Randomized Min Cut

Probability

Events, probabilities,

conditional probabilities,

union bound

Random variables,

expected value,

linearity of expectation

Markov's inequality

Algorithms

Approx. 3-SAT,

Big idea: we only want the average

Linearity of Expectation

$E[X+Y] = E[X] + E[Y]$ (ridiculously useful)

"average sum = sum of averages"

let $Y_i = \begin{cases} 1 & \text{is ith clause satisfied} \\ 0 & \text{otherwise} \end{cases}$ $(i = 1, \ldots, m)$

$E[Y_i] = 7/8$ (1 clause case)

$E\left[\begin{matrix}\text{\# clauses} \\ \text{satisfied}\end{matrix}\right] = E\left[\sum_{i=1}^{m} Y_i\right] = \sum_{i=1}^{m} E[Y_i] = 7/8\, m$

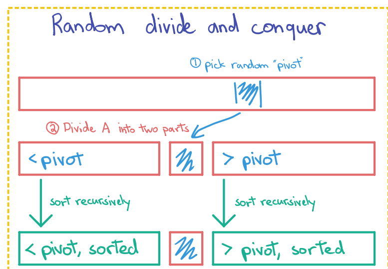
let $Y_i = \begin{cases} 1 & \text{is ith clause satisfied} \\ 0 & \text{otherwise} \end{cases}$

$Z = \sum_{i=1}^{m} Y_i$ = # clauses satisfied

$E[Z] = \frac{1}{2} E[Z \mid x_1 = T] + \frac{1}{2} E[Z \mid x_1 = F]$

$\Rightarrow$ either $E[Z \mid x_1 = T]$ or $E[Z \mid x_1 = F]$ is $\geq E[Z]$!

QuickSort,



Expected running time of QuickSort

For each $i, j \in [n]$ w/ $i < j$,

$X_{ij} = \begin{cases} 1 & \text{if ith smallest element is compared to jth sma}\ldots \\ 0 & \text{otherwise} \end{cases}$

$\sum_{i<j} X_{ij}$ = total # comparisons made by the algorithm

$\approx$ total running time

$E\left[\sum_{i<j} X_{ij}\right]$ = average running time

$E\left[\sum_{i<j} X_{ij}\right] = \sum_{i,j} E[X_{ij}]$

Linearity of Expectation

$E[X+Y] = E[X] + E[Y]$ (ridiculously useful)

"average sum = sum of averages"

now analyze $E[X_{ij}]$ in isolation

$E[X_{ij}] = Pr[X_{ij} = 1]$

$= Pr\left[\begin{matrix}\text{ith element is} \\ \text{compared to jth}\end{matrix}\right]$

$\boxed{i}\ \boxed{\ }\ \boxed{\ }\ \boxed{\ } \cdots \boxed{\ }\ \boxed{j}$

ith elem is compared w/ jth

$\Leftrightarrow$ ith or jth elem selected before any element in between

happens w/ probability $\frac{2}{j-i+1}$

$E[X_{ij}] = \frac{2}{j-i+1}$

$E\left[\sum_{i<j} X_{ij}\right] = \sum_{i,j} E[X_{ij}]$ by linearity of expectation

$= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \leq \sum_{j=1}^{n} \sum_{i=1}^{j} \frac{2}{j-i+1}$

$\leq 2n\left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}\right)$

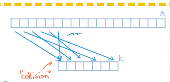
$\leq O(n \log n)$ nth harmonic number

Quickselect (HW)

Probability

Hash Functions,

Universal Hash Functions



$h : [n] \to k$ s.t. for all pairs $i, j \in [n]$,

$$\Pr[h(i) = h(j)] \le \frac{1}{k}$$

(much weaker than ideal hash)

e.g. $h(x) = ((ax + b) \bmod p) \bmod k$

where
- p prime, $> k$
- a in $\{1, \dots, p\}$ unif. random
- b in $\{0, \dots, p-1\}$ unif. random

Markov's Inequality

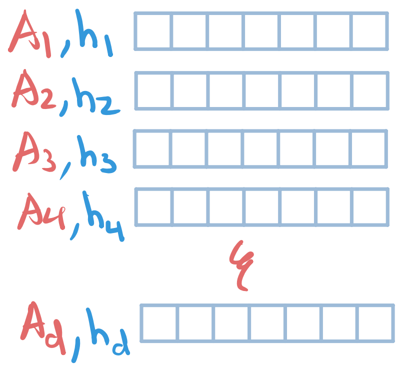
Markov's Inequality (special case)

Let $X \ge 0$ be nonnegative RV

$$\Pr[X \ge 2E[X]] \le \frac{1}{2}$$

Algorithms

Frequency Estimation, Heavy Hitters (in streams)



estimate f_i w/ $\min_{j=1,\dots,d} A_j[h_j(i)]$

Probability

k-wise independence, higher order moments

(generalizating universal hash functions and Markov's inequality)

Algorithms

Hash tables w/ chaining + linear probing

k-wise independent hash functions

$h : [U] \to [m]$

s.t. for any k keys $x_1, x_2, \dots, x_k \in [U]$,
k values $v_1, v_2, \dots, v_k \in [U]$,

$P[h(x_1)=v_1, h(x_2)=v_2, \dots, h(x_k)=v_k] = \frac{1}{m^k}$

e.g. $h(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{k-1} x^{k-1} \bmod p$

where $p \geq U$ prime,

$a_0, a_1, \dots, a_{k-1} \in [p]$ indep, uniformly random

k-wise independent random variables

$Pr[X_{i_1}=y_1, X_{i_2}=y_2, \dots, X_{i_k}=y_k]$

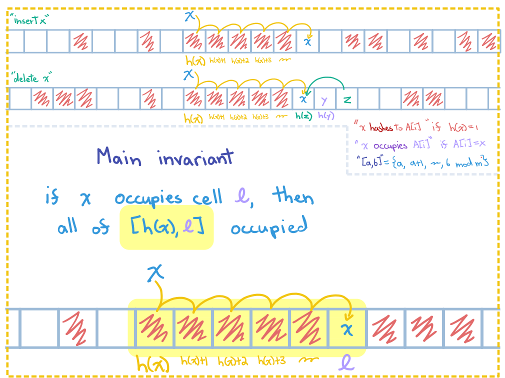
$= Pr[X_{i_1}=y_1] Pr[X_{i_2}=y_2] \dots Pr[X_{i_k}=y_k]$

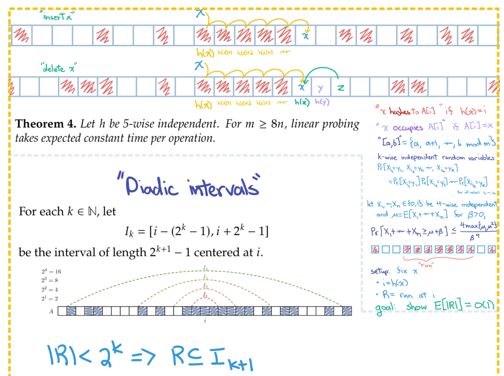
for all values $y_1, \dots, y_k$

Important lemma

let $X_1, \dots, X_n \in \{0,1\}$ be 4-wise independent and $\mu = E[X_1 + \dots + X_n]$. then for $\beta > 0$,

$$Pr[X_1 + \dots + X_n \geq \mu + \beta] \leq \frac{4 \max\{\mu, \mu^2\}}{\beta^4}$$

