CS 5330: Randomized Algorithms

Week 1
Jan 15 '19

Puzzle of the Day: Three Dice

A: 1, 1, 6, 6, 8, 8
B: 2, 2, 4, 4, 9, 9
C: 3, 3, 5, 5, 7, 7

Rules:
1) Alice chooses a die.
2) Bob chooses a die.
3) Both roll. Higher number wins.

What is probability Alice/Bob wins?
[Assume both choose optimally.]

Note: $E[A] = E[B] = E[C]$.

Why randomized algorithms?

Magic!!
Choose a random
answer and
shazam!

→ Faster

E.g., Sampling
"How many Singaporeans like soup?"
Run a poll...

or

Find a MST

→ Simpler

Decide if p is prime.
Polynomial identity testing
Nuts-and-bolts

→ Other

Existence proofs
Derandomization
Security /Privacy

Goals

1. Tools and techniques to help YOU!

2. Neat algorithms!

Topics

1) Classical algorithms
   - QuickSort
   - Treaps
   - Min Cut
   - Etc.

2) Hashing
   - Chaining
   - Open Addressing
   - Cuckoo Hashing
   - Bloom Filter
   - Tabulation Hashing

3) Sampling
   - Polling
   - Data analysis

4) Markov Chains \ Random Walks
   - MCMC/Metropolis

5) Other

Load balancing
Backoff protocols
Expert algorithms
Preserving privacy

Admin:

① Weekly problem sets

② Experimental Evaluation

③ Midterm (toward end of semester)

④ Project

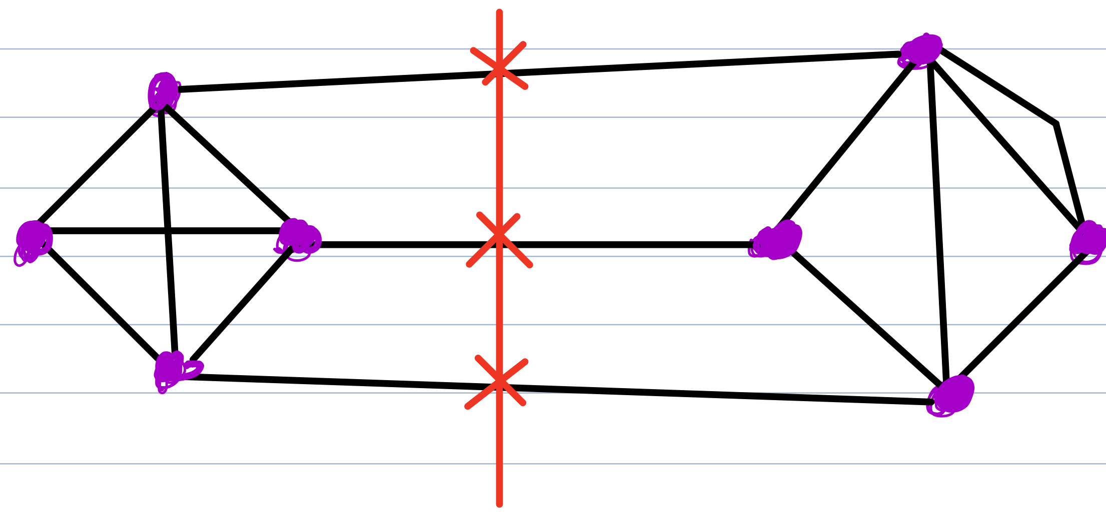
Book: Mitzenmacher, Upfal:
Probability and Computing

Assumed Background:

1) Basic Probability → $E[X]$, $Var[X]$, $Pr(X|Y)$
Linearity of expectation
Independence, etc, etc

2) Undergrad Algorithms → Sorting, graphs
MST, Shortest paths
Recurrences, big-O, etc

Randomized Min Cut



Graph $G = (V, E)$ $\quad$ n nodes
m edges

Cut: divedes nodes V into 2 sets V_1, V_2
$V = V_1 \cup V_2$

Size of cut = # edges from V_1 to V_2
$\text{size}(V_1, V_2) = |\{(u,v) : u \in V_1,\ v \in V_2\}|$

