Design and Analysis of Algorithms I

Contraction Algorithm

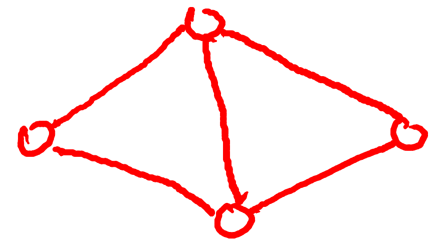
Overview

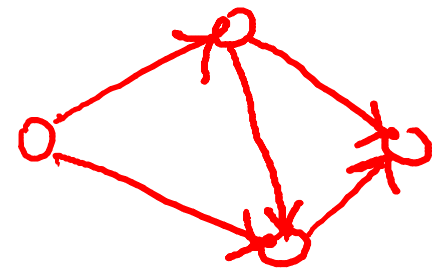
Goals for These Lectures

- Further practice with randomized algorithms
  - In a new application domain (graphs)
- Introduction to graphs and graph algorithms

Also: "only" 20 years ago!

Graphs





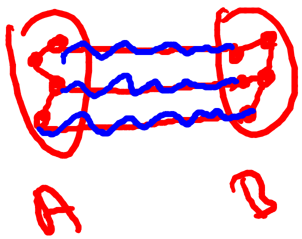
Two ingredients

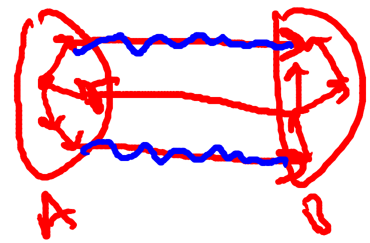
- vertices aka nodes (V)
- edges (E) = pairs of vertices
  - can be undirected [unordered pair] or directed [ordered pair] (aka arcs)

Examples: road networks, the Web, social networks, precedence constraints, etc.

Cuts of Graphs

Definition: a cut of a graph (V,E) is a partition of V into two non-empty sets A and B.





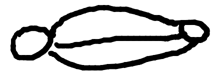
Definition: the crossing edges of a cut (A,B) are those with:
- one endpoint in each of (A,B) [undirected]
- tail in A, head in B [directed]

Tim Roughgarden

Roughly how many cuts does a graph with n vertices have?

- ○ n
- ○ n^2
- ○ 2^n
- ○ n^n

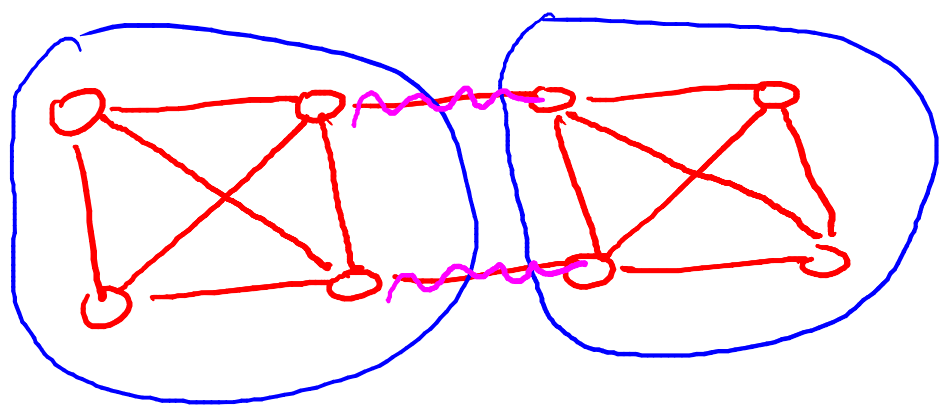
The Minimum Cut Problem

Input: an undirected graph $G = (V, E)$.
[parallel edges  allowed]
[see other video for representation of input]

Goal: compute a cut with fewest number of crossing edges. (a *min cut*)

What is the number of edges crossing a minimum cut in the graph shown below?

- 1
- 2
- 3
- 4



A Few Applications

- identify network bottlenecks / weaknesses
- community detection in social networks
- image segmentation
  - input = graph of pixels
  - use edge weights [(u,v) has large weight ⟺ "expect" u,v to come from same object]

hope: repeated min cuts identifies the primary objects in picture.