CS 113: Mathematical Structures for Computer Science

Dr. Francis Parisi

Pace University

Fall 2018

Trees

Chapter 13

Trees

- Fundamental Properties of Trees
  - Degrees of the Vertices of a Tree
- Rooted and Spanning Trees
  - Binary Search Trees
  - Decision Trees
  - Constructing Spanning Trees
  - Depth-First Search Trees
  - Breadth-First Search Trees
- The Minimum Spanning Tree Problem
  - Kruskal's Algorithm
  - Prim's Algorithm

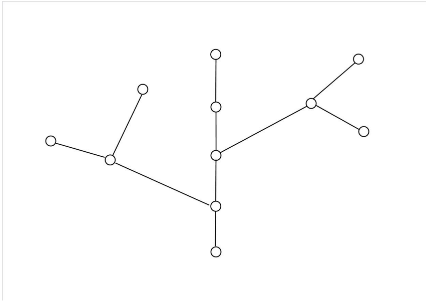
Trees – Fundamental Properties of Trees

Trees are a class of graphs and while simple they are an important class

Let's define what is a tree ...

Definition

A **tree** is a connected graph with no cycles



Trees – Fundamental Properties of Trees

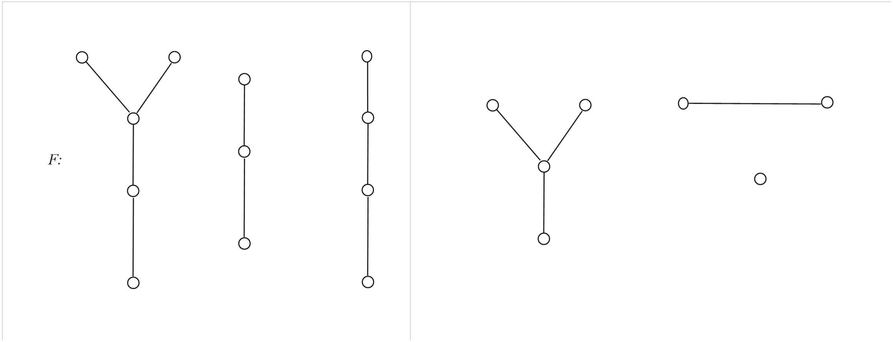
Definition

A graph containing no cycles is called a **forest**

- A connected forest is a tree
- Every forest is a tree but not every tree is a forest
- Each component of a forest is a tree

Trees – Fundamental Properties of Trees

Below is a graph F with six components, each component is a tree



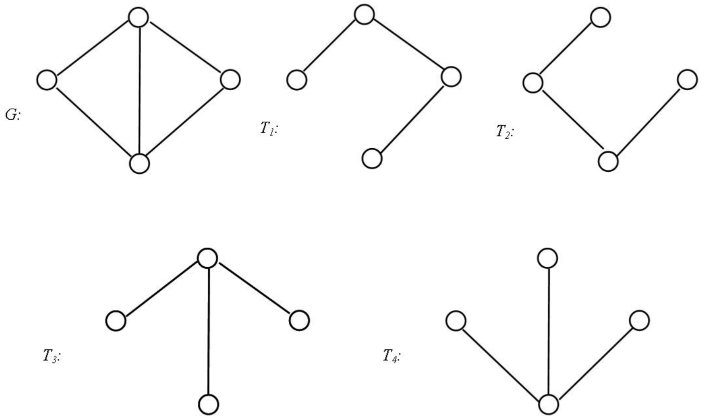
Trees – Fundamental Properties of Trees

Definition

A **spanning tree** T of a connected graph G is a spanning subgraph of G that is a tree, that is, $V(T)V(G)$. If G is a tree then G has only one spanning tree – itself

For the connected graph G on the next slide, all of the trees T_1, T_2, T_3, T_4 are spanning trees of G

Trees – Fundamental Properties of Trees



Trees – Fundamental Properties of Trees

Theorem

Every nontrivial tree contains at least two leaves

Proof.

. . . $\square$

Theorem (13.6)

Every tree of order n has size $n-1$

Proof.