Goal: find min-sized cut

[Network design, relicbility, etc]

Solutions:

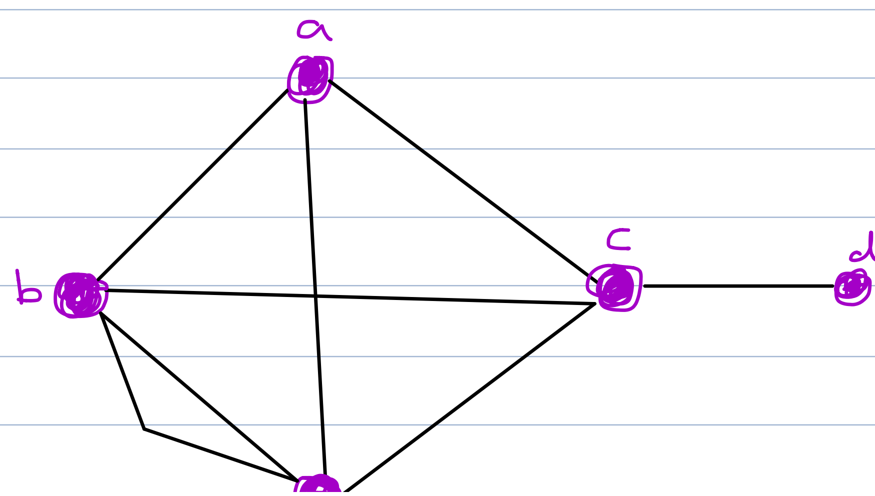
1) Enumerate all cuts: 2^n possible cuts

2) Max-Flow / Min-cut: Reasonable algs: $O(n^3)$
Relatively complicated

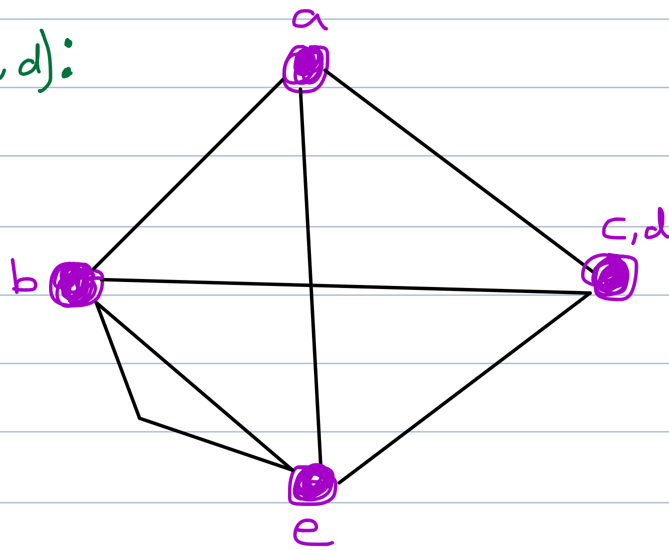
Today Karger (and Stein) Min Cut
Beautiful, simple algorithm
Magic!

Basic idea: Repeat until only 1 cut left:
① Choose $e \in E$ u.a.r.
② Exclude e.
uniformly at random

