Oct 7

Randomized algorithms

Deterministic algorithms : for a given input, same behaviour
same output.

Randomized algorithm : • takes some random decisions (not arbitrary)
• Output can be random
• running time can be random

{ Output needs to be correct with high probability.
Running time needs to be small with high probability

Example: randomized quick sort
randomized second minimum.

Randomized algorithms are often faster and simpler than the deterministic ones.

There are examples, where the best known deterministic algorithm takes exponential time, but there is a randomized algorithm with polynomial time.

Why should we accept an algorithm which can give wrong answers at times ?

- probability of a wrong answer can be made as small as you want (trade off with time)

- Even if your algorithm is deterministic, there can be a hardware error

Approximate Median:

Given n numbers, and a parameter ε, find a number whose

$$\text{rank} \in \left[\frac{n}{2}(1-\varepsilon), \frac{n}{2}(1+\varepsilon) \right]$$

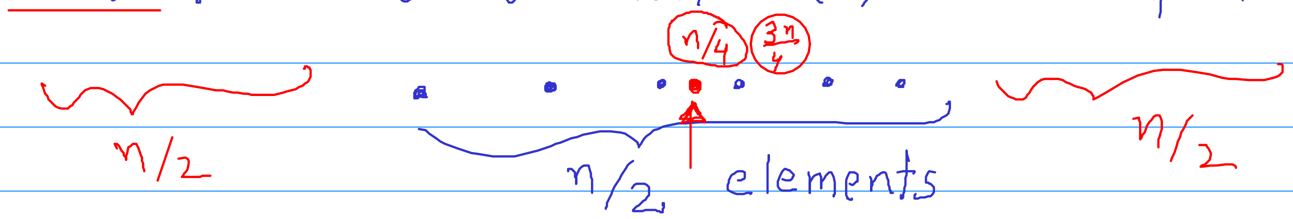
$n = 100 \quad \varepsilon = 1/5 \qquad \text{rank} \in [40, 60]$

Best deterministic algorithm:

Exact median : $O(n)$

Approximate median : $\text{rank} \in \left[\frac{n}{3}, \frac{2n}{3}\right]$

Claim : better than $O(n)$ is not possible.



If you see only $n/2$ elements, the best guarantee you can give is an element with rank between $n/4$ & $3n/4$.

Can we do better than $O(n)$?

Randomized algorithm $\approx O(\log n)$

Idea: Random sampling

Can judge many properties from a small sample.

Algorithm:

Select k numbers uniformly randomly (with replacement)

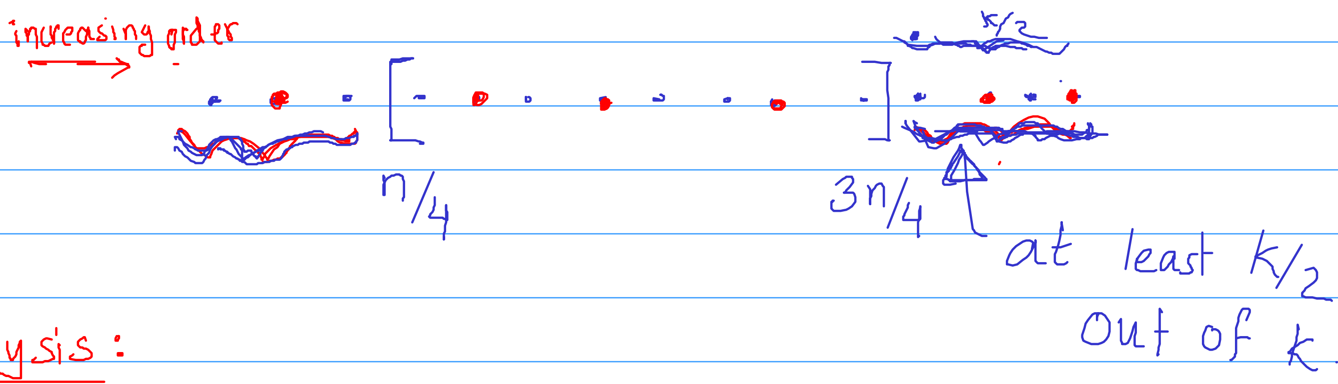
Output the median of these k numbers.

