Design and Analysis of Algorithms I

Contraction Algorithm

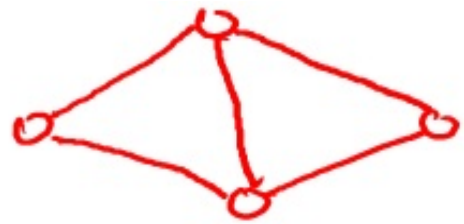
Overview

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



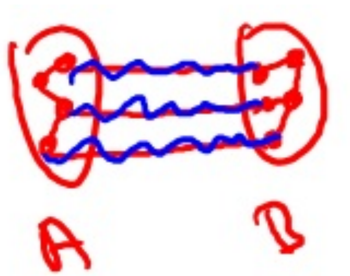
Two ingredients

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

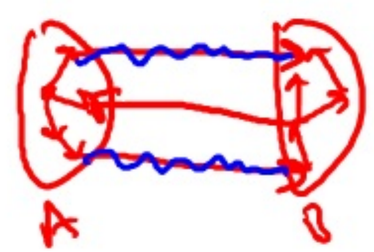
Examples: road networks, the Web, social networks, precedence constraints, etc.

Cuts of Graphs

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



[undirected]



[directed]

Definition: the crossing edges of a cut (A, B) are those with:

- the one endpoint in each of (A, B) [undirected]
- tail in A, head in B [directed]

Ter
ver

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

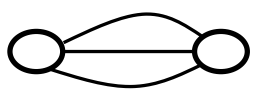
- n
- n^2
- 2^n
- n^n

Actually 2^n - 2, for non-empty rule.

two sets, in total n vertices,
in each set, with or without a node.
but at least one node in each set.

The Minimum Cut Problem

- INPUT: An undirected graph G = (V, E).

[Parallel  edges allowed]

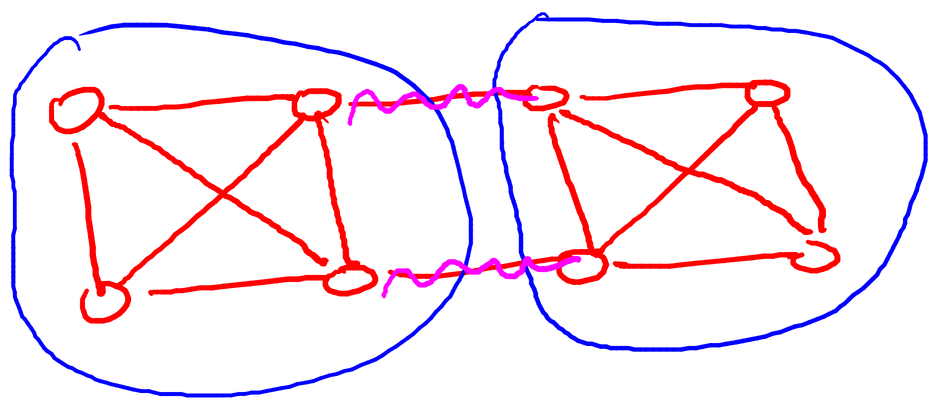
[See other video for representation of the input]

- GOAL: Compute a cut with fewest number of crossing edges. (a min cut)

Ter
ver

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

- 1
- 2
- 3
- 4



A Few Applications

- indentify 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 some object]

  <u>hope</u>: repeated min cuts identifies the primary objects in picture.