COL 758: Advanced Algorithms

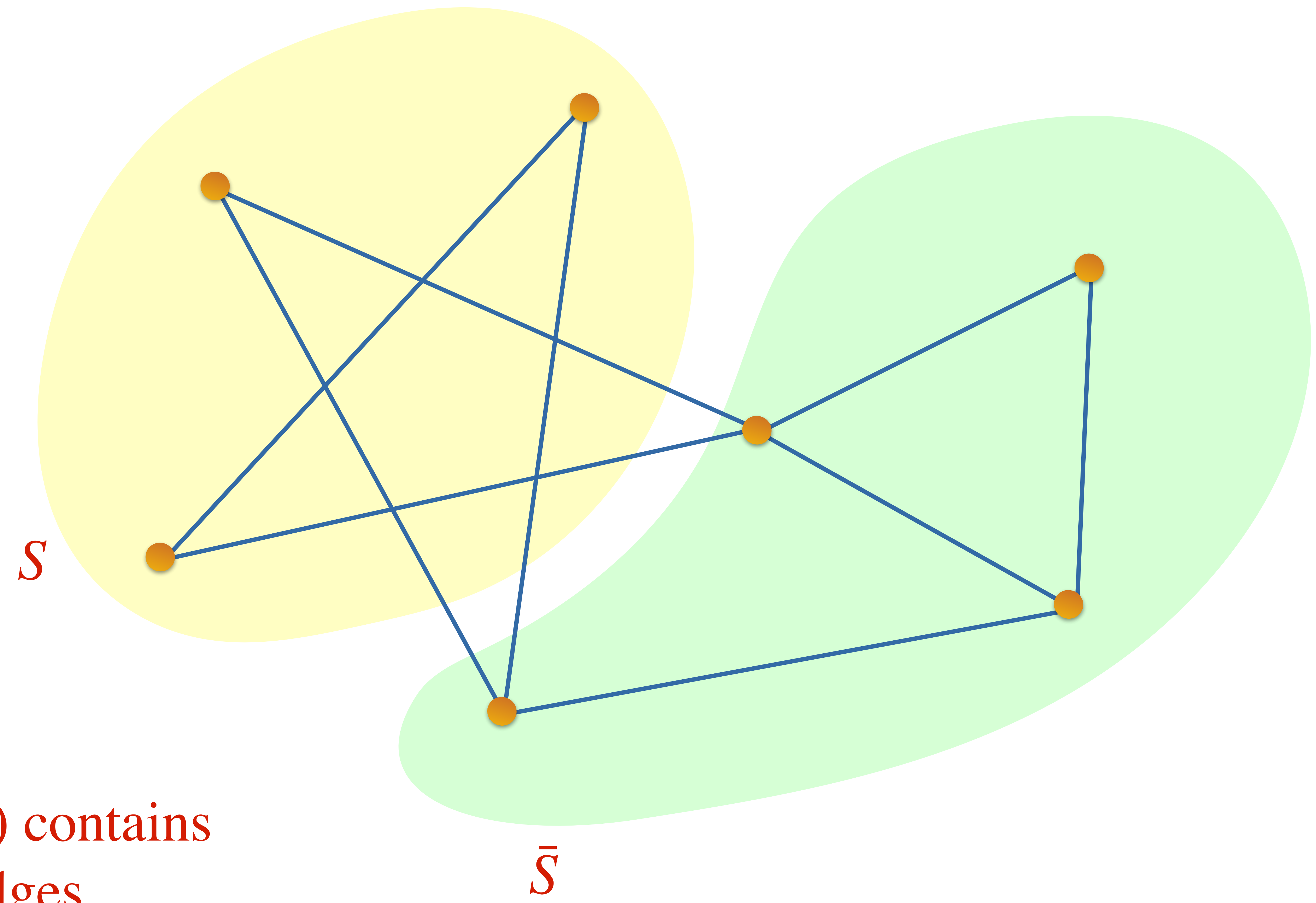
Lectures 15-17

Global Min-Cut in $O(n^4 \log n)$ time

What is a Cut ?

- Given a graph G, a **cut** is a partition $(S, \bar{S})$ of vertices of G.
- The **edges of cut** $(S, \bar{S})$, denoted by $E(S, \bar{S})$, is the set of all edges in G with exactly one endpoint in S.