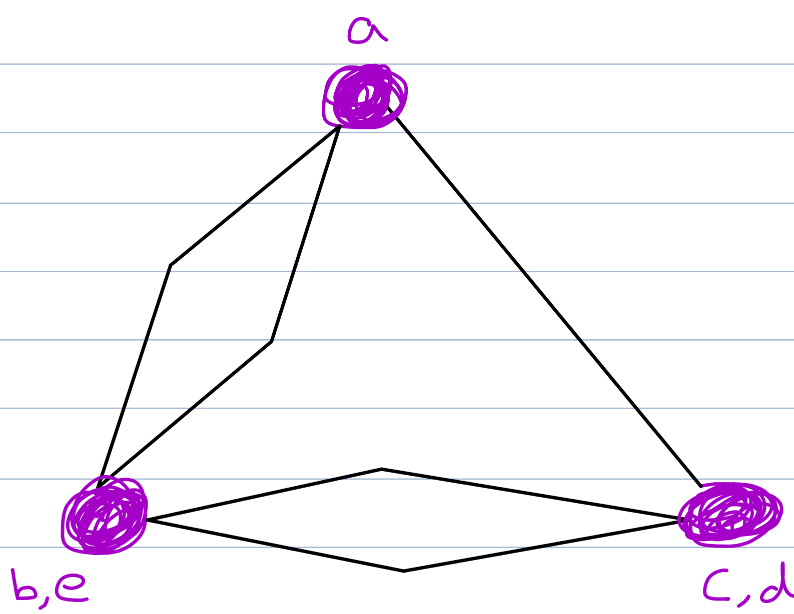
Operation: edge contraction



Contract (c, d):



Contract (b, e):



On contract(u,v):

1) Create new node uv or (u,x) or (v,x)

2) Replace every edge (X,u) or (X,v) with (X, uv)

3) Delete all edges (u,v) or (v,u)

Keep duplicate edges!

Alg: Collapse(G, n, t)

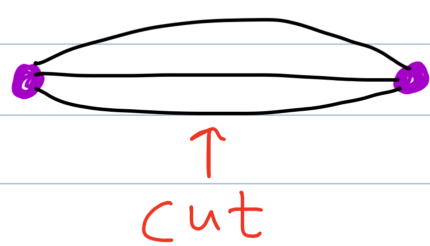
Repeat n-t times:

a) Choose edge e u.a.r.

b) Contract(e)

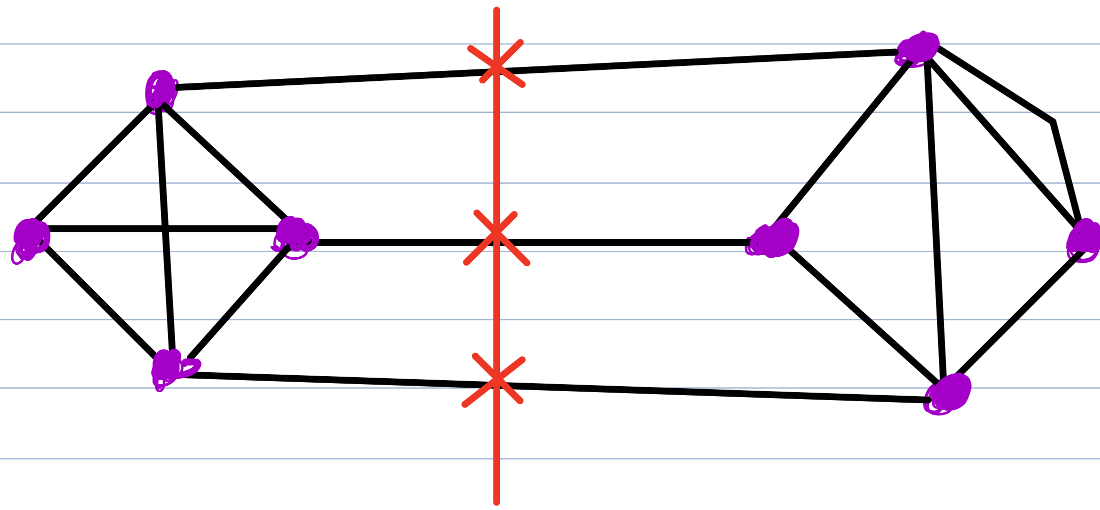
Return G.

After Collapse(G,n,2): 2 nodes left!



Return only cut.

Why might this work?



→ Works if no edge in cut is chosen

Ex: $|\text{Cut}| = 3$, $m = 15$ $\Rightarrow$ $1/5$ choose edge in cut

→ min cut is, by definition, small!
Unlikely to choose...

Lemma: if size of min cut is k, then $m > nk/2$

Proof: Every node has degree $\geq K$.

(Otherwise cut $< K$, ie, just that node.)

$$m = \frac{1}{2} \sum_{u \in V} \deg(u)$$ ← counts every edge twice

$$\geq \frac{1}{2} \sum_{u} K$$

$$\geq \frac{1}{2} n K$$

Let C = a min cut.

