CMPSCI 575/MATH 513

Combinatorics and Graph Theory

Lecture #8: Trees
(Tucker Sections 3.1, 3.2)
David Mix Barrington
23 September 2016

Trees

- The Variety of Tree Definitions
- Characterizing Trees
- Counting Nodes and Leaves
- The Number of Trees on n Labels
- DFS, BFS, and Spanning Trees
- Searching State Graphs
- Traversals of Trees

The Variety of Tree Definitions

- Today we'll talk about trees, a concept so important to computer science that it is covered extensively in CS 187 (data structures) and CS 250 (discrete math).
- Much of what we'll say today is review of that material, but there will be one new result about counting labeled trees.
- To start, we must deal with a number of competing definitions of "tree".

The Variety of Tree Definitions

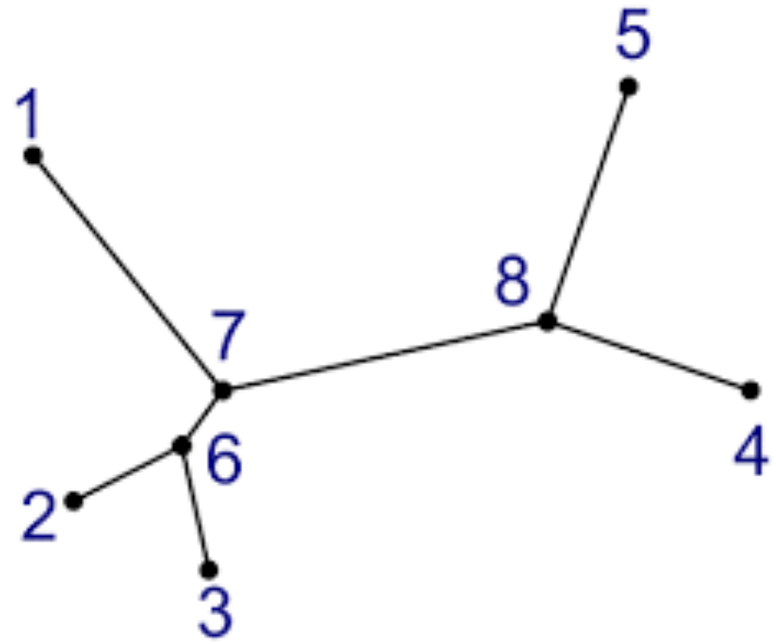
- Primarily, we will view a tree as a type of graph, one that is **connected** and **acyclic.** The latter means that there is no simple path (trail) from a node to itself.
- When we choose a root for such a tree, we find that there is a unique way to direct every edge away from the root, forming a directed graph called a **rooted tree**.
- In a rooted tree we use "family" language to describe nodes as parents, children, etc.

Trees and Recursion

- A rooted tree can be defined as a node (the root) with edges to zero or more more nodes (its children), each of which is the root of a rooted tree.
- The tree is **binary** if every node has either zero or two children (though sometimes we allow one child).
- A **leaf** is a node with no children, and other nodes are called **internal nodes**.

An Example

- Here is an 8-node tree, with 7 edges.
- If we made 7 the root, it would have children 1, 6, and 8, and the other nodes would be grandchildren.
- If 5 were the root it would have an only child and nodes 2 and 3 would be at **depth** 4.



Characterizing Trees

- Here are four properties of a connected graph:
- (a) acyclic: there are no circuits
- (b) ∃x:∀y: unique path from x to y
- (c) ∀x:∀y: unique path from x to y
- (d) minimally connected: removing any edge disconnects the graph
- We can prove that (a) → (d) → (b ↔ c) → (a), and each step of this is pretty easy.

Characterizing Trees

- ¬(d) → ¬(a): If you can remove (x, y) and leave a connected graph, you have a circuit
- ¬(c) → ¬(d): If you have distinct paths from x to y, you can remove an edge on one of them and still have the other to connect the graph
- (c) ↔ (b): (c) → (b) is trivial, and given (b) you can use those paths to prove (c)
- ¬(a) → ¬(c): A circuit yields two paths.