. . . $\square$

Theorem 13.6 can be stated as follows: *If T is a connected graph of order n and size m, then $m = n - 1$*

Trees – Fundamental Properties of Trees

Theorem

Every graph of order n and size $n-1$ containing no cycles is a tree

Proof.

$\ldots$ $\square$

Corollary

If F is a forest of order n and size m having k components, then $m = n - k$

Trees – Fundamental Properties of Trees

Theorem

Every connected gapgh of order n and size $n-1$ is a tree

Theorem

If G is a graph of order n and size m satisfying any two of the three properties then G is a tree:

$$(1)\ G \text{ is connected, } (2)\ G \text{ has no cycles, } (3)\ m = n - 1$$

Theorem

A graph G is a tree if and only if G has a unique $u - v$ path for every two vertices u and v in G

Trees – Fundamental Properties of Trees

Degrees of the Vertices of a Tree

Let T be a tree of order $n \geq 2$ and size m having maximum degree $\Delta(T) = k$. Suppose that T has n_i vertices of degree i for $i = 1, 2, \ldots, k$. The number n_1 is the number of leaves in T. Then we have

$$\sum_{i=1}^{k} n_i = n_1 + n_2 + \cdots + n_k = n \text{ and}$$

$$\sum_{i=1}^{k} i n_i = 1 \cdot n_1 + 2 \cdot n_2 + \cdots + k \cdot n_k = 2m = 2(n-1) = 2n - 2.$$

The second sum is the First Theorem of Graph Theory.

Trees – Fundamental Properties of Trees

Degrees of the Vertices of a Tree

Using the prior sums we can formulate a theorem

Theorem

Let T be a tree of order $n \geq 2$ with maximum degree k having n_i vertices of degree i for $i = 1, 2, \ldots, k$. Then the number of leaves in T is

$$n_1 = 2 + n_3 + 2n_4 + 3n_5 + \cdots + (k-2)n_k.$$

Trees – Fundamental Properties of Trees

Degrees of the Vertices of a Tree

Proof.

Since

$$\sum_{i=1}^{k} in_i = 2n - 2 = 2\left(\sum_{i=1}^{k} n_i\right) - 2,$$

it follows that

$$n_1 + 2n_2 + 3n_3 + \cdots + kn_k = 2n_2 + 2n_2 + 2n_3 + \cdots + 2n_k - 2$$

and so

$$n_1 = 2 + n_3 + 2n_4 + 3n_5 + \cdots + (k-2)n_k.$$

□

Trees – Rooted and Spanning Trees

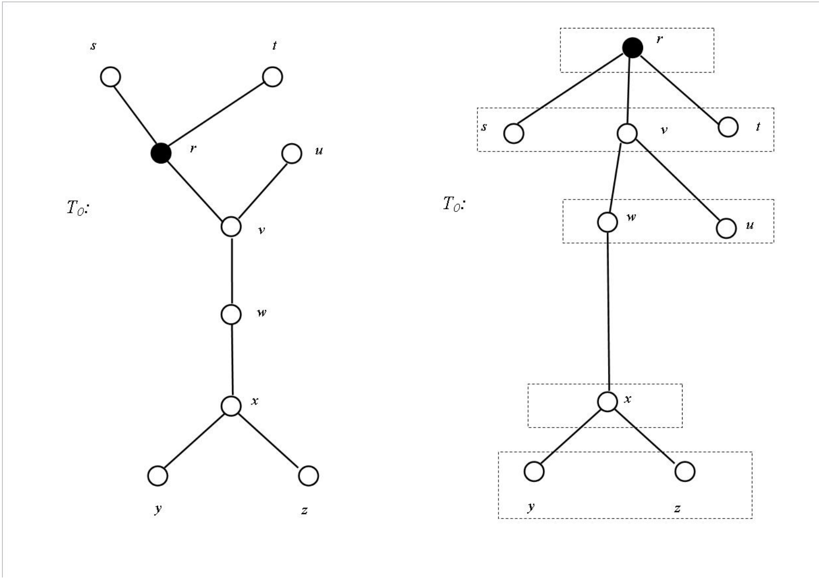
The idea of a rooted tree comes into play when we are concerned with a particular vertex r in a tree T and the distance from r to each vertex in T

Definition

A tree T is called a **rooted tree** if T contains a designated vertex r, called the **root** of T. The tree T is then said to be **rooted at** r.