Example:



$E(S, \bar{S})$ contains four edges

How many cuts are there in G ?

Ans: $\left(\dfrac{2^n - 2}{2}\right)$

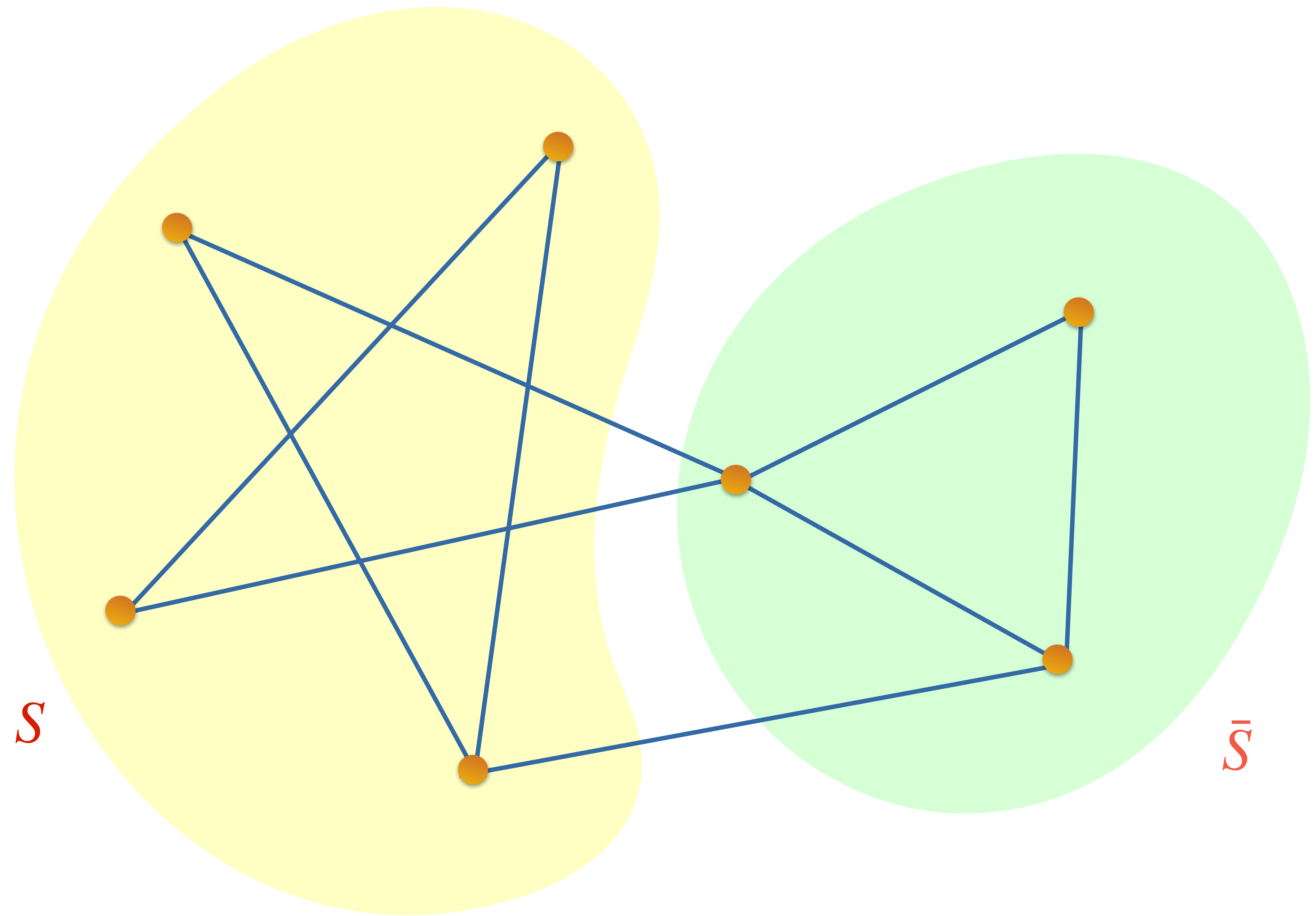
(A Cut is non-trivial partition of vertices in two sets).

Global Min-Cut

Given: A graph G with n vertices.

Output: A subset S of vertices in which the set of edges leaving S, denoted $E(S, \bar{S})$, is minimised.

