Contraction Algorithm

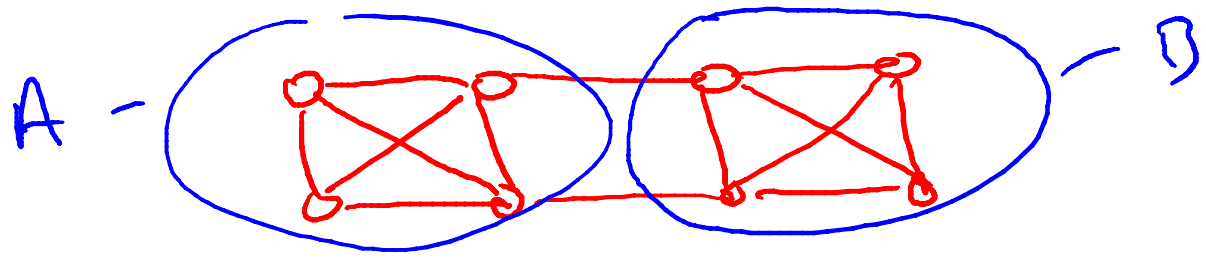
The Analysis

Design and Analysis of Algorithms I

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)



Random Contraction Algorithm

[due to Karger, early 90s]

While there are more than 2 vertices:

- pick a remaining edge (u,v) uniformly at random
- merge (or "contract") u and v into a single vertex
- remove self-loops

return cut represented by final 2 vertices.

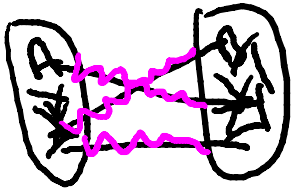
The Setup

Question: what is the probability of success?

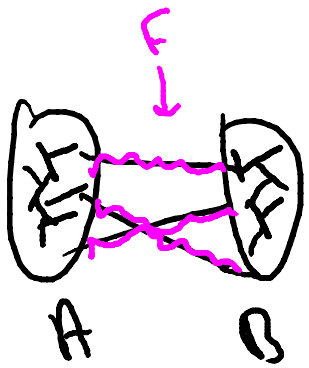
Fix a graph $G=(V,E)$ with n vertices, m edges.

Fix a minimum cut (A,B).

Let k = # of edges crossing (A,B). (call these edges F)



What Could Go Wrong?



① Suppose an edge of F is contracted at some point $\Rightarrow$ algorithm will not output (A, B).

② Suppose only edges inside A or inside B get contracted $\Rightarrow$ algorithm will output (A, B).

Thus: $\Pr[\text{output is } (A,B)] = \Pr[\text{never contracts an edge of } F]$

Let S_i = event that an edge of F contracted in iteration i.

Goal: compute $\Pr[\neg S_1 \wedge \neg S_2 \wedge \neg S_3 \wedge \cdots \wedge \neg S_{n-2}]$.

What is the probability that an edge crossing the minimum cut (A, B) is chosen in the first iteration (as a function of the number of vertices n, the number of edges m, and the number k of crossing edges)?

$\Pr[S_1] = \frac{\#\text{ of crossing edges}}{\#\text{ of edges}}$

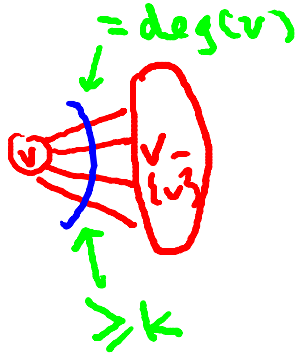
$= k/m$

- ○ k/n
- ○ k/m
- ○ k/n^2
- ○ n/m

The First Iteration

Key Observation: degree of each vertex is at least k.