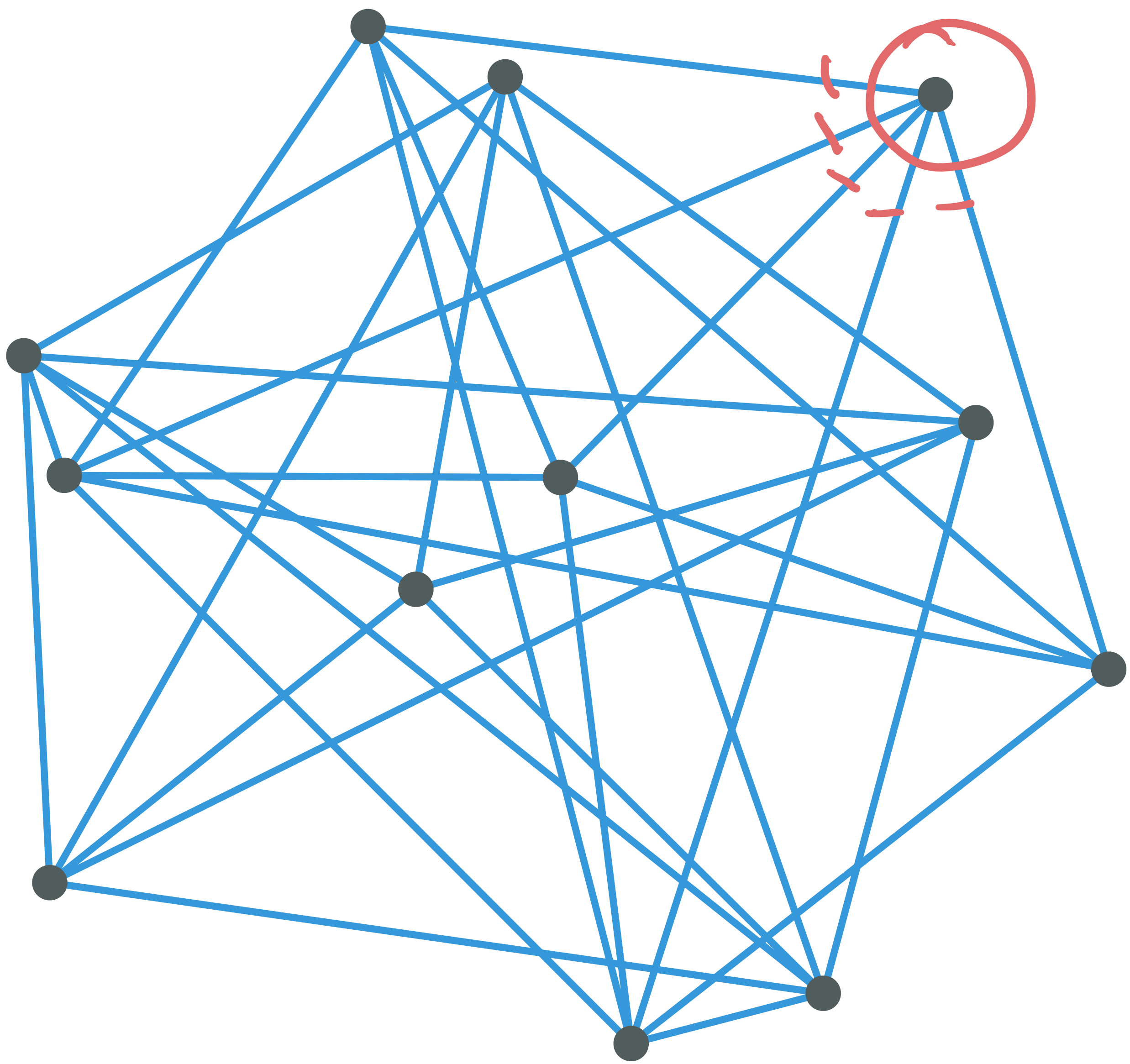




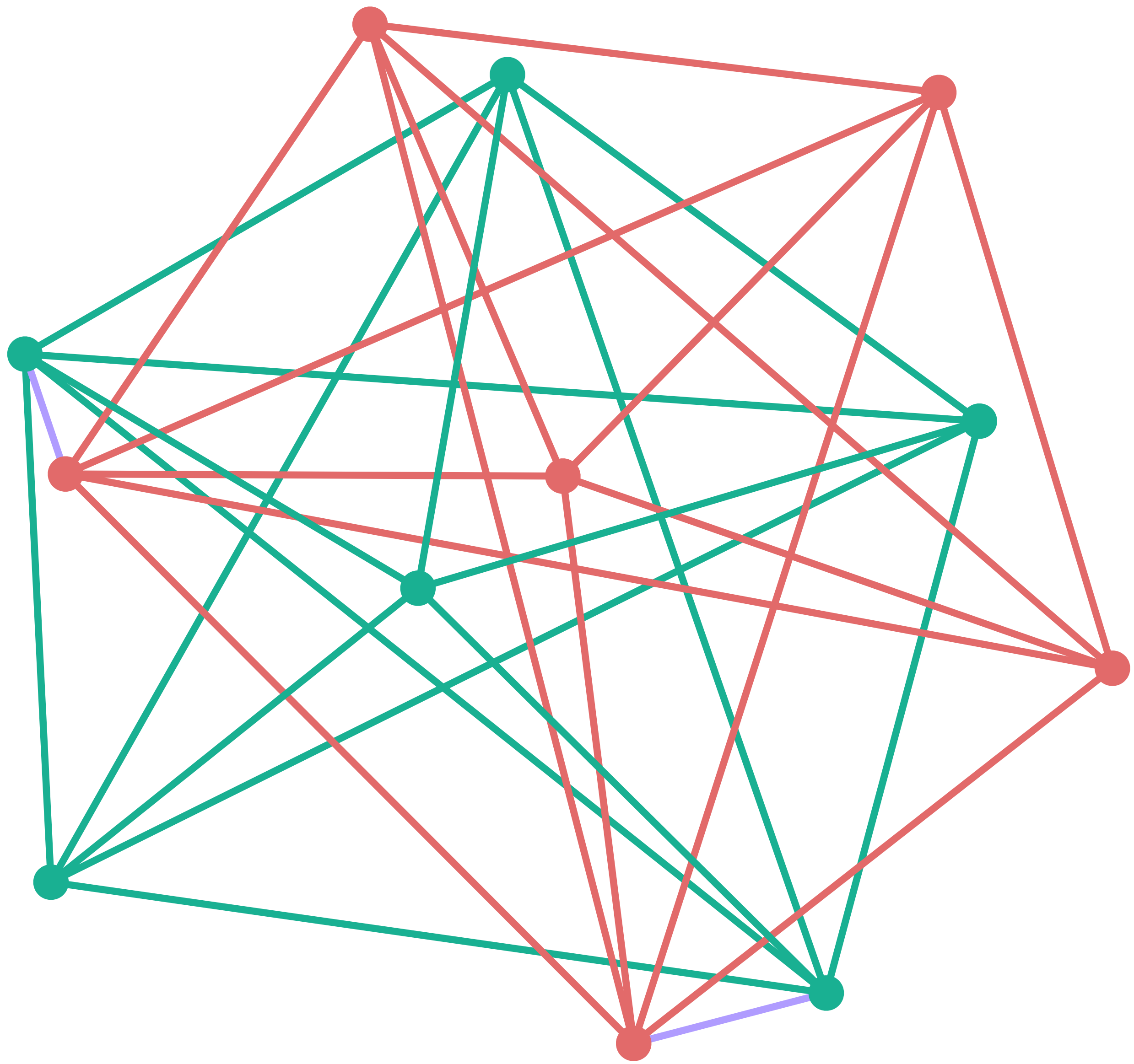
Today

Probability	Algorithms
Galton-Watson processes	Random Contractions for min cut

Goal: disconnect graph by removing min # edges



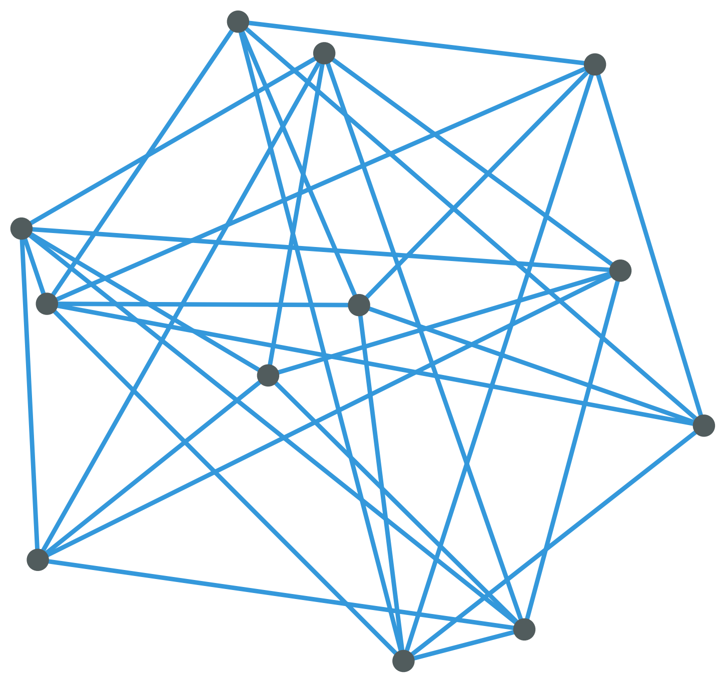
Goal: disconnect graph by removing min # edges