Want: Rank of the output number $\in \left[\frac{n}{2}(1-\varepsilon), \frac{n}{2}(1+\varepsilon)\right]$

with probability $1 - 1/n$.

What should be k?

Let's take $\varepsilon = 1/2$.



Analysis:

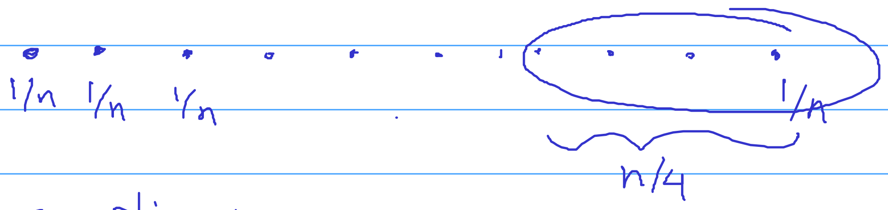
Probability that the median of the k selected numbers has rank $> 3n/4$.

$=$ probability that at least $k/2$ selected numbers have rank $> 3n/4$

Oct 11

Let $a_1, a_2, \dots, a_k$ be the k sampled numbers.

$$\left[\text{Probability that } a_i \text{ has rank} \geq \frac{3n}{4}\right] = \frac{n}{4} \times \frac{1}{n} = \frac{1}{4}$$



Equivalent question:

tossing a coin k times

where $\Pr[\text{heads}] = 1/4$, $\Pr[\text{tails}] = 3/4$

$$\Pr[\text{at least } k/2 \text{ heads}]$$

$$= \sum_{j=k/2}^{k} \binom{k}{j} \cdot \left(\frac{1}{4}\right)^j \cdot \left(\frac{3}{4}\right)^{k-j}$$

$$\leq \left(3/4\right)^{k/2}$$

$$\Pr\left[\text{output's rank is not in } [n/4, 3n/4]\right] \leq 2 \cdot \left(3/4\right)^{k/2}$$

take $k = 6 \log n$

$$= 2 \cdot \left(3/4\right)^{3 \cdot \log n}$$

$$\leq 2 \cdot \left(\frac{1}{2}\right)^{\log n}$$

$$= \frac{2}{n}$$

$$\Pr\left[\text{output's rank} \in [n/4, 3n/4]\right] \geq 1 - 2/n.$$

What should be k in terms of acceptable error ε and probability of success $1-\delta$?

$$k = O\left(\frac{1}{\varepsilon^2} \log 1/\delta\right)$$ HW

Probability bound

$$P = \sum_{j=k/2}^{k} \binom{k}{j} \left(\frac{1}{4}\right)^{j} \left(\frac{3}{4}\right)^{k-j}$$

independent of n

Binomial theorem : for some $x \in (0,1)$

$$\left(\frac{1}{4} + \frac{3}{4}x\right)^k = \sum_{j=0}^{k} \binom{k}{j} \left(\frac{1}{4}\right)^{j} \left(\frac{3}{4}\right)^{k-j} x^{k-j}$$

$$\geq \sum_{j=k/2}^{k} \binom{k}{j} \left(\frac{1}{4}\right)^{j} \left(\frac{3}{4}\right)^{k-j} x^{k-j}$$

$x^{k/2}, x^{k/2-1}, \ldots, x^0$

Use $x^r \geq x^{k/2}$ for $r \leq k/2$,

$$\geq x^{k/2} \cdot \sum_{j=k/2}^{k} \binom{k}{j} \left(\frac{1}{4}\right)^{j} \left(\frac{3}{4}\right)^{k-j}$$

$$= x^{k/2} \cdot P$$

$$\Rightarrow \quad P \cdot x^{k/2} \leq \left(1/4 + 3/4\, x\right)^k$$

$$P \leq \frac{\left(1/4 + 3/4\, x\right)^k}{x^{k/2}} = \left[\frac{\left(1/4 + 3/4\, x\right)^2}{x}\right]^{k/2}$$

minimize this

Bound holds for every $x \in (0,1)$

Apply calculus to find the best value of x.

$$p \leq (3/4)^{k/2}$$

Without replacement $\binom{n}{k}$ subsets are equally prob.

