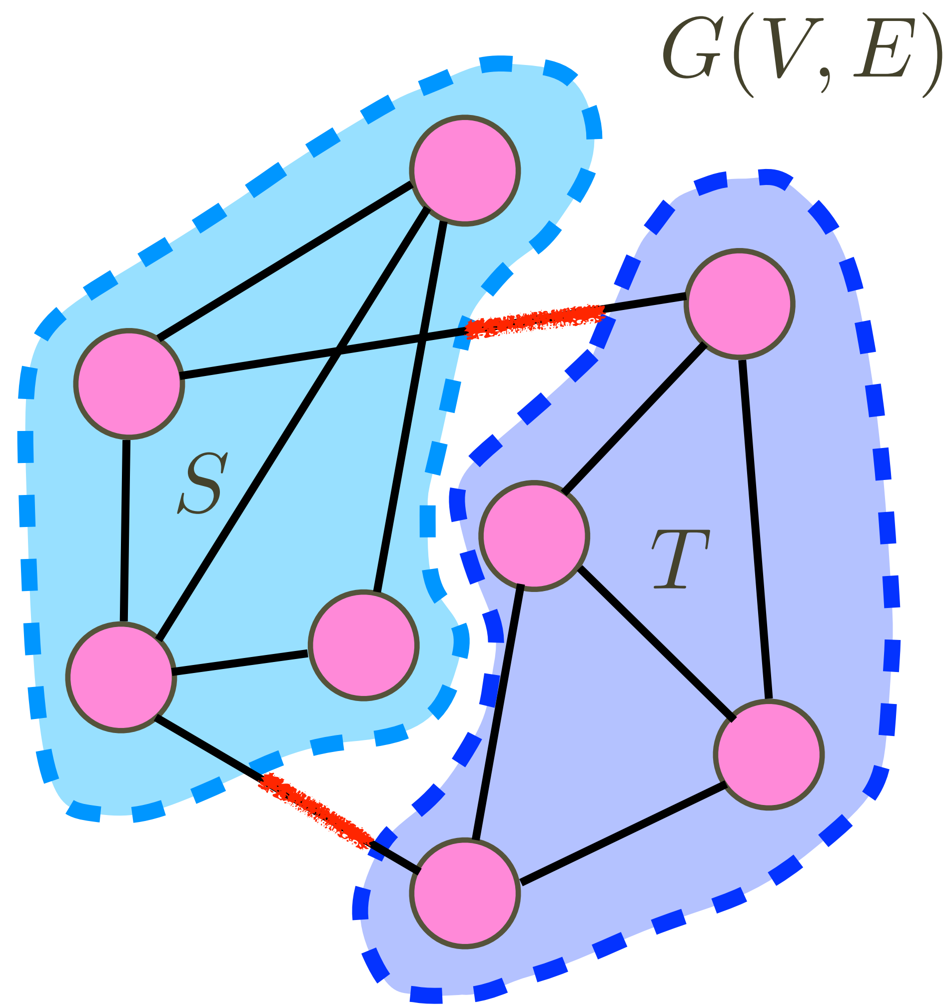
# Randomized Algorithms

南京大学

尹一通

# Min-Cut

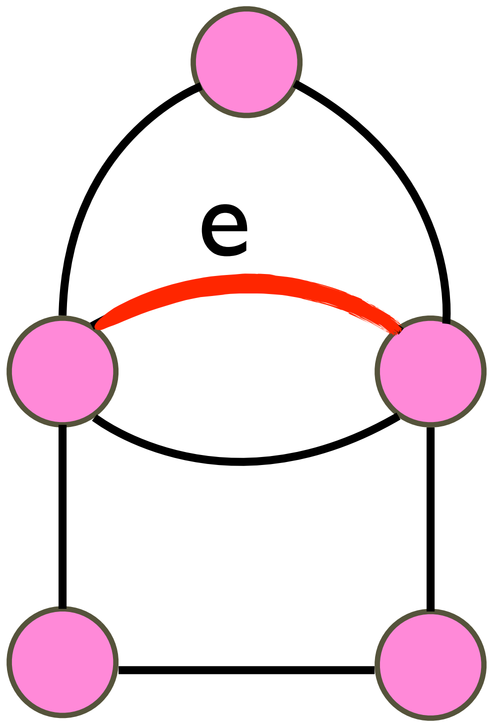
- Partition $V$ into two parts: $S$ and $T$
- minimize the cut $|C(S,T)|$
- many important applications (e.g. parallel computing)
- deterministic algorithm:
  - max-flow min-cut
  - best known upper bound: $O(mn + n^2 \log n)$




$C(S,T)$
$= \{uv \in E \mid u \in S \text{ and } v \in T\}$

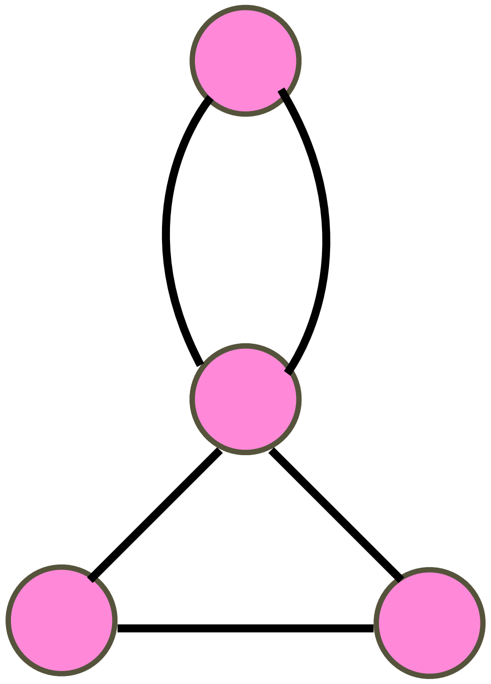
# Contraction

- multigraph $G(V, E)$
- multigraph: allow parallel edges
- for an edge e, contract(e) merges the two endpoints.



