Announcements/Reminders:

- P4 due tonight

Last class:

- Graphs (cont'd)
  - Using Edge Representations
  - Searches/Traversals: Depth-First Search

Today:

- Graphs (cont'd)
  - Breadth-First Search
  - Dijkstra's Shortest Path Algorithm
Breadth-First Search (BFS)

What kind of questions can we answer?

Using a queue:
BFS Examples

BFS node visit order beginning at A:

Graph 1:

Graph 2:

BFS spanning tree starting at A:

Graph 1:

Graph 2:
Graph Algorithms

What kind of questions do we want to ask about graphs?

Cycle Detection

Path Detection
Finding Shortest Paths: Dijkstra's Algorithm

Problem:

Solution Strategy:

Example:

Priority Queue:  
Costs and predecessors:
Dijkstra's Algorithm (cont'd)

Pseudocode:

Given: Start node $s$

foreach node $u$ reachable from $s$:
    init $u$.dist to infinity
    init $u$.predecessor = null
    init $u$.visited = false

$PQ = \text{new priority queue}$
$PQ.insert(<0, s>)$

while $PQ$ is not empty:

    $<u$.dist, $u> = PQ.removeMin()$

    foreach unvisited neighbor $v$ of $u$:

        // if $v$.dist can be improved i.e.
        if $v$.dist $> u$.dist + cost($u, v$):

            // update $v$.dist to be the path thru $u$
            $v$.dist = $u$.dist + cost($u, v$)

            // set $v$'s predecessor
            $v$.predecessor = $u$

            insert $<v$.dist, $v> \text{ into } PQ$
            OR update if $v$ is already in $PQ$

        mark $u$ as visited

A finished node will never be visited again!