$$\sum_{j \geq k/2} \frac{\binom{n/4}{j}\binom{3n/4}{k-j}}{\binom{n}{k}}$$

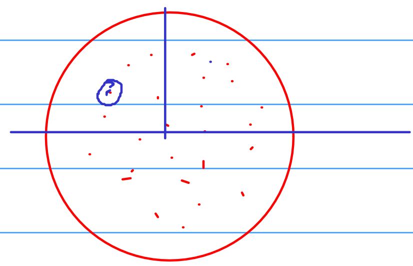
HW

Why do we want high probability like $1 - 1/n$

$$\left(\frac{9}{10}\right)^{\ell} \qquad \left(1 - \frac{1}{n^2}\right)^n \approx 1 - \frac{1}{n}$$

Choosing 1 out of n uniformly randomly

$\equiv$ Choosing $\log_2 n$ random bits

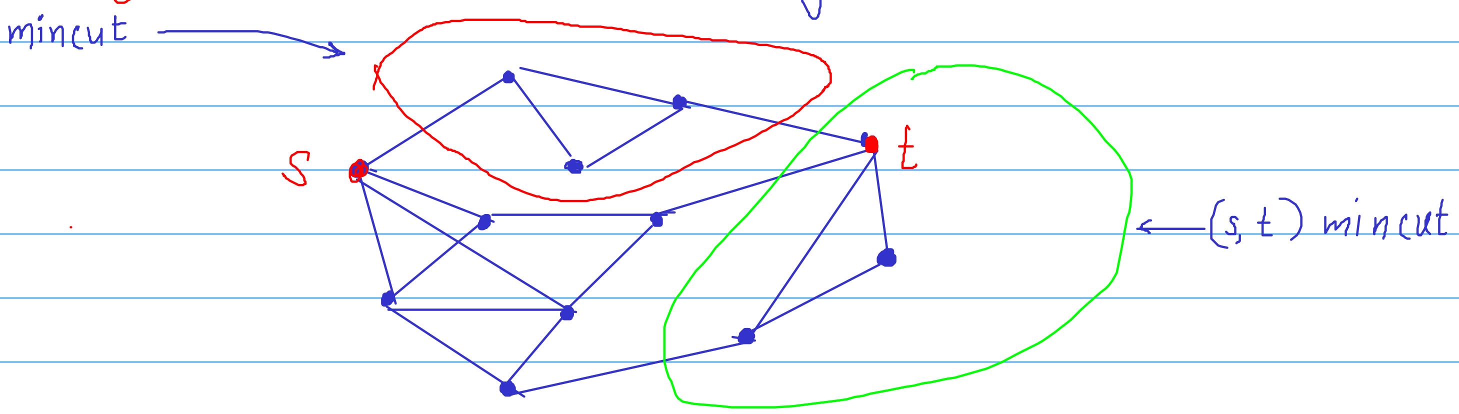


Global Minimum Cut

Cut : partition of vertices into two sets

say, $V = A \cup B$

cut edges : edges connecting A to B.



Minimum Cut : Given an undirected graph, find a cut with minimum number of cut edges

(S, t)-minimum cut: (max flow)

Given an undirected graph with two special vertices s, t

find a cut which separates s and t and has minimum number of edges.

$O(mn)$

- Minimum cut can be solved by using the (s, t)-minimum cut algorithm $n-1$ times. HW

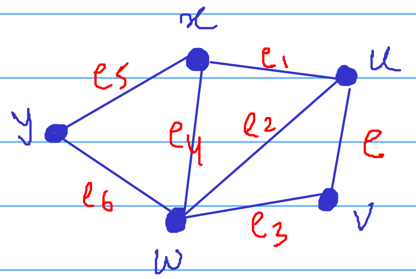
$O(mn^2)$

Karger 1992 : a simple randomized algorithm for minimum cut. $O(n^2 \log^c n)$

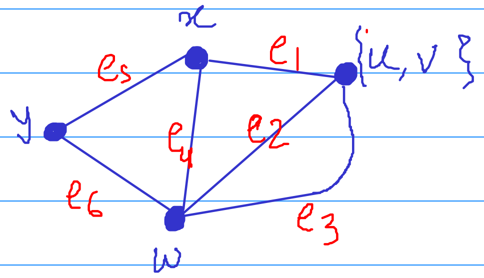
Contraction of an edge $e = (u,v)$

→ delete e and any parallel edges to e.
→ combine u and v to form a single vertex.