Counting Nodes and Leaves

- It's easy to show by induction that an n-node tree has exactly n-1 edges.
- Similarly, an m-ary tree with i internal nodes must have exactly n = mi + 1 nodes and so the number of leaves is (m-1)i + 1.
- A balanced m-ary tree of depth h has exactly m^h leaves and $(m^h-1)/(m-1)$ internal nodes.
- We can use this in applications.

Binary Search

- If we have n items in an array and they are in order, we can find a given item (or prove its absence) in ceiling(log(n+1)) queries. This is because the items are at leaves of a balanced binary tree.
- In a binary search tree we store an item at each node of a binary tree, not necessarily balanced. If we don't find an item at a node, we know which of the two subtrees to search. This is O(log n) time if we are balanced.

How Many Trees on n Labels?

- Here's another counting problem, solved by Cayley in 1889. If we have n nodes, labeled 1 through n, how many ways are there to combine them into a tree?

- For n=1 and n = 2 all trees are isomorphic.

- For n=3, any of the three nodes could be in the middle, so there are 3 trees.

- For n=4, there are 4 choices of a root for a star graph, and 12 ways to arrange a line.

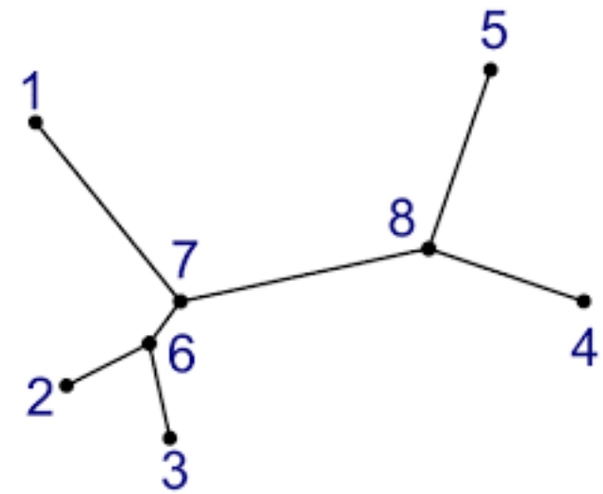
- We have $f(2) = 2^0$, $f(3) = 3^1$, and $f(4) = 4^2$.

How Many Trees on n Labels?

- Cayley's theorem says that there are exactly n^{n-2} trees on n labels.
- Our proof constructs a bijection between labeled trees with n nodes and **Prufer sequences**, which are sequences of n-2 labels.
- We need to go from tree to sequence and from sequence to tree.

Tree to Sequence

- The first element of the sequence is the label of the neighbor of the lowest numbered leaf, here 7.
- We then delete the leaf and continue. The neighbor of the new lowest numbered leaf is 6.
- Now the lowest leaf is 3, with neighbor 6, followed by 4, with neighbor 8, 5 with neighbor 8, and 6 with neighbor 7, for 7-6-6-8-8-7.



Sequence to Tree

2-7-3-3-6

- To illustrate the map in the other direction, we'll build the 7-node tree with Prufer sequence 2-7-3-3-6.

1——2

- We know that 1, 4 and 5 are leaves because they are not in the sequence.
- 1 is the lowest leaf and thus has neighbor 2.

Sequence to Tree

2-7-3-3-6

- 1 is the lowest leaf and thus has neighbor 2.

1——2——7

- Now we consider 2, 4, and 5 to be the leaves. The lowest is 2, which thus has neighbor 7.

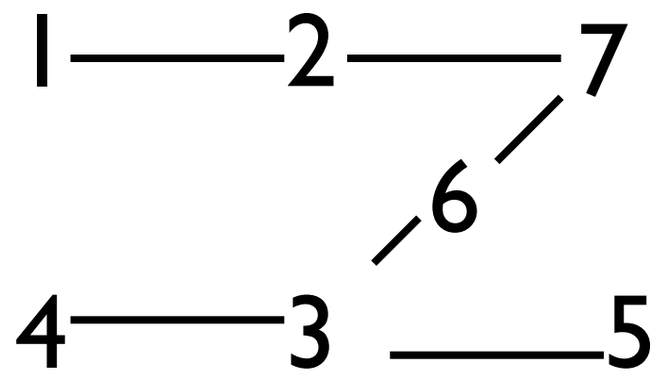
4——3

- Now the leaves are 4, 5, and 7. The lowest is 4, which has neighbor 3. This makes our leaves 5 and 7.
- 3 is still in the sequence and thus still an internal node.