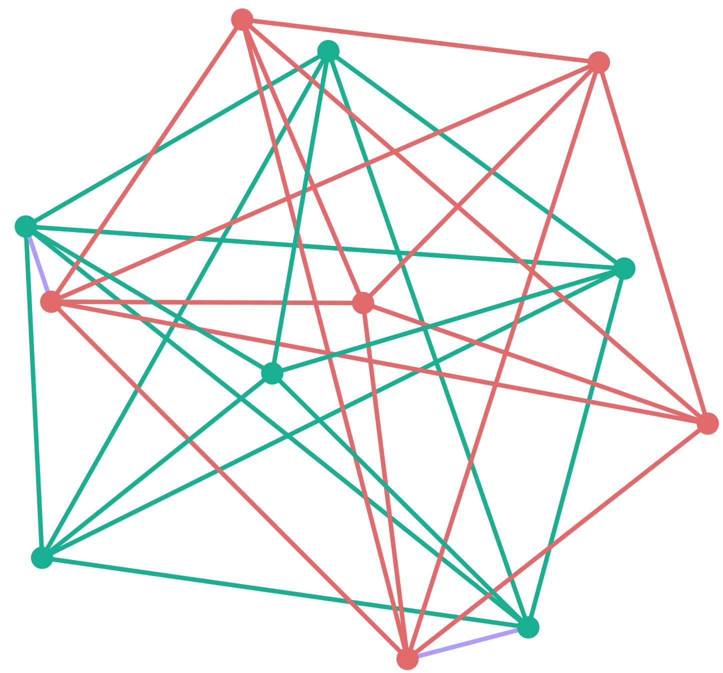


Global Minimum Cut

Input: Connected, Undirected Graph

Goal: disconnect graph by removing min # edges





Also, "minimum weight cut" w/ edge capacities

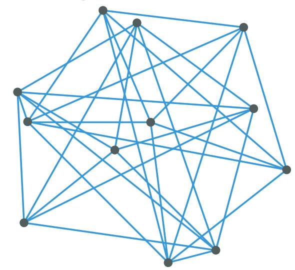
Global min-cut via max flow

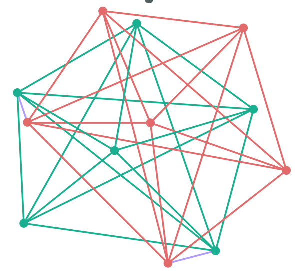
① for each pair $s,t \in V$

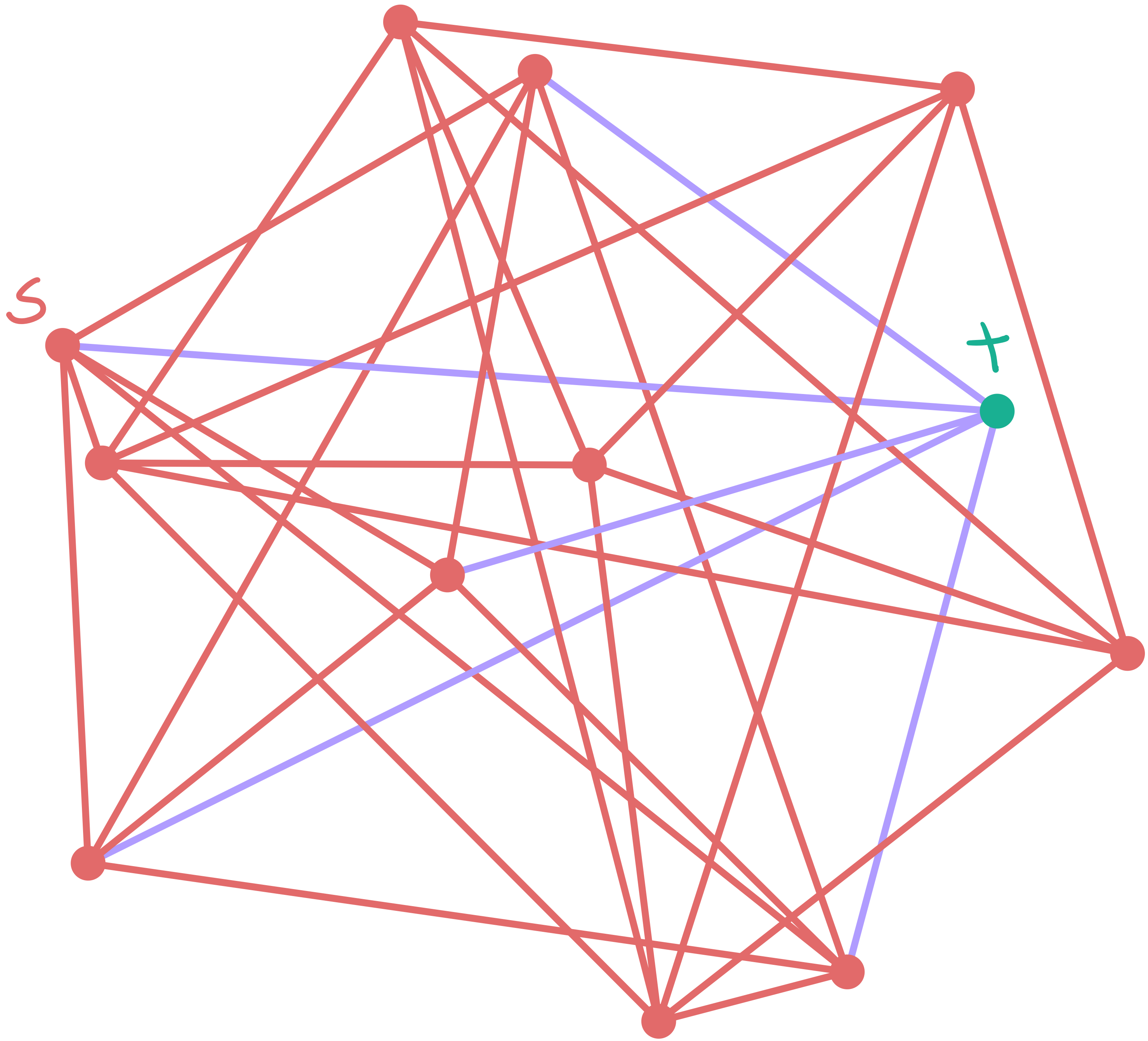
   ⓐ compute min (s,t)-cut

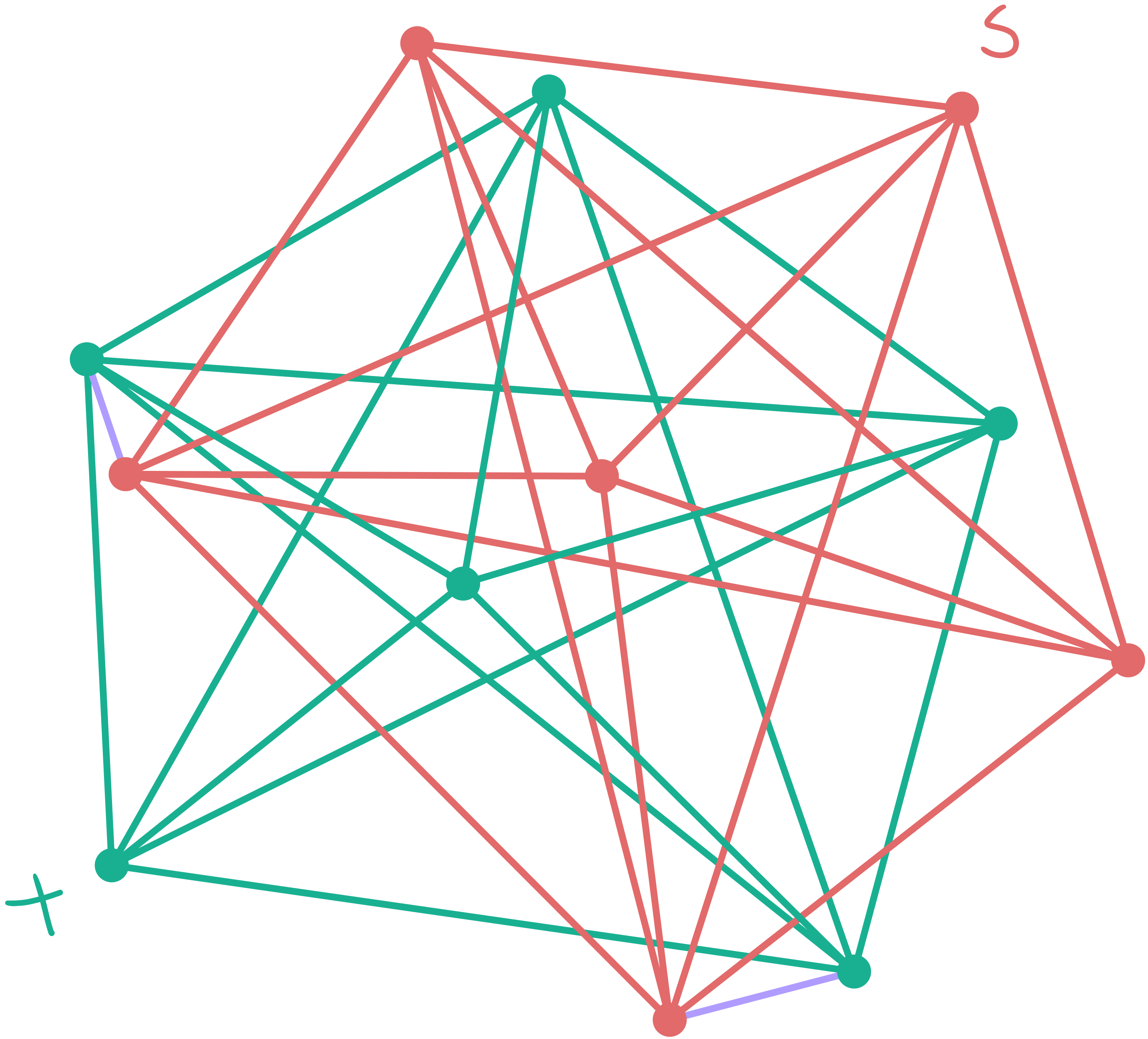
② return min (s,t)-cut over all s,t

Goal: disconnect graph by removing min # of edges






s
t



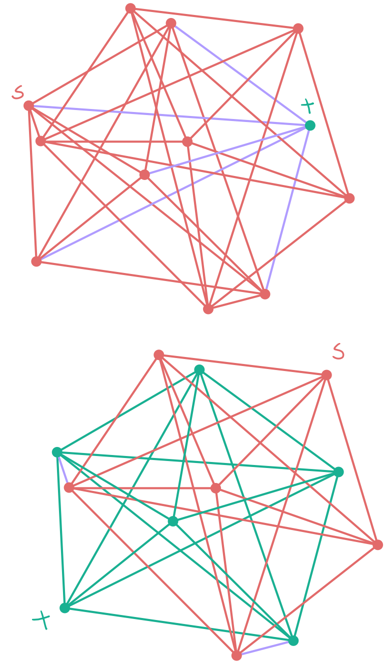
s
t

Global min-cut via max flow

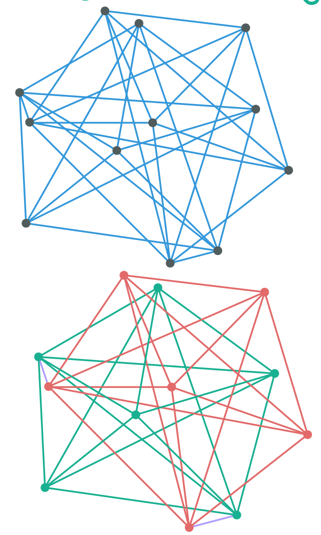
① for each pair $s,t \in V$

ⓐ compute min (s,t)-cut

② return min (s,t)-cut over all s,t



Goal: disconnect graph by removing min # of edges



Running time

$\binom{n}{2} \cdot MF(m,n)$ $\qquad m^{1+o(1)}$

"$MF(m,n)$" = time for max flow w/ m edges, n vertices

Augmenting paths $O(\lambda m)$

Shortest Aug. Paths $O(m^2 n)$

Fattest Aug. Paths $O(m(m+n\log n)\log\lambda)$

Better?

Fix s, try all t

$n \cdot MF(m,n)$

Question:

Max # min-cuts in a graph?

0

$O(1)$

constant

62%

$n^{O(1)}$

polynomial

0

not voting

27.5%

$n^{\log^{O(1)} n}$

subexponential

6

2^n

exponential

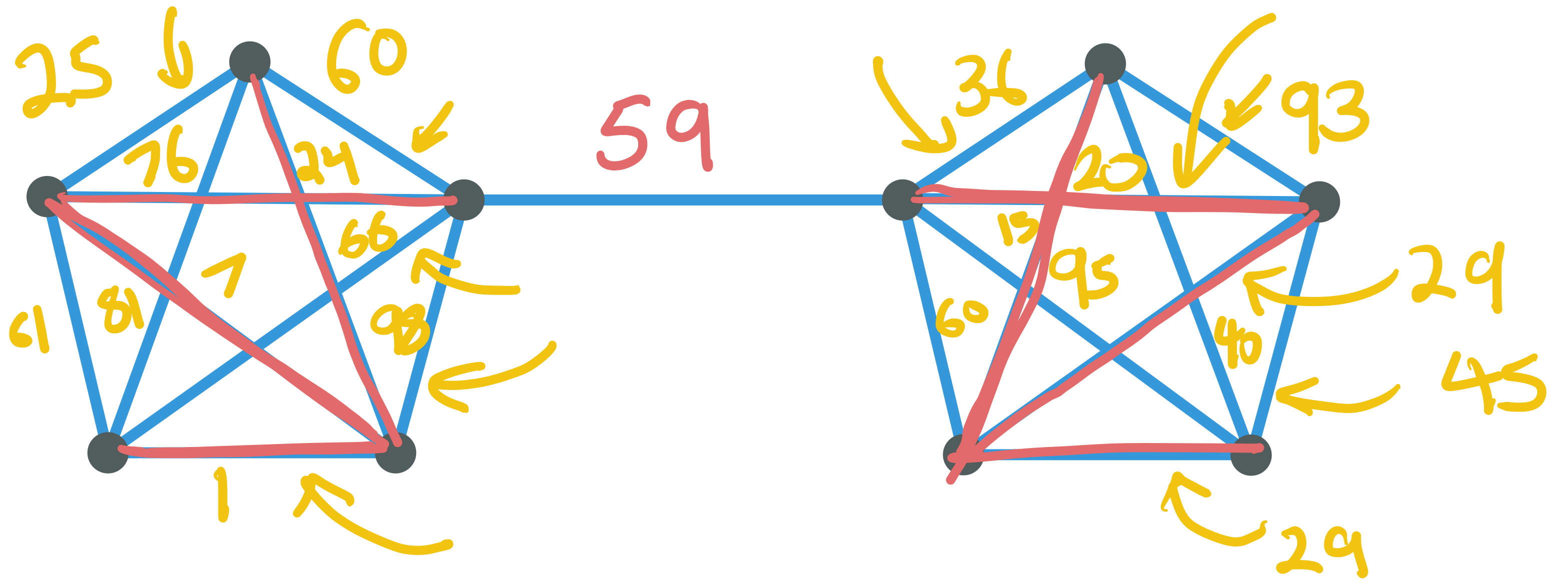
1 ~~4~~ no vote

3 liars

Randomly Sampling the Min-Cut

algorithm:

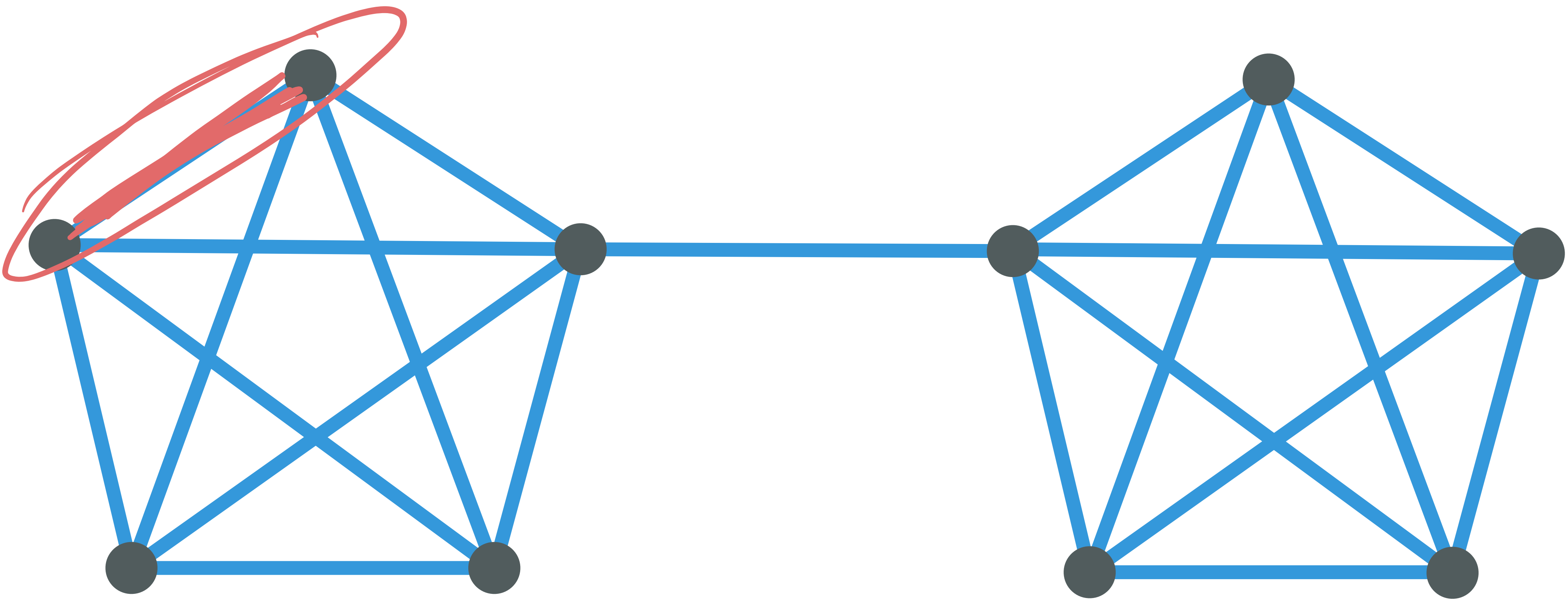
Repeatedly sample edges in proportion to their capacities until there is only one cut from which we have not yet sampled any edges. Return this cut.



or equivalently:

Independently assign every edge $e \in E$ a weight $w_e \in [0,1]$ uniformly at random. Build the minimum weight spanning tree T w/r/t w. Let e be the heaviest edge in T. Return the cut induced by the two components of $T - e$.

Independently assign every edge $e \in E$ a weight $w_e \in [0,1]$ uniformly at random. Build the minimum weight spanning tree T w/r/t w. Let e be the heaviest edge in T. Return the cut induced by the two components of $T - e$.



Randomized contractions

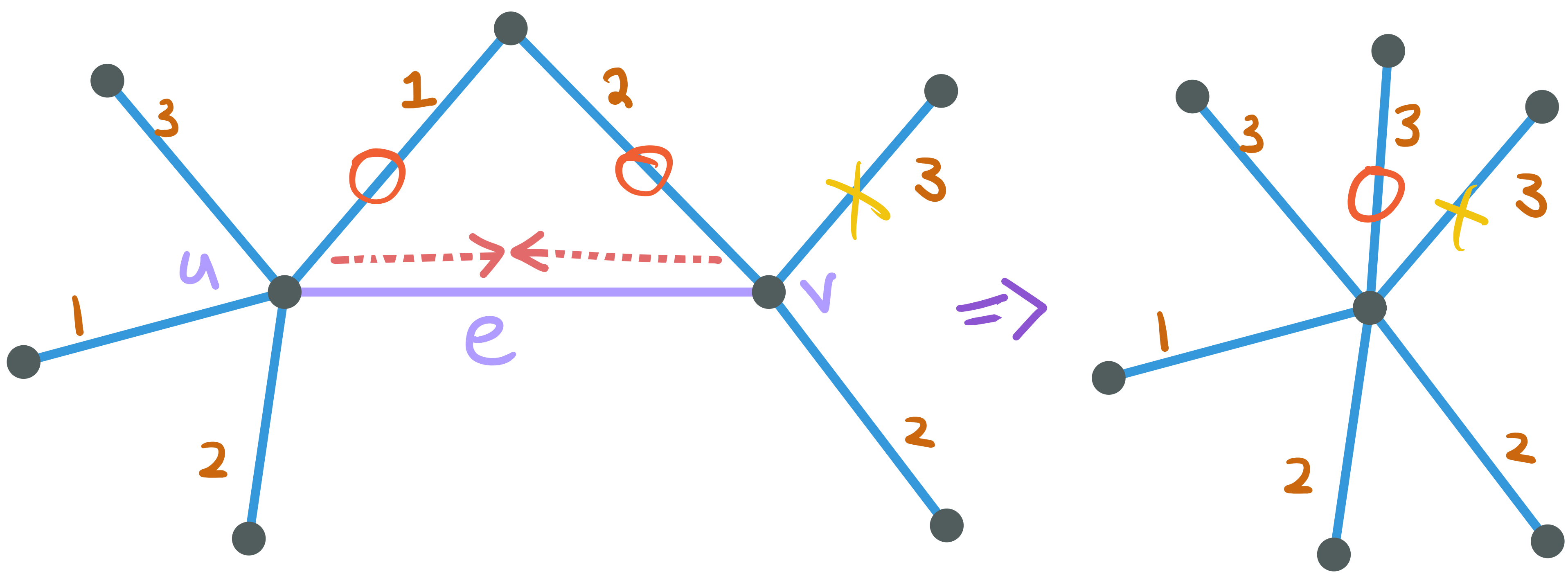
Input: $G=(V,E)$, weights $c \in \mathbb{R}^{E}_{>0}$
(capacities)

