Project Orientation

Version Control / Subversion

Graphs

And other fun stuff!

Krysta Yousoufian

CSE 331 Section, 2/2/2012

With material from Marty Stepp and others

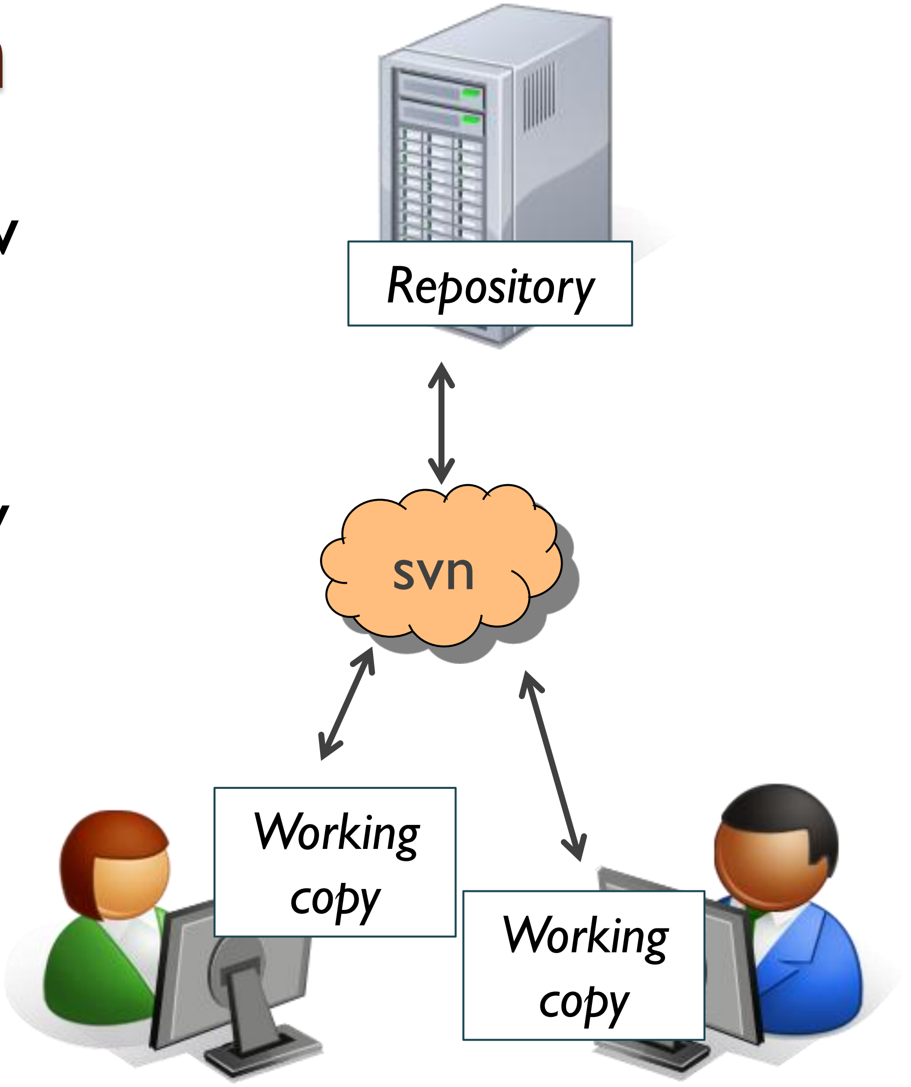
Announcements

- Pick up HW2 and solutions after class
  - Questions? Best to ask TA who graded that problem
  - Zach: questions 1, 4
  - Jackson: question 2
  - Laure: question 3
  - I'll try to answer questions, but I didn't grade
- Reading quiz: due 2am Friday (i.e. tonight)
- HW4: due Thursday, Feb. 9th
  - **Try to do a commit this weekend**
- Midterm: Monday, Feb. 13th

VERSION CONTROL

Organization

- Don't worry how repo stores files
- Don't try to edit repo files directly (outside SVN)



Setup Actions

1. Install version control software

2a. One person per team:

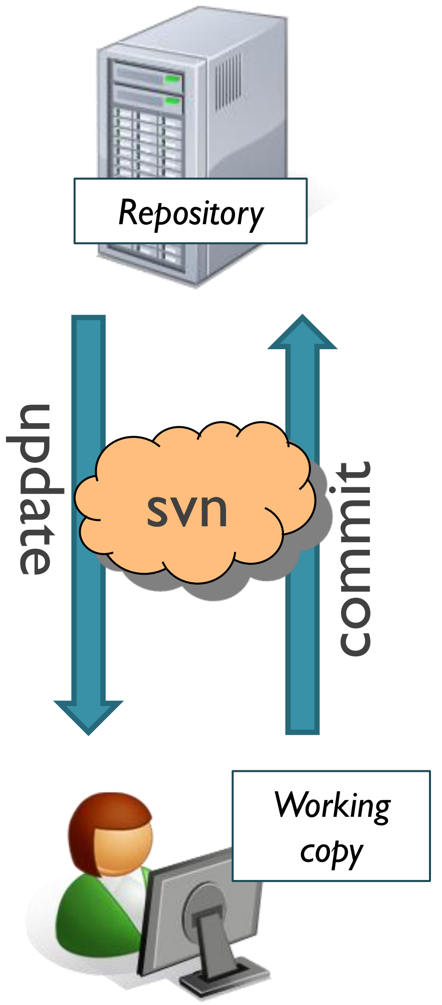
- Create the repository
- Import a new project

2a. Everyone else: checkout the repository

Common Actions

Everyday commands:

