Minimum Spanning Trees

Reading

- Section 9.6

Outline

- Minimum Spanning Trees
- Prim's Algorithm
- Kruskal's Algorithm

- Extra:Distributed Shortest-Path Algorithms

A File Sharing Problem

- Say a bunch of users want to distribute a file amongst themselves.
- Between each pair of users there is a cost of sending information.
- How can we most efficiently get the file to every user?

A Kevin Bacon Problem

- Everyone knows that every actor is 6 degrees from Kevin Bacon.
- Instead of trying to minimize the distance, what if we asked how many films are needed so that every actor is connected to every other actor.

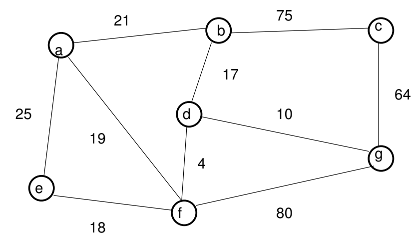
Spanning Trees

- Can think of each of these problems as pruning edges of a graph to arrive at a tree.
- *Every* connected graph contains a tree stucture. (Probably many different trees.)
  - From definition: A tree is an acyclic graph. Remove an edge from every cycle in a graph, and you have a tree.

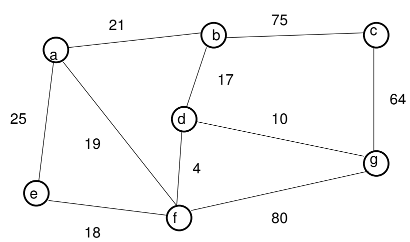
Minimum Spanning Trees

- Input: A graph G=(V, E) with weights w_e.
- Output: A set of edges $T \subset E$, such that G'=(V, T) is a tree and the sum of edge weights in T is minimized.
- Alternately: Remove as much weight as possible while still remaining connected.

A Spanning Tree



A Minimum Spanning Tree



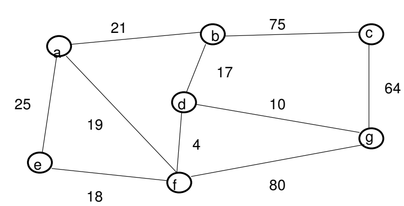
Recall Dijkstra's Algorithm

- For each node, maintain best known shortest path
- At each iteration, node with least value is at its true minimum, so fix that node and update neighbors.

Prim's Algorithm

- For each node, maintain best known edge cost connecting that node to tree.
- At each iteration, node with least cost can be added to tree. Then update neighbors.

Prim's algorithm example



	known	cost
A		
B		
C		
D		
E		
F		
G		

Runtime

- Exact same as Dijkstra's algorithm
- One iteration for each vertex, and each iteration must find min over all vertices, so O(____).
- Also have to update costs so total of ___.
- If we use a heap to find the minimum, then each of ___ updates takes O(____), and each find takes O(____), so __________.

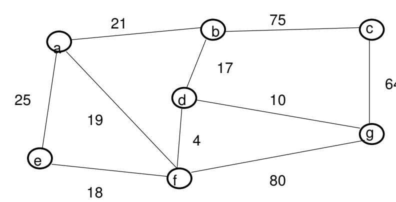
Question

- Which is better, heap or no heap?

Kruskal's Algorithm

- Idea: We want small weight edges that don't form a cycle.
- So start with edge of smallest weight and add edges of increasing weight that don't cause cycle, until we have connected the graph.

Kruskal's algorithm example



Implementing Kruskal's

- Needed to repeatedly get least cost edge. Use…?
- Needed to check if two vertices are already in same component. Use…?
- How do we know when to stop?

Kruskal's pseudocode

- Kruskal's Min Spanning Tree(G):
  - Build a heap H from edges
  - While # tree edges < N-1
    - (u, v) = deleteMin(H)
    - If find(v) != find(u)
      - Mark (u, v) as tree edge
      - Union(v, u)

Runtime

- Loop could execute ____ times, and heap operations take ______.
- Same as Dijkstra's with a heap, but much faster in practice. Why?

Fun Topic: Internet Routing

- Experiment: Open a command line an type 'traceroute www.yahoo.com'
- How does your computer know how to get to Yahoo's computer?

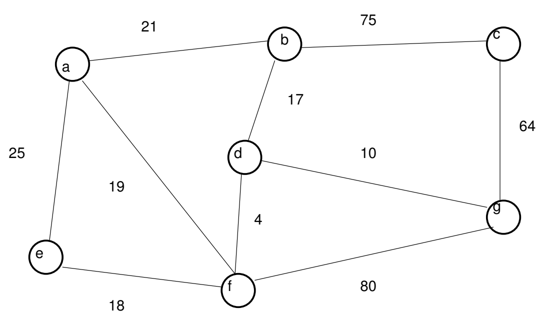
The Internet Graph

- Computers are nodes, edges are links.
- Weight links by latency, bandwidth, load, economic factors, etc.
- Then run Dijkstra's algorithm, and you'll know best route to all other computers.
- Problem?

Solution: Link-State

- Every computer knows its neighbors. Send that neighbor information to every other node.
- Flooding: Each node sends received packets to all neighbors. Eventually all nodes reached.

Example



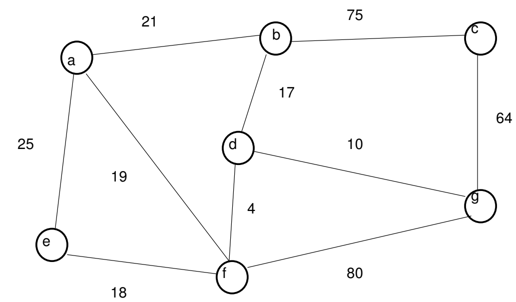
Problems with Link-State

- Problems?

Another Solution: Distance Vector

- Instead of trying to share complete network topology, just tell neighbors about routes.
- A node advertises the best known value of the shortest path from it to all other nodes.

Example



Comparison

- Distance Vector:
  - less data sent
  - Intuitively seems right that you shouldn't need a map of the whole network
  - But: can easily get inconsistent views of the network, leading to very bad behavior, like infinite loops
- Distance vector used for 20 years in Internet. Link-State common now.

Scale

- Not used in small networks, like our campus network
- Not used across backbone providers—network topology is a corporate secret!

Moral

- Graphs algorithms have:
  - Very rich theory
  - Compelling applications in almost every domain of CS, and many beyond.