(might create parallel edges)



Contract e

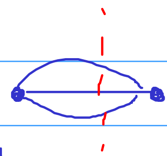


G $\qquad$ G/e

Mincut

Input : Multigraph G (no self loops)

Output : a cut in G

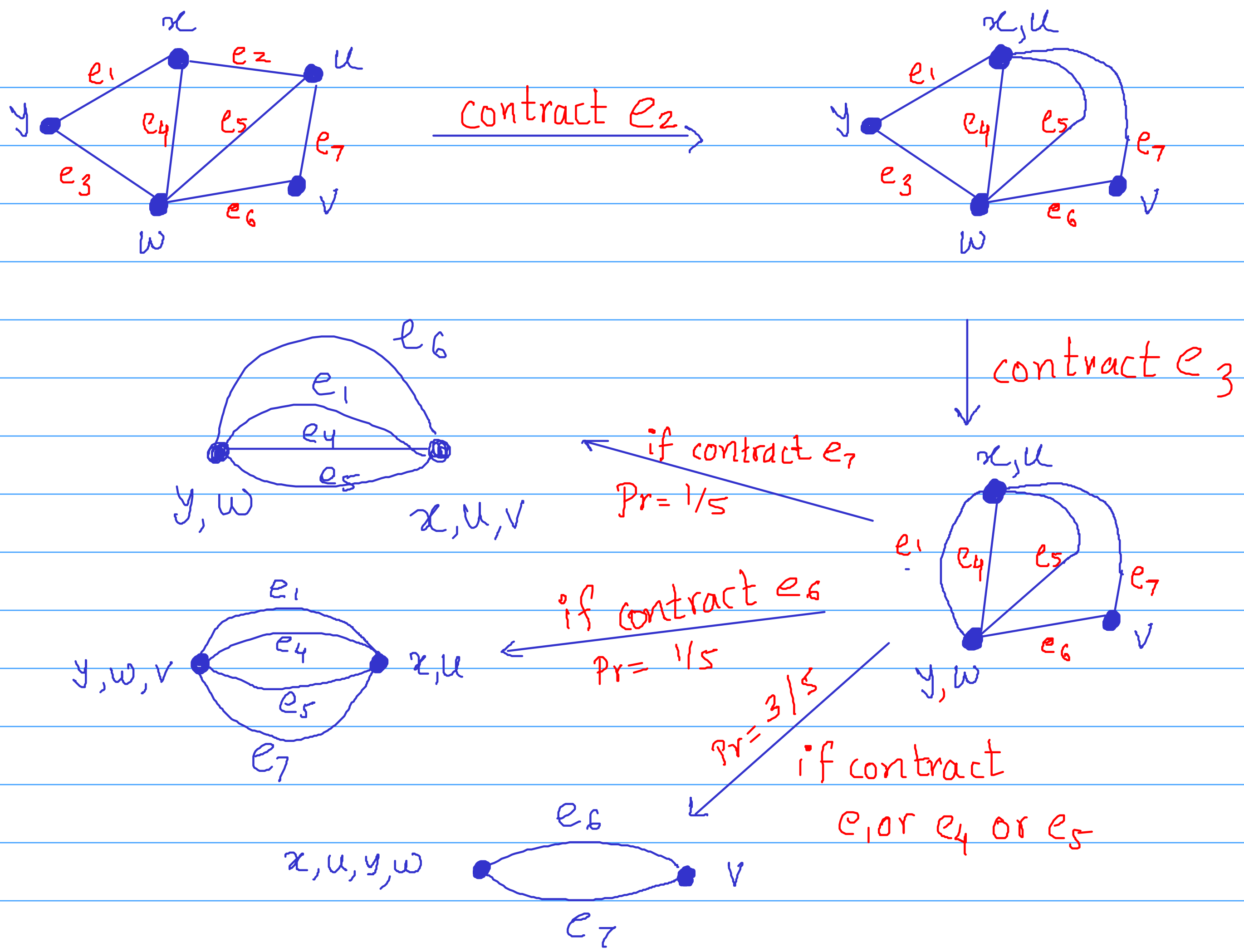


1. If G has only two vertices, output the only possible cut

2. else Pick an edge randomly uniformly from $G \to e$

3. $G \leftarrow G/e$ (contract e)

4. return Mincut (G)



In the last step, with probability $3/5$ we get a cut with 2 edges
with probability $2/5$ we get a cut with 3 edges

Observation: Any cut in G/e is also a cut in G, with the same cut edges.

