Major Themes

- Trees
- Spanning trees
- Binary trees
- Binary search trees
- Applications

But first: Review Graphs

Types of Graphs

- Directed graph
- Weighted graph (directed or undirected)
- Simple graph
- Special graphs
  - Cycle $C_n$
  - Wheel $W_n$
  - Complete graph $K_n$
  - A Complete Bipartite $K_{x,y}$
- Connected graph
- **Tree: a connected graph with no cycles**
- **Forest: a graph with no cycles**
Which of the following are trees?

G₁  G₃
G₂  G₄

Tree: a connected graph with no cycles

- A tree with n vertices has n-1 edges
- A connected graph with n vertices and n-1 edges is a tree.
- Every tree with at least 2 vertices has 2 or more leaves
  - A leaf is a vertex of degree 1.

Subgraphs of K₃ – which of these are trees?

A Spanning Trees

Spanning Tree: A subgraph that is a tree and contains all the vertices of a given graph.

- Every connected graph has at least 1 spanning tree.
- Every weighted connected graph has a minimum spanning tree
Kruskal’s Minimum Spanning Tree Algorithm

• Order the edges of G, by increasing weight
• G’ starts out as the graph with all the vertices of G and none of the edges
• Keep adding edges to G’, at each iteration adding the lowest weighted edge that does not introduce a cycle
• Stop when the graph becomes connected.
• G’ is now the minimum spanning tree.

Prim’s Minimum Spanning Tree Algorithm

• G’ starts out as the graph with the least weighted edge in G and the 2 vertices incident with this edge
• Select the lowest weighted edge that connects a vertex in G’ to any of the remaining vertices of G; add edge and vertex to G’ and repeat until all the vertices of G are added to G’
• G’ is now the minimum spanning tree.
**Rooted Tree**

- A *rooted tree* has a distinguished vertex called the *root*.
- Rooted trees are usually drawn with the root at the top and all other vertices connected by edges growing downward.
- Each edge in a rooted tree connects a *parent* vertex with a *child* vertex below it.

**Binary Tree**

- A rooted tree.
- Each vertex has at most 2 children.
- The children of a vertex are distinguished as *left* and *right* child.
- The $k^{th}$ *rank* of a binary tree is the collection of vertices at distance $k$ from the root.
- In a *complete binary tree* every vertex has exactly two children, except for the leaves in the bottom rank.
- The height of a tree is the highest rank of its leaves.

**Turn these into rooted trees**

$G_1$ and $G_4$

**More about binary trees:**

- How many leaves in a complete binary tree?
- If a tree has height 5 and 50 vertices, is it possible for it to be:
  - a binary tree?
  - a complete binary tree?
- Repeat with a tree of height 5 with 100 vertices.
- Example 10.5.3 – ancestral trees.
Binary Search Trees

A binary search tree has a datum associated with each node such that in the ordering of the data set, the datum occurs earlier than any of the data downward and to the right and occurs later than any of the data downward and to the left.

Example: Subsets of the set \{a,b,c\} as a binary decision tree

Check Yourself 10.5 #4

- Place baa at the root and draw the binary search tree for aaa, ab, baa, baba

Fake coin puzzle

- You have some gold coins and a balance scale. One of the coins is a fake and weighs less (gold is HEAVY). How can you determine which coin is fake with the least number of weighings?