Unit-4
Tree and Graph

Course: MCA

Subject: Data and File Structure

NONLINEAR DATA STRUCTURE

- 2 primary types:
- Trees
- Graphs
- **All trees are graphs, but not all graphs are trees**
- Recursion is useful and is the easiest way to process them.
- It helps keep track of what's been processed and what remains

Graphs

- Graphs can have multiple references in and multiple references out (whereas tree node only has one reference in)

- Graphs can be directed or undirected and cyclic or acyclic

Trees

- Single parent
- 0 or more children
- A node with no children is called a "leaf"
- The topmost node is called the "root"
- N-ary trees
- Binary trees

N-ary Trees

- The best example of an n-ary tree is your computer's directory system.

- It has a single starting point and then 0 or more branches.

Binary Trees and Binary Search Trees

- Binary search trees allow for fast insertion and removal of elements
- They are specially designed for fast searching
- A binary tree consists of two nodes, each of which has two child nodes
- All nodes in a binary search tree fulfill the property that:
  - Descendants to the left have smaller data values than the node data value
  - Descendants to the right have larger data values than the node data value

BST Tree Nodes

- All nodes in a binary search tree fulfill the property that:
  - Descendants to the left have smaller data values than the node data value
  - Descendants to the right have larger data values than the node data value

BST (Balanced Binary Trees)

- Balanced tree: each node has approximately as many descendants on the left as on the right
- If a binary search tree is balanced, then adding an element takes $O(\log(n))$ time
- If the tree is unbalanced, insertion can be slow
- Perhaps as slow as insertion into a linked list

Traversing a Tree

- Which were used for the binary tree can now be used for the general.
- When the general tree has been represented as a binary tree, the algorithms l tree.
- In-order traversals make no sense when a general tree is converted to a binary tree.
- In the general tree each node can have more than two children so trying to insert the parent node in between the children is rather difficult, especially if there is an odd number of children.
- Pre - order
- This is a process where the root is accessed and processed and then each of the subtrees is preorder processed. It is also called a depth-first traversal.

Traversing a Tree

- Pre-order Traversal

- In this way the resulting printout has all nodes at any given level starting in the same tab column.

- It is relatively easy to draw lines to produce the original general tree except that the tree is on its side with it's root at the left rather than with the root at the top.

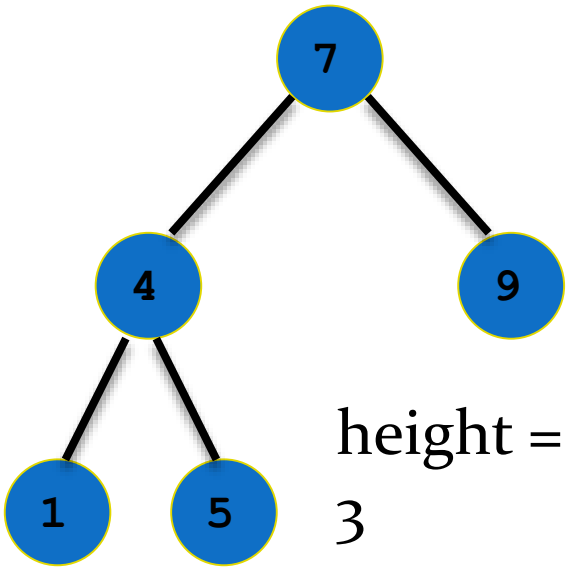
Tree Traversal

- Tree traversal schemes include
  - Preorder traversal (root, left, right)
  - Inorder traversal (left, root, right)
  - Postorder traversal (left, right, root)

  - Preorder generates prefix expression,(polish notation) from an expression trees
  - Inorder generates a sorted ordering
  - Postorder generates a post fix expression, also useful for node deletion

Height of a BST

- Insert 7
- Insert 4
- Insert 1
- Insert 9
- Insert 5
- It's a complete tree!



height = $\lfloor \log(5) \rfloor + 1 =$ 3

BSTs with heights O(log n)

- It would be ideal if a BST was always close to a full binary tree
- It's enough to guarantee that the height of tree is $O(\log n)$
- To guarantee that we have to make the structure of the tree and insertion and deletion algorithms more complex
- e.g. AVL trees (balanced), 2-3 trees, 2-3-4 trees (full but not binary), **red–black** trees (if red vertices are ignored then it's like a full tree)

What is a graph?