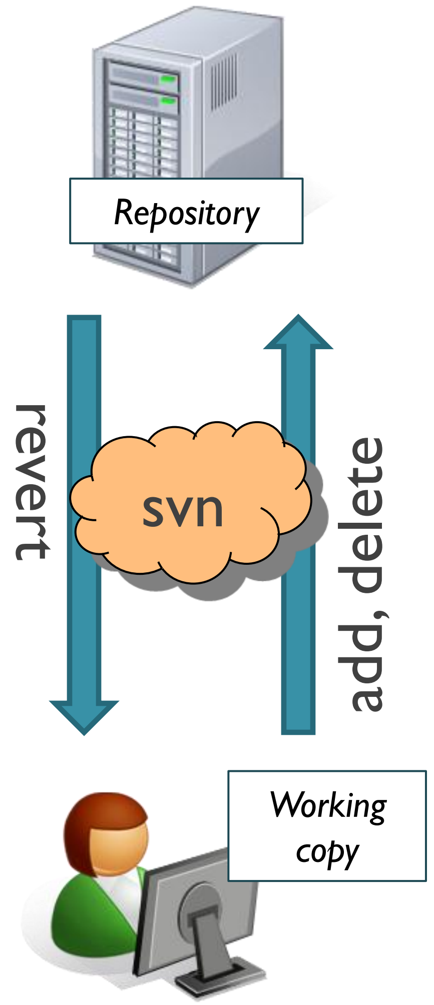
- Update
  - Merge others' changes *from* repo *into* your working copy
- Commit / checkin
  - Merge changes *into* repo *from* your working copy
  - May need/want to update first



Common Actions

Slightly less frequent commands:

- Add, delete
  - add or delete a file in repo
  - Local additions/deletions not propagated otherwise
- Revert
  - Erase your local changes to a file
- Resolve, diff, merge
  - Handle a conflict – two users editing the same code



Subversion

- One version control system
- Simple, free
- There are lots of others
  - Git, Mercurial, Hg, …
- Several ways to run:
  - Command-line
  - GUI (TortoiseSVN, NautilusSVN)
  - **Subclipse**: plugin for Eclipse
- Good reference:
  - http://svnbook.red-bean.com/

Using Subclipse

- Follow setup, checkout instructions on HW4
- Package Explorer > select project > right-click > Team
  - Commit
  - Update: "Update to HEAD"
  - etc.

Subclipse Demo

Very important!

- Do a commit this weekend!
- Some students had configuration issues last quarter
- This is how you'll submit your project
- Fix any issues now, not at 10:59pm Thursday

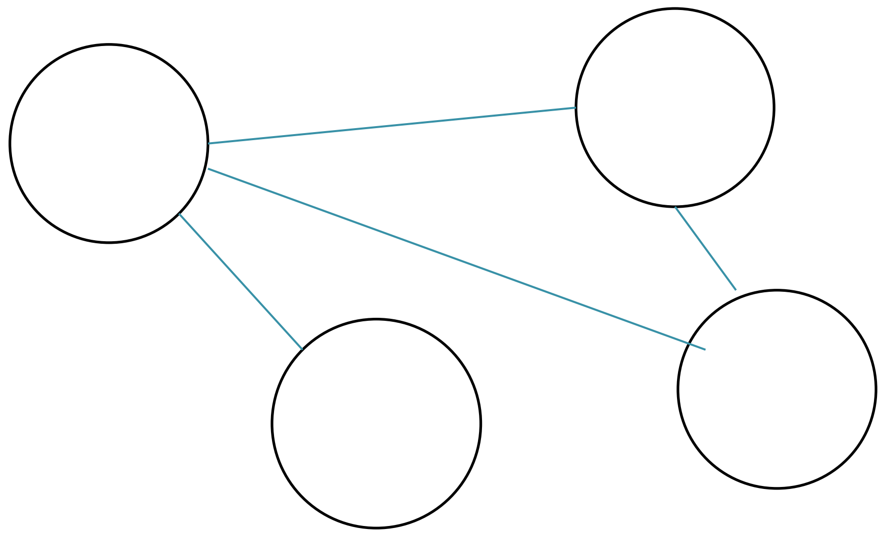
Commit Errors

- “Malformed network data error”
  - Make sure you have Subclipse v1.6
  - In Eclipse, go to Window -> Preferences -> Team -> SVN, and under "SVN Interface," change "JavaHL" to "SVNKit.
- Other errors?
  - If Google can’t solve it, post **precise** error message or screenshot on discussion board
  - If Google can solve it, post answer to discussion board

GRAPHS

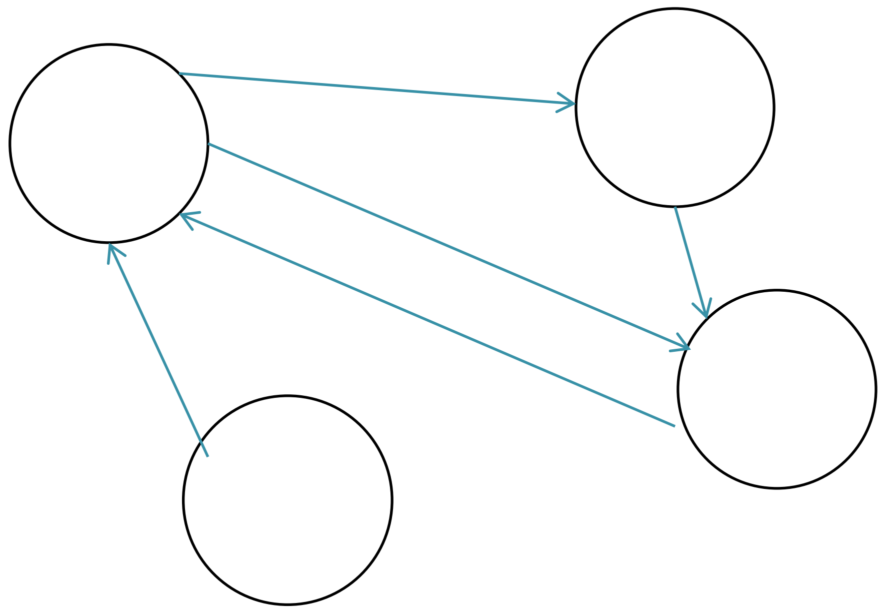
Graphs

- **Graph:** a collection of nodes and edges
- **Node:** a point on the graph
- **Edge:** connects two nodes



