CS330 Activities for Lecture 23 *

10.1 Trees Introduction

- **A** tree is a connected undirected graph with no simple circuits
- **Theorem** - An undirected graph is a tree if and only (iff) there is a unique simple path between any two of its vertices.
- **Theorem**: A tree with \( n \) vertices has \( n-1 \) edges (inductive proof)
- **A rooted tree** is a tree in which one vertex has been designated as root and every vertex is directed away from the root.
- From here on, all trees are rooted unless said otherwise.

- Relationships between vertices. Given a vertex \( v \):
  - The *parent* of \( v \) is the first vertex along the path from \( v \) to the root; the parent is unique. The root has no parent.
  - An *ancestor* of \( v \) is its parent or its parent’s parent, etc., up to the root.
  - A vertex can be the parent of many *children*. The *descendants* of \( v \) are its children, its children’s children, etc. Two vertices are *siblings* if they have the same parent.
  - An *interior vertex* is a vertex with children. A *leaf* is a vertex with no children. The *frontier* of a tree is the set of all its leaves.
  - The *subtree rooted at vertex* \( v \) is the tree consisting of \( v \) and its descendants.
  - In an *n–ary tree*, every internal vertex has \( \leq n \) children.
  - A *full n–ary tree* is an \( n \)-ary tree where every internal vertex has exactly \( n \) children.
  - A *labeled* tree is a where data is attached to the vertices or edges.

- **Distances**
  - The *level* of a vertex is the length of the (unique) path from the root to this vertex. (The root has level 0.)
  - The *height* of a vertex is the length of the longest path from the vertex to one of its descendant leaves.
  - The *height* of a rooted tree is the maximum of the levels of all vertices. (I.e. the length of the longest path from the root to any leaf.)
  - A *balanced tree* if one where all the leaves are at 2 adjacent levels.

**Questions**

1. For the tree shown here, what is the root? What are the leaf vertices? Interior vertices? What are the ancestors of \( e \)? Descendants of \( c \)? How are \( b \) and \( c \) related? (Obscure: If this were a family tree, how would \( d \) and \( g \) be related?)

* Some material from MATH230 Dr. Robert Ellis IIT-AMAT, zyBooks Disc. Math., and Disc. Math. and its Appl., Rosen (7th Ed.) is included under the academic/nonprofit Fair Use exception of US Copyright Law. This document must NOT be distributed beyond participants in IIT’s CS330.
2. Why are family trees not always rooted trees as described here? (Hint: Richard III married Anne Neville and Richard’s mother was named Cecily Neville.)

3. Characterize a basketball championship playoff bracket in terms of trees.

4. Here’s how a chain letter works. The initial person sends out some number \(N\) of copies of the letter, which contains some instructions and a list of \(\leq M\) names (initially, just one name, the person who starts the chain letter). The instructions say if there are \(< M\) names on the list, add your name and send out \(N\) copies of the letter. If there are \(M\) names on the list, send a dollar to the person at the top of the list, remove that name, then add your name and send out \(N\) copies of the letter. Who gets money from the chain letter, and how much do they get?

10.2 Tree Applications

- Trees are good for storing hierarchical information — management organization charts, lineages, sequences of decisions, nested pieces of information. The textbook shows game trees and variable-length coding trees, but we’ve also seen parse trees (where vertices are nonterminals and children are the results of a grammar rule application).

Questions

5. (a) What grammar rules are being used in the parse tree below left? (b) Draw a parse tree for the derivation

\[
E \rightarrow T + T \rightarrow F + T \rightarrow C + T \rightarrow 2 + F \rightarrow 2 + F * F \rightarrow 2 + V * F \rightarrow 2 + x * F \rightarrow 2 + x * V \rightarrow 2 + x * y
\]

\[
\frac{S}{0} \rightarrow S \rightarrow S \rightarrow 1 \rightarrow \lambda
\]

† So says Wikipedia’s Wars of the Roses entry.
6. Another tree application is **operator trees**, which represent expressions. Interior nodes represent operations like +, -, etc. and leaves are basic operands (like numerals and variables). Draw an operator tree for \( \times \cdot y + 3 / \left( y / 4 \right) \).

7. A game tree represents a game situation and its possible outcomes (textbook shows a tic-tac-toe game). Vertices are game situations, edges are possible plays from a situation. The illustration here shows a subtree of all possible tic-tac-toe game trees. (a) What does the root of the tree of all possible tic-tac-toe games look like? (b) How many edges lead out of it? (c) How many edges lead out from the children of the root? From their children? (d) In the true game tree, a path from the root ends if you get to a complete game (win, lose, or tie). If we ignore that and imagine games always continuing until you fill the board entirely, how large is the full game tree?
8. In an encoding tree, the edges are labeled with bits, the leaves are letters or other symbols, and a path from the root to a leaf gives you the bitstring that encodes that letter. E.g., in the tree below from the book, 0 encodes a, 100 encodes i, 101 is for t, and so on. (a) What is the bitstring that encodes Mxyzptlk? (b) Why can’t we label interior nodes with letters? (c) If the goal is to find short bitstrings to encode a character string, what does this say about the relative frequency of the letter a vs the letter, oh, m? (d) If on average, the letters a, i, t, and r appear 4%, 3%, 2%, and 1% of the time, then how many bits on average do we use to encode a string that uses only those letters?

![Diagram of an encoding tree](image-url)