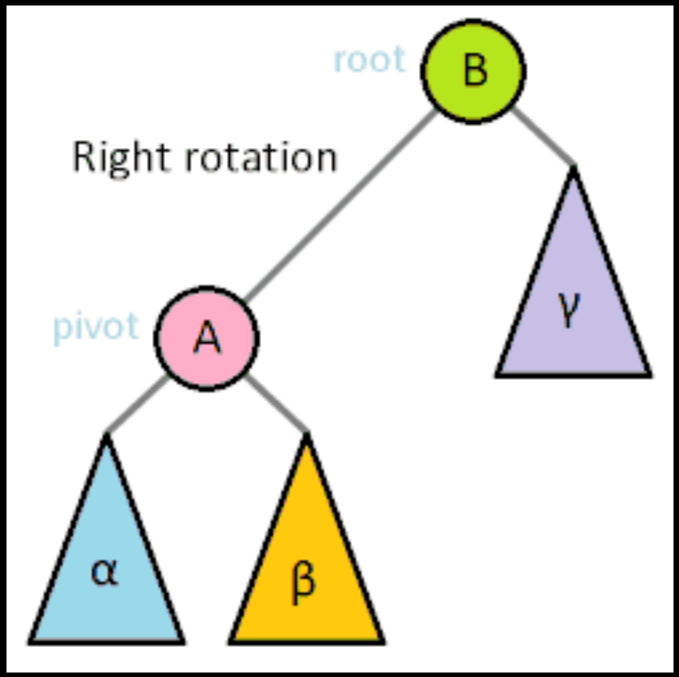
Binary Search Tree

- A **BST** is a data structure where each node has a parent and no more than 2 children.
- The left node MUST be smaller than the right and smaller than the parent node.
- A **balanced BST** happens if all nodes have 2 children
- **Depth** is important to runtime.



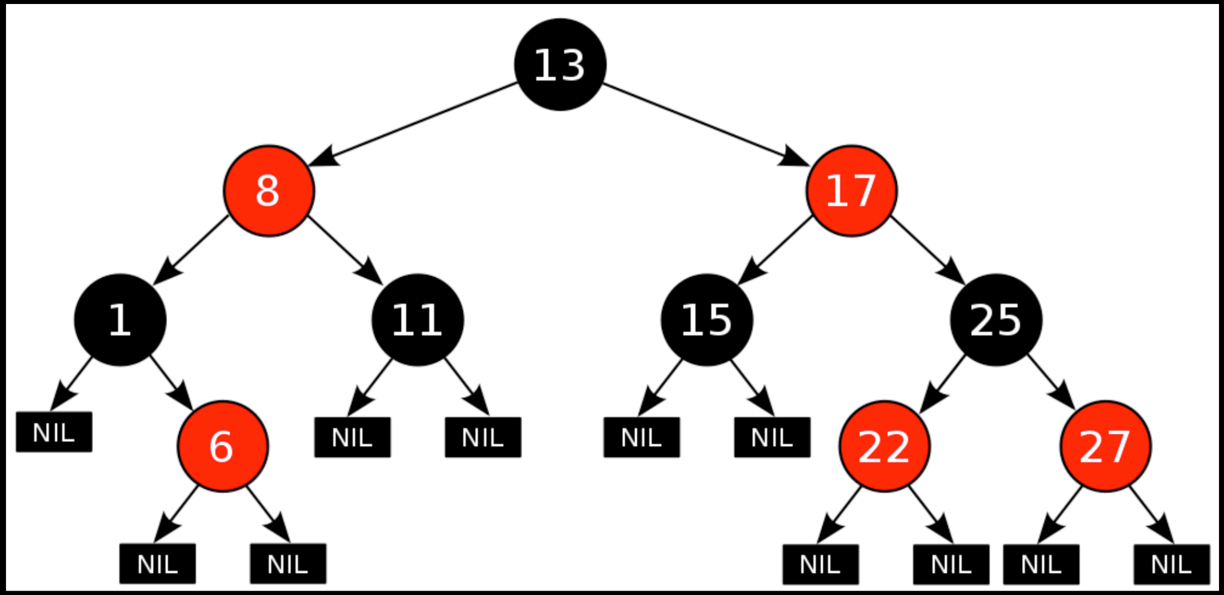
AVL Trees

- **Self balancing** BST
- Heights of two child subtrees of any node **differ by at most one.**
- Lookup, insertion, and deletion take O(logn).
- Use **tree rotations** to rebalance the tree.



Red Black Trees

- Each node has an extra bit — coloring it red or black.
- Adding extra bit is a good design strategy.
- VERY similar to AVL trees — but use colors to balance.



1. All leaves are black.
2. If a node is red, then both its children are black.
3. Every path from a given node to any of its descendants contain the same number of black nodes.

Splay Tree

Self balancing binary tree with added operation

Continuously brings accessed nodes closer to root — thus keeping lookup time to O(logn) — which is super fast.

Operation called splay brings element that is accessed closer to the root.