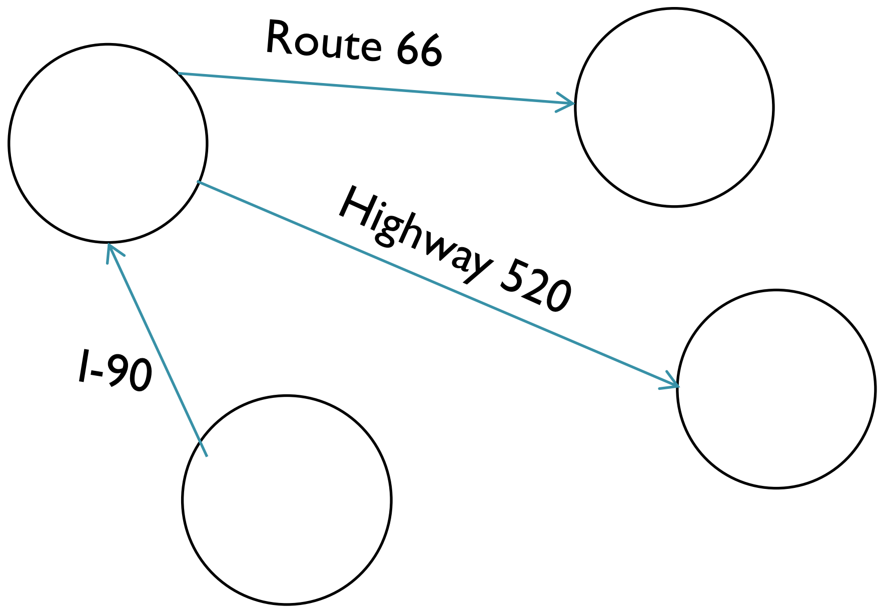
Directed Graphs

- Edges are one way
  - `e1 = <A, B>` is an edge from A to B
  - `e2 = <B, A>` is an edge from B to A
  - Graph can contain `e1, e2,` neither, or both