degree = # of incident edges



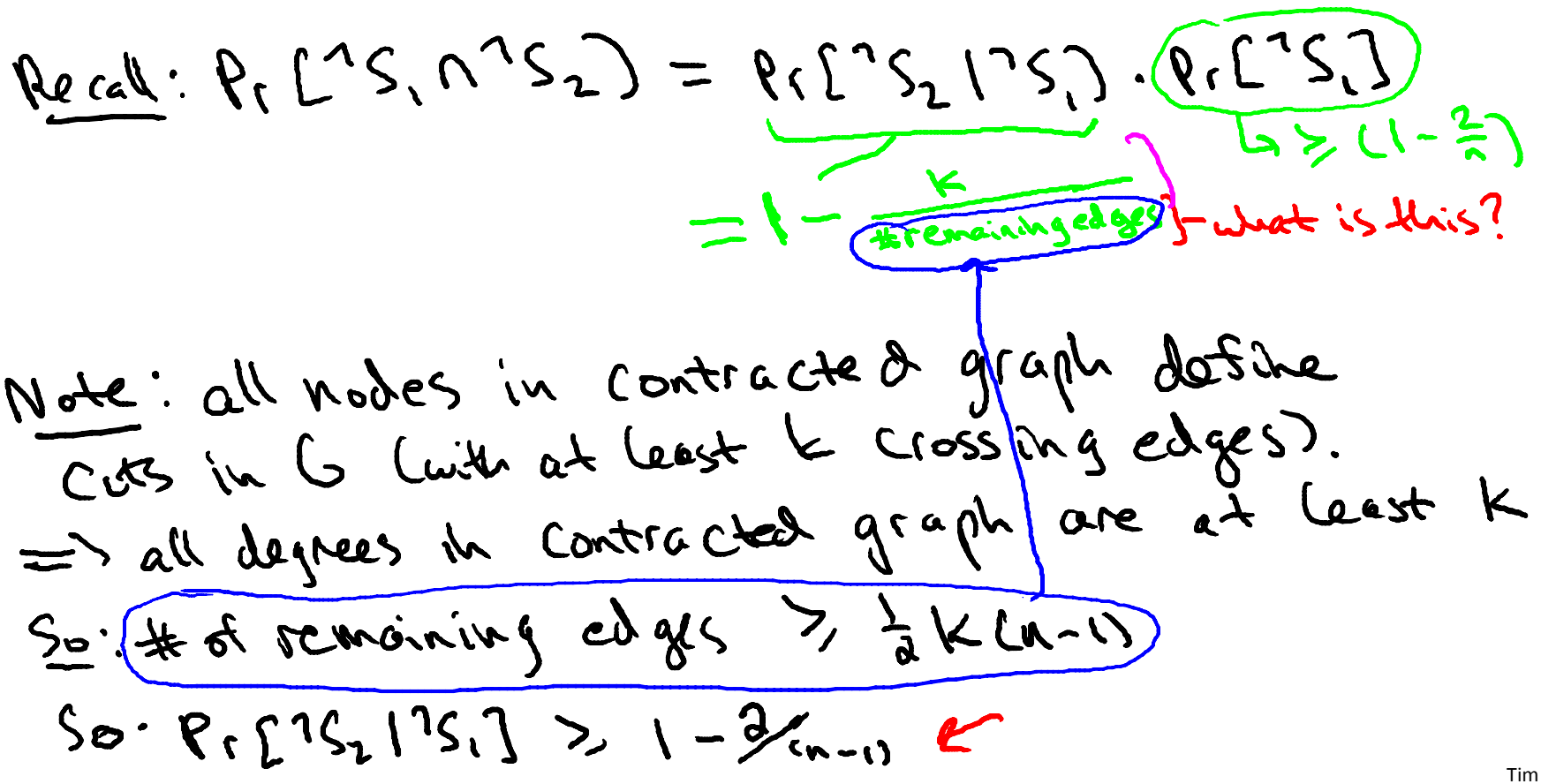
Reason: each vertex v defines a cut $(\{v\}, V-\{v\})$.

Since $\sum_v \text{degree}(v) = 2m$, we have $m \geq \frac{kn}{2}$

($\sum_v \text{degree}(v) \geq kn$)

Since $Pr[S_1] = \frac{k}{m}$, $Pr[S_1] \leq \frac{2}{n}$.

The Second Iteration



All Iterations

In general:

$$\Pr[\neg S_1 \wedge \neg S_2 \wedge \neg S_3 \wedge \cdots \wedge \neg S_{n-2}] =$$

$$\Pr[\neg S_1] \cdot \Pr[\neg S_2 \mid \neg S_1] \cdot \Pr[\neg S_3 \mid \neg S_1 \wedge \neg S_2] \cdot \cdots \cdot \Pr[\neg S_{n-2} \mid \neg S_1 \wedge \cdots \wedge \neg S_{n-3}]$$

$$\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\left(1 - \frac{2}{n-2}\right) \cdots \left(1 - \frac{1}{n-(n-4)}\right)\left(1 - \frac{1}{n-(n-3)}\right)$$

$$= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdot \cdots \cdot \frac{2}{4} \cdot \frac{1}{3} = \frac{2}{n(n-1)} \geq \frac{1}{n^2}.$$

Problem: low success probability! (But: non-trivial)

recall $\propto 2^n$ cuts!

Repeated Trials

Solution: run the basic algorithm a large number N times, remember the smallest cut found.

Question: how many trials needed?

Let T_i = event that the cut (A, B) is found on the ith try.

$\Rightarrow$ by definition, different T_i's are independent

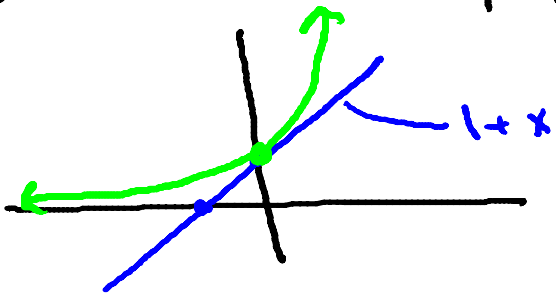
So: $Pr[\text{all } N \text{ trials fail}] = Pr[\neg T_1 \wedge \neg T_2 \wedge \cdots \wedge \neg T_N]$

$= \prod_{i=1}^{N} Pr[\neg T_i]$ (by independence!) $\leq \left(1 - \frac{1}{n^2}\right)^N$

Repeated Trials (con'd)

$\Pr[\text{all trials fail}] \leq \left(1 - \frac{1}{n^2}\right)^N$

Calculus fact: $\forall$ real numbers x, $1 + x \leq e^x$.



So: if we take $N = n^2$, $\Pr[\text{all fail}] \leq \left(e^{-\frac{1}{n^2}}\right)^{n^2} = \frac{1}{e}$

If we take $N = n^2 \ln n$, $\Pr[\text{all fail}] \leq \left(\frac{1}{e}\right)^{\ln n} = \frac{1}{n}$.

Running time: polynomial in n and m but slow ($\Omega(n^2 m)$)

But: can get big speedups (to roughly $O(n^2)$) with more ideas.