$K = |C|$

$$\Pr(\text{Choose edge in } C \text{ in step } 1) = K/m$$

$$\leq K/(nK/2)$$

$$\leq 2/n$$

After step 1? Assume $e \notin C$:

→ Cut C unchanged

→ $n \to n-1$ (one fewer node)

→ min cut still K ← prove it!!

$\forall u: \deg(u) \geq K$

→ # edges $\geq (n-1)K/2$

$$\Pr(\text{choose } e \in C \mid \text{step 1 good}) = \frac{K}{m'} \le \frac{K}{(n-1)K/2}$$

$$\le \frac{2}{n-1}$$

$$\Pr(\text{choose } e \in C \mid \text{step 1,2 good}) \le K \Big/ (n-2)K/2$$

$$\le \frac{2}{n-2}$$

Define: E_j = event do <u>not</u> choose edge $e \in C$ in iteration j

$$\boxed{\begin{aligned}\Pr(E_j \mid E_1, E_2, \dots, E_{j-1}) &\le K \Big/ (n-j+1)K/2 \\ &\le \frac{2}{n-j+1}\end{aligned}}$$

Review: $\Pr(A \overset{\text{and}}{\wedge} B) = \Pr(A)\cdot\Pr(B \mid A)$

Chain rule (induction):

$$\Pr(X_1 \wedge X_2 \wedge \dots \wedge X_t) = \Pr(X_1)\Pr(X_2 \mid X_1)\Pr(X_3 \mid X_1, X_2)\cdots\Pr(X_t \mid X_1, X_2, \dots, X_{t-1})$$

Pr(no edge chosen in C)

$$= \Pr(E_1 \wedge E_2 \wedge \cdots \wedge E_{n-3})$$

$$= \Pr(E_1)\Pr(E_2 \mid E_1)\Pr(E_3 \mid E_1, E_2)\cdots\Pr(E_{n-3} \mid E_1, \dots, E_{n-4})$$

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

$$= \left(\frac{\cancel{n-2}}{n}\right)\left(\frac{\cancel{n-3}}{n-1}\right)\left(\frac{\cancel{n-4}}{\cancel{n-2}}\right)\left(\frac{\cancel{n-5}}{\cancel{n-3}}\right)\cdots\left(\frac{\cancel{3}}{\cancel{5}}\right)\left(\frac{2}{\cancel{4}}\right)\left(\frac{1}{\cancel{3}}\right)$$

$$= \prod_{j=3}^{n} \frac{j-2}{j}$$

$$= 2/(n)(n-1) \qquad = \Omega(1/n^2)$$

Is that good enough?

Amplification

some constant

KargerCut:

Run Collapse$(G, n, 2)$ $\frac{c \cdot n^2 \cdot \ln(n)}{2}$ times

Return smallest cut.

→ each run is independent

→ $Pr(\text{run fails}) \leq \left(1 - \frac{2}{n^2}\right)$

$Pr(\text{KargerCut fails}) = Pr(\text{all runs fail})$

$$\leq \left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2} \cdot c \ln(n)}$$

$$\leq e^{-c \ln(n)}$$

$$\leq \frac{1}{n^c}$$

Fact $\frac{1}{4} \le \left(1-\frac{1}{x}\right)^x \le e^{-1}$ very useful!!

$$e^x = 1 + x + \frac{x^2}{2} + \cdots \ge 1 + x$$

$$e^{-x} = 1 - x + \frac{x^2}{2} - \cdots \ge 1 - x$$

$$\le 1 - x + \frac{x^2}{2} \le 1 - \frac{x}{2} \text{ if } x \le 1$$

$$\Rightarrow e^{-1} \ge (1-x)^{1/x}$$

"With high probability" $\Rightarrow \ge 1 - 1/n^c$

for any constant c

Error is polynomially small

$\rightarrow$ as small as you want $\rightarrow$ choose c

$\rightarrow$ pay for it in algo. cost

Error scales with problem:

$\rightarrow$ Bigger n, smaller error

"With $\ge 99\%$ Probability, you don't fail"