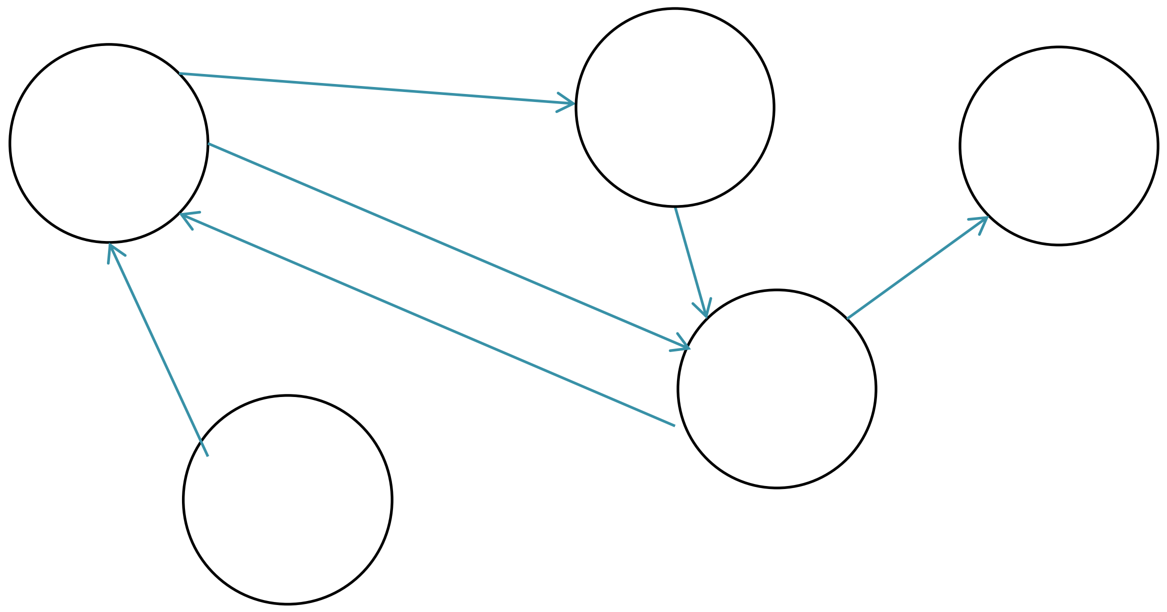
Labeled Graphs

- Label on every edge



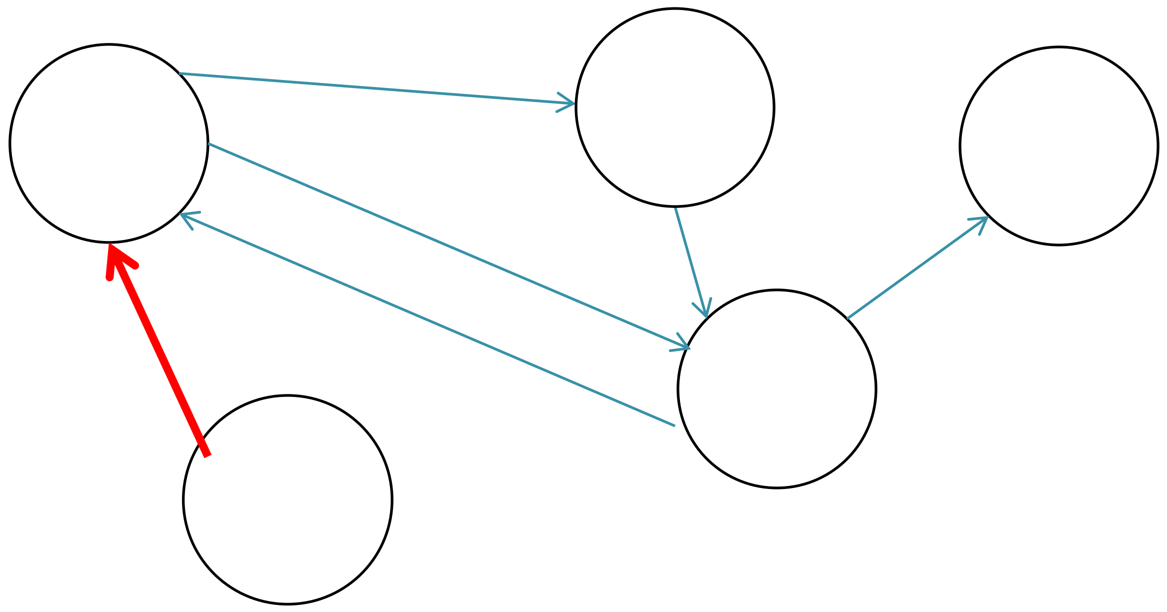
Path

- Formally: a sequence of edges `<n1,n2>, <n2,n3>, …, <n_i-1,n_i>`
- Informally: a route through the graph formed by following edges



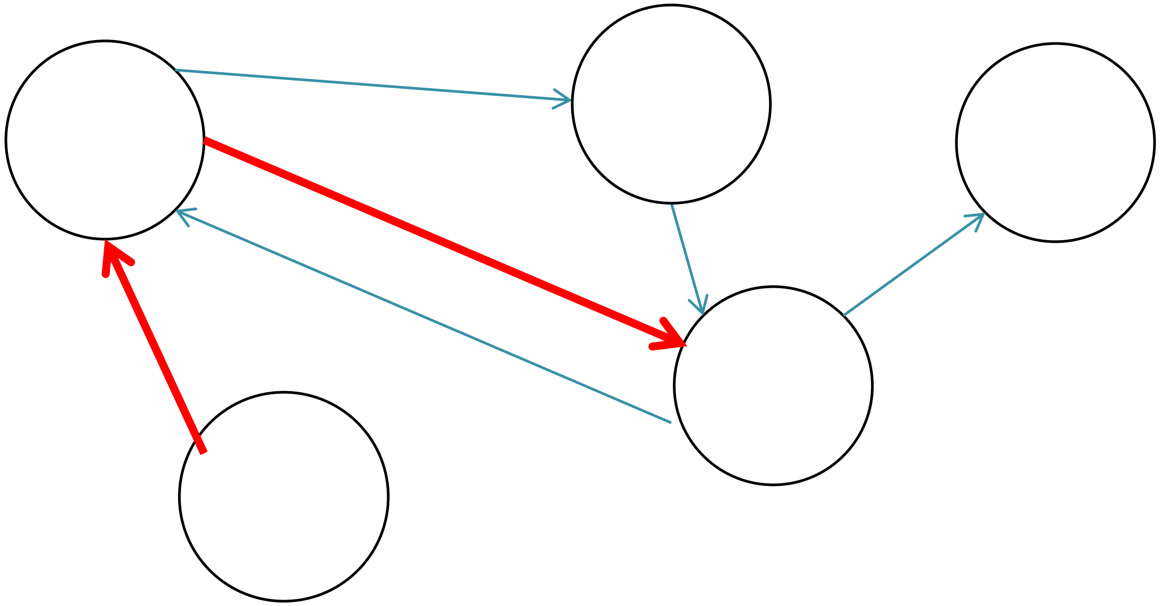
Path

- Formally: a sequence of edges `<n1,n2>, <n2,n3>, …, <n_i-1,n_i>`
- Informally: a route through the graph formed by following edges



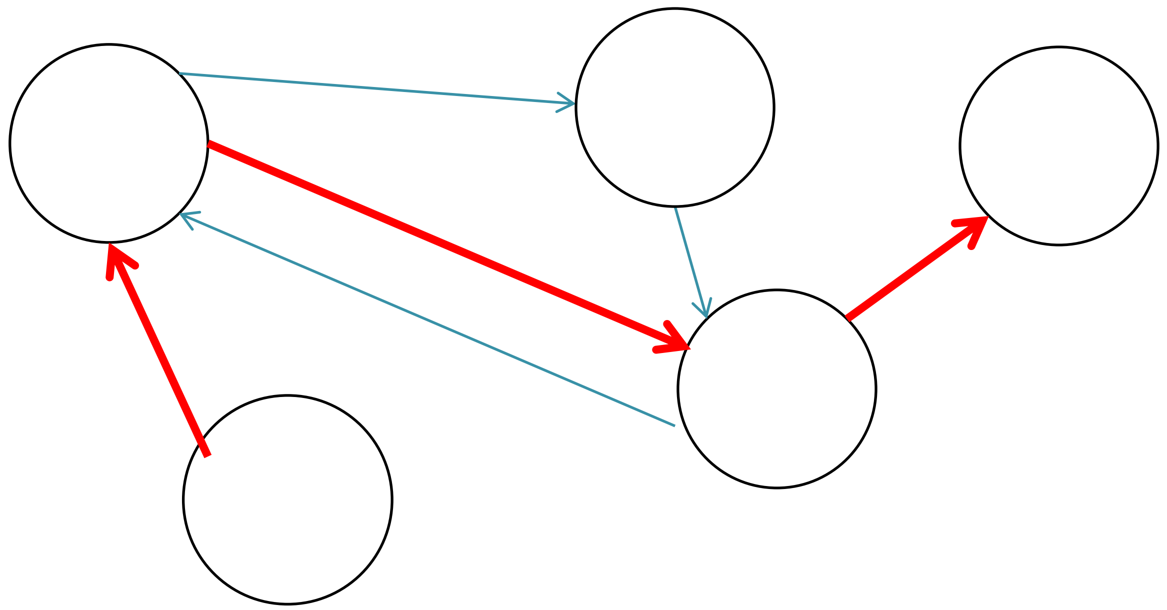
Path

- Formally: a sequence of edges `<n1,n2>`, `<n2,n3>`, …, `<n_i-1,n_i>`
- Informally: a route through the graph formed by following edges