Example:



Here, $(S, \bar{S})$ is a min-cut, and

$E(S, \bar{S})$ contains three edges.

Approach

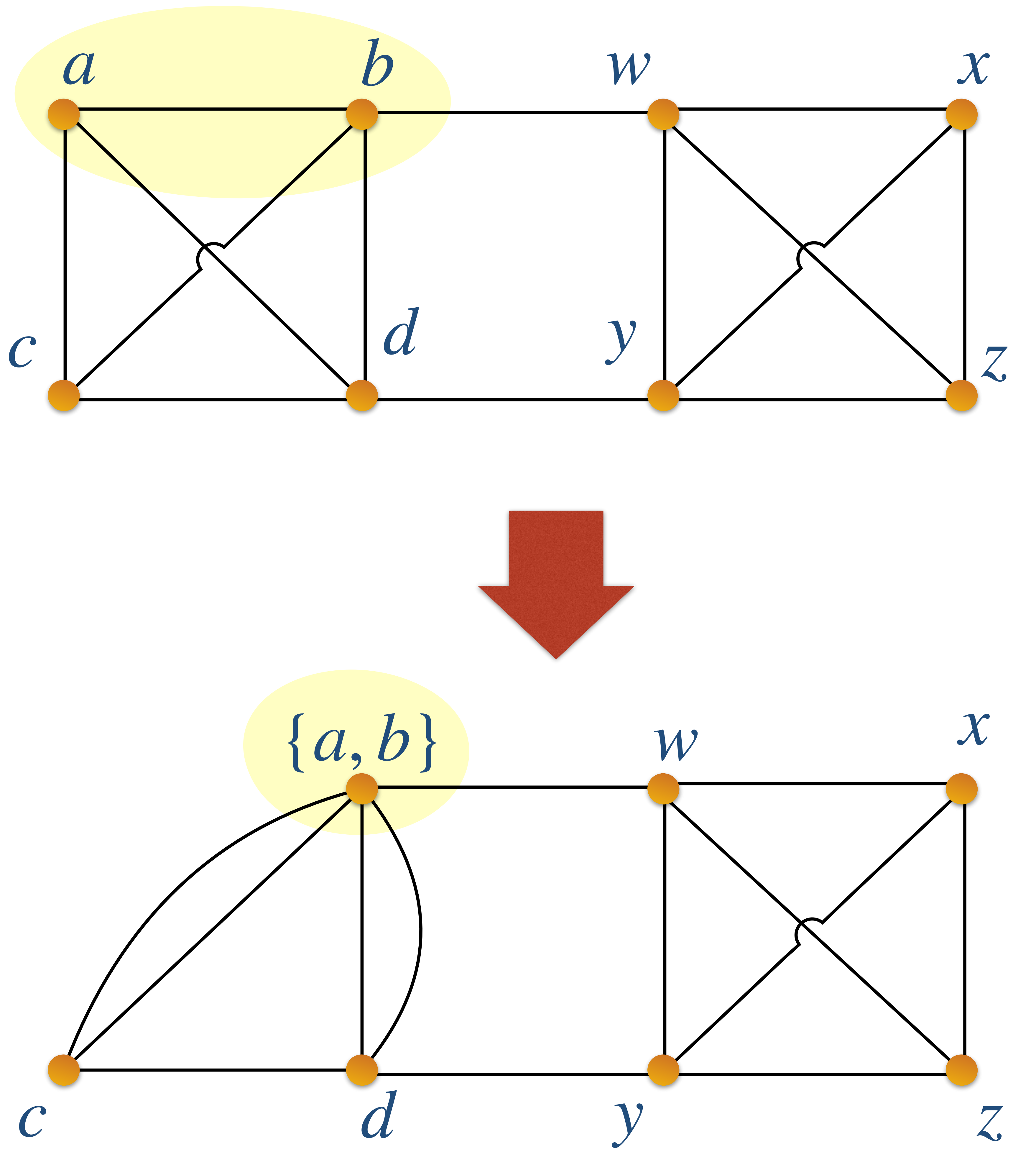
Try to merge endpoint of a <u>random-edge</u> into a super node.

Repeat this process till we are left with two supernodes.

Merge Operations:

Don't allow self-loops

Allow multiple edges between two vertices.