$\Rightarrow$ $n = 10$: Probably no one fails

$n = 100$: about 1 fails

$n = 1000$: about 10 fail

"With $\ge 1 - \frac{1}{n^2}$ Probability, you don't fail"

$\Rightarrow$ $\forall n$, no one fails

Conclusion: Karger Cut finds a min cut
with high probability: $Pr \geq 1 - \frac{1}{n^c}$

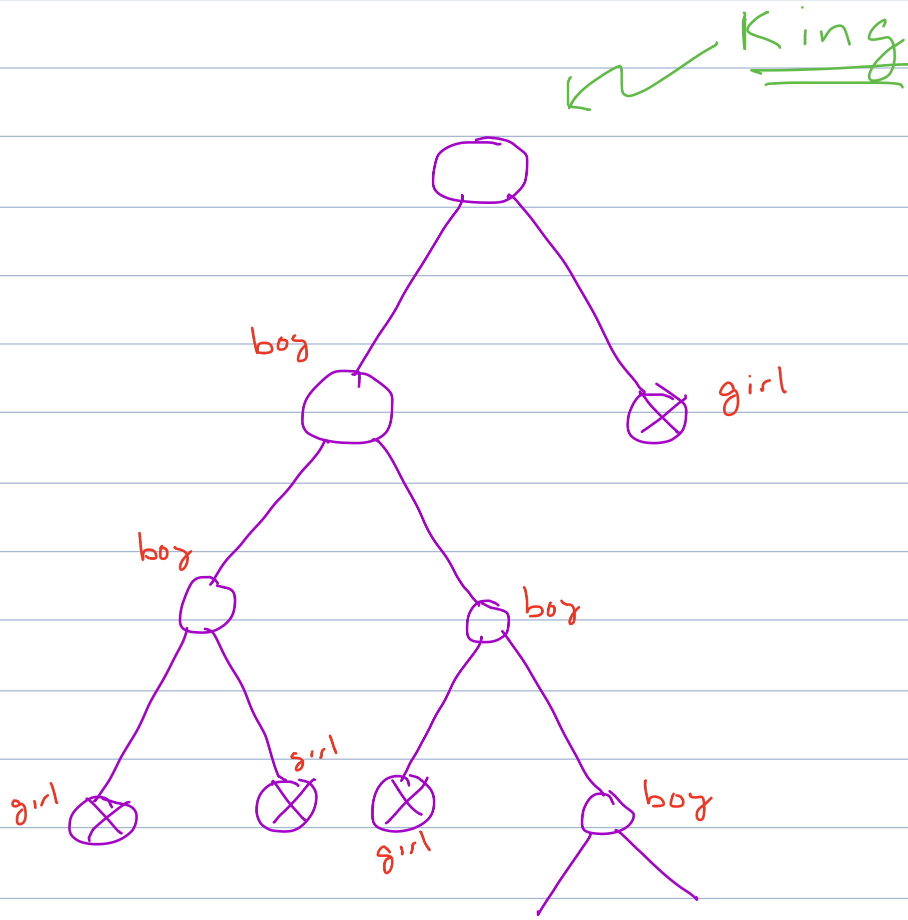
Time: Each iteration:

1) Choose random edge: $O(n)$
2) Contract: $O(n)$

Total cost: $O(n^4 \log n)$

Faster version to come...

Branching Process

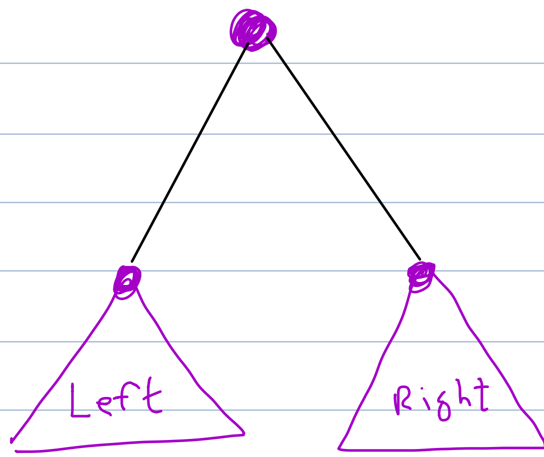


King wants line of male descendents [Sorry, Kings were sexist]

Rules:

1) Every person has $\underline{2}$ children
2) $Pr[boy] = 1/2$

Question: how likely is the King to have d generations of male lineage?