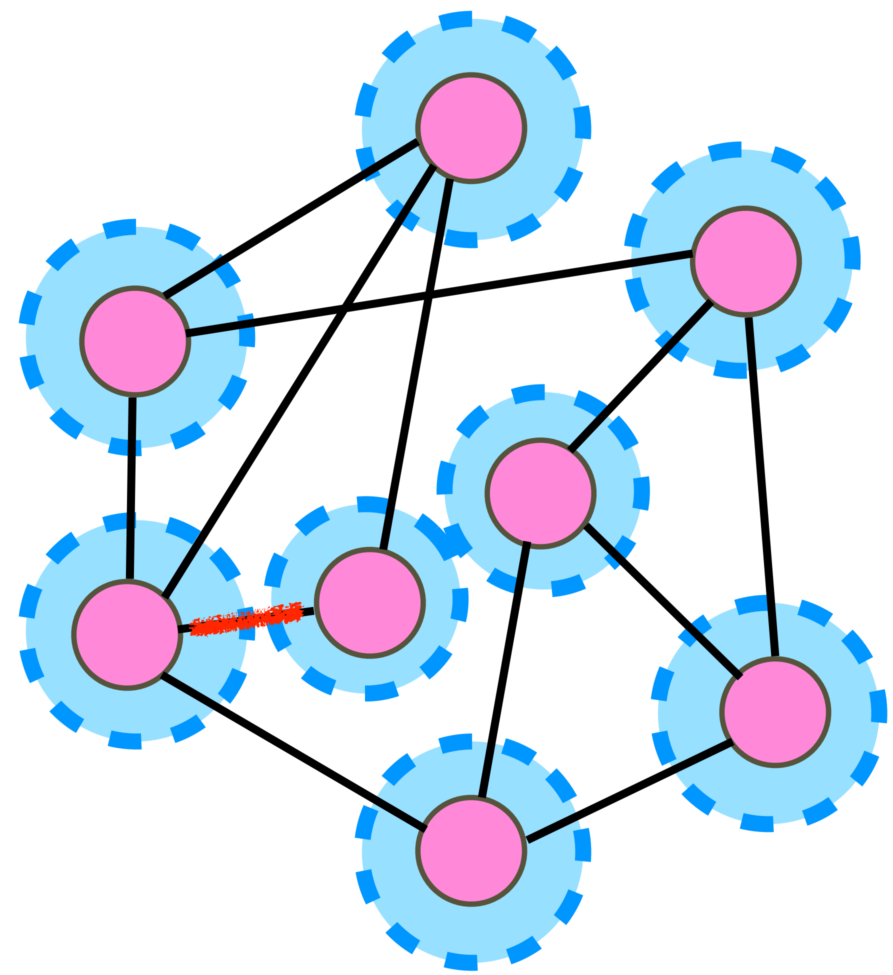
# Contraction

- multigraph $G(V, E)$
- multigraph: allow parallel edges
- for an edge e, contract(e) merges the two endpoints.



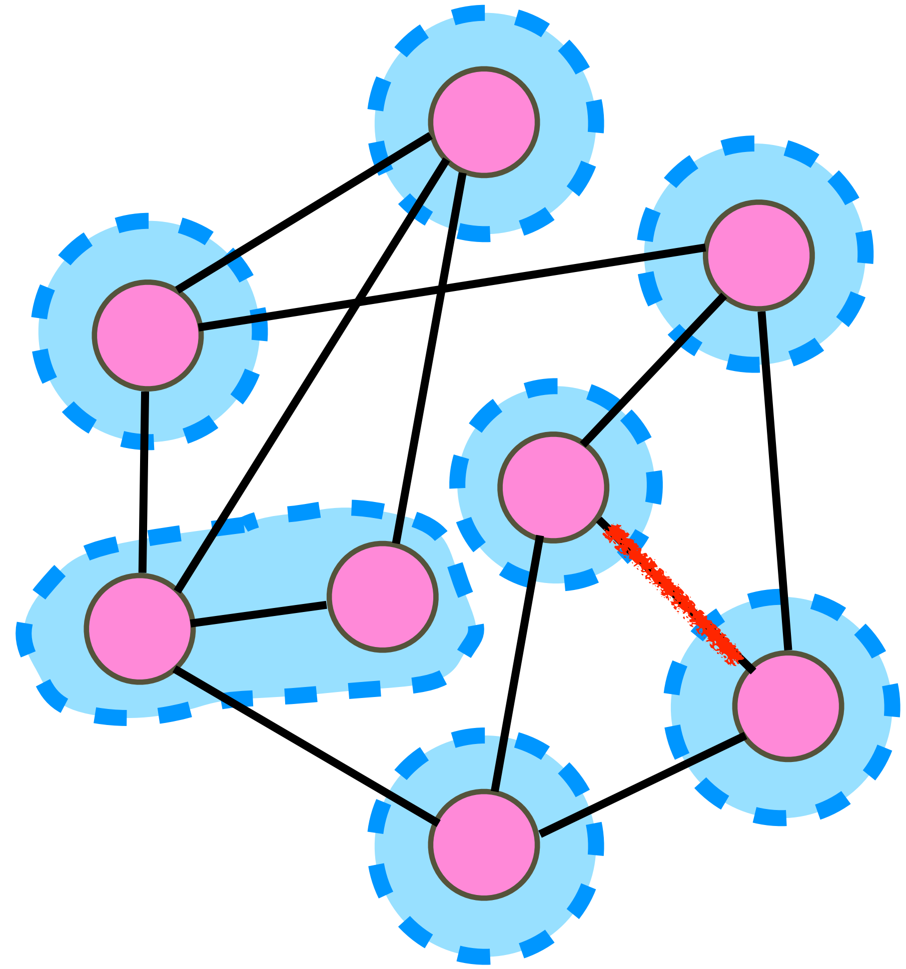
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
| --- |
| while $\|V\|>2$ do |
| choose a uniform $e \in E$ ; |
| contract($e$); |
| return remaining edges; |