The **height** of a rooted tree with root r is the largest integer h for which there is a vertex v of T at level h. In other words, it's the maximum distance from r to another vertex of T.

Trees – Rooted and Spanning Trees



Trees – Rooted and Spanning Trees

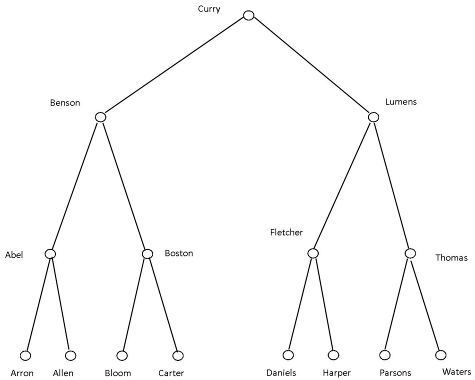
Binary Search Tree

Definition

For a list of names arranged alphabetically a **binary search tree** is a binary tree whose vertex set is the set of names on the list such that for every vertex v of T the vertices in the left subtree of v precede it alphabetically and those vertices in the right subtree of v follow v alphabetically.

Trees – Rooted and Spanning Trees

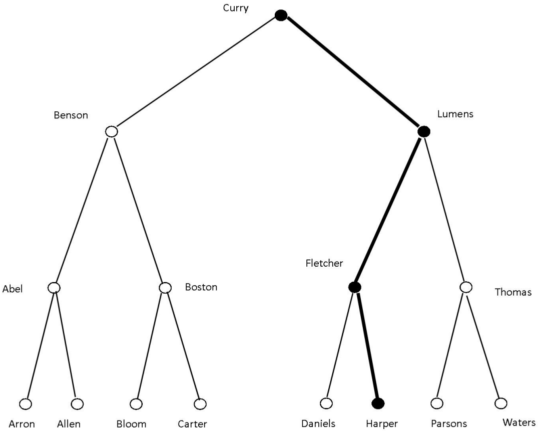
Binary Search Tree



Suppose we want to find *Harper*

Trees – Rooted and Spanning Trees

Binary Search Tree

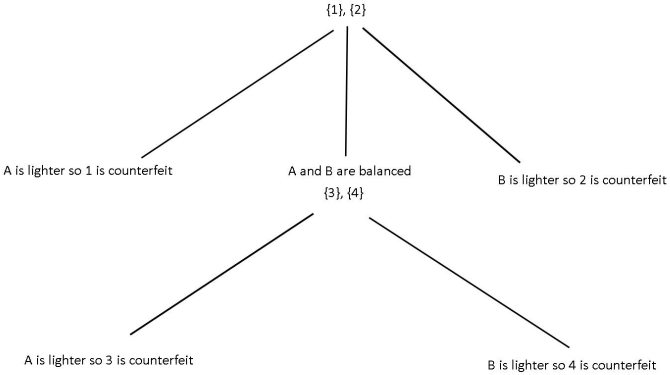


Trees – Rooted and Spanning Trees

Decision Trees Rooted trees are useful for finding the solution to decision problems

- Suppose we have four coins numbered 1,2,3,4
- Exactly one coin is counterfeit and weighs less than the real coins
- How can we solve this problem as a sequence of weighings with a minimum number of weighings?
- We can use a **decision tree**

Trees – Rooted and Spanning Trees



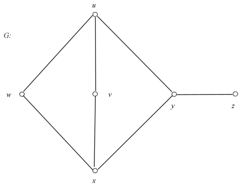
Trees – Rooted and Spanning Trees

Constructing Spanning Trees

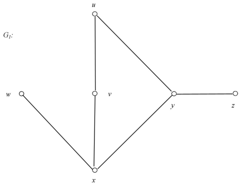
- There two ways we can construct a spanning tree T from a graph G
- We can delete an appropriate edge from G
- Begining with the vertex set $V(G)$ we add edges

