CSCI 136: Data Structures and Advanced Programming
Lecture 23
Trees, part 2

Instructor: Dan Barowy
Williams

Topics
Tree terminology

Your to-dos
1. Read before Wed: Bailey, Ch 14.4.
2. Lab 7 (partner lab), due Tuesday 11/8 by 10pm.

Announcements
CSCI 136 final exam
Saturday, December 17 at 1:30pm
Room TBD
A tree is a recursive data structure that stores information hierarchically. A tree is either:

- empty (i.e., $\emptyset$), or
- a node containing a value and references to one or more trees.

The empty tree:

A non-empty binary tree:

Terminology:

The topmost node is called the root.

Properties of trees:

Connected: every node in a tree is reachable by following a single unique path starting from the root node.
Properties of trees

# edges: a tree having \( n \) vertices always has \( n-1 \) edges.

Terminology

The nodes at the bottom of a tree are called **leaves**.

Terminology

Any node that is not a leaf is an **interior node**.

Terminology

A **node** may have **children**.
Terminology

A **node** may have **children**.

**Terminology**

A **node** may have **children**.

**Terminology**

A **node** that has **children** is called the **parent** of those children.

**Terminology**

For a **given node**, all of the nodes above it are called **ancestors**.
Properties of trees

**Single ancestor:** every node in a tree has at most one ancestor.

![Diagram showing a single ancestor property]

Terminology

For a **given node**, all of the nodes below it are called **descendants**.

![Diagram showing descendants]

Properties of trees

**Subtrees:** the descendents of every tree (except the empty tree) are also trees.

![Diagram showing subtrees]

Terminology

The **degree** of a tree is the maximum number of **children** had by any node.

![Diagram showing degree]

Degree of this tree: 2
Degree 2 trees are common: we call them **binary trees**.
A tree that is missing no leaves is **full**.

A **path** is a sequence of edges between **two nodes**.

**Cycle-free**: no path will ever revisit the same node.