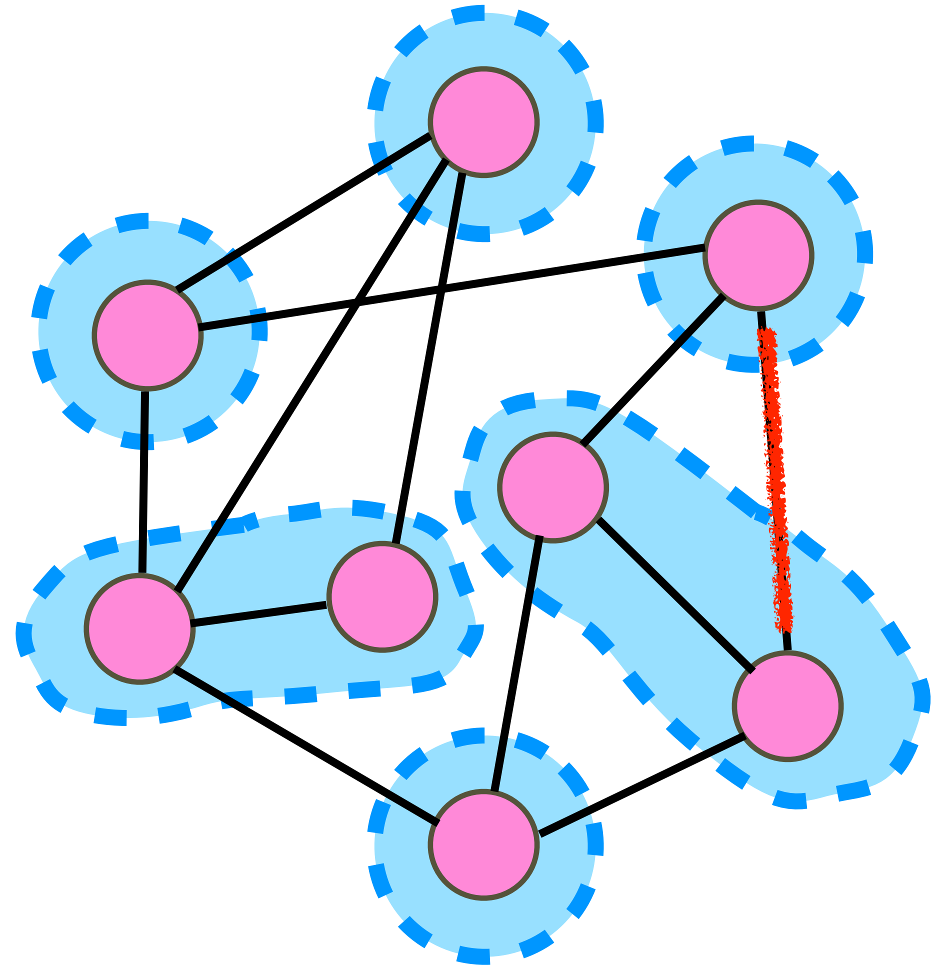
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
|---|
| while $\|V\|>2$ do |
|     choose a uniform $e \in E$ ; |
|     contract($e$); |
| return remaining edges; |



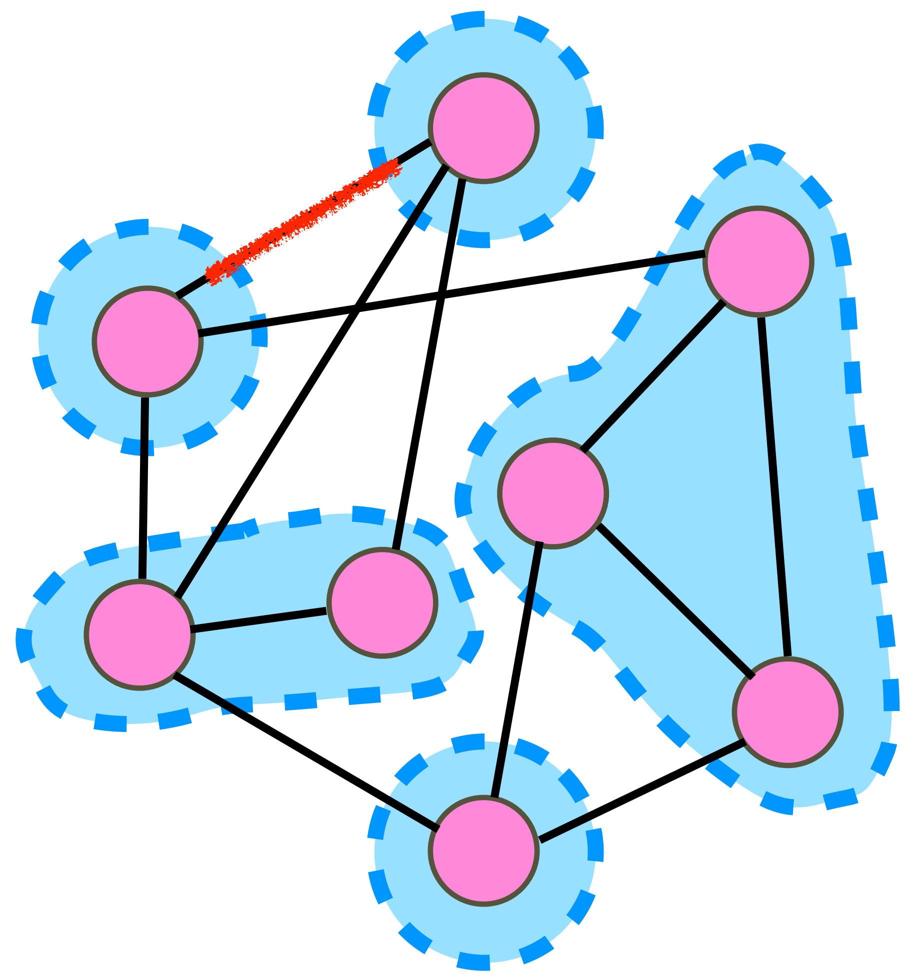
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
|---|
| while $\|V\|>2$ do<br>    choose a uniform $e \in E$ ;<br>    contract($e$);<br>return remaining edges; |