Trees – Rooted and Spanning Trees

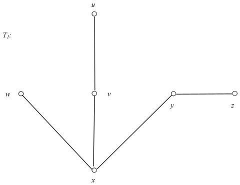
Connected graph G



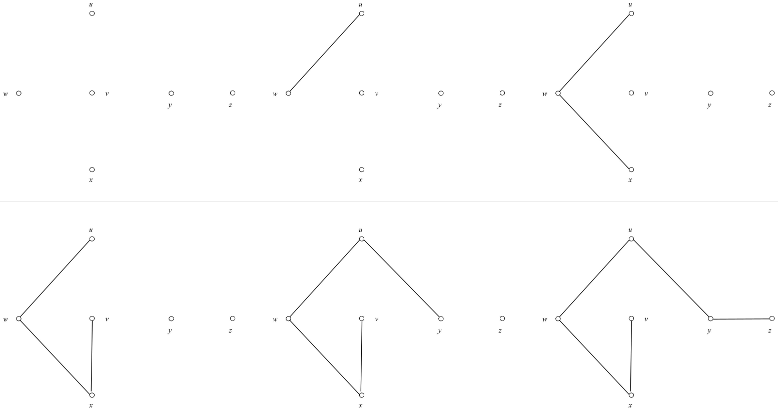
Delete uw



Delete uy



Trees – Rooted and Spanning Trees



Trees – Rooted and Spanning Trees

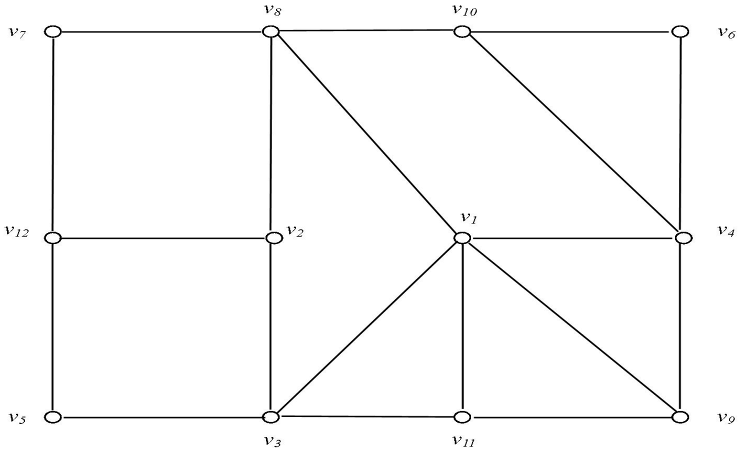
Depth-First Search Trees

- Begin at some vertex v
- Visit all the vertices of G proceeding deeply into G with a path P
- If P contains all the vertices of G then P is a spanning tree and a depth-first search tree
- If P does not contain all the vertices of G then backtrack until we reach vertex v' and construct a new path P'
- We continue until we have a spanning tree

Trees – Rooted and Spanning Trees

Example

Construct a depth-first search tree T rooted at v_1 for the graph G



Trees – Rooted and Spanning Trees

Breadth-First Search Trees

- Similar to a breadth-first search tree except
- Start with a tree T_1 rooted at some vertex v
- If T_1 does not contain all the vertices, then move to a neighbor of v with the smallest subscript and add that edge, and so on

Example

Let's revisit the graph G and create a breadth-first search tree

Trees – Minimum Spanning Tree Problem

Definition

A **minimum spanning tree** of a connected weighted graph G is a spanning tree of G whose weight is minimum among all spanning trees of G