With good probability

<u>a given min-cut is preserved</u>

on performing

"a single contraction"

Let $(S, \bar{S})$ be some fixed min-cut of G.

Suppose $\delta = |E(S, \bar{S})|$ is the number of edges leaving set S.

Claim: $m \geq \frac{n\delta}{2}$, or $2|E(G)| \geq n \cdot$ size of global min-cut.

Proof: For each v_i, we have $deg(v_i) \geq \delta$.

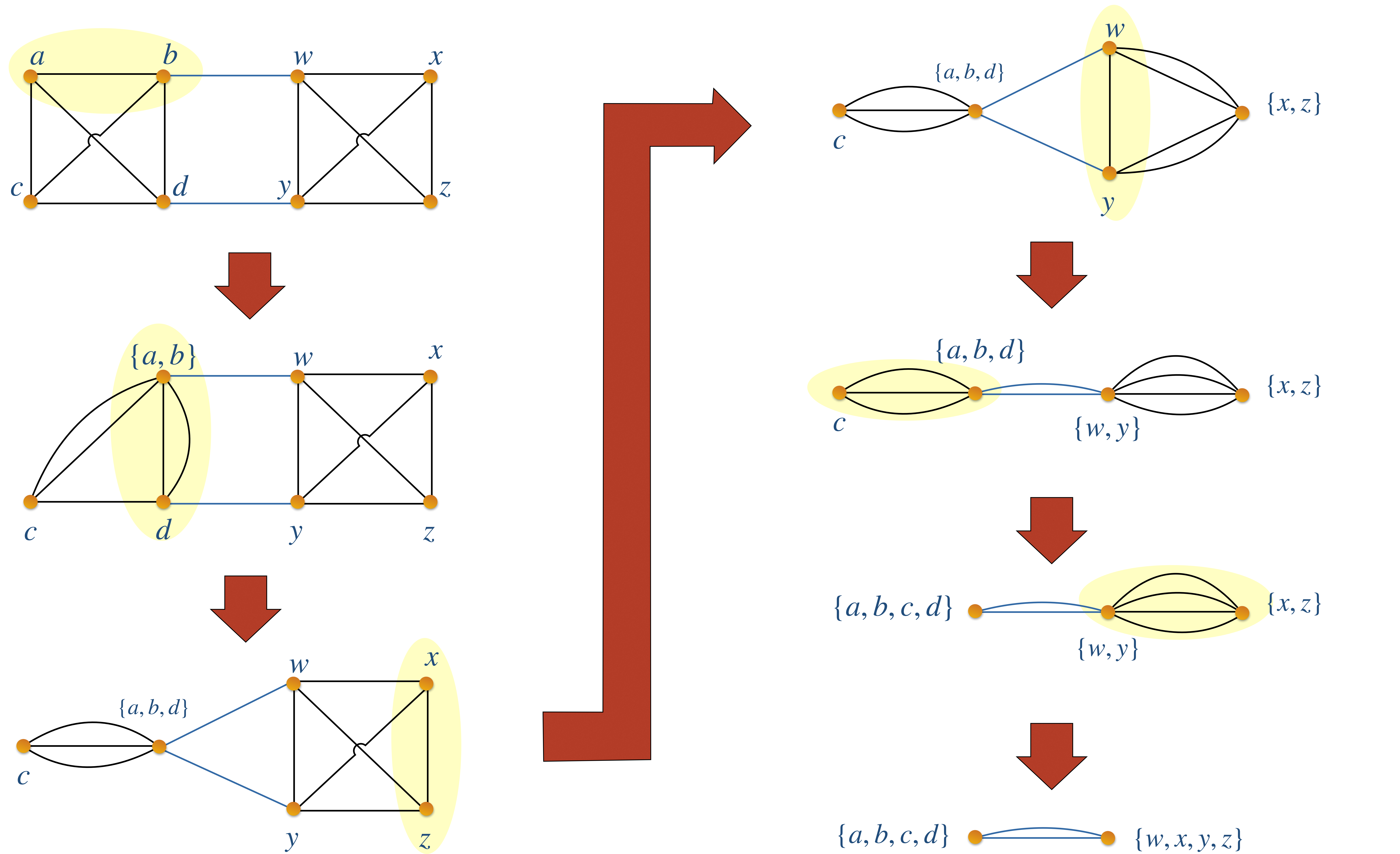
Bcoz if $deg(v_i) < \delta$ then $\left(v_i, V - \{v_i\}\right)$ will be a smaller min-cut.