- A data structure that consists of a set of nodes (*vertices*) and a set of edges that relate the nodes to each other
- The set of edges describes relationships among the vertices

Formal definition of graphs

- A graph *G* is defined as follows:

G=(V,E)

***V(G):* a finite, nonempty set of vertices**

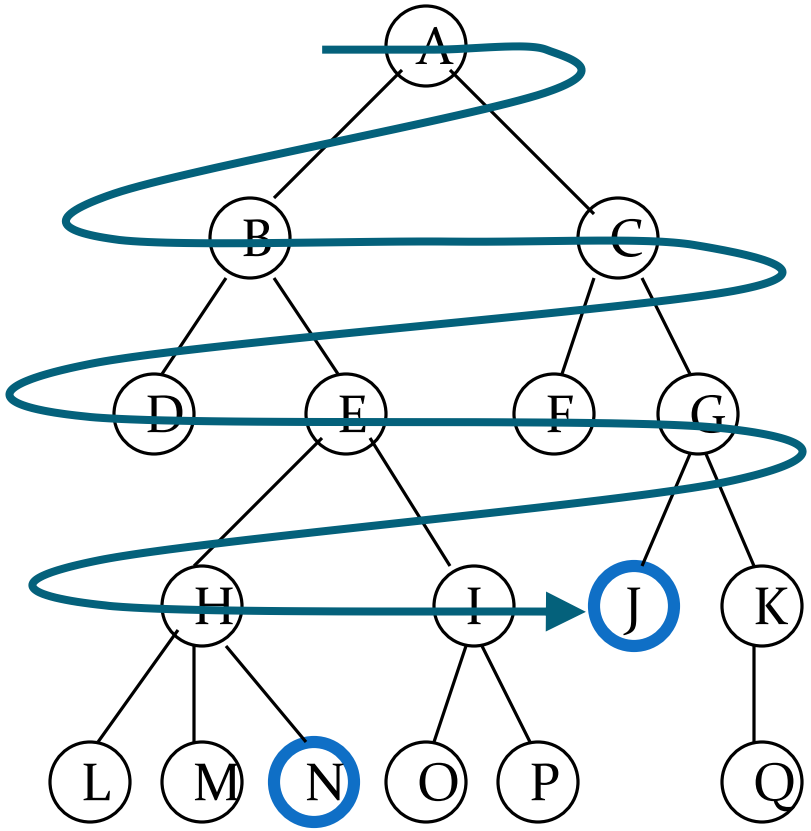
***E(G):* a set of edges (pairs of vertices)**

Directed vs. undirected graphs (cont.)

- When the edges in a graph have a direction, the graph is called *directed* (or *digraph*)
- When the edges in a graph have no direction, the graph is called *undirected*.

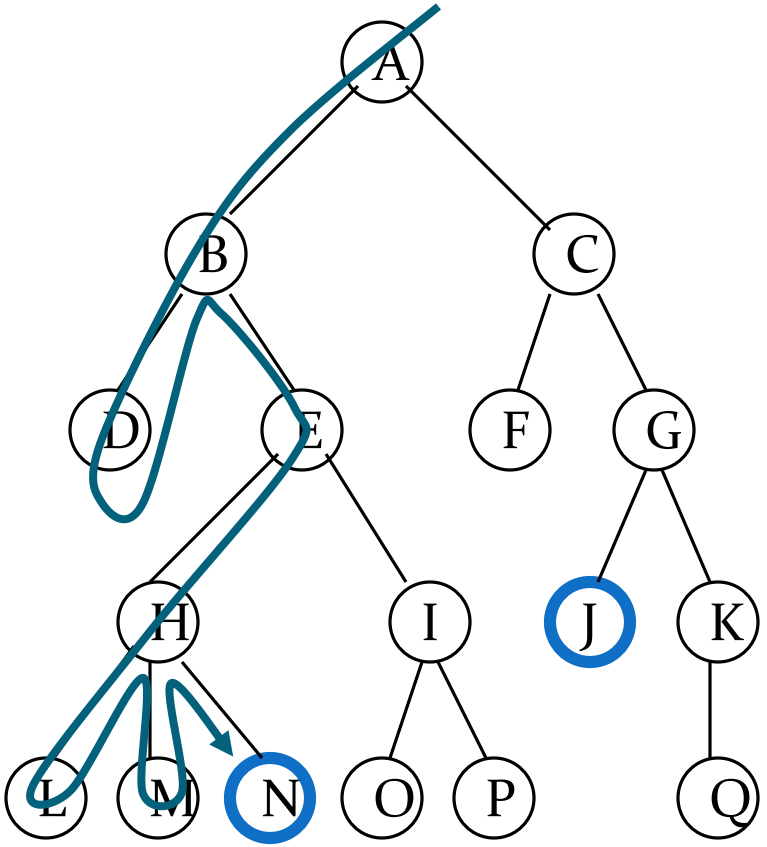
- *Warning*: if the graph is directed, the order of the vertices in each edge is important !!

