

# CSCA48 SUMMER 2017

## WEEK 6 - TREES

Brian Harrington

University of Toronto

June 5 - 9, 2017



UNIVERSITY OF  
**TORONTO**  
SCARBOROUGH

# ADMIN

- Term Test # 1
  - Covering everything up to now (ADTs - Linked Lists)
  - Closed book, no aids allowed
  - Focus on ADTs

# GRAPHS

- We've seen that objects can link to each-other
- People objects link to parents
- Node objects link to nodes
- General term for collection of linked objects `graph`
  - Linked list - directed acyclic unary graph
    - directed - You can go one way, but not (necessarily) backwards
    - acyclic - There are no loops (cycles)
    - unary - Each object is linked to *one* other object
  - Doubly Linked List - undirected acyclic unary graph
  - Family Tree - directed acyclic (hopefully) binary graph
  - Social Network - undirected n-ary graph

# TREES

- Trees are a special type of graph
- directed (usually), acyclic graphs
  - Each node has exactly 1 parent (Except the root, which has none), but one parent may have many children
  - There is a path from every node to the root
  - There are no `cycles` - no paths form loops
  - Result: There is exactly 1 path from every node to the root

# TREES: SOME DEFINITIONS

- **Leaf:** A node with no children
- **Internal Node:** A node which is neither a leaf, nor the root
- **Parent/Child:** If A is B's parent, B is A's child
- **Descendants:** A node's children, it's children's children, etc
- **Siblings:** Nodes with the same parent
- **Subtree:** The tree rooted at a node's child (the child plus all of that child's dependants)

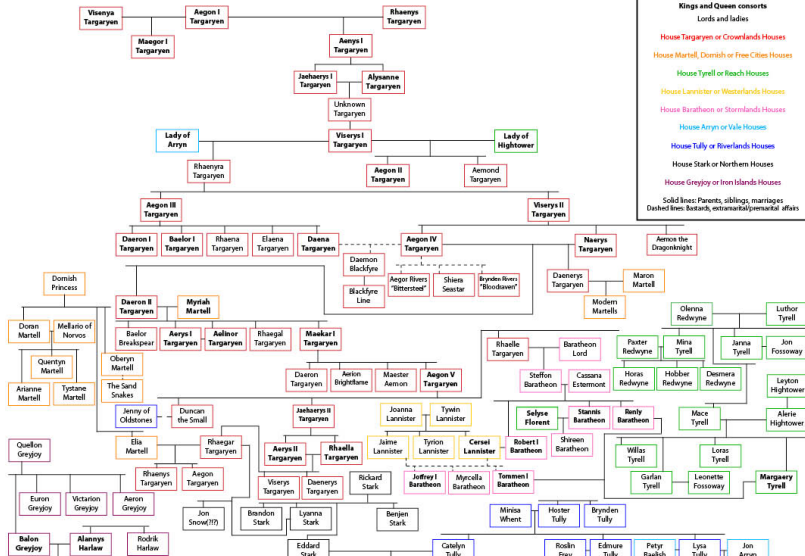
## TREES: MORE DEFINITIONS

- **Branching Factor:** The maximum number of children for any node
- **Path:** A sequence of nodes  $n_1, n_2, \dots, n_k$  such that  $n_i$  is the parent of  $n_{i+1}$
- **Length of a Path:** The number of nodes in a path
- **Height of a Tree:** The longest path from the root to a leaf
- **Depth of a node:** The number of nodes on the path from the root to that node (including that node)
- **NOTE:** These definitions may change slightly from source to source. If you're reading online, make sure to check the definitions.

# BINARY TREES

- Binary tree: a tree with a branching factor of 2 (each node has a maximum of two children)
- Call the children `left` and `right` for convenience

Solid lines: Parents, siblings, marriages  
Dashed lines: Bastards, extramarital/premarital affairs

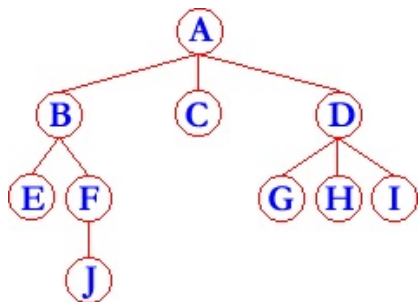




# TREE TRAVERSALS

- When we want to `traverse` a tree:
  - `traverse`: to visit every node once
- Preorder: first visit the node, then its children (from left to right)
- Postorder: first visit all the children (left to right), then visit the node
- Inorder: visit left subtree, then the node, then right subtree
  - Only makes sense for binary trees
- Depth First: Visit all nodes in one subtree before starting on the next subtree
- Breadth First: Visit all nodes at depth  $n$  before visiting nodes at depth  $n + 1$

# TREE TRAVERSALS



- Preorder: ABEFJCDGHI
- Postorder: EJFBCGHIDA
- Inorder: not a binary tree