A graph contains nodes and edges

Seattle

San Francisco

Salt Lake City

Dallas

Chicago
+ 350 students in no romantic and/or sexual relationship
From: “Chains of Affection: The Structure of Adolescent Romantic and Sexual Networks”, American Journal of Sociology, by Peter Bearman of (Columbia), James Moody (Ohio State), and Katherine Stovel (U. of Washington);
Graphs

• A graph can be thought of as either of:
  – a collection of edges
    • Each edge represents some relationship
  – for each node, a collection of neighbors
    • The neighbors are those connected by an edge
Operations on a graph

Creation:
• Create an empty graph

Querying:
• Look up a node: Does it exist? What are its neighbors?
• Look up an edge (= a pair of nodes): does it exist? (You know the nodes it connects.)
• Iterate through the nodes or edges

Modification:
• Add/remove a node
• Add/remove an edge
networkx Graph Library

• Used in Homework 4

```python
import networkx as nx

G = nx.Graph()
G.add_node(1)
G.add_node(2)
G.add_edge(1, 2)
print(G.nodes())
print(G.edges())
```
Installing networkx Graph Library

• Used in Homework 4

• To install networkx:
  – Open up "canopy" (can search for it)
  – Then select tools-> package manager
  – Click on "available Packages" and search for "networkx".
  – The button next to networkx will say something like "free“, click on that button.
  – It will ask if you want to install, say yes.
  – To check if you have it installed, type “import network x” in IDLE (nothing should happen, but if it is NOT installed you will get an error message).
import networkx as nx
import matplotlib.pyplot as plt

g = nx.Graph() # Creates a graph

g.add_node(1)   # Adds node 1
g.add_node(3)
g.add_node(2)
print g.nodes()

g.add_edge(1, 2) # Adds edge from node 1 to node 2
g.add_edge(1, 3)
print g.edges()

assert len(g.nodes()) == 3
assert len(g.edges()) == 2

nx.draw(g) # Draw the graph
plt.show() # Show the graph in a separate window