Breadth-first searching[1]



- A breadth-first search (BFS) explores nodes nearest the root before exploring nodes further away
- For example, after searching A, then B, then C, the search proceeds with D, E, F, G
- Node are explored in the order A B C D E F G H I J K L M N O P Q
- J will be found before N

Depth-first searching[2]



- A depth-first search (DFS) explores a path all the way to a leaf before backtracking and exploring another path
- For example, after searching A, then B, then D, the search backtracks and tries another path from B
- Node are explored in the order A B D E H L M N I O P C F G J K Q
- N will be found before J

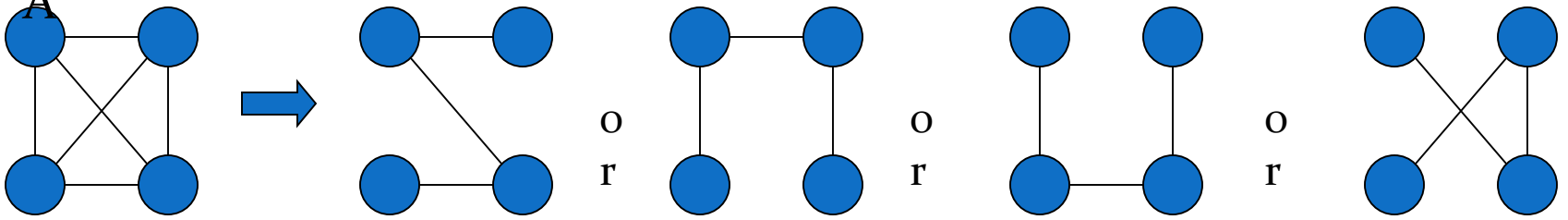
Spanning Trees[3]

A spanning tree of a graph is just a subgraph that contains all the vertices and is a tree.

A graph may have many spanning trees.

Graph A

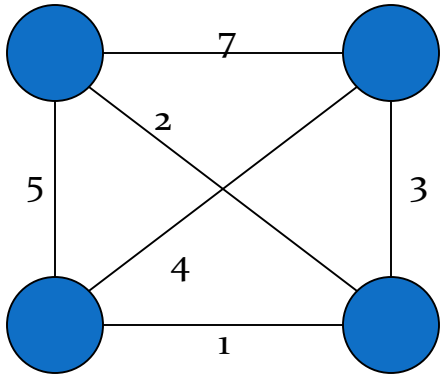
Some Spanning Trees from Graph A



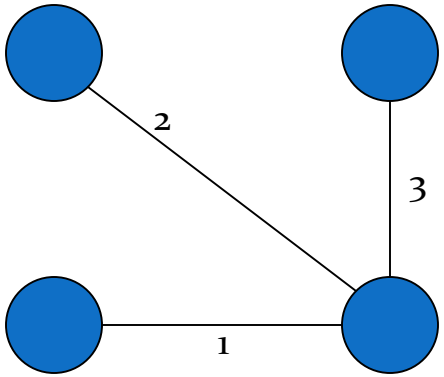
Minimum Spanning Trees[4]

The Minimum Spanning Tree for a given graph is the Spanning Tree of minimum cost for that graph.

Complete Graph



Minimum Spanning Tree



References

- An introduction to Datastructure with application by jean Trembley and sorrenson
- Data structures by schaums and series –seymour lipschutz
- http://en.wikipedia.org/wiki/Book:Data_structures
- http://www.amazon.com/Data-Structures-Algorithms
- http://www.amazon.in/Data-Structures-Algorithms-Made-Easy/dp/0615459811/
- http://www.amazon.in/Data-Structures-SIE-Seymour-Lipschutz/dp

- **List of Images**

1. http://www.algolist.net/Algorithms/Graph/Undirected/breadth-first_search
2. http://www.expertsmind.com/questions/depth-first-search-30153061.aspx
3. http://en.wikipedia.org/wiki/Book:Data_structure/spanning tree
4. http://en.wikipedia.org/wiki/Book:Data_structure