Let P_d = Prob. of d generations

P_d = Pr (Left boy has $d-1$ gen. $\longrightarrow \frac{1}{2} \cdot P_{d-1}$

or

Right boy has $d-1$ gen.) $\longrightarrow \frac{1}{2} P_{d-1}$

Recall (inclusion-exclusion): $Pr(A \text{ or } B) = Pr(A) + Pr(B) - Pr(A \wedge B)$

independent children

$$P_d = \frac{1}{2} P_{d-1} + \frac{1}{2} P_{d-1} - \frac{1}{4} P_{d-1}^2$$

$$= P_{d-1} - \frac{1}{4} P_{d-1}^2$$

Change of variables: $z_d = 1/p_d$ $\qquad p_d \le 1 \Rightarrow z_d \ge 1$

$p_1 = 1 \Rightarrow z_1 = 1$

$$\frac{1}{z_d} = \frac{1}{z_{d-1}} - \frac{1}{4z_{d-1}^2} = \frac{4z_{d-1} - 1}{4z_{d-1}^2}$$

$$z_d = \frac{4z_{d-1}^2}{4z_{d-1} - 1} \le z_{d-1} + 1$$

$(z+1)(4z-1) = 4z^2 + 4z - z - 1$
$\ge 4z^2$ if $z \ge 1$

$\le d$

$\Rightarrow p_d \ge 1/d$

Chance of surviving 10 generations $\ge 10\%$

Back to min-cut:

Problem: after many iterations, pr(success) gets smaller.

↳ if fail, repeat <u>entire</u> alg.

Idea: stop early!

Collapse(G, n, t):

$$\Pr(\text{success}) = \left(\frac{\cancel{n-2}}{n}\right)\left(\frac{\cancel{n-3}}{n-1}\right)\cdots\left(\frac{t}{\cancel{t+2}}\right)\left(\frac{t-1}{\cancel{t+1}}\right)$$