① while $|E| > 1$:

Ⓐ sample $e \in E$ in proportion to c

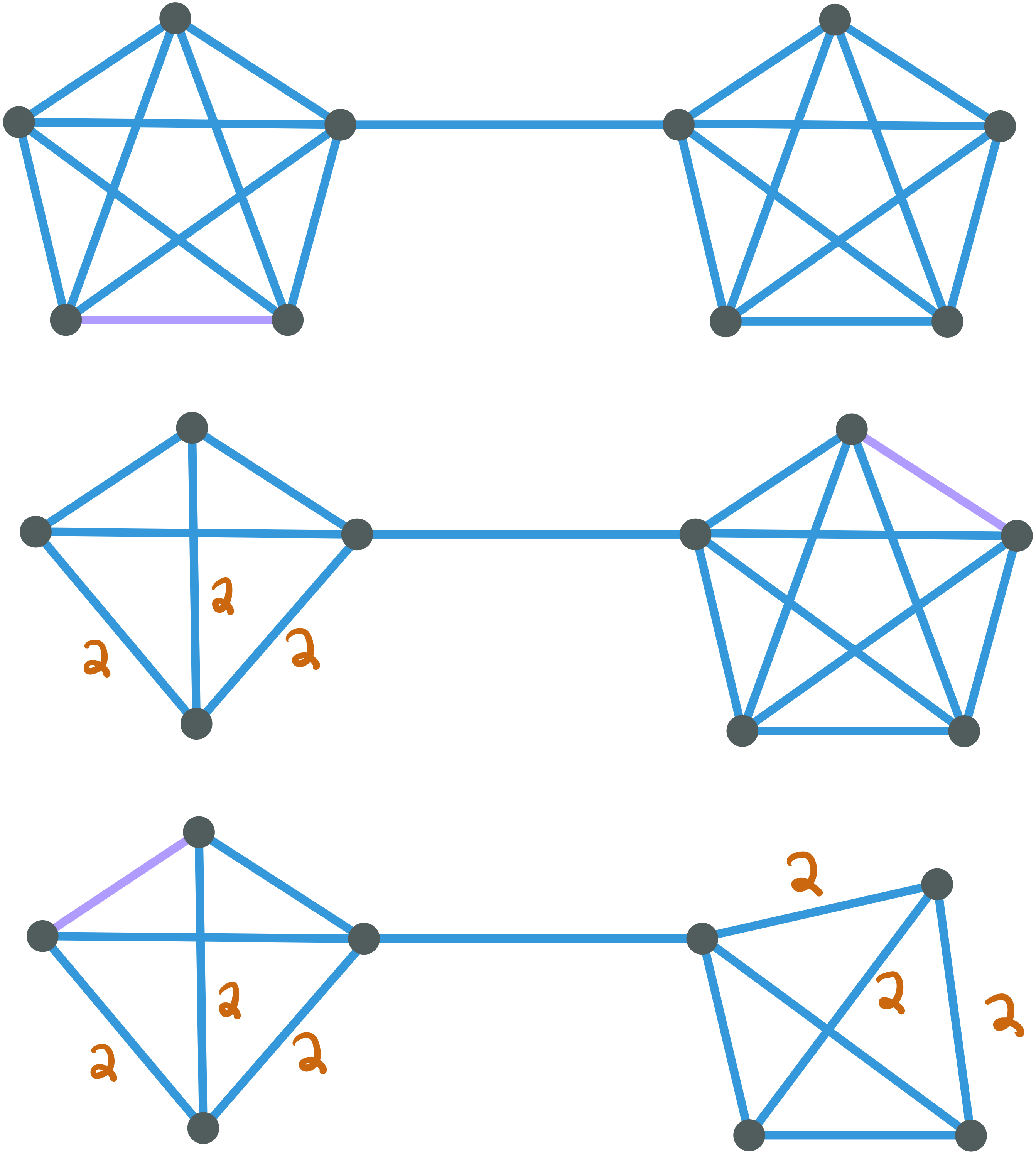
Ⓑ "contract" e

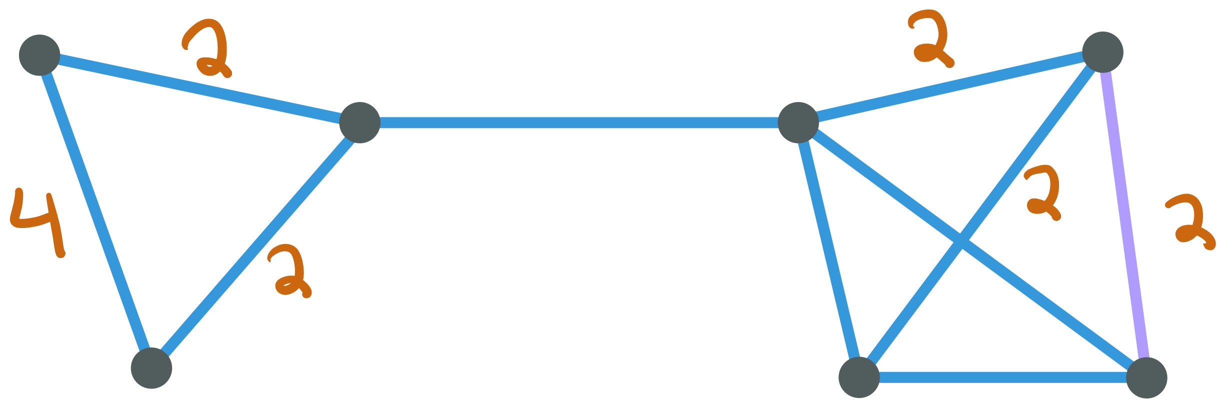
② return set of edges (in original graph) contracted to remaining edge "contracting an edge"



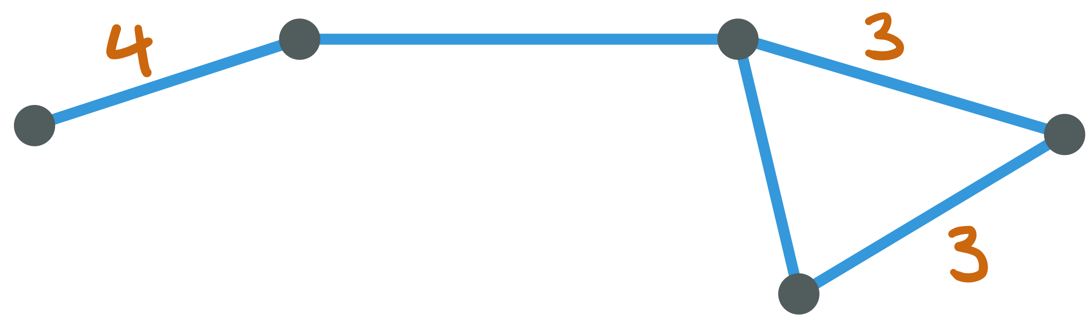
combine endpoints into a single vertex

sum capacities of parallel edges


2
2
2
2
2
2
2
2
2


2
4
2
2
2
2


2
4
2
3
3


4
3
3

Randomized contractions

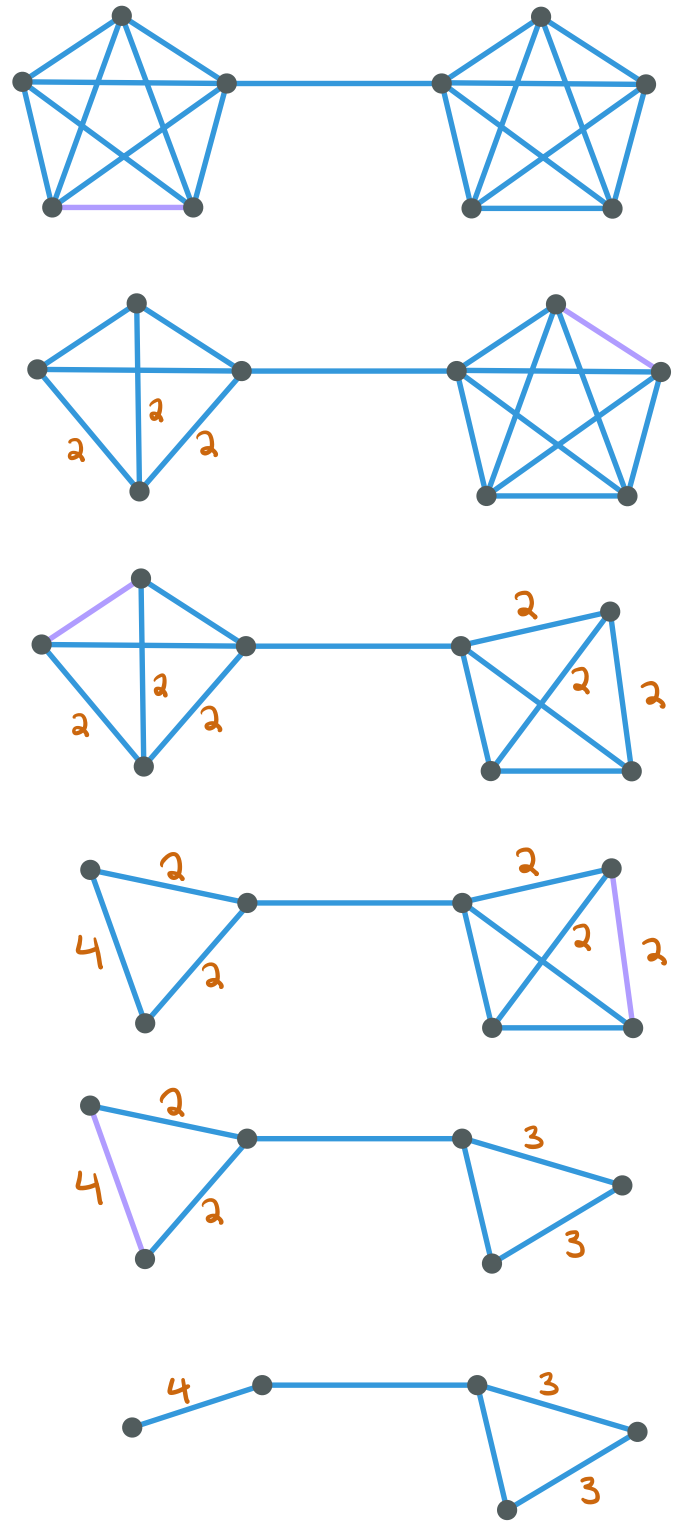
Input: $G=(V,E)$, weights $c \in \mathbb{R}^E_{>0}$ (capacities)

