



2007-04-24
Lecture #19
(Material on Test 3)

Today's Outline
Graphs

Graphs

Chapter 12 in your book

Similar to when we talked about trees, 1st we must talk about terminology for graphs

Vertices = nodes in our graph (similar to tree nodes, linked list nodes, etc.)

Arcs = directed lines connecting two vertices

Edges = undirected lines connecting two vertices

Lines = general term to refer to both Arcs and Edges

Graph = collection vertices that are connected by a collection of Lines

Graphs

Directed Graph = aka “digraph” ... collection of vertices that are connected by a collection of arcs

Real World Examples:

MapQuest or Google Maps ... roads could be one-way or go both directions

Airport flights, could be one-way or round-trip

Neural Network Artificial Intelligence (20 Q, Chess)

Graphs

Undirected Graph = collection of vertices that are connected by a collection of edges

Real World Examples:

United States Map ... each state touches 0 or many other states

Note: An Undirected graph is actually a “digraph”

Graphs

Why choose a graph over a tree or linked list?

- **different number of ways out of each vertice**
- **no real starting point or endpoint**
- **can go backwards, forwards, up, down, basically anywhere there is an edge**

Graphs

Adjacent Vertices = aka “neighbors”, or vertices with an edge that directly connects them

Path = sequence of vertices in which each vertex is “adjacent” to the next one

Note: Both Directed and UnDirected graphs have paths, but you must follow the arcs on the digraph

Graphs

Cycle = path consisting of at least 3 vertices that starts and ends with the same vertex

Loop = special case of a cycle where a single arc starts and ends on the same vertex

Graphs

Connected Vertices = 2 vertices are “connected” if there is a path between them

Connected Graph = there is a path from any one vertex to any other vertex in that graph ignoring direction

DiGraphs in particular have 2 additional breakdowns

Strongly Connected = taking into consideration the arc directions, there is a path from any one vertex to any other vertex

Weakly Connected = there is at least 2 vertices that are not connected

Graphs

Disjoint Graphs = 2 graphs are disjoint if they are not connected

InDegree of a vertex = number of arcs going into vertex

OutDegree of a vertex = number of arcs going out of vertex

Degree of a vertex = indegree + outdegree

Graphs

What kind of operations could you perform on a Graph?

- **Add Vertex (Ex: Add a City)**
- **Delete Vertex (Ex: Delete a City)**
- **Add Arc (Ex: Add a Road)**
- **Delete Arc (Ex: Delete a Road)**
- **Find Vertex (Ex: Find a City given a starting point)**
- **Traverse Graph (Ex: List all cities)**

Graphs

Traversals

1.) Depth First Traversal

- All a node's descendants are processed before moving to an adjacent node

Stack Algorithm:

- a.) Push first vertex onto the stack
- b.) Process top vertex, Pop Stack, Push all adjacent vertexes onto stack
- c.) Repeat Step b.) until stack is empty

Recall Queue is in LIFO (last in first out)

Graphs

Traversals

2.) Breadth First Traversal

- process all adjacent vertices before going to next level

Queue Algorithm:

a.) Enqueue first vertex

b.) Dequeue first vertex, process it, Enqueue all adjacent vertices

c.) Repeat b.) until Queue is empty

Recall Queue is in FIFO (first in first out)

Final Homework 6

Mini-MapQuest Program

- **Read a list of cities and one-way roads between them from a file**
- **Insert them into a graph using an Adjacency list**
- **User selects 2 cities**
 - **you use the Dijkstra's Shortest Path Algorithm to compute the shortest distance between them**