Sequence to Tree

2-7-3-3-6

- The leaves are 5 and 7.
- 5 is lowest and has neighbor 3.
- Now 3 and 7 are leaves.
- 3 is lowest and has neighbor 6.
- We connect the final leaves, 6 and 7, and we are done.
- The reverse process gives us 2-7-3-3-6.



DFS, BFS, and Spanning Trees

- Any connected graph has one or more **spanning trees**, which are subgraphs that form trees containing all the nodes.
- The familiar DFS and BFS algorithms each produce a spanning tree, with edges from each newly discovered vertex to the vertex from which it was first found.
- These algorithms also test for connectedness because they search the whole connected component of the start node.

DFS, BFS, and Spanning Trees

- Running a DFS or BFS takes O(e) time, where e is the number of edges. (If, that is, we store the graph as a list rather than as a matrix.)
- The DFS and BFS trees can then be used to answer other questions about the graph.
- The BFS tree contains shortest paths from the start node to every other node.
- But DFS is in general more space-efficient.

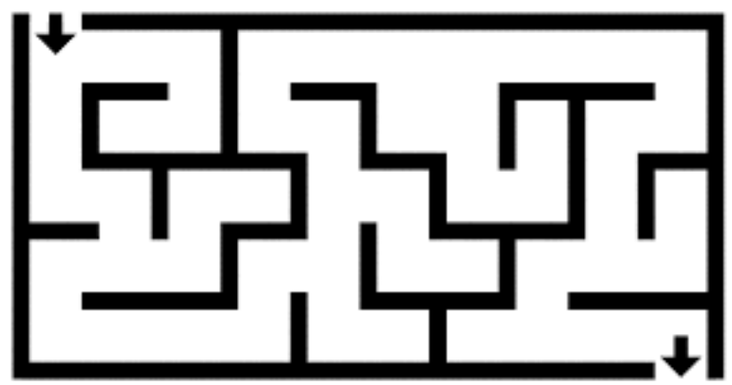
Searching State Graphs

- Many solitaire puzzles can be modeled by a directed graph, where each node represents a state and an edge represents a legal move.
- Once you have the graph, you can DFS or BFS to find a path to the (or a) goal node.
- Or if you can't represent the whole graph, you can explore it node by node, though you might waste time revisiting.
- DFS in this setting is called **backtrack search**.

Searching a Maze

- A maze like the one below may be modeled as a graph, a subgraph of a Manhattan grid graph like we saw on HW#1. We make a node for the center of each square, and an edge whenever two squares have no wall between them.

Many mazes, like this one, have the property that the walls form only two connected components. This turns DFS into the "right-hand rule".