$$= \frac{t(t-1)}{n(n-1)}$$

Choose $t = \dfrac{n}{\sqrt{2}} + 1$

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

$\Pr(\text{Collapse}(G, n, \frac{n}{\sqrt{2}}+1) \text{ succeeds}) \geq 1/2$

FastCut(G, n) → if $n \le 8$, use brute force

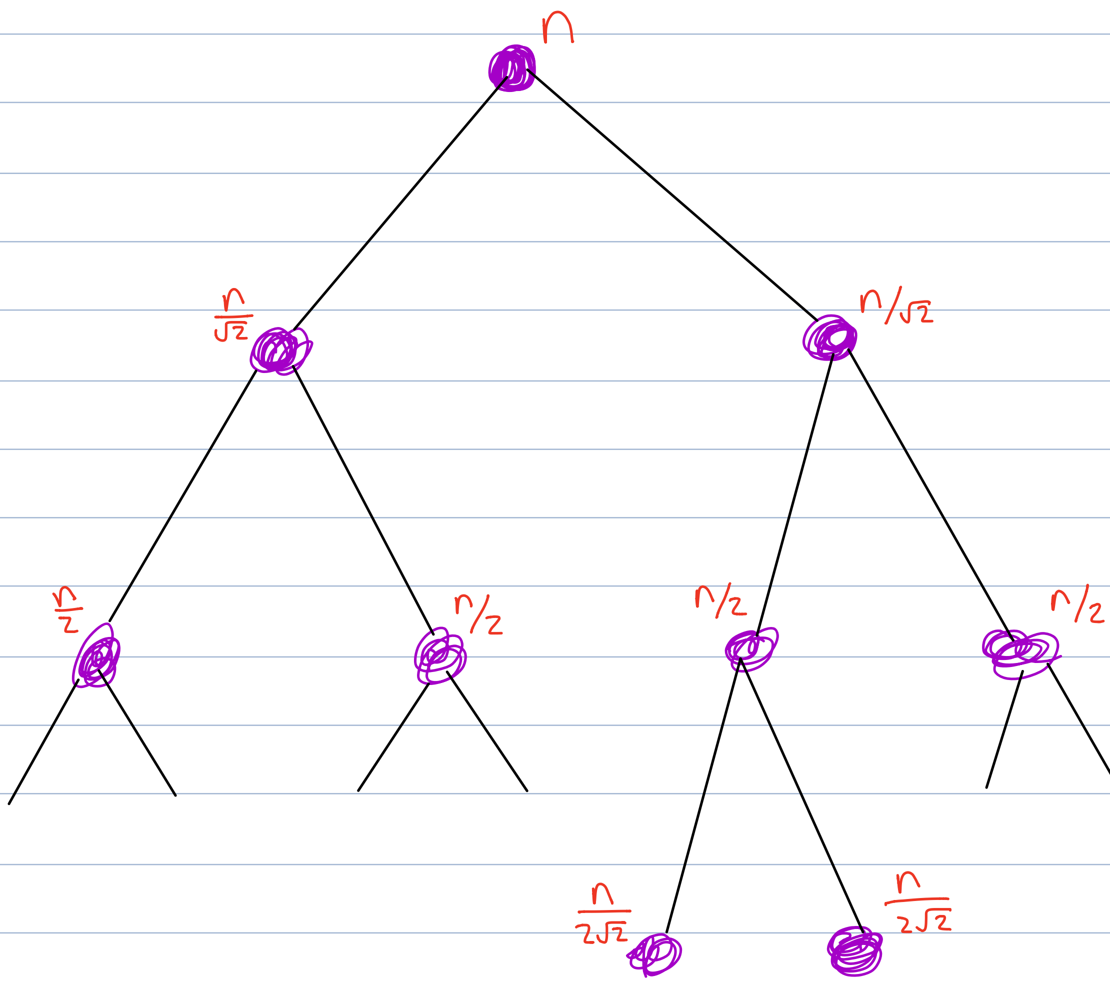
① $G_1 = \text{Collapse}(G, n, \frac{n}{\sqrt{2}} + 1)$

$C_1 = \text{FastCut}(G_1, \frac{n}{\sqrt{2}} + 1)$

② $G_2 = \text{Collapse}(G, n, \frac{n}{\sqrt{2}} + 1)$

$C_2 = \text{FastCut}(G, \frac{n}{\sqrt{2}} + 1)$

③ Return $\min(C_1, C_2)$



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

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

Prob correct?

→ Tree has height $\leq 2\log n$ ($\leq n^2$ leaves)

$$h(n) = 1 + h\left(\frac{n}{\sqrt{2}}\right)$$
$$= 1 + 1 + h\left(\frac{n}{2}\right)$$
$$= h\left(\frac{n}{2}\right) + 2$$
$$= 2\log n$$

→ Each edge represents a Contract$(G, \ell, \frac{\ell}{\sqrt{2}}+1)$

↳ succeeds w.p. $\geq 1/2$

→ Alg succeeds if root→leaf path succeeds at every step

→ Same as branching problem!

$\Pr[\text{succeeds } 2\log n \text{ deep}] \geq 1/2\log n$

Conclusion: FastCut succeeds w.p. $\geq 1/2\log n$

Final Alg: Repeat FastCut $2c\log^2 n$ times.
Return smallest cut.

$$\Pr(\text{fail}) = \left(1 - \frac{1}{2\log n}\right)^{2c\log^2 n} \leq e^{-c\log n} \leq 1/n^c$$

$\Rightarrow$ succeeds w.h.p.

Time: $O(n^2 \log^3 n)$

Idea 1

1. For each edge, choose a random weight.
   (Repeat if not unique.)
2. Find an MST
3. Cut heaviest edge in MST
   Return that cut.

Claim: Finds min-cut w.p. $\geq 2/n(n-1)$

Why?

Idea 2.

KargerCut implies: Every graph has at most $\frac{1}{2}n(n-1)$ min cuts.

Why?

Idea 3

If you run FastCut with 3 repeats (instead of 2) then Pr(success) is much higher!

This is not a good idea. Why?