Path

- Formally: a sequence of edges `<n1,n2>`, `<n2,n3>`, …, `<n_i-1,n_i>`
- Informally: a route through the graph formed by following edges



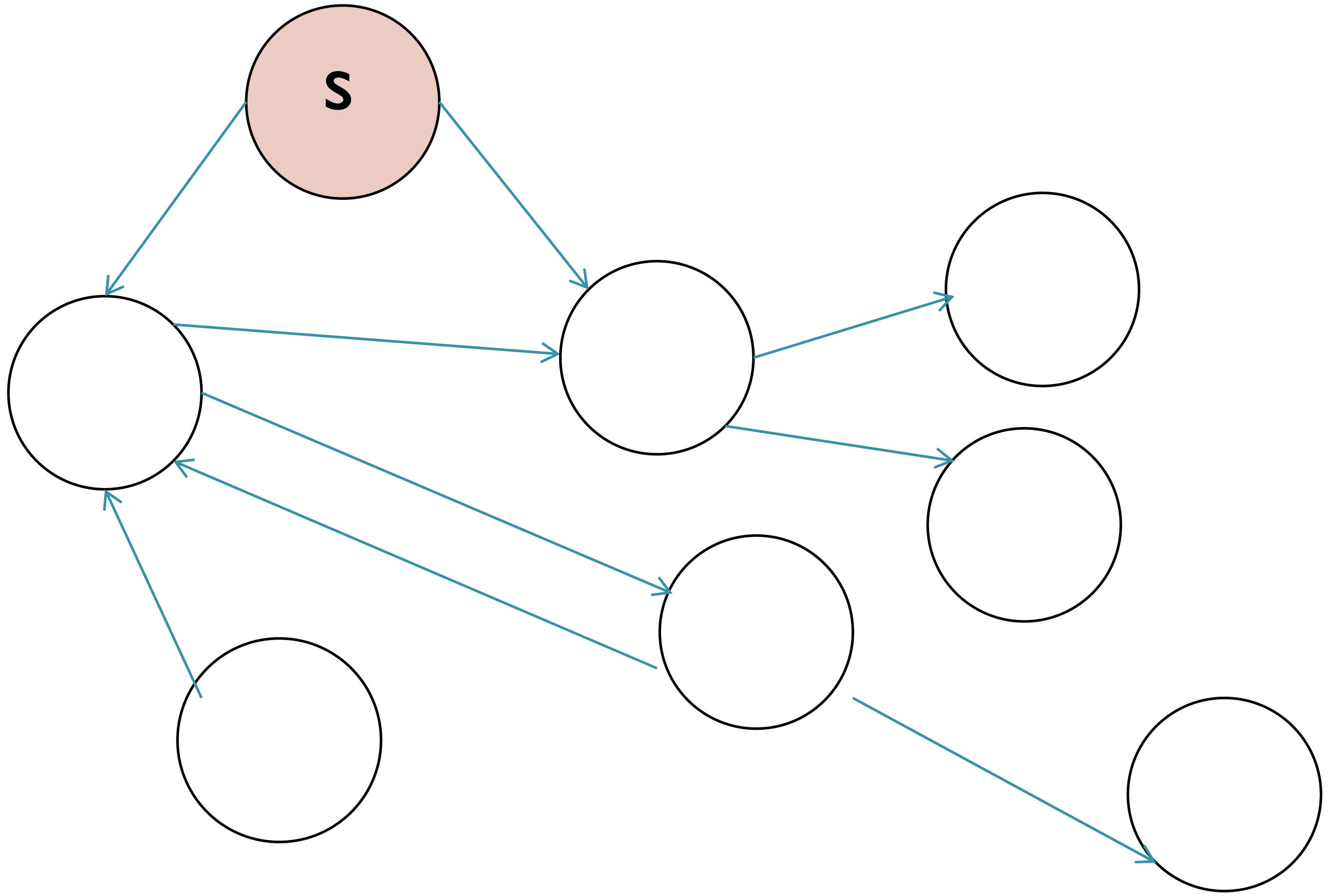
What can you do with a graph?

- What nodes are reachable (have a path) from some node?
- What is the shortest path (fewest edges) between two nodes?
- If edge labels represent costs, what is the minimum-cost path between two nodes?
  - Not necessarily the path with fewest edges!