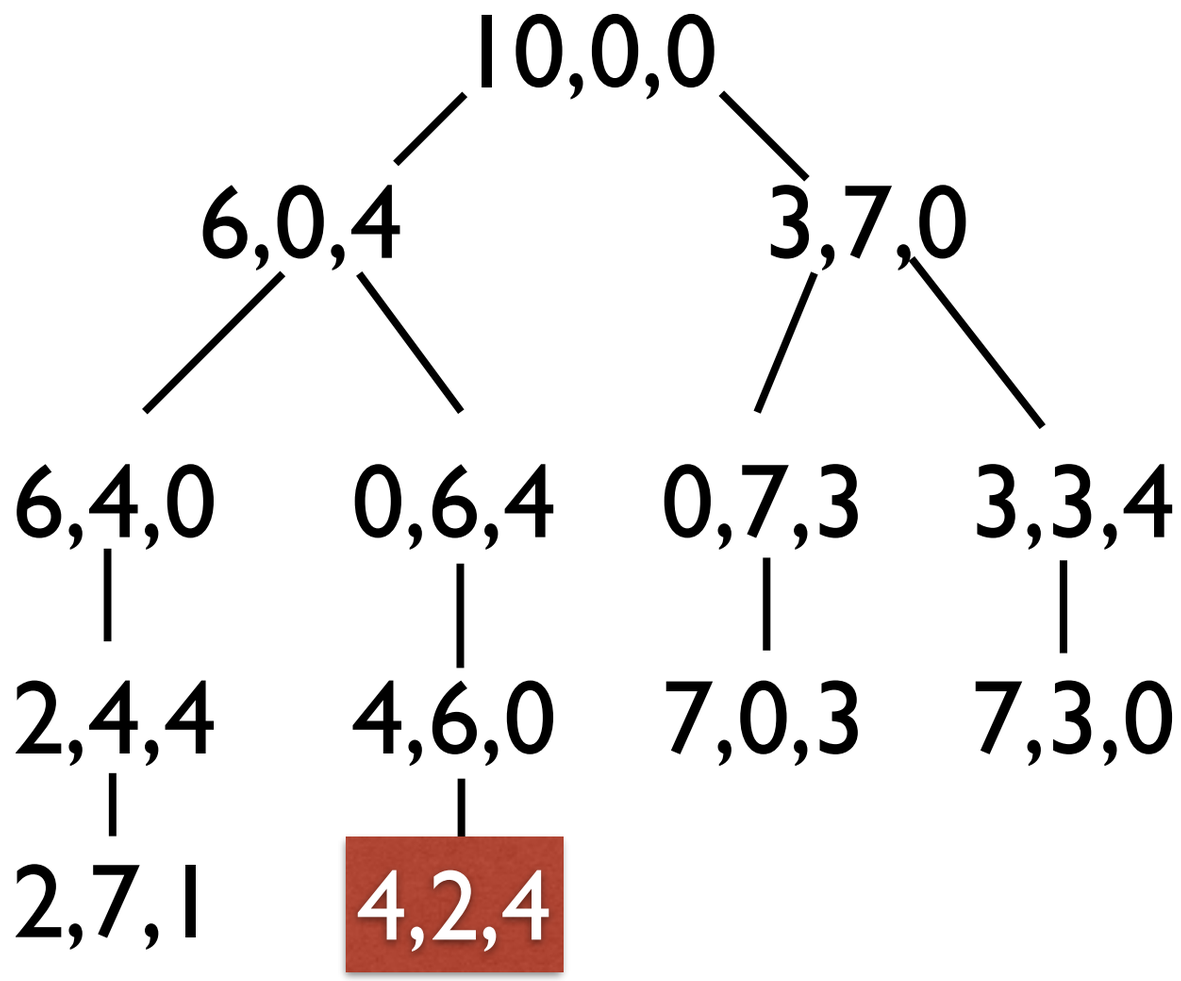


Pitcher Pouring

- You have three pitchers, a full one of size 10 and empty ones of size 7 and 4, and you need to measure 2 units of water. You may pour from one pitcher to another until the former is empty or the latter is full.
- A state of the system is a triple (i, j, k) saying how much is in each pitcher. We start at (10, 0, 0) and want to get to (i, 2, k) or (i, j, 2).
- We just make a graph, and a BFS will find us the shortest possible sequence of moves.

Pitcher Pouring

- For the first two moves, there are multiple options, but in the third and fourth moves most choices lead to previously seen states.



- We reach a goal state in four moves.