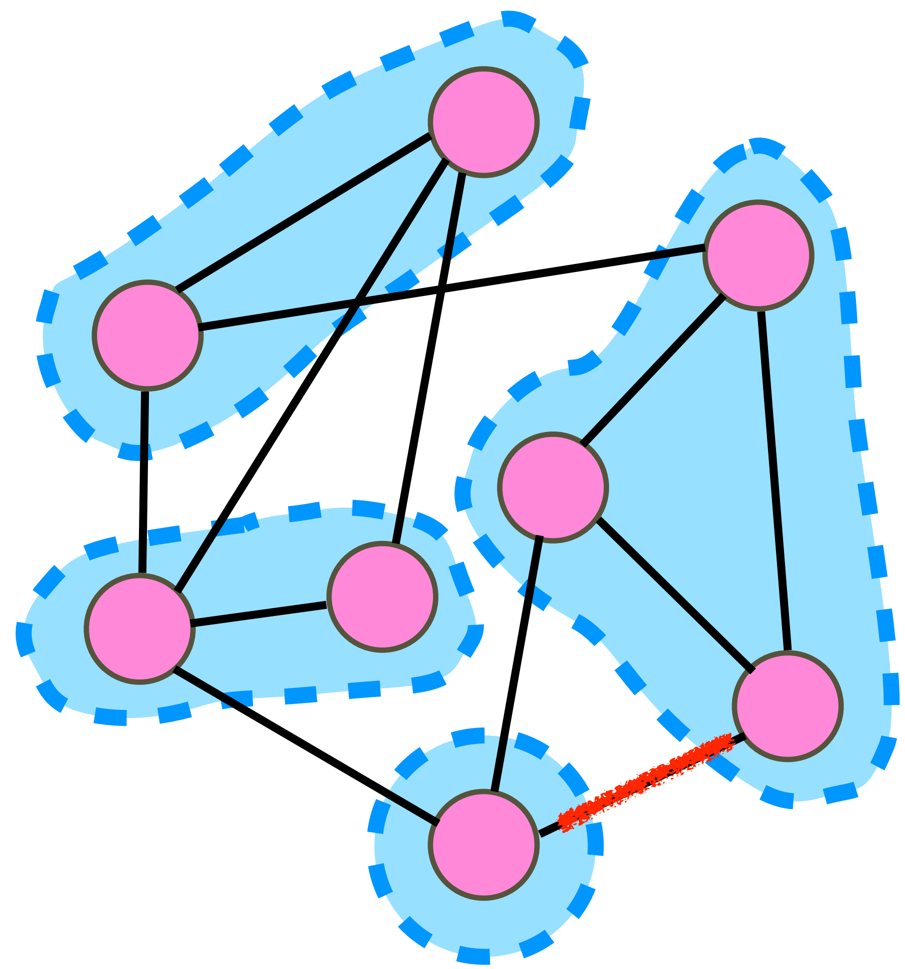
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
| --- |
| while $\|V\|>2$ do |
|     choose a uniform $e \in E$ ; |
|     contract($e$); |
| return remaining edges; |



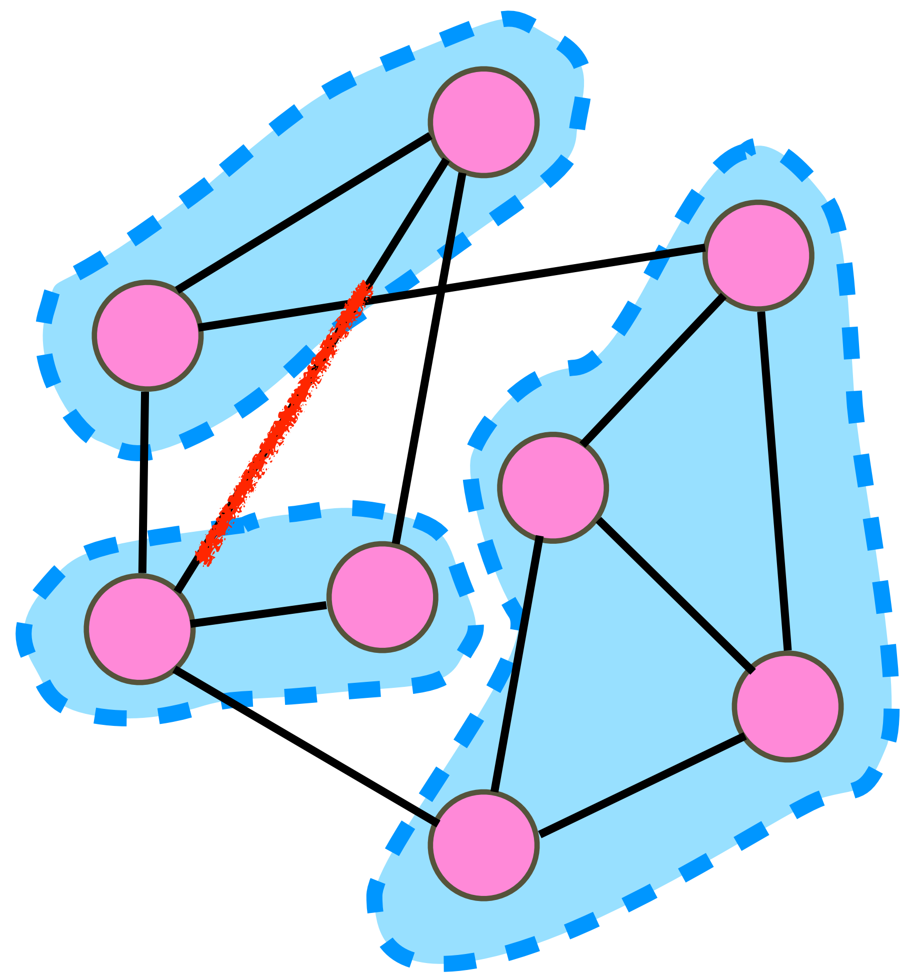
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
|---|
| while $\|V\|>2$ do |
|     choose a uniform $e \in E$ ; |
|     contract($e$); |
| return remaining edges; |