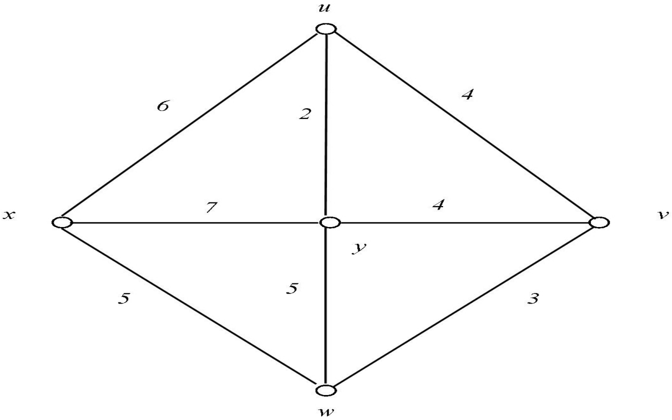
- Just as we saw with weighted graphs the weight of edge e of G is $w(e)$
- The **weight** $w(H)$ of a subgraph H of G is the sum of the weights of the edges

$$w(H) = \sum_{e \in E(H)} w(e)$$

Trees – Minimum Spanning Tree Problem

Kruskal's Algorithm

- Select any edge with smallest weight
- Then select any remaining edge with the smallest weight
- Continue until we produce a spanning tree

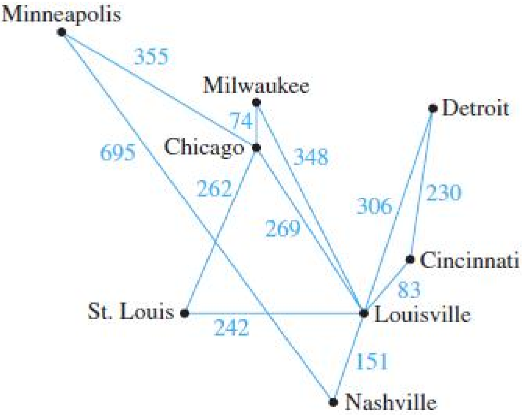


Trees – Minimum Spanning Tree Problem

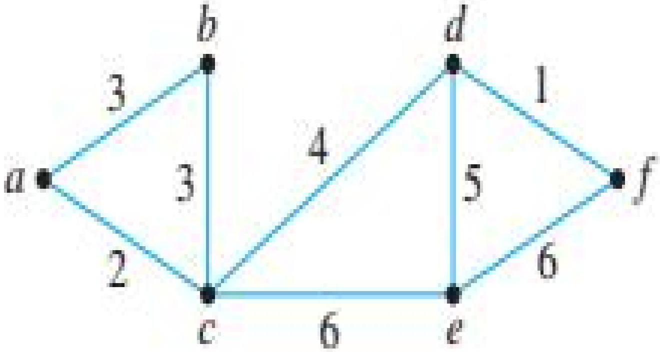
Prim's Algorithm

- Select a vertex r in G and an edge with smallest weight incident r
- Select a second edge with smallest weight incident to exactly vertex already in T
- Proceed until we have a minimum spanning tree

Trees – Minimum Spanning Tree Problem



Trees – Minimum Spanning Tree Problem



Trees – Key Results

- Every nontrivial tree contains at least two leaves
- Every tree of order n has size $n-1$
- If T is a connected graph of order n and size m containing no cycles, then $m = n-1$
- Every graph of order n and size $n-1$ containing no cycles is a tree
- If F is a forest of order n and size m having k components, then $m = n-k$
- Every connected graph of order n and size $n-1$ is a tree
- If G is a graph of order n and size m satisfying any of the two following three properties, then G is a tree: (1) G is connected, (2) G has no cycles and (3) $m = n-1$
- A graph G is a tree if and only if G has a uniques $u-v$ path for every two vertices u and v in a tree G

Trees – Key Results

- Let T be a tree of order $n \geq 2$ with maximum degree k having n_i vertices of degree i for $i = 1, 2, \ldots k$. The the number of leaves in T is
  $n_1 = 2 + n_3 + 2n_4 + 3n_5 + \cdots + (k-2)n_k$.
- A graph G contains a spanning tree if and only if G is connected
- **The Depth-First Search Algorithm:** This algorithm constructs a depth-first search tree in a connected graph
- **The Breadth-First Search Algorithm:** This algorithm constructs a breadth-first search tree in a connected graph
- **Kruskal's Algorithm:** an algorithm that produces a minimum spanning tree in a connected weighted graph by constructing a forest of increasing size
- **Prim's Algorithm:** an algorithm that produces a minimum spanning tree in a connected weighted graph by constructing a forest of increasing size