Missionaries and Cannibals

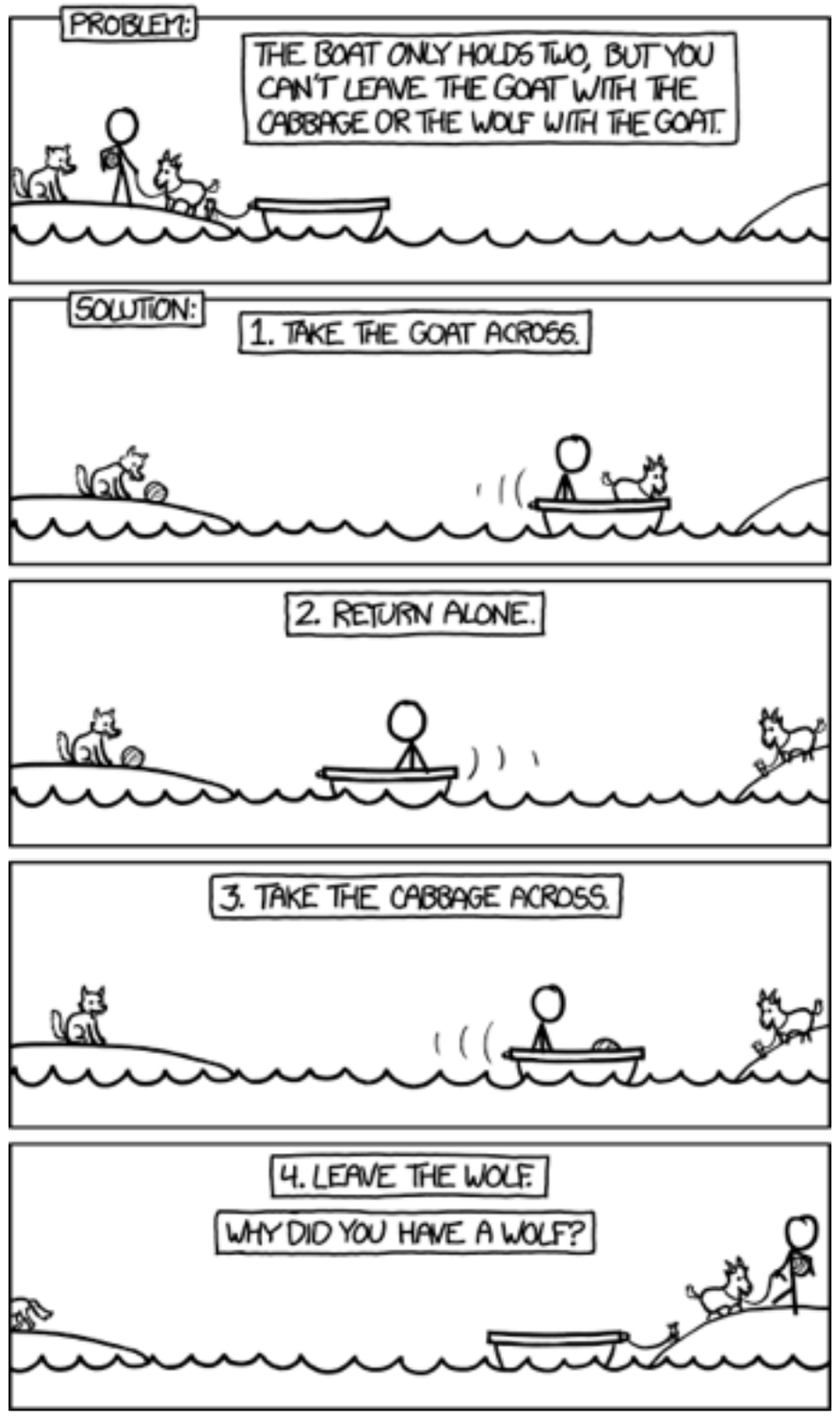
- This politically incorrect puzzle has three missionaries and three cannibals on one shore of a river, and a two-person boat.
- The forbidden states are to have one or missionaries outnumbered by cannibals on one shore.
- A state consists of the number of m's and the number of c's on the near shore, plus a bit to say where the boat is.

Missionaries and Cannibals

- We call the start state $(3,3)^*$, and the goal state (0,0).
- The forbidden states are (1, 0), (1, 2), (1, 3), (2,0), (2, 1), (2,3), and the same states starred. The move set is {m, mm, mc, c, cc}.
- Tucker gives a solution with eleven moves.
- This puzzle is small enough to graph by hand and find the path by eye, but a BFS would be guaranteed to find the shortest path.

Mandatory xkcd Reference

- In this classic problem there are 16 states, and some are excluded.
- If we wanted to finish we could return with the goat, take the wolf, and return for the goat leaving the wolf and cabbage safely together.



xkcd.com/1134

Tree Traversals

- We recall the definitions of **pre-order**, **in-order**, and **post-order** traversals, recursively on the structure of a rooted tree.
- In a binary search tree, the order of stored elements is in-order.
- In a DFS or BFS, we view the elements in pre-order applied to the spanning tree.
- Arithmetic expressions are usually written by humans in infix notation and converted to postfix to be evaluated by machines.