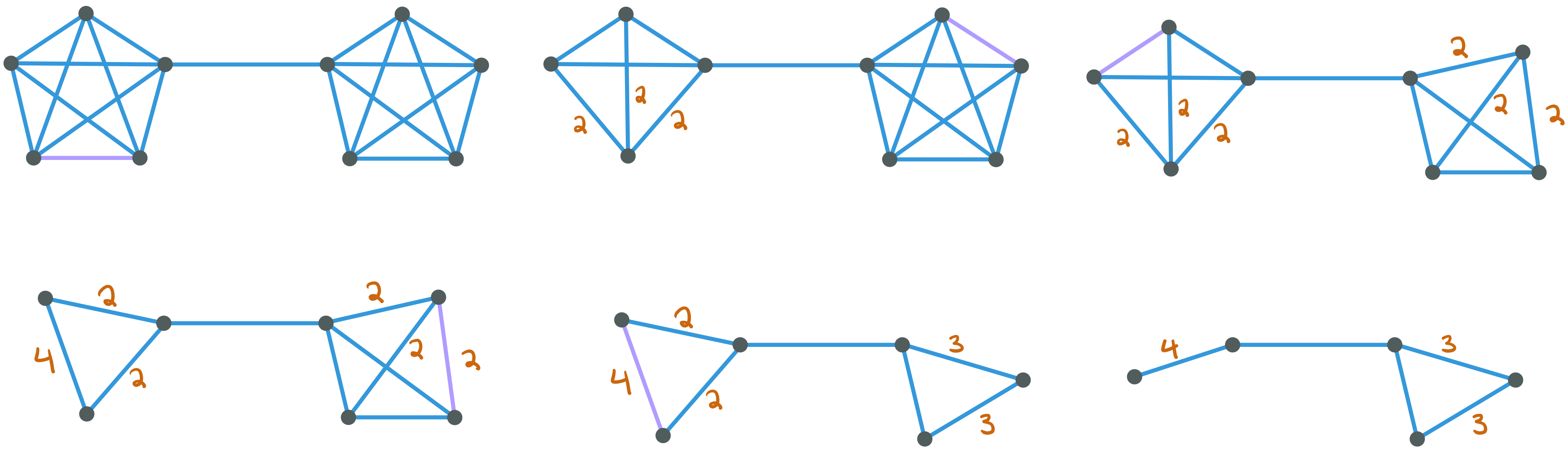
① while $|E|>1$

Ⓐ sample $e \in E$ in proportion to c

Ⓑ "contract" e

② return set of edges (in original graph) contracted to remaining edge



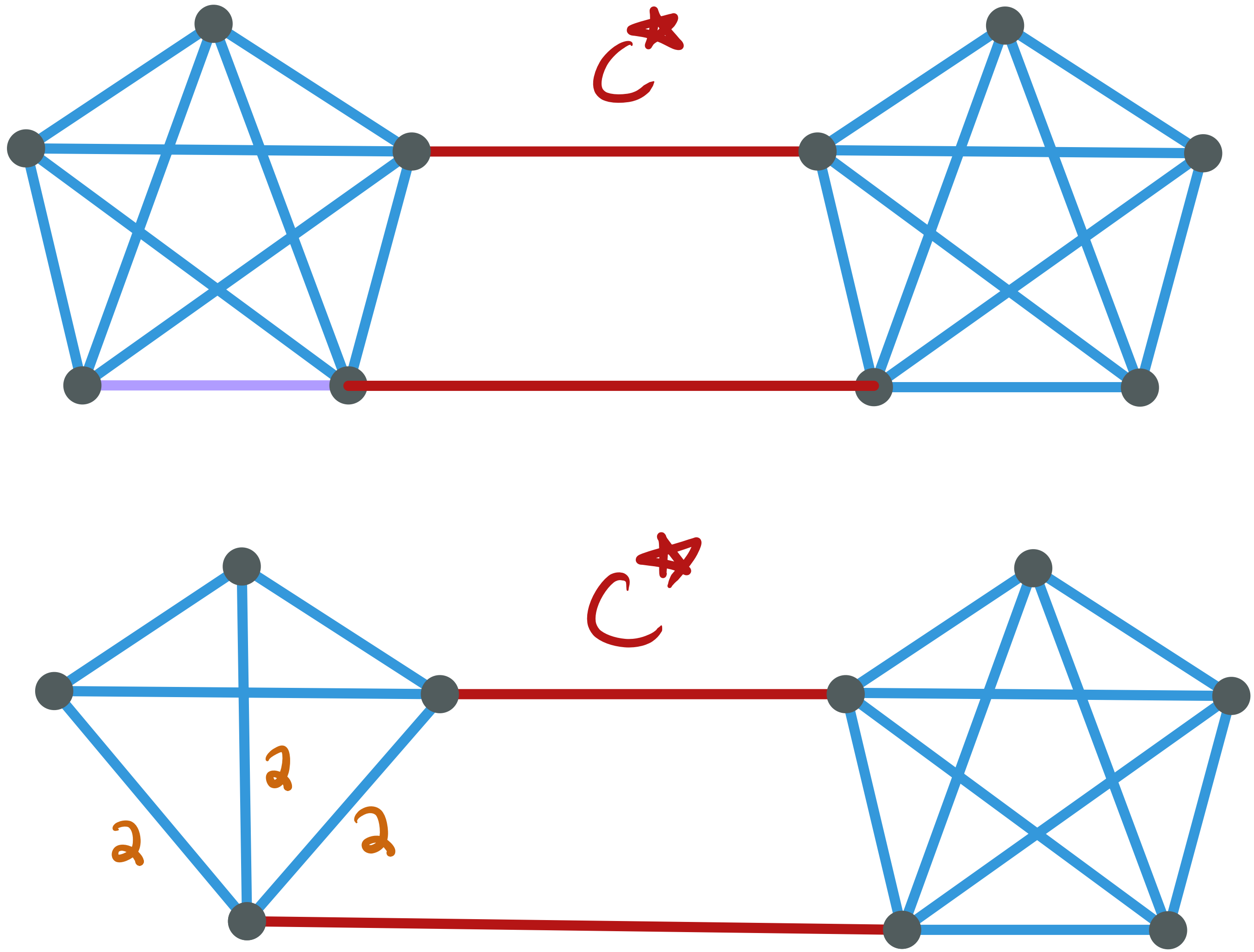


Analysis

Fix a min-cut C^*

$P[\text{output } C^*]$?

Obs. Contracting a non-min-cut edge preserves the min-cut.



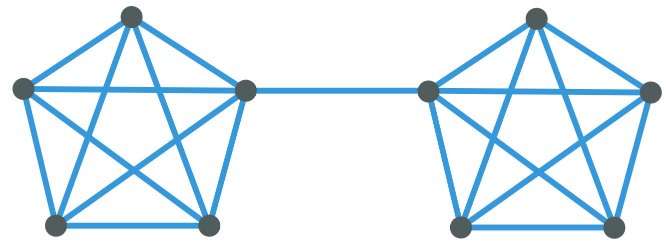
We output C^* iff every sampled edge is not in* C^*

* in the contracted image of C^*

Goal: analyze probability that each sampled edge avoids C^*

Let $\lambda = \sum_{e \in C^*} c(e)$ be capacity of min-cut

Lemma: $\sum_{e \in E} c(e) \geq \frac{\lambda n}{2}$

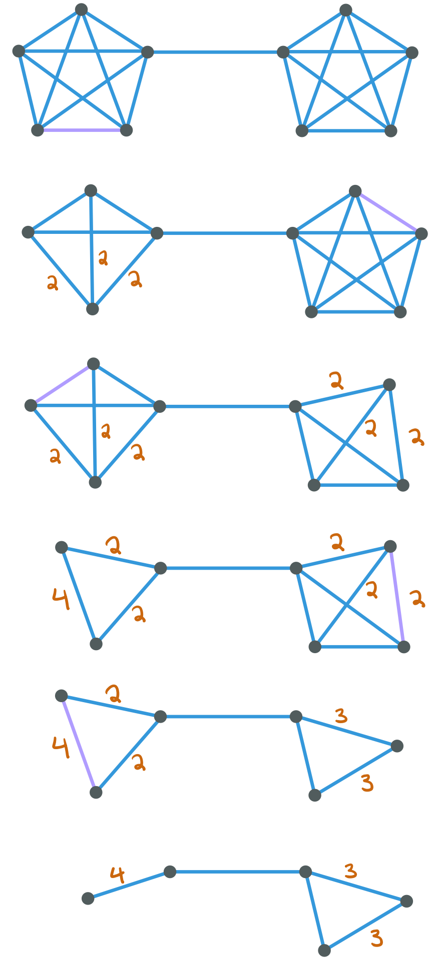


$$\sum_{e \in E} c(e) = \frac{1}{2} \sum_{v} \deg_c(v) \geq \frac{\lambda n}{2}$$

Lemma: $\sum_{e \in E} c(e) \geq \frac{\lambda n}{2}$

Lemma let $e \in E$ be sampled in proportion to c. Then $\Pr[e \in C^*] \leq \frac{2}{n}$

Theorem $\Pr[C^* \text{ returned}] \ge \frac{1}{\binom{n}{2}}$



E_k = event that C^* not sampled after k iterations

want to show $P[E_{n-2}] \ge \frac{2}{\binom{n}{2}}$

$$\Pr[E_2] = \Pr[E_2, E_1]$$
$$= \underbrace{\Pr[E_2 | E_1]}_{\left(1 - \frac{2}{n-1}\right)} \cdot \underbrace{\Pr[E_1]}_{\left(1 - \frac{2}{n}\right)}$$

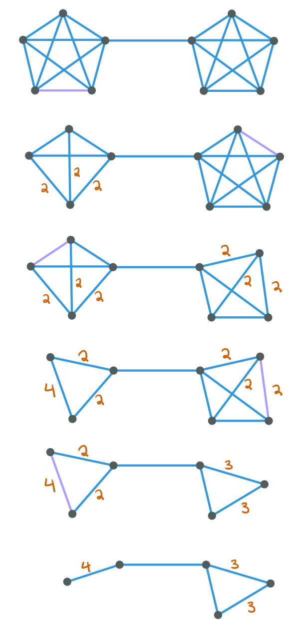
Lemma let $e \in E$ be sampled in proportion to c. Then $\Pr[e \in C^*] \le \frac{2}{n}$

$$P[E_{n-2}] = \Pr[E_{n-2} \mid E_{n-3}] \Pr[E_{n-3} \mid E_{n-4}] \cdots \Pr[E_1]$$

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

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

$$= \frac{2!\,(n-2)!}{n!} = \frac{1}{\binom{n}{2}}$$



Recall:

Lemma 2. let $e \in E$ be sampled in proportion to c.

Then $\Pr[e \in C^*] \le \frac{2}{n}$

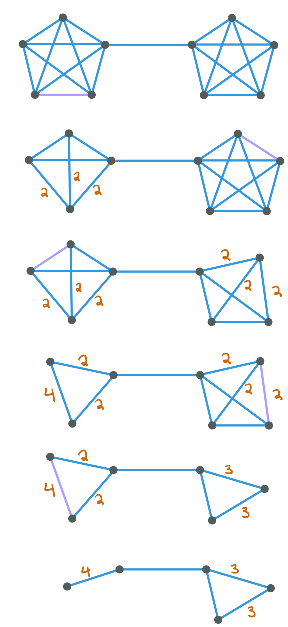
$$P[E_{n-2}] = P[E_1]\, P[E_2 \mid E_1] \cdots P[E_{n-2} \mid E_{n-1}]$$

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

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

$$= \frac{(n-2)!\,2!}{n!}$$

$$= \frac{1}{\binom{n}{2}}$$



Recall:

Lemma 2. let $e \in E$ be sampled in proportion to c.

Then $\Pr[e \in C^*] \leq \frac{2}{n}$

Question:

Max # min-cuts in a graph?

A

0

$O(1)$

constant

B

82%

$n^{O(1)}$

polynomial

17% died

C

0

$n^{\log^{O(1)} n}$

subexponential

D

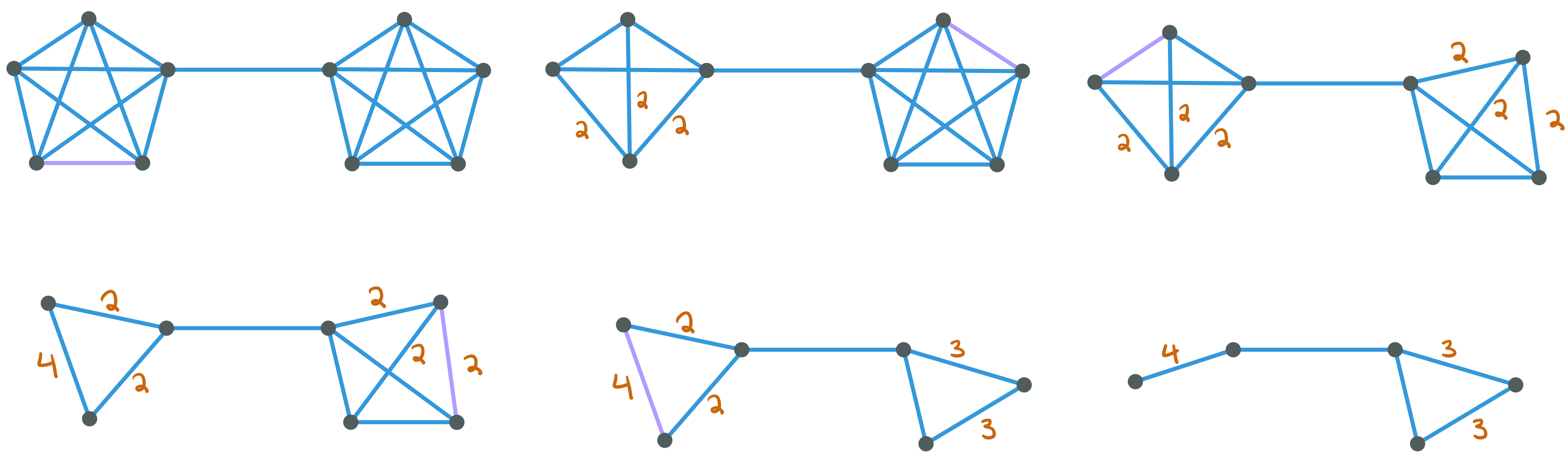
1

2^n

exponential

why?