KD Trees

A multidimensional tree that is utilized for space partitioning and packing problems.

Great for multidimensional search.

Every leaf is a k dimensional point.

Also great for point sets.

Minimum Spanning Trees

A subset of edges that connect all vertices of a graph — MST is the minimum collection of them.

Known for MST problem — a **greedy** algorithm.

MSTs are used in **Prim's** and **Krustal's algorithms.**

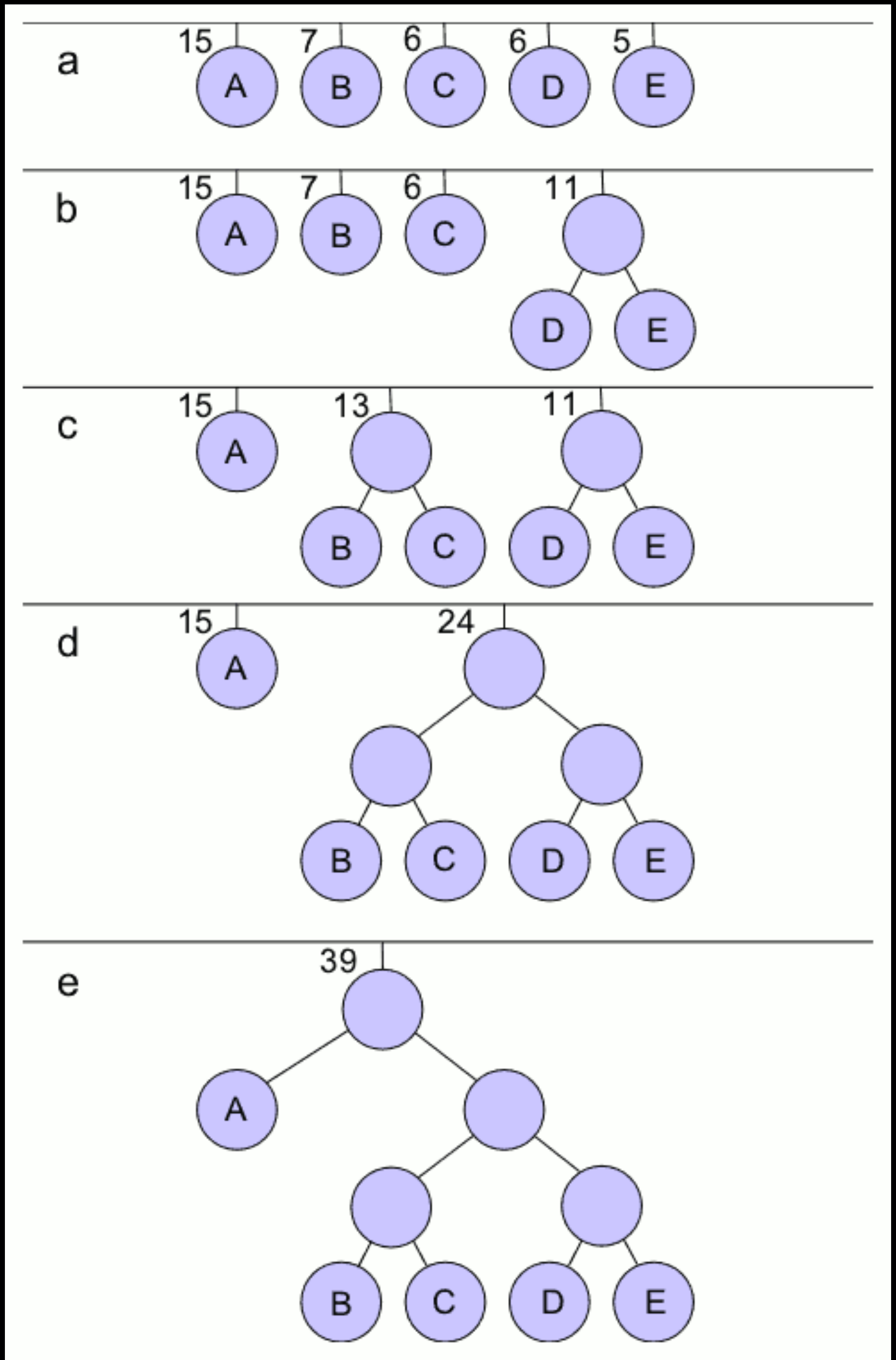
Cycle and **cut** property.

MSTs are important

Taxonomy
Cluster analysis
Network layout
Circuit design
Decision trees and learning
… more

Huffman Tree

- Used for Huffman Coding problems — precedes it by one year
- Used for data compression
- It’s a frequency sorted binary tree for encodings
- Not overly used today but fun to think about



Heap Structure

We'll talk about these next week :) but they are a type of tree!!!

Good resources

- 6.006 OCW
- Geeks for Geeks is great
- Cracking the Coding Interview is good
- Next week I talk about heaps and heap sorting
- https://github.com/ninalutz/AlgorithmsWorkshops