So, $2m = \sum_{i=1}^{n} deg(v_i) \geq n\delta = n \cdot$ size of global min-cut

Therefore,

$$Prob\left(\text{random edge} \in E(S, \bar{S})\right) = \frac{\delta}{m} \leq \frac{\delta}{n\delta/2} = \frac{2}{n}$$

Example: Merge Operations that preserves a Min-Cut



Let $(S, \bar{S})$ be some fixed min-cut of G.

Then,

$$Prob\Big(\text{Min-cut } (S, \bar{S}) \text{ is preserved after first merge }\Big) \;\geq\; 1 - \frac{2}{n}$$

Similarly,

$$Prob\Big(\text{Min-cut } (S, \bar{S}) \text{ is preserved after ALL merges }\Big)$$

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

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

$$= \frac{(n-2)!}{(n!)/2} \;\geq\; \frac{1}{n^2}$$

Randomized Algorithm

Min-Cut(G)

For $i = 1$ **to** $cn^2 \log n$:

Initialise H to G;

Repeat $(n-2)$ **times:**

Pick a random edge e in H;

$H \leftarrow$ Contract(e, H);

$(S_i, \bar{S}_i) \leftarrow$ Two super-nodes in H;

$k = \arg\min \left\{ |E(S_i, \bar{S}_i)| \;\middle|\; 1 \le i \le cn^2 \log n \right\}$

Return Cut $(S_k, \bar{S}_k)$;

Success probability $\geq \dfrac{1}{n^2}$

Time complexity = $O(n^4 \log n)$

(Bcoz time of one contraction is $O(n)$)

Success Probability is at least:

$$1 - \left(1 - \frac{1}{n^2}\right)^{cn^2 \log n} \approx 1 - \frac{1}{n^c}$$

Question: How can we "improve run-time"

of Global-Min-Cut algorithm

Let $(S, \bar{S})$ be some fixed min-cut of G.

Then, $$Prob\Big(\text{Min-cut } (S, \bar{S}) \text{ is preserved after first merge}\Big) \geq 1 - \frac{2}{n}$$

Question: What is success probability if we stop at a graph with k supernodes?

$$n \longrightarrow (n-1) \longrightarrow (n-2) \longrightarrow \cdots \longrightarrow (k+1) \longrightarrow k$$

Solution:

$$\begin{aligned}
&\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\left(1 - \frac{2}{n-2}\right)\cdots\left(1 - \frac{2}{k+1}\right) \\
&= \frac{(n-2)!/(k-2)!}{(n!)/(k!)} \\
&= \frac{k(k-1)}{n(n-1)} \approx \frac{k^2}{n^2}
\end{aligned}$$

Remark:

If $k = n/2$, then

success prob $\approx 1/4$

Observation

$$n \longrightarrow (n-1) \longrightarrow \cdots \longrightarrow (n/2) \longrightarrow (n/2-1) \longrightarrow \cdots \longrightarrow 3 \longrightarrow 2$$

After first $n/2$ merges, min-cut is preserved with probability $\geq 1/4$

Probability of preserving min-cut decreases to $\approx 1/n^2$

Here, success probability is high, so, we do not need large number of repetitions

When n decreases, more repetitions are needed to boost success probability

Cost of single merge is $\Omega(n)$

When n decreases, merge can be done more efficiently.

Observation:

When graph is smaller, then failure probability is higher, and at this stage more repetitions are needed to boost up probability, however, these can be done efficiently because merge will take lesser time.