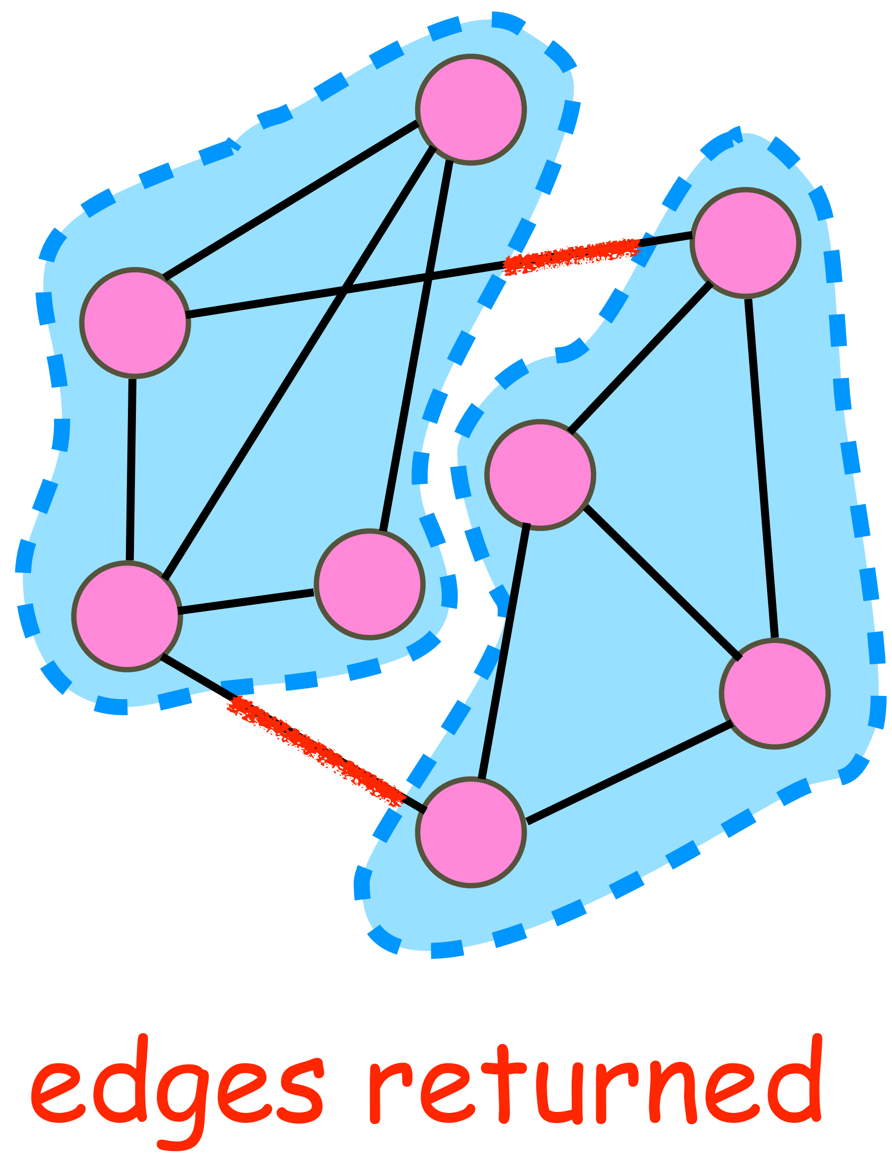
# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
| --- |
| while $\|V\|>2$ do<br>    choose a uniform $e \in E$ ;<br>    contract($e$);<br>return remaining edges; |



# Karger's min-cut Algorithm

| MinCut ( multigraph $G(V,E)$ ) |
| --- |
| while $\|V\|>2$ do |
|     choose a uniform $e \in E$ ; |
|     contract($e$); |
| return remaining edges; |



MinCut ( multigraph $G(V,E)$ )

while $|V|>2$ do

    choose a uniform $e \in E$ ;

    contract($e$);

return remaining edges;

**Theorem** (Karger 1993):

$\Pr[\text{a min-cut is returned}] \geq \frac{2}{n(n-1)}$

repeat independently

for $n(n-1)/2$ times

and return the smallest cut

$$
\begin{aligned}
& \Pr[\text{fail to finally return a min-cut}] \\
= \quad & (\Pr[\text{fail to find a min-cut in a running}])^{n(n-1)/2} \\
\leq \quad & \left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)/2} \\
< \quad & \frac{1}{e}
\end{aligned}
$$

**MinCut ( multigraph $G(V,E)$ )**

while $|V|>2$ do
    choose a uniform $e \in E$ ;
    contract($e$);
return remaining edges;

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$

$i$-th round:
$$G_i = \text{contract}(G_{i-1}, e_{i-1})$$

$$\left.\begin{array}{c} C \text{ is a min-cut in } G_{i-1} \\ e_{i-1} \notin C \end{array}\right\} \Rightarrow C \text{ is a min-cut in } G_i$$

$C$: a min-cut of $G$

$$\Pr[C \text{ is returned}] = \Pr[e_1, e_2, \ldots, e_{n-2} \notin C]$$

chain rule:
$$= \prod_{i=1}^{n-2} \Pr[e_i \notin C \mid e_1, e_2, \ldots, e_{i-1} \notin C]$$

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$ $\quad$ *i*-th round: $G_i = \mathrm{contract}(G_{i-1}, e_{i-1})$

$$\left.\begin{array}{c} C \text{ is a min-cut in } G_{i-1} \\ e_{i-1} \notin C \end{array}\right\} \Rightarrow C \text{ is a min-cut in } G_i$$

$C$: a min-cut of $G$

$$\Pr[C \text{ is returned}] = \prod_{i=1}^{n-2} \Pr[e_i \notin C \mid e_1, e_2, \ldots, e_{i-1} \notin C]$$

$e_1, e_2, \ldots, e_{i-1} \notin C \Rightarrow C$ is a min-cut in $G_i$

$C$ is a min-cut in $G(V,E)$ $\Rightarrow |E| \geq \frac{1}{2}|C||V|$

**Proof:** degree of $G \geq |C|$

**MinCut ( multigraph $G(V,E)$ )**

while $|V|>2$ do

  choose a uniform $e \in E$ ;

  contract($e$);

return remaining edges;

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$  $i$-th round: $G_i = \text{contract}(G_{i-1}, e_{i-1})$

