UNIT 6C
Organizing Data: Trees and Graphs

Trees
Trees

- A **tree** is a hierarchical data structure.
  - Every tree has a **node** called the **root**.
  - Each node can have 1 or more nodes as **children**.
  - A node that has no children is called a **leaf**.
- A common tree in computing is a **binary tree**.
  - A binary tree consists of nodes that have at most 2 children.
  - A **complete binary tree** has the maximum number of nodes on each of its levels.
- Applications: data compression, file storage, game trees

Binary Tree

The root has the data value 84.
There are 4 leaves in this binary tree: 13, 37, 50, 98.
This binary tree is not complete.
Binary Trees: Implementation

- One common implementation of binary trees uses nodes like a linked list does.
  - Instead of having a “next” pointer, each node has a “left” pointer and a “right” pointer.
- We could also use an array.

Binary Search Tree (BST)

- A binary search tree (BST) is a binary tree such that
  - All nodes to the left of any node have data values less than that node
  - All nodes to the right of any node have data values greater than that node
Using a BST

• How would you search for an element in a BST?

Graphs
Graphs

- **A graph** is a data structure that consists of a set of vertices and a set of edges connecting pairs of the vertices.
  - A graph doesn’t have a root, per se.
  - A vertex can be connected to any number of other vertices using edges.
  - An edge may be bidirectional or directed (one-way).
  - An edge may have a weight on it that indicates a cost for traveling over that edge in the graph.

- Applications: computer networks, transportation systems, social relationships
Undirected and Directed Graphs

Graphs in Ruby

```
inf = 1.0/0.0

graph = 
[ [ 0, 6, 7, 5 ],
 [ 6, 0, 4, inf ],
 [ 7, 4, 0, 3],
 [ 5, inf, 3, 0 ] ]
```
### An Undirected Weighted Graph

#### Vertices:
Pitt, Erie, Will, S.C., Harr, Scr, Phil.

#### Edges:
- Pitt to Erie: 10
- Pitt to Will: 7
- Pitt to S.C.: 8
- Pitt to Harr: 9
- Pitt to Scr: 11
- Pitt to Phil: 7
- Erie to Will: 12
- S.C. to Harr: 6
- S.C. to Scr: 4
- S.C. to Phil: 3
- Harr to Scr: 5
- Harr to Phil: 11
- Scr to Phil: 0

#### Weight Matrix:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0</strong></td>
<td>0</td>
<td>10</td>
<td>∞</td>
<td>8</td>
<td>7</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td><strong>1</strong></td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>7</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>∞</td>
<td>12</td>
<td>0</td>
<td>6</td>
<td>∞</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>0</td>
<td>9</td>
<td>4</td>
<td>∞</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>7</td>
<td>∞</td>
<td>∞</td>
<td>9</td>
<td>0</td>
<td>∞</td>
<td>11</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>∞</td>
<td>7</td>
<td>4</td>
<td>∞</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>∞</td>
<td>5</td>
<td>∞</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

---

### Original Graph

- Pitt to Erie: 12
- Pitt to S.C.: 8
- Pitt to Harr: 9
- S.C. to Harr: 7
- S.C. to Phil: 4
- Harr to Phil: 11
- Scr to Phil: 0

---

15110 Principles of Computing,
Carnegie Mellon University - CORTINA
The total costs of the shortest path from Pittsburgh to every other location using only edges from the original graph.

Graph Algorithms

- There are algorithms to compute the minimal spanning tree of a graph and the shortest paths for a graph.
  - We will see these later on in the semester.
- There are algorithms for other graph operations:
  - If a graph represents a set of pipes and the number represent the maximum flow through each pipe, then we can determine the maximum amount of water that can flow through the pipes assuming one vertex is a “source” (water coming into the system) and one vertex is a “sink” (water leaving the system)
  - Many more graph algorithms... very useful to solve real life problems.