any min-cut is output w/ prob $\geq \frac{1}{\binom{n}{2}}$



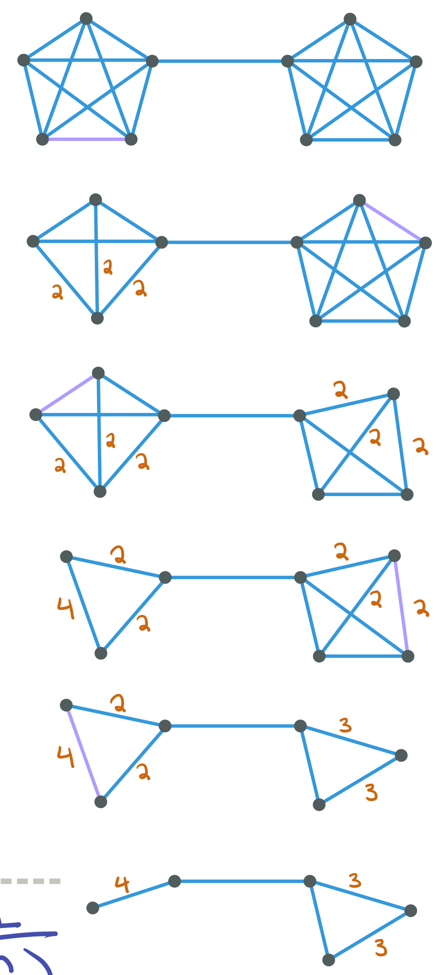
Random contractions algo takes $O(m)$ time

Rerun algo $O(n^2\log(n))$ times to ensure success w/h/p

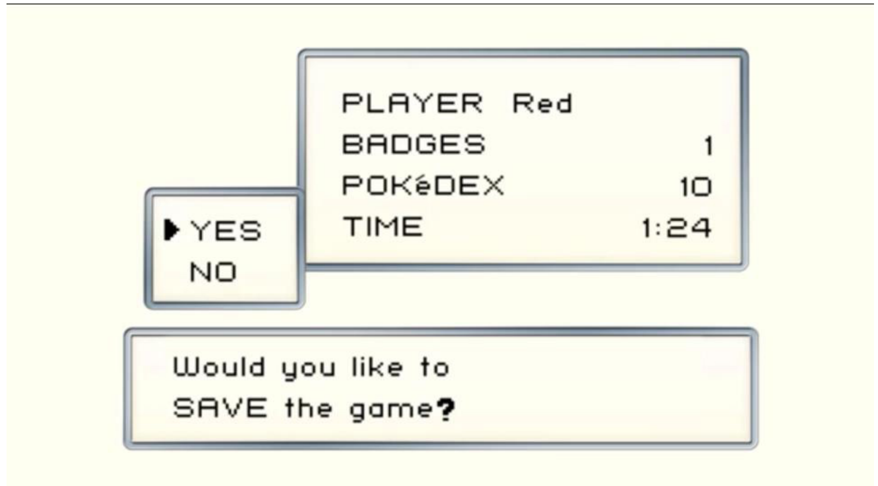
$\Rightarrow O(mn^2\log(n))$ running time

$$\left(1-\frac{1}{k}\right)^k \approx e^{-1}$$

Better?



Theorem $\Pr[C^* \text{ returned}] \geq \frac{1}{\binom{n}{2}}$


PLAYER Red
BADGES 1
POKéDEX 10
TIME 1:24
▶YES
NO
Would you like to
SAVE the game?

Amplification by branching

Instead of restarting at the beginning, restart partway through

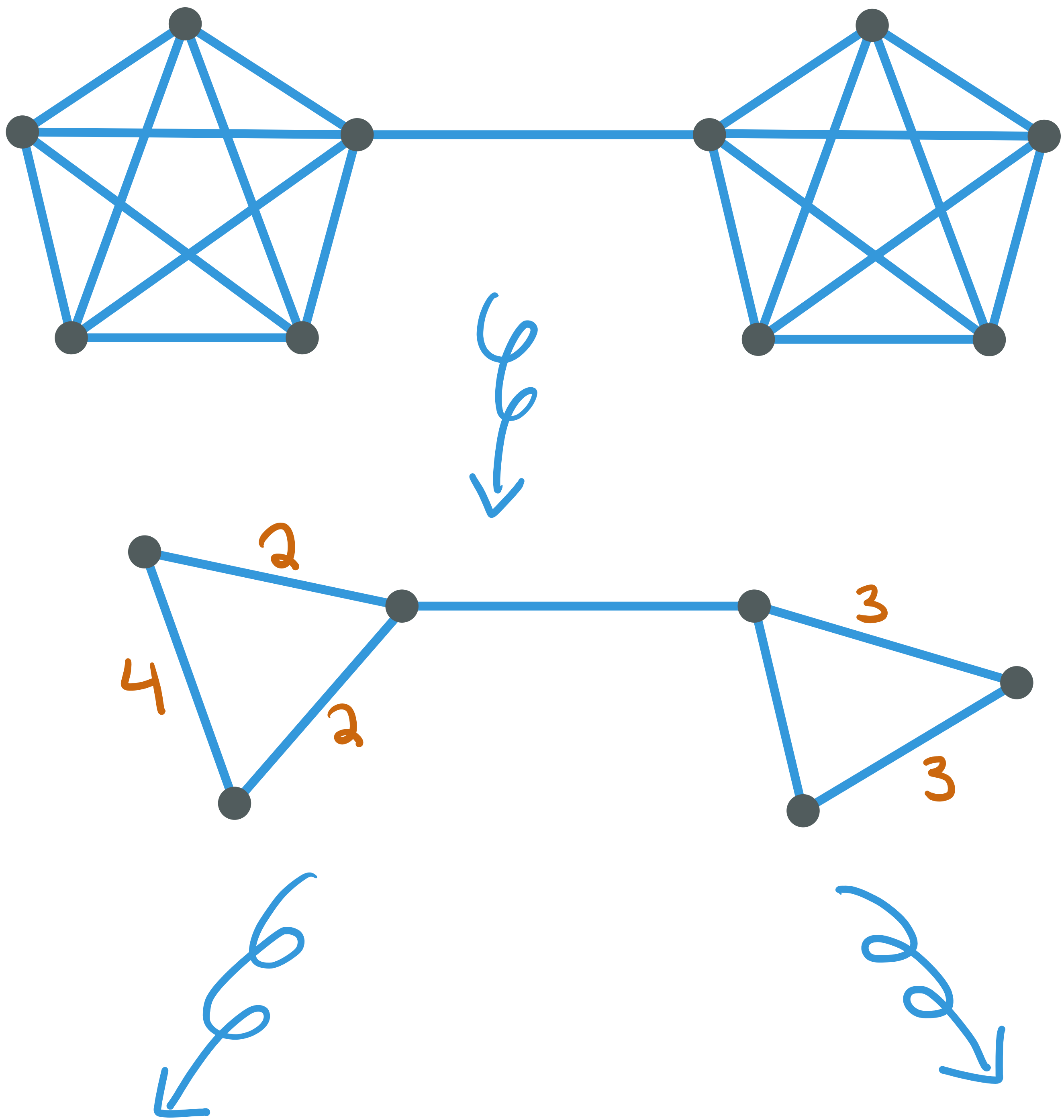
High-level algorithm

① randomly contract until prob. of ruining the min-cut is ~½

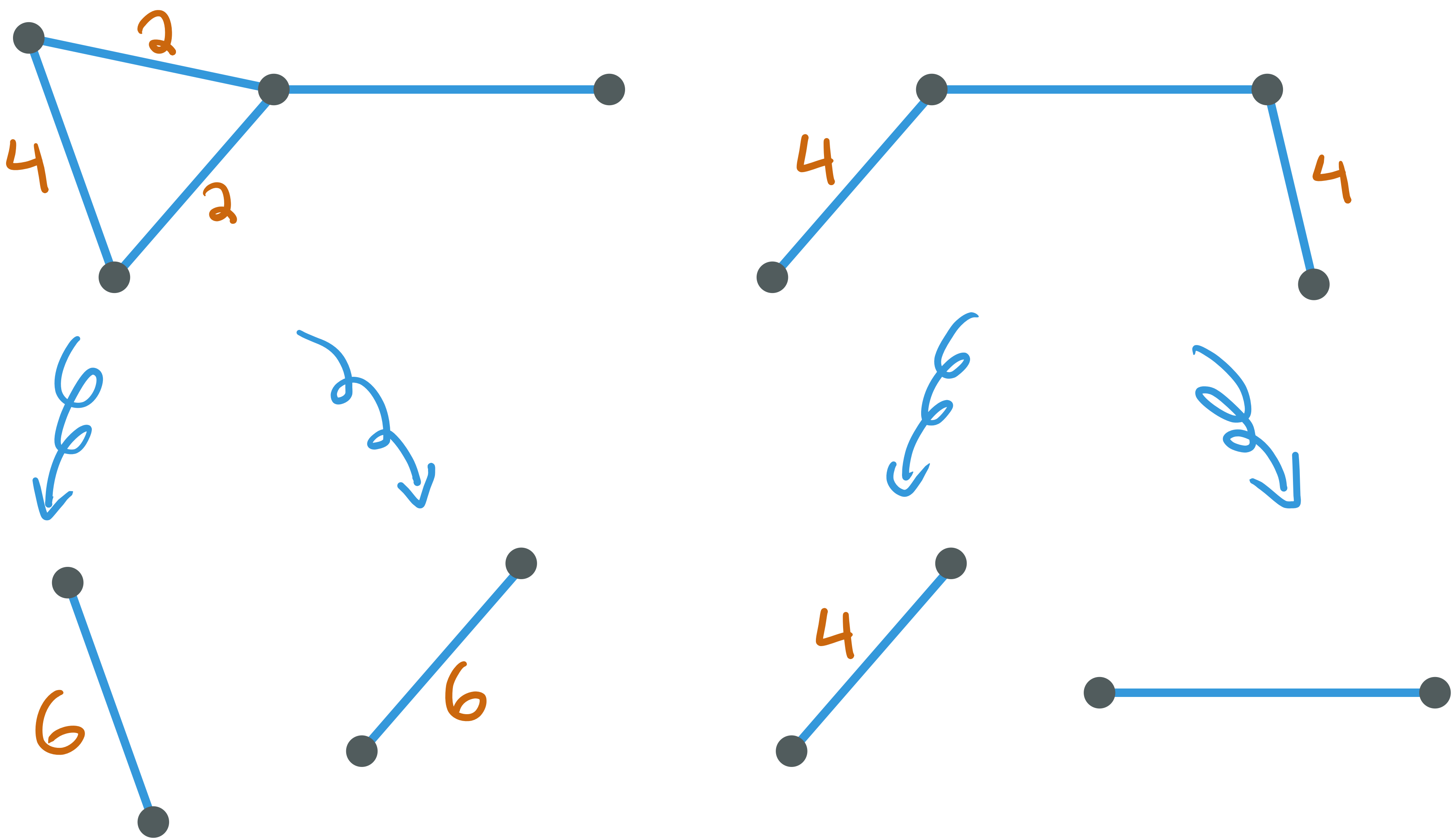
② *Branch:*

make two copies of current graph

recurse on both copies



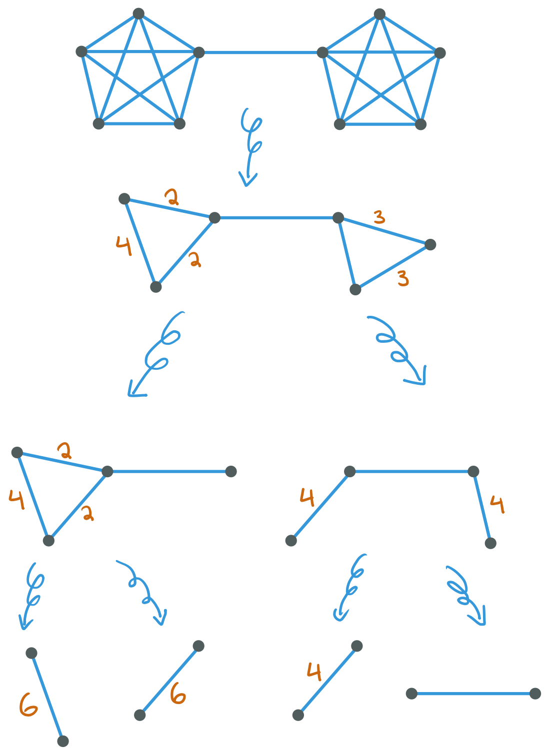
φ
2
4
2
3
3
φ
φ



2
4
2
4
4
6
6
4

Amplification by branching

Instead of restarting at the beginning, restart partway through



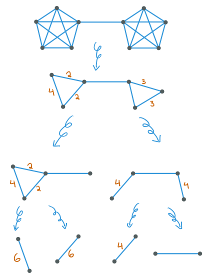
High-level algorithm

① randomly contract until prob. of ruining the min-cut is ~ ½

② Branch:

make two copies of current graph

recurse on both copies



High-level algorithm

① randomly contract until prob. of ruining the min-cut is ~ ½

② Branch:

make two copies of current graph

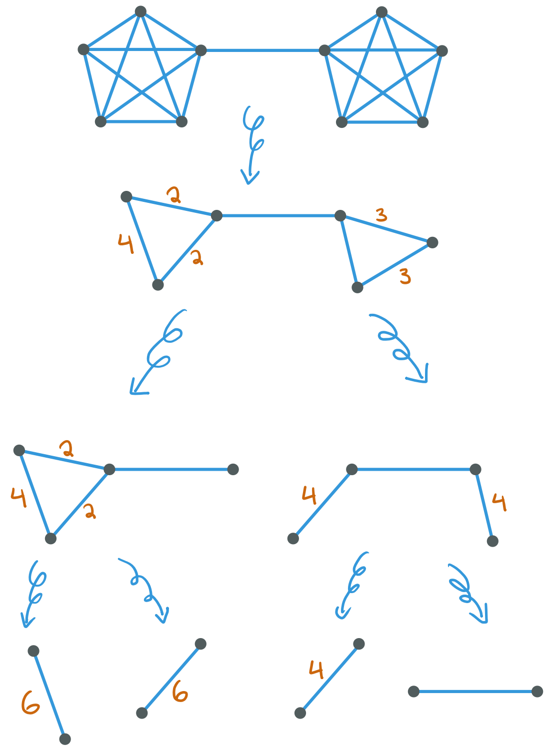
recurse on both copies

`branching-contractions`$(G = (V, E),\ c)$

1. Let $n = |V|$. If $n \leq 3$ then compute the minimum cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:

   A. Sample $e \sim c$ and contract e.

3. $C_1 \leftarrow$ `branching-contractions`(G, c)
4. $C_2 \leftarrow$ `branching-contractions`(G, c)
5. Uncontract and return the minimum of C_1 and C_2.



High-level algorithm

① randomly contract until prob. of ruining the min-cut is ~ ½

② *Branch:*

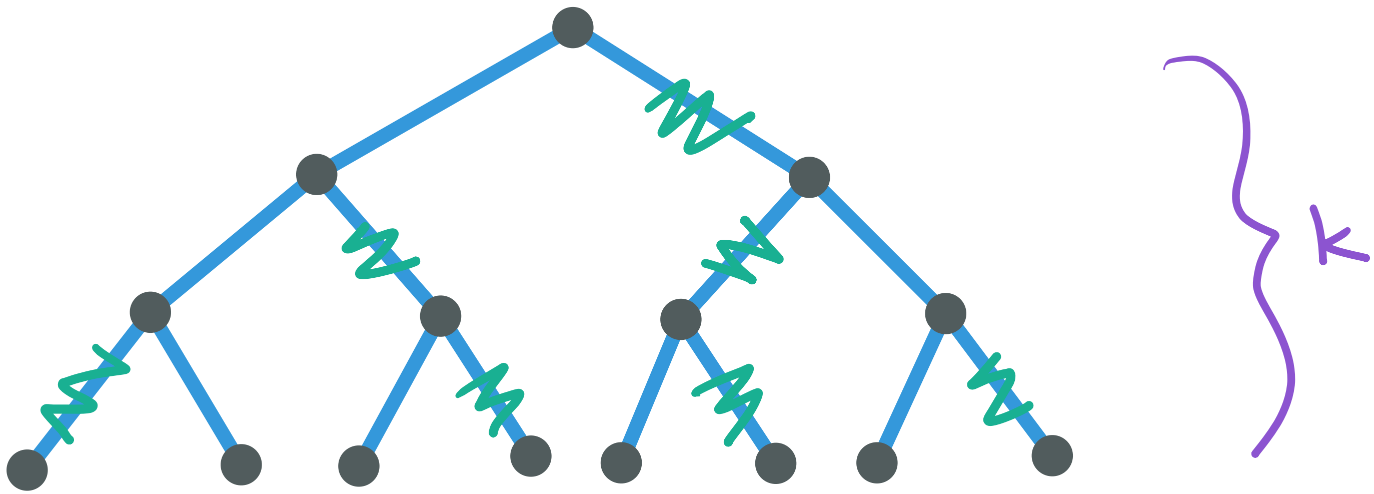
make two copies of current graph

recurse on both copies

Claim: branching-contractions returns min-cut w/ probability $\geq \frac{1}{O(\log n)}$

Claim: branching-contractions returns min-cut w/ prob. $\geq \frac{1}{O(\log n)}$

Need the following fact (proved later)



let T = complete binary tree of height k,
w/ every edge deleted w/ prob $\leq 1/2$

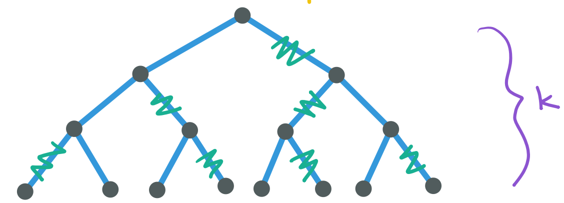
Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

"Galton-Watson process"

branching-contractions($G = (V, E)$, c)

1. Let $n = |V|$. If $n \leq 3$ then compute the min-cut by brute force.
2. Until $|V| = \lceil \frac{n}{\sqrt{2}} \rceil + 1$:
   A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ branching-contractions(G,c).
4. $C_2 \leftarrow$ branching-contractions(G,c).
5. Uncontract and return the minimum of C_1 and C_2.

"Galton-Watson process"



let T = complete binary tree of height k, w/ every edge deleted w/ prob $\leq 1/2$

Fact: P[some leaf connected to root] $\geq \frac{1}{k}$

Claim:

returns min-cut w/ prob. $\geq \frac{1}{O(\log n)}$

Proof:

High-level idea:

model recursive calls as tree

- nodes = subproblems
- edges = recursive subcalls
- delete parent edge of nodes/subproblems that fail before recursive call

Q: what is probability of edge being deleted

Fix $C^* =$ min-cut.

suppose we contract $n-k$ edges

Recall

$E_k =$ event that C^* not sampled after k iterations

$\Pr[E_{i+1} \mid E_i] = 1 - \frac{2}{n-i}$

$$\Pr[E_{n-k}] = \Pr[E_1]\Pr[E_2 \mid E_1] \cdots \Pr[E_{n-k} \mid E_{n-k-1}]$$

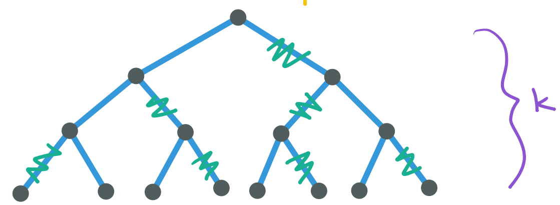
$$= \left(\frac{n-2}{n}\right) \cdot \left(\frac{n-3}{n-1}\right) \cdots \left(\frac{k-1}{k+1}\right)$$

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

`branching-contractions`$(G = (V, E),\ c)$

1. Let $n = |V|$. If $n \leq 3$ then compute the min-cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:
   A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ `branching-contractions`(G, c).
4. $C_2 \leftarrow$ `branching-contractions`(G, c).
5. Uncontract and return the minimum of C_1 and C_2.

"Galton-Watson process"



let $T =$ complete binary tree of height k, w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

$$k = \lceil n/\sqrt{2} \rceil + 1 \Rightarrow P[E_{n-k}] \geq \frac{1}{2}$$

Claim: returns min-cut w/ prob. $\geq \frac{1}{O(\log n)}$

Proof:

High-level idea:

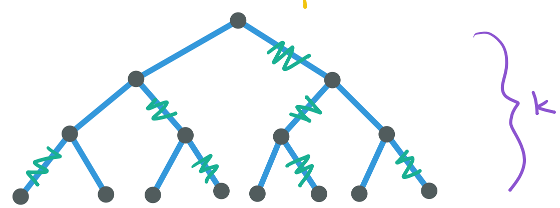
model recursive calls as tree

- nodes = subproblems
- edges = recursive subcalls
- delete parent edge of nodes/subproblems that fail before recursive call $\Big\}$ Prob. $\leq 1/2$

branching-contractions($G=(V,E)$, c)

1. Let $n = |V|$. If $n \leq 3$ then compute the min-cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:
   A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ branching-contractions(G,c).
4. $C_2 \leftarrow$ branching-contractions(G,c).
5. Uncontract and return the minimum of C_1 and C_2.

"Galton-Watson process"



let T = complete binary tree of height k, w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

$P[E_{n-n/\sqrt{2}}] \geq \frac{1}{2}$

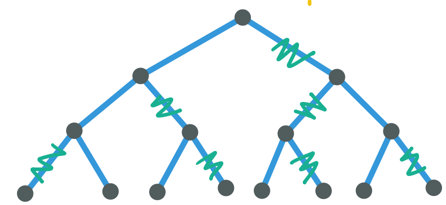
model recursive calls as tree

- nodes = subproblems
- edges = recursive subcalls
- delete parent edge of nodes/subproblems that fail before recursive call } Prob. $\leq 1/2$

`branching-contractions`$(G=(V,E),\ c)$

1. Let $n=|V|$. If $n \leq 3$ then compute the min-cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:
   A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ `branching-contractions`(G,c).
4. $C_2 \leftarrow$ `branching-contractions`(G,c).
5. Uncontract and return the minimum of C_1 and C_2.

"Galton-Watson process"

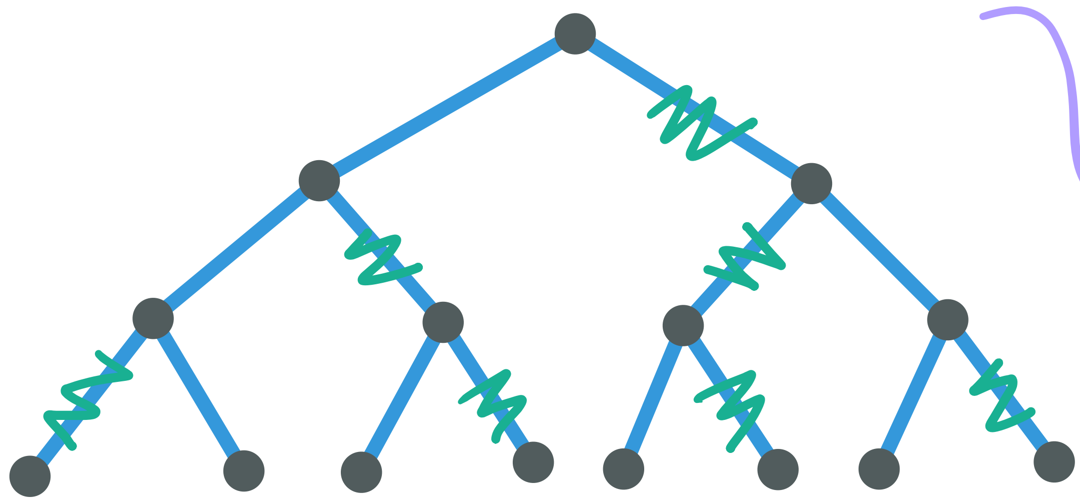


} k

let T = complete binary tree of height k, w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