$C$ is a min-cut in $G_i$ $\Rightarrow$ $|E(G_i)| \geq \frac{1}{2}|C||V(G_i)|$

$$\Pr[e_i \in C] = \frac{|C|}{|E(G_i)|} \leq \frac{2|C|}{|C||V(G_i)|} = \frac{2}{(n-i+1)}$$

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$ *i*-th round: $G_i = \text{contract}(G_{i-1}, e_{i-1})$

$$\left.\begin{array}{c} C \text{ is a min-cut in } G_{i-1} \\ e_i \not\in C \end{array}\right\} \Rightarrow C \text{ is a min-cut in } G_i$$

$C$: a min-cut of $G$

$$\Pr[C \text{ is returned}] = \prod_{i=1}^{n-2} \Pr[e_i \not\in C \mid e_1, e_2, \ldots, e_{i-1} \not\in C]$$

$\Downarrow$ $C$ is a min-cut in $G_i$

$$C \text{ is a min-cut in } G(V,E) \Rightarrow |E| \geq \frac{1}{2}|C||V|$$

$$\geq \prod_{i=1}^{n-2}\left(1 - \frac{2}{(n-i+1)}\right)$$

$$= \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} = \frac{2}{n(n-1)}$$

MinCut ( multigraph $G(V,E)$ )

while $|V|>2$ do

choose a uniform $e \in E$ ;

contract($e$);

return remaining edges;

**Theorem** (Karger 1993)**:**

For any min-cut $C$,

$$\Pr[C \text{ is returned}] \geq \frac{2}{n(n-1)}$$

running time: $O(n^2)$

# Number of Min-Cuts

**Theorem** (Karger 1993)**:**

For any min-cut $C$,

$$\Pr[C \text{ is returned}] \geq \frac{2}{n(n-1)}$$

**Corollary**

The number of distinct min-cuts in a graph of $n$ vertices is at most $n(n\text{-}1)/2$.

# An Observation

| MinCut ( multigraph $G(V,E)$ ) |
|---|
| while $\|V\|> t$ do<br>    choose a uniform $e \in E$ ;<br>    contract($e$);<br>return remaining edges; |

$C$: a min-cut of $G$

$$\Pr[e_1, \ldots, e_{n-t} \not\in C] = \prod_{i=1}^{n-t} \Pr[e_i \not\in C \mid e_1, \ldots, e_{i-1} \not\in C]$$

$$\geq \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} = \frac{t(t-1)}{n(n-1)}$$

only getting bad when $t$ is small

# Fast Min-Cut

**Contract** ( $G, t$ )

while $|V|>t$ do

    choose a uniform $e \in E$ ;

    contract($e$);

return current multigraph $G$;

**FastCut** ( $G$ )

if $|V|\leq 6$ then return a min-cut by brute force;

else:   ($t$ to be fixed later)

    $G_1$= Contract($G,t$);

    $G_2$= Contract($G,t$);

(independently)

    return min{FastCut($G_1$), FastCut($G_2$)};

FastCut ( $G$ )

if $|V| \le 6$ then return a min-cut by brute force;

else: ($t$ to be fixed later)

$G_1$= Contract($G$,$t$);

$G_2$= Contract($G$,$t$);

(independently)

return min{FastCut($G_1$), FastCut($G_2$)};

$C$: a min-cut in $G$

$E$: no edge in $C$ is contracted during Contract($G$,$t$)

$$\Pr[E] = \prod_{i=1}^{n-t} \Pr[e_i \not\in C \mid e_1, \ldots, e_{i-1} \not\in C]$$

$$\ge \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \quad \ge \frac{t(t-1)}{n(n-1)} \quad \ge \frac{1}{2}$$

choose $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$

| FastCut ( $G$ ) |
| --- |
| if $\|V\| \leq 6$ then return a min-cut by brute force; |
| else: set $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$ |
| $G_1$ = Contract($G$,$t$); (independently) |
| $G_2$ = Contract($G$,$t$); |
| return min{FastCut($G_1$), FastCut($G_2$)}; |

$C$: a min-cut in $G$

$E$: no edge in $C$ is contracted during $\mathrm{Contract}(G,t)$

$$\Pr[E] \geq \frac{1}{2}$$

$$p(n) = \Pr[C = \mathrm{FastCut}(G)]$$

$$= 1 - (1 - \Pr[E]\Pr[C = \mathrm{FastCut}(G_1) \mid E])^2$$

$$\geq 1 - \left(1 - \tfrac{1}{2} p\left(\left\lceil 1 + \tfrac{n}{\sqrt{2}} \right\rceil\right)\right)^2$$

| FastCut ( $G$ ) |
|---|
| if $\|V\| \leq 6$ then return a min-cut by brute force;<br>else: set $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$<br>$G_1$= Contract($G$,$t$);<br>$G_2$= Contract($G$,$t$); (independently)<br>return min{FastCut($G_1$), FastCut($G_2$)}; |

$C$: a min-cut in $G$

$$p(n) = \Pr[C = \mathrm{FastCut}(G)] \geq 1 - \left(1 - \frac{1}{2}p\left(\left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil\right)\right)^2$$

by induction: $p(n) = \Omega\left(\frac{1}{\log n}\right)$

running time: $T(n) = 2T\left(\left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil\right) + O(n^2)$

by induction: $T(n) = O(n^2 \log n)$

**FastCut ( $G$ )**

if $|V|\le 6$ then return a min-cut by brute force;

else: set $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$

$G_1$= Contract($G$,$t$);

$G_2$= Contract($G$,$t$);

(independently)

return min{FastCut($G_1$), FastCut($G_2$)};

**Theorem** (Karger-Stein 1996)**:**

FastCut runs in time $O(n^2 \log n)$ and returns a min-cut with probability $\Omega(1/\log n)$.

repeat *independently* for $O(\log n)$ times

total running time: $O(n^2 \log^2 n)$

returns a min-cut with probability $1\text{-}O(1/n)$

# Primality Test

Input: positive integer n
Output: "yes" if n is prime
"no" if n is composite

~~for $k = 1, 2, \ldots, \sqrt{n}$~~
~~check whether $n \mid k$~~

efficient?

efficient algorithm: running time is polynomial of the length of input

$(\log n)^{O(1)}$ time for primality test

# Fermat's little theorem

$n$ is prime

$\Downarrow$

$$\forall\ a \in \{1, 2, \ldots, n-1\}$$
$$a^{n-1} \equiv 1 \pmod{n}$$

This proof requires the most basic elements of group theory.