Modified Algorithm

Fast-Min-Cut(G)

Initialise H_1 and H_2 to G;

Repeat $(n - n/\sqrt{2})$ **times:**

Pick a random edge e in H_1 and Contract(e, H_1);

Pick a random edge e in H_2 and Contract(e, H_2);

$(C_1, \bar{C}_1) \leftarrow$ Fast-Min-Cut(H_1);

$(C_2, \bar{C}_2) \leftarrow$ Fast-Min-Cut(H_2);

Return Smaller of the two cuts $(C_1, \bar{C}_1)$ and $(C_2, \bar{C}_2)$;

<u>Observation:</u>

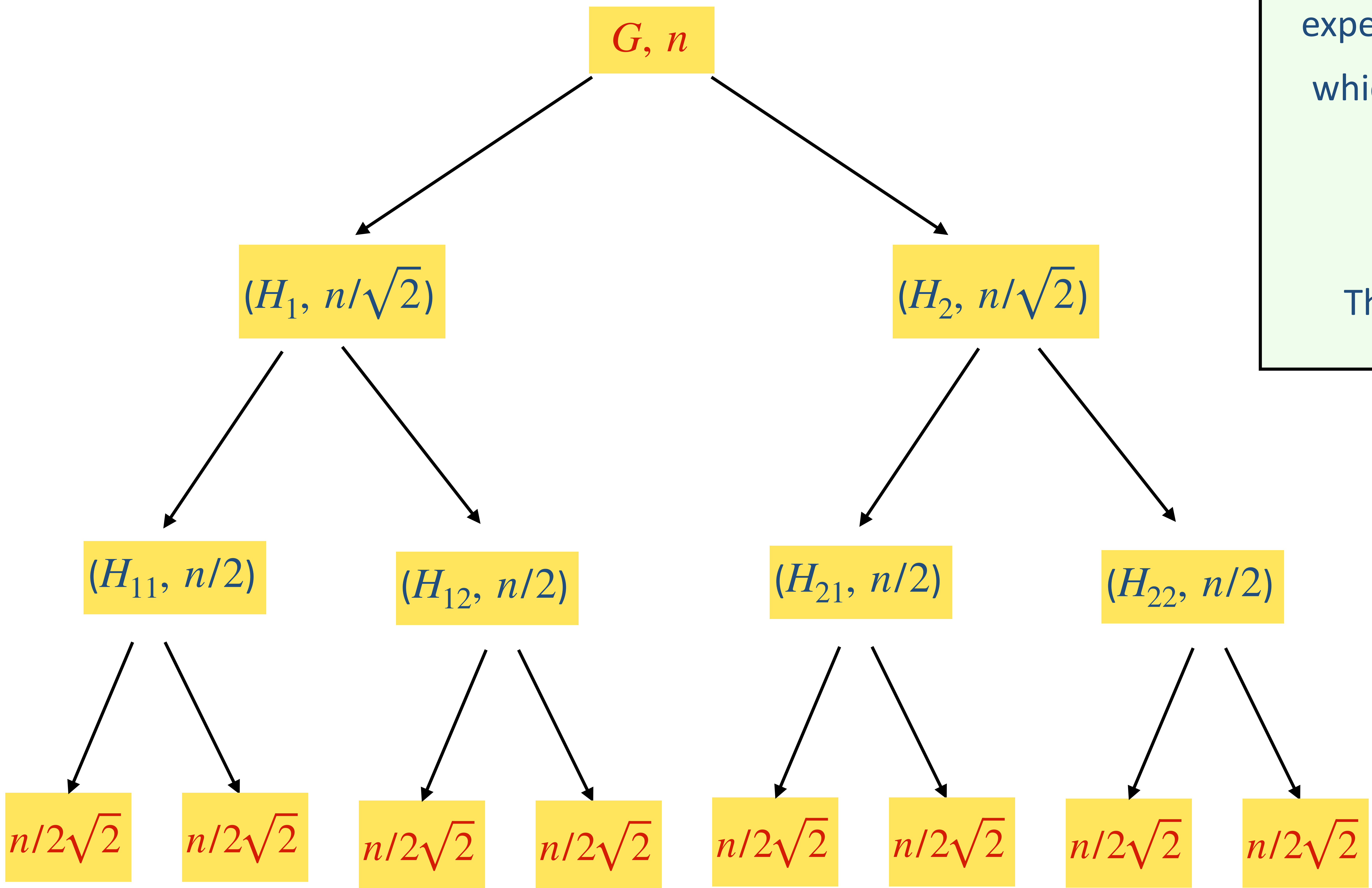
The size of graph H_1, H_2

is $n/\sqrt{2}$

<u>Question:</u>

Why do we split and parallelise at $n/\sqrt{2}$, that is, when the number of vertices remaining is $1/\sqrt{2}$ times that of the size of input graph.