$P[E_{n-n/\sqrt{2}}] \geq \frac{1}{2}$



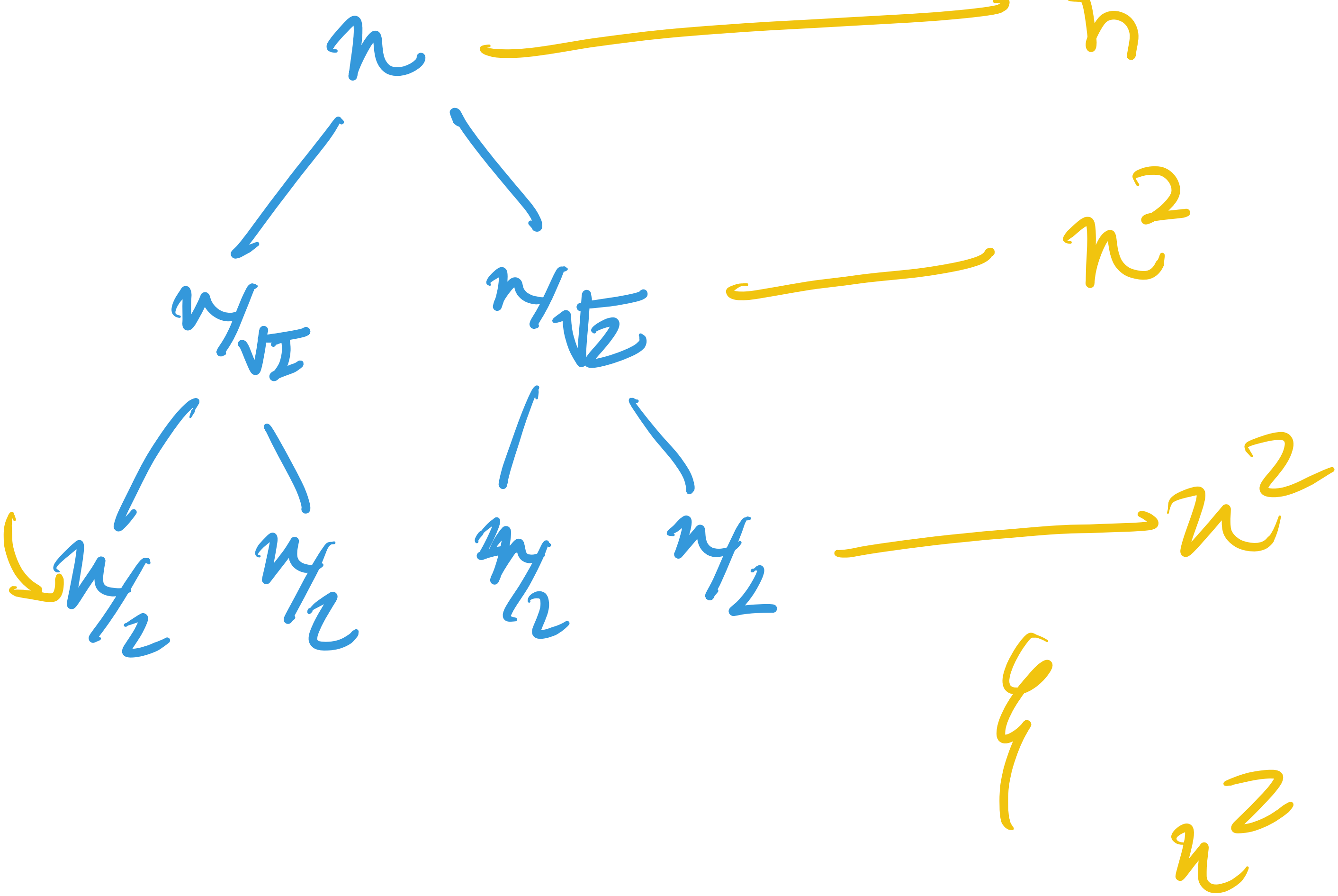
} height $= O(\log n)$

algo correct <=> root-to-leaf path

Fact => prob success $\geq \Omega(1/\log n)$

Running time

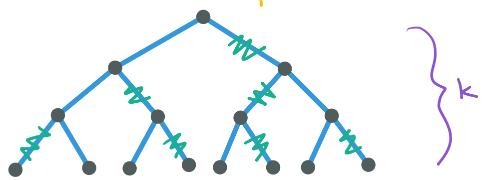
$$T(n) \le 2T\left(\frac{n}{\sqrt{2}} + 1\right) + n^2$$



`branching-contractions`$(G = (V, E),\ c)$

1. Let $n = |V|$. If $n \le 3$ then compute the min-cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:
   A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ `branching-contractions`(G, c).
4. $C_2 \leftarrow$ `branching-contractions`(G, c).
5. Uncontract and return the minimum of C_1 and C_2.

"Galton-Watson process"



let T = complete binary tree of height k, w/ every edge deleted w/ prob $\le 1/2$

Fact: P[some leaf connected to root] $\ge \frac{1}{k}$

Running time

$$T(n) \le 2T\left(\frac{n}{\sqrt{2}}+1\right) + n^2$$

$$n \qquad\qquad n^2$$

$$n/\sqrt{2} \quad n/\sqrt{2} \qquad\qquad 2(n/\sqrt{2})^2$$

$$n/2 \quad n/2 \quad n/2 \quad n/2 \qquad\qquad 4(n/2)^2 = O(n^2)$$

$$\vdots$$

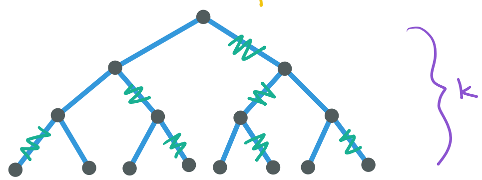
$$+\; 2^i\left(n/2^{i/2}\right)^2 = O(n^2)$$

$$O(n^2 \log n)$$

`branching-contractions`$(G=(V,E),\ c)$

1. Let $n = |V|$. If $n \le 3$ then compute the min-cut by brute force.
2. Until $|V| = \left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1$:
   - A. Sample $e \sim c$ and contract e in G.
3. $C_1 \leftarrow$ `branching-contractions`(G, c).
4. $C_2 \leftarrow$ `branching-contractions`(G, c).
5. Uncontract and return the minimum of C_1 and C_2.

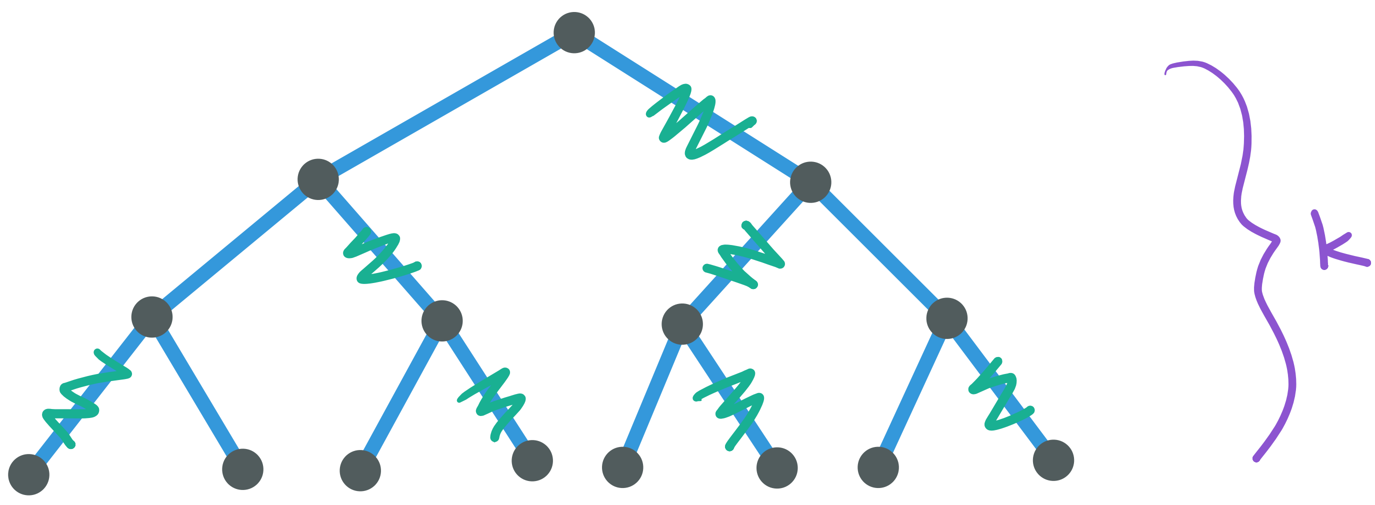
"Galton-Watson process"



let T = complete binary Tree of height k, w/ every edge deleted w/ prob $\le 1/2$

Fact: $P[\text{some leaf connected to root}] \ge \frac{1}{k}$

Now we prove the fact:



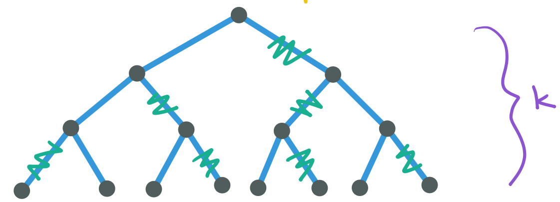
let T = complete binary tree of height k,
w/ every edge deleted w/ prob $\leq 1/2$

Fact: P[some leaf connected to root] $\geq \frac{1}{k}$

"Galton-Watson process"

$p_i \overset{\text{def}}{=}$ prob that a particular node at level i has path to a subleaf

"Galton-Watson process"

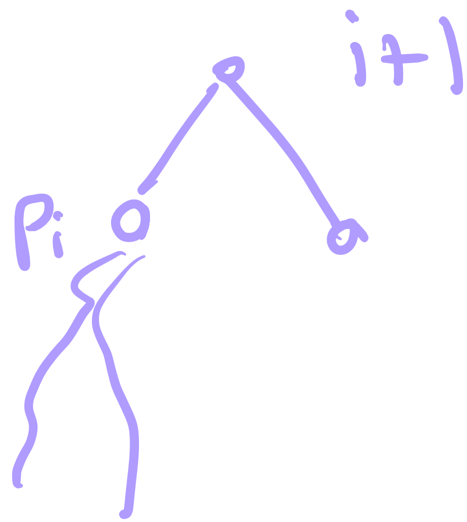


let T = complete binary tree of height k,
w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

$p_0 = 1$

$p_{i+1} =$



$p_i \overset{\text{def}}{=}$ prob that a particular node at level i has path to a subleaf

$$p_0 = 1$$

$$p_{i+1} = 1 - \left(\frac{1}{2} + \frac{1}{2}(1-p_i)\right)^{2}$$

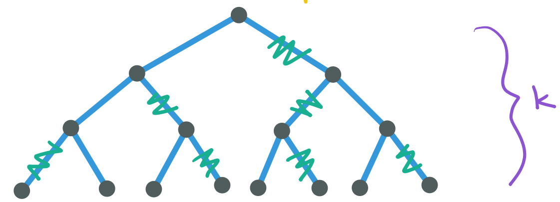
$$= 1 - \left(1 - \frac{p_i}{2}\right)^2 = p_i\left(1 - p_i/4\right)$$

$$p_0 = 1$$

$$p_1 = \frac{3}{4}$$

$$p_2 = \frac{39}{64} \geq \frac{1}{2}$$

"Galton-Watson process"



let T = complete binary tree of height k, w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

$p_{i+1} = p_i(1 - p_i/4)$

claim: $p_k \geq 1/k \quad \forall k$

$p_0 = 1, \quad p_1 = 3/4, \quad p_2 \geq \frac{39}{64}$

let $f(x) = x(1 - x/4)$

f is increasing for $x \leq 2$

$$p_{i+1} = p_i(1 - p_i/4)$$

claim: $p_k \geq 1/k \quad \forall k$

$p_0 = 1, \quad p_1 = 3/4, \quad p_2 \geq \frac{39}{64}$

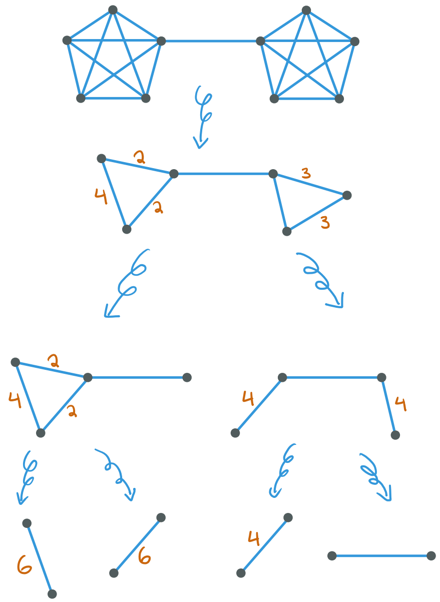
let $f(x) = x(1 - x/4)$

f is increasing for $x \leq 2$

$$p_{k+1} \geq \frac{1}{k}\left(1 - \frac{1}{4k}\right) = \frac{1}{k} - \frac{1}{4k^2} \overset{(*)}{\geq} \frac{1}{k+1}$$

$$(*) \quad \frac{1}{k} - \frac{1}{k+1} = \frac{1}{k^2+1} \geq \frac{1}{4k^2}$$

All put together:



High-level algorithm

① randomly contract until prob. of ruining the min-cut is ~ 1/2

② Branch:

make two copies of current graph

recurse on both copies

$O(n^2 \log n)$ time

succeeds w/ probability $\Omega\left(\frac{1}{\log n}\right)$

repeat $O(\log n)$ times to get constant prob.