Thus, the final output is a valid cut in G.

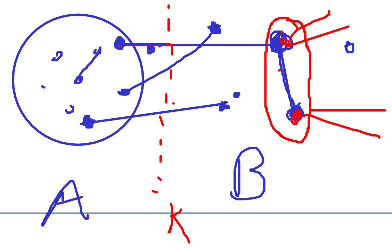
Which cut will be output? $2^{n-1}-1$ cuts

Is every cut equally likely to be output?

A cut survives if none of its edges get contracted.

smaller the cut, more chances of survival

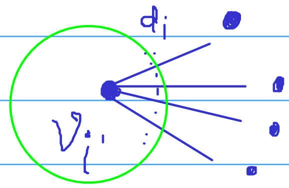
Oct 14



Let's fix a minimum cut (A,B)

what is the probability that (A,B) will be output?

Let k be the number of cut edges in (A,B)



$\Pr[\text{cut } (A,B) \text{ survives the first contraction}]$

$$= \frac{m-k}{m} = 1 - \frac{k}{m} \geq 1 - \frac{2}{n}$$

$$m = \frac{1}{2}\sum_{i=1}^{n} d_i$$

Obs: $d_i \geq k$

$$\Rightarrow m = \frac{1}{2}\sum d_i \geq \frac{1}{2}kn$$

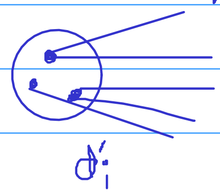
$$\Rightarrow \frac{2}{n} \geq \frac{k}{m}$$

Observation: Each contraction reduces the number of vertices by 1.

Given that (A,B) survived the first contraction, probability that it will survive second contraction

$$= 1 - \frac{k}{m'} \geq 1 - \frac{2}{n'} = 1 - \frac{2}{n-1}$$

$$\because m' = \frac{1}{2}\sum_{i=1}^{n'} d_i' \geq \frac{1}{2}k(n') \Rightarrow \frac{k}{m'} \leq \frac{2}{n'}$$



d_i' is the size of a cut in the current graph.

Every cut in the current graph is also a cut in the original graph with same cut edges $\Rightarrow d_i' \geq k$

$\Pr[(A,B) \text{ survives first two contractions}]$

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

$$\boxed{\Pr(S_1 \text{ and } S_2) = \Pr(S_1) \cdot \Pr(S_2 \mid S_1)}$$

$\Pr[(A,B) \text{ survives first } n-2 \text{ contractions}]$

$$= \Pr(S_1 \text{ and } S_2 \text{ and } \cdots \text{ and } S_{n-2})$$

$$= \Pr(S_1) \cdot \Pr(S_2 \mid S_1) \cdot \Pr(S_3 \mid S_1, S_2) \cdots \Pr(S_{n-2} \mid S_1, S_2 \cdots S_{n-3})$$

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

$$= \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{n-4}{\cancel{n-2}} \cdot \frac{n-5}{\cancel{n-3}} \cdots \frac{1}{3}$$

$$= \frac{2 \cdot 1}{n(n-1)} = \frac{2}{n(n-1)}$$

Probability that output is a minimum cut

$$\geq \frac{2}{n(n-1)} \cdot \text{number of min cuts}$$

$$\geq 2/n(n-1)$$

Boosting the success probability

Repeat the algorithm k times

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

$$\Pr[\text{success in at least one trial}] \geq 1 - \left(1 - \frac{2}{n(n-1)}\right)^k$$

$$1 - x \leq e^{-x}$$

$$\Pr[\text{success}] \geq 1 - e^{-2/n(n-1) \cdot k}$$

Choose $k = n^2 \Rightarrow \Pr \geq 1 - e^{-2}$

$$k = n^2 \log n \Rightarrow \Pr \geq 1 - e^{-2 \log n} = 1 - \frac{1}{n^2}$$

$$\text{Running time} = O(n^2 \cdot m \cdot \log n)$$

There are clever implementations of the contraction approach with running time $O(n^2 \log^c n)$

Motwani Raghavan
Chapter 10

Questions:

① Does success probability increase if you are happy with an approximate minimum cut ?

② Can you use contraction approach for minimum (s,t) - cut ?