The idea is to recognise that the set $G = \{1, 2, \ldots, p-1\}$, with the operation of multiplication (taken modulo $p$), forms a group. The only group axiom that requires some effort to verify is that each element of $G$ is invertible. Taking this on trust for the moment, let us assume that $a$ is in the range $1 \le a \le p-1$, that is, $a$ is an element of $G$. Let $k$ be the order of $a$, so that

$$a^k \equiv 1 \pmod{p}.$$

By Lagrange's theorem, $k$ divides the order of $G$, which is $p-1$, so $p-1 = km$ for some positive integer $m$. Then

$$a^{p-1} \equiv a^{km} \equiv (a^k)^m \equiv 1^m \equiv 1 \pmod{p}.$$

**The invertibility property**

To prove that every element $b$ of $G$ is invertible, we may proceed as follows. First, $b$ is relatively prime to $p$. Then Bézout's identity assures us that there are integers $x$ and $y$ such that

$$bx + py = 1.$$

Reading this equation modulo $p$, we see that $x$ is an inverse for $b$, since

$$bx \equiv 1 \pmod{p}.$$

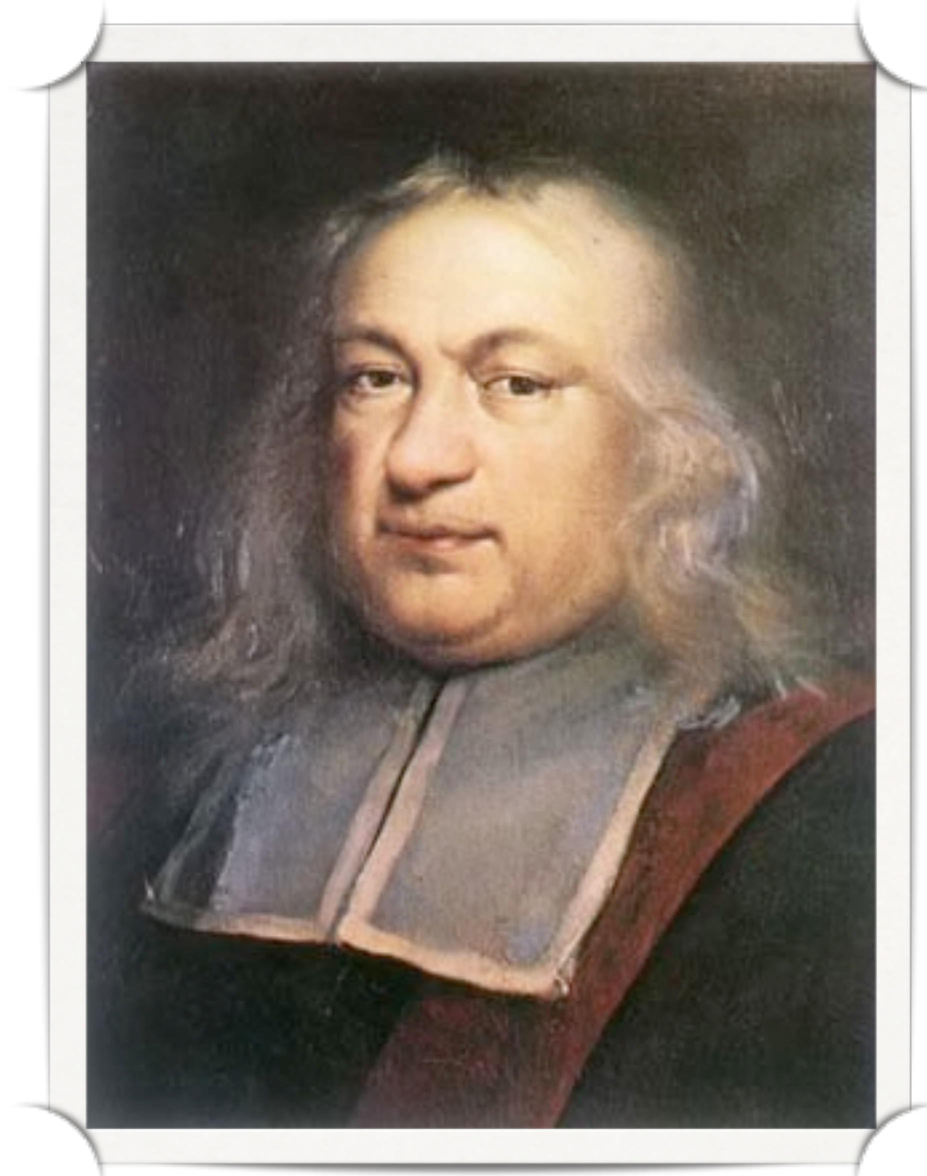
Therefore every element of $G$ is invertible, so as remarked earlier, $G$ is a group.

Pierre de Fermat

*amateur mathematician*

$$\exists\ a \in \{1, 2, \ldots, n-1\}$$
$$a^{n-1} \not\equiv 1 \pmod{n}$$

$n$ is composite



compositeness
~~Primality~~ test:

find such a *proof* $a$

- Does it exist?
- How to find it?

## Fermat Test

- Choose a random $a \in \{1, 2, \ldots, n-1\}$.
- If $a^{n-1} \not\equiv 1 \pmod{n}$, return non-Carmichael **composite**.
- Else return **probably prime**. or Carmichael

## Carmichael number

$n$ is composite, and

$$\forall\, a \in \{1, 2, \ldots, n-1\} \quad a^{n-1} \equiv 1 \pmod{n}$$

fools the Fermat test

## Fermat Test

- Choose a random $a \in \{1, 2, \ldots, n-1\}$.
- If $a^{n-1} \not\equiv 1 \pmod{n}$, return **composite**.
- Else return **probably prime**.