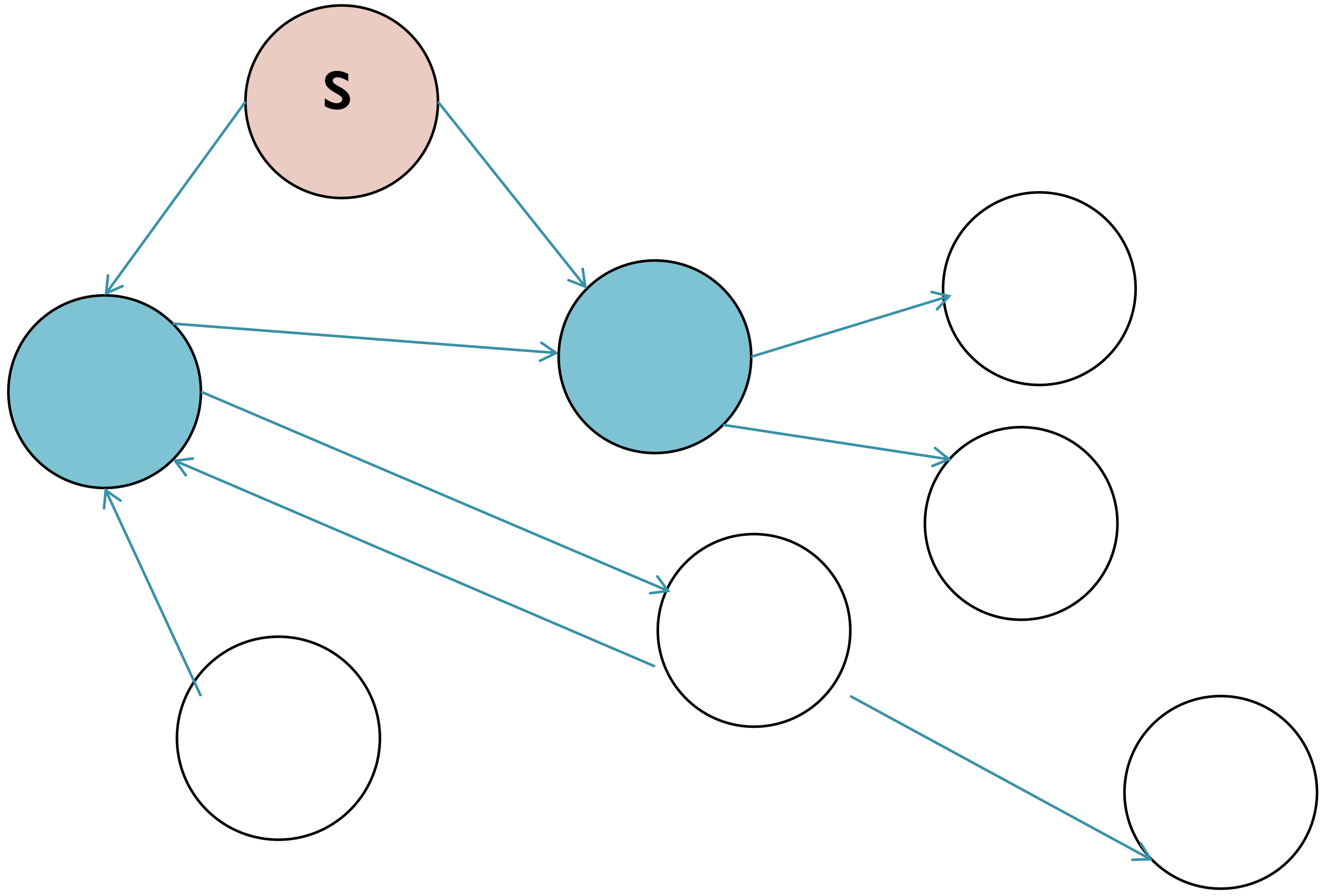
Breadth-first search (BFS)

- An algorithm for traversing a graph
  - Aside: contrast with depth-first search (DFS)
- Given a starting node s:
  - Visit all neighbors of s (direct edge from s)
  - Visit neighbors of neighbors
  - Visit neighbors of neighbors of neighbors
  - …
- If searching for a path to some node t, stop when you find t