$n$ is prime: return **probably prime**

$n$ is non-Carmichael composite:

good $a$ $a^{n-1} \not\equiv 1$ bad $a$ $a^{n-1} \equiv 1$

$n$ is Carmichael:

incorrectly return **probably prime**

## Fermat Test

- Choose a random $a \in \{1, 2, \ldots, n-1\}$.
- If $a^{n-1} \not\equiv 1 \pmod{n}$, return **composite**.
- Else return **probably prime**.

multiplicative group modulo $n$

$$\mathbb{Z}_n^* = \{a \mid 1 \leq a \leq n-1 \wedge \gcd(a, n) = 1\}$$

$$B = \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv 1 \pmod{n}\}$$

closed under multiplication mod $n$

⇒ $B$ is subgroup of $\mathbb{Z}_n^*$ ⇒ $|\mathbb{Z}_n^*| \mid |B|$

## Fermat Test

- Choose a random $a \in \{1, 2, \ldots, n-1\}$.
- If $a^{n-1} \not\equiv 1 \pmod{n}$, return **composite**.
- Else return **probably prime**.

$n$ is prime: return **probably prime**

$n$ is non-Carmichael composite:

return **composite** with probability $\geq 1/2$

$n$ is Carmichael:

incorrectly return **probably prime**

Fermat's proof of compositeness:

$$a^{n-1} \not\equiv 1 \pmod{n}$$ incomplete

Another proof of compositeness:

nontrivial square root of 1

$$a^2 \equiv 1 \pmod{n} \qquad a \not\equiv \pm 1 \pmod{n}$$

**Theorem**

If $n$ is prime, 1 does not have nontrivial square root.

$$a^2 \equiv 1 \pmod{n} \Rightarrow (a+1)(a-1) \equiv 0 \pmod{n}$$

$$\Rightarrow (a+1)(a-1) \mid n$$

Fermat's proof of compositeness:

$$a^{n-1} \not\equiv 1 \pmod{n}$$ incomplete

Another proof of compositeness:

$$a^2 \equiv 1 \pmod{n}$$

$$a \not\equiv \pm 1 \pmod{n}$$

for composite $n$:

find an $a \in \{1, 2, \ldots, n-1\}$ such that $a^{n-1} \not\equiv 1 \pmod{n}$

or $a \not\equiv \pm 1 \pmod{n}$ but $a^2 \equiv 1 \pmod{n}$

## Miller-Rabin test

choose a random $a \in \{1, 2, \ldots, n-1\}$;

decompose $n-1$ as $n-1 = 2^t m$ with odd $m$;

compute $a^m, a^{2m}, \ldots, a^{2^i m}, \ldots, a^{2^t m} \pmod{n}$;

Fermat test

if $a^{n-1} = a^{2^t m} \not\equiv 1 \pmod{n}$, return **composite**;

if $\exists i, a^{2^i m} \equiv 1$ but $a^{2^{i-1} m} \not\equiv \pm 1$, return **composite**;

nontrivial square root of 1

else return **probably prime**;

$n$ is prime: return **probably prime**

$n$ is non-Carmichael composite: Fermat test

return **composite** with probability $\geq 1/2$

$n$ is Carmichael: ?

**Miller-Rabin test**

choose a random $a \in \{1, 2, \ldots, n-1\}$;

decompose $n-1$ as $n-1 = 2^t m$ with odd $m$;

compute $a^m, a^{2m}, \ldots, a^{2^i m}, \ldots, a^{2^t m} \pmod{n}$;

Fermat test

if $a^{n-1} = a^{2^t m} \not\equiv 1 \pmod{n}$, return **composite**;

if $\exists i, a^{2^i m} \equiv 1$ but $a^{2^{i-1} m} \not\equiv \pm 1$, return **composite**;

nontrivial square root of 1

else return **probably prime**;

let $j$ be the maximum such $j$ satisfying:

$$\exists b \in \mathbb{Z}_n^* \text{ s.t. } b^{2^j m} \equiv -1 \pmod{n}$$

let $B = \{a \in \mathbb{Z}_n^* \mid a^{2^j m} \equiv \pm 1 \pmod{n}\}$

is a proper subgroup of $\mathbb{Z}_n^*$ for Carmichael $n$

## Miller-Rabin test

choose a random $a \in \{1, 2, \ldots, n-1\}$;

decompose $n-1$ as $n-1 = 2^t m$ with odd $m$;

compute $a^m, a^{2m}, \ldots, a^{2^i m}, \ldots, a^{2^t m} \pmod{n}$;

Fermat test

if $a^{n-1} = a^{2^t m} \not\equiv 1 \pmod{n}$, return **composite**;

if $\exists i, a^{2^i m} \equiv 1$ but $a^{2^{i-1} m} \not\equiv \pm 1$, return **composite**;

nontrivial square root of 1

else return **probably prime**;

$n$ is prime: return **probably prime**

$n$ is non-Carmichael composite: Fermat test

return **composite** with probability $\geq 1/2$

$n$ is Carmichael: all bad $a \in$ a proper subgroup

return **composite** with probability $\geq 1/2$

# Nondeterminism

**Miller-Rabin test**

For any composite $n$, there exist (over half) $a \in \{1, 2, \ldots, n-1\}$

$$a^{n-1} \not\equiv 1 \pmod{n} \quad \text{or}$$

$$\exists i, a^{2^i m} \equiv 1 \pmod{n} \text{ but } a^{2^{i-1} m} \not\equiv \pm 1 \pmod{n}$$

where $m$ is odd and $n - 1 = 2^t m$.

efficiently verifiable

certificate (*proof*, *witness*) of compositeness of $n$

$$\text{COMPOSITE} \in \mathbf{NP}$$

$$\text{PRIME} \in \text{co-}\mathbf{NP}$$

in 2002, PRIME ∈ P (AKS test)

Randomization: redundancy of certificates

# Randomized Algorithms

"algorithms which use randomness in computation"

How?

- To hit a witness.
- To fool an adversary.
- To simulate random samples.
- To construct a solution.
- To break symmetry.
- ... ...