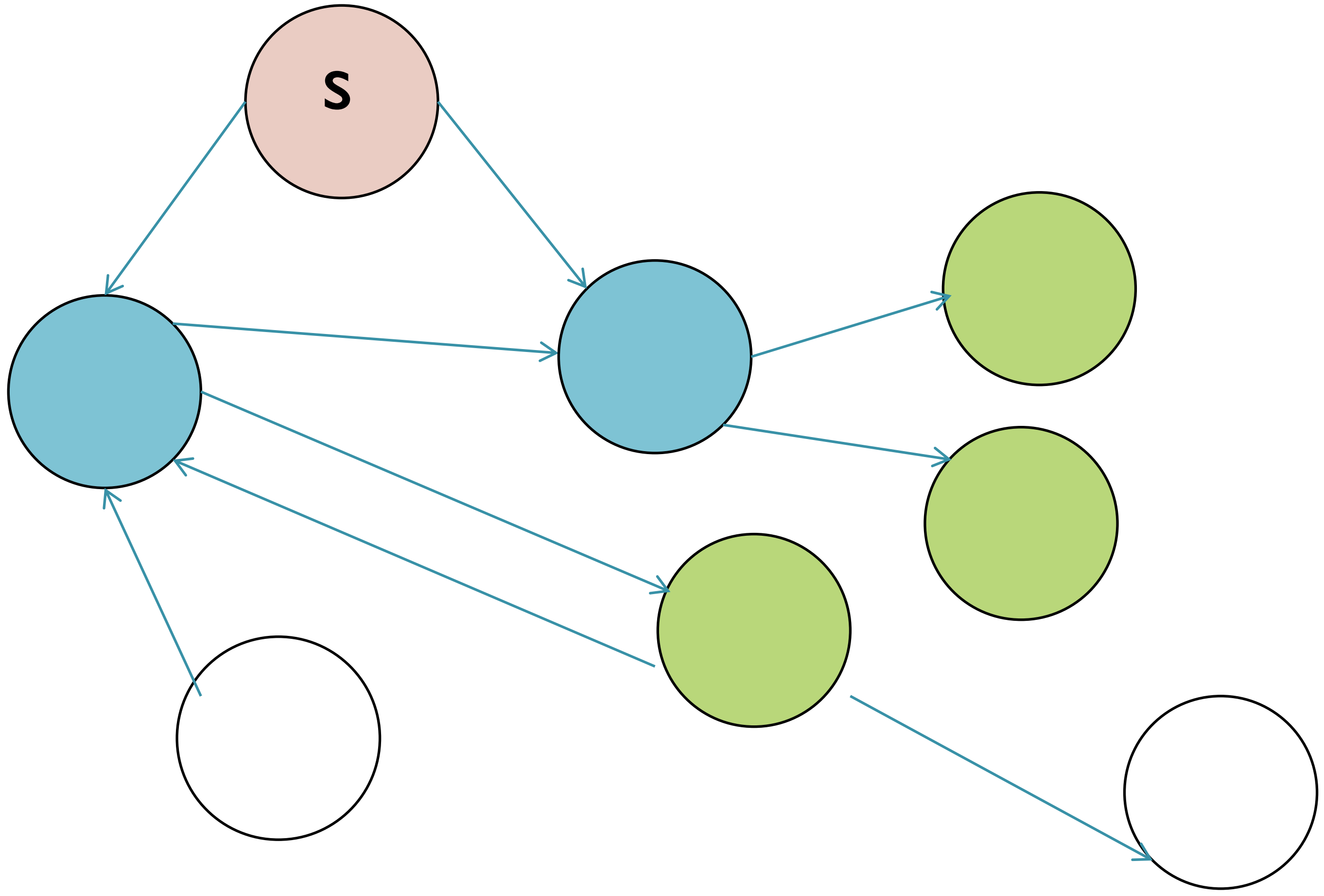
BFS



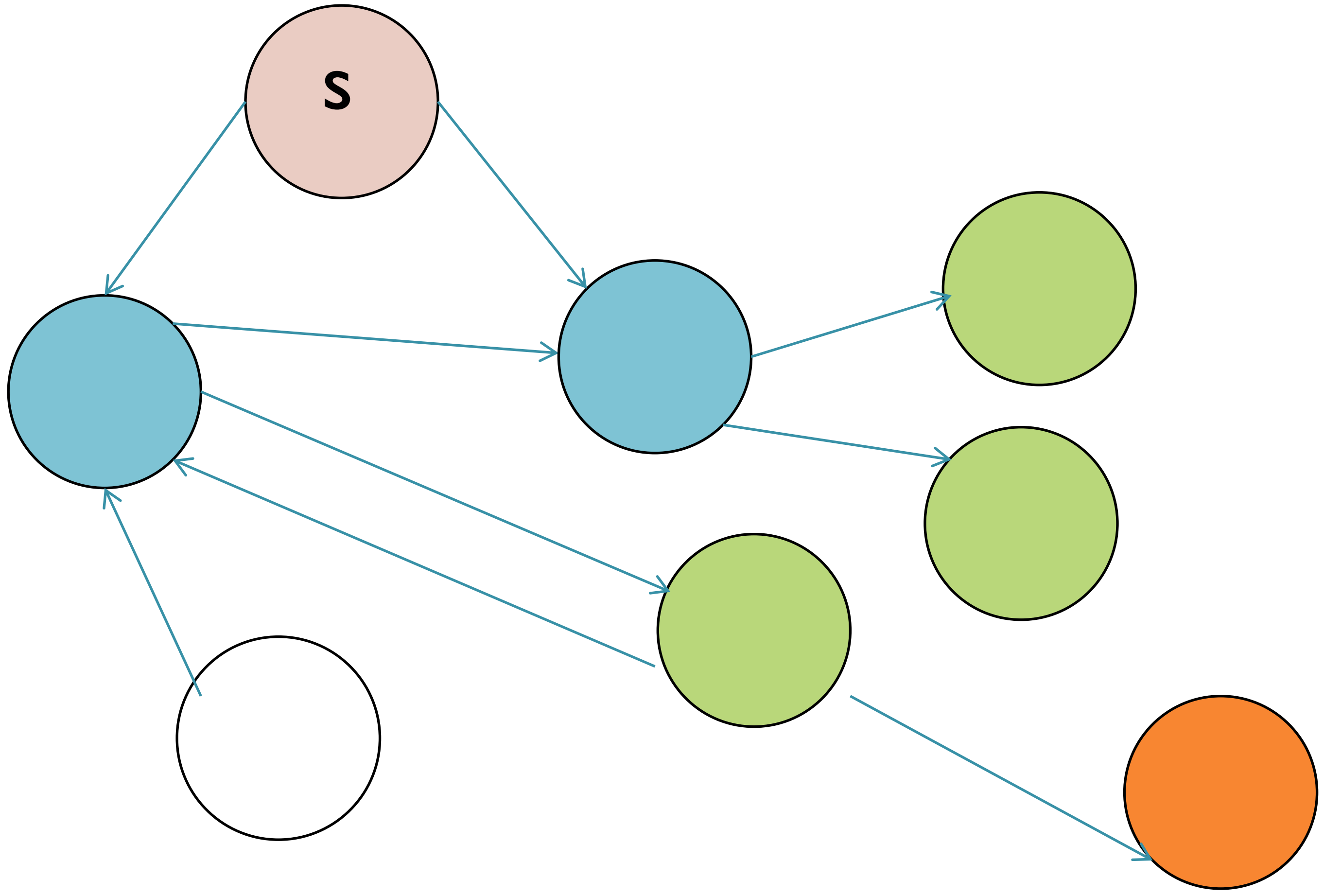
BFS



BFS



BFS

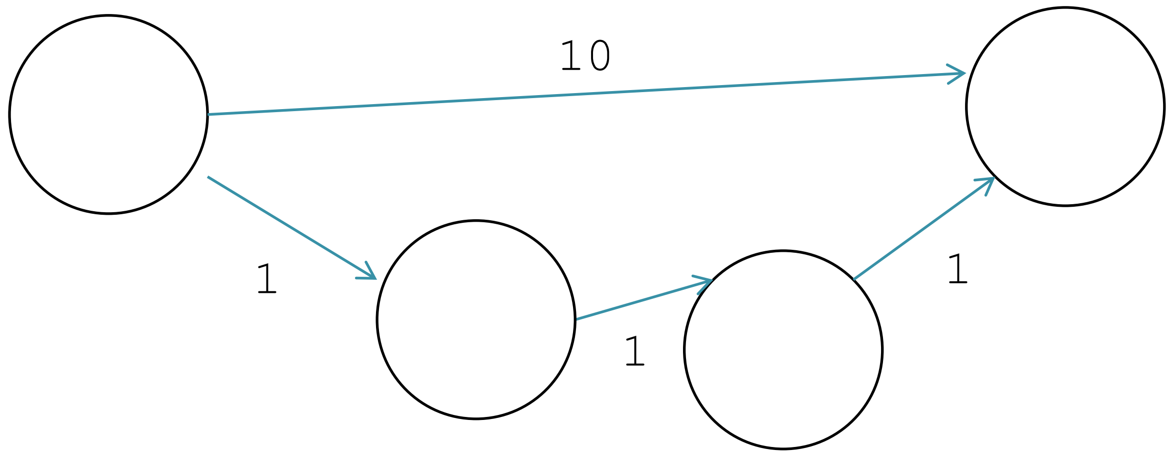


Dijkstra's algorithm

- Labels on edges are "costs"
    - Money, distance, time, etc.
- Find path from s to t of lowest cost
- Not necessarily the shortest path

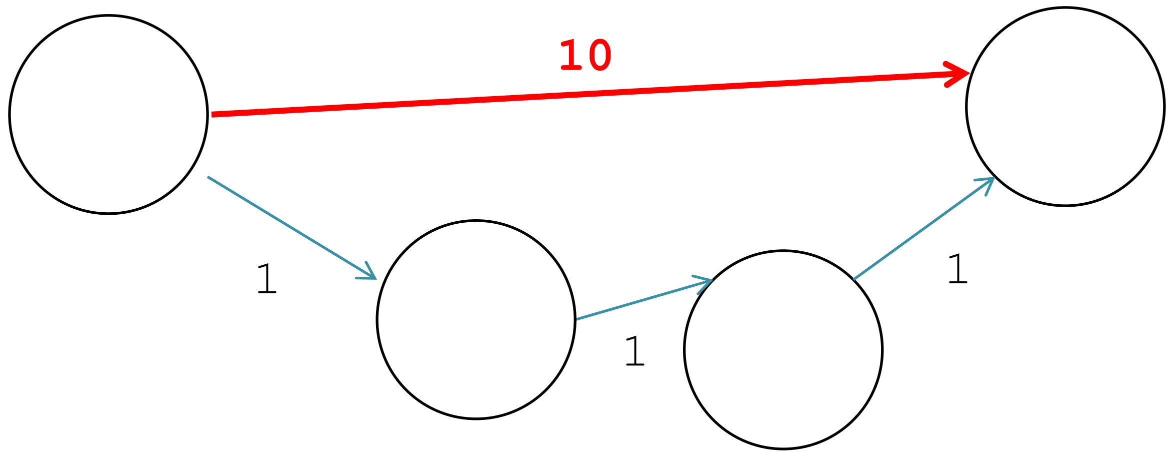
Dijkstra's algorithm

- Labels on edges are "costs"
  - Money, distance, time, etc.
- Find path from s to t of lowest cost
- Not necessarily the shortest path



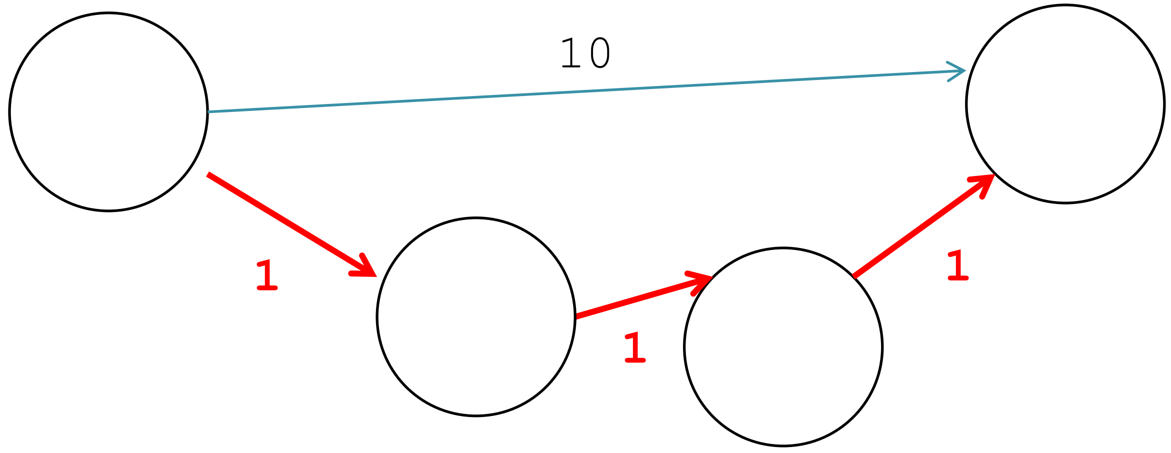
Dijkstra's algorithm

- Labels on edges are "costs"
  - Money, distance, time, etc.
- Find path from s to t of lowest cost
- Not necessarily the shortest path



Dijkstra's algorithm

- Labels on edges are "costs"
  - Money, distance, time, etc.
- Find path from s to t of lowest cost
- Not necessarily the shortest path



Ring buffer

- Implementation of queue