Recursion Tree



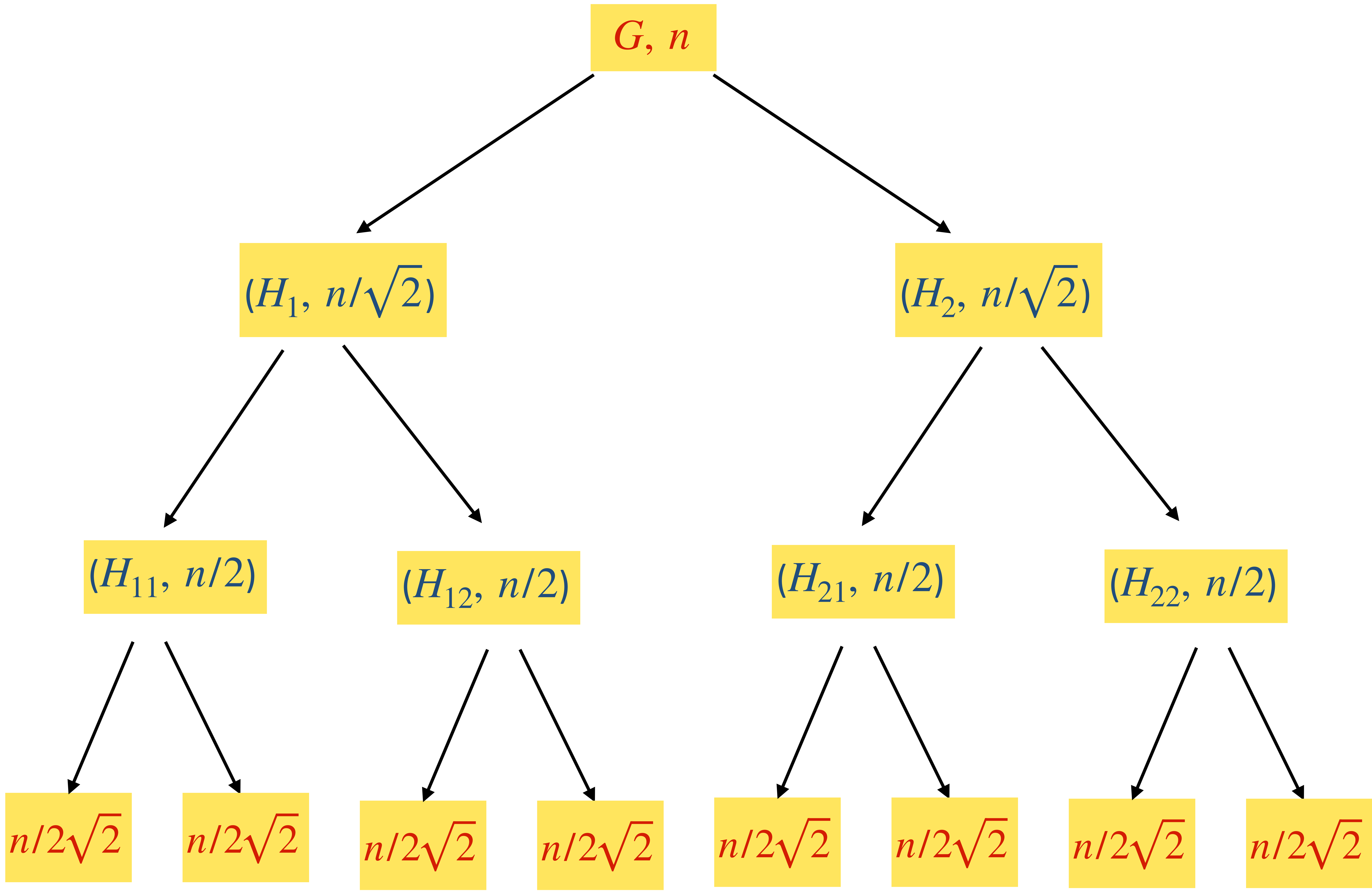
<u>Remark:</u>

If size of H_1 and H_2 was (n/a), then the expected number of graphs in level 1 which preserves the min-cut of G is $2/a^2$.

This expression is 1 for $a = \sqrt{2}$

Recursion Tree

Expected number of graphs in a level preserving min-cut of G



$$2 \cdot \left(\frac{1}{\sqrt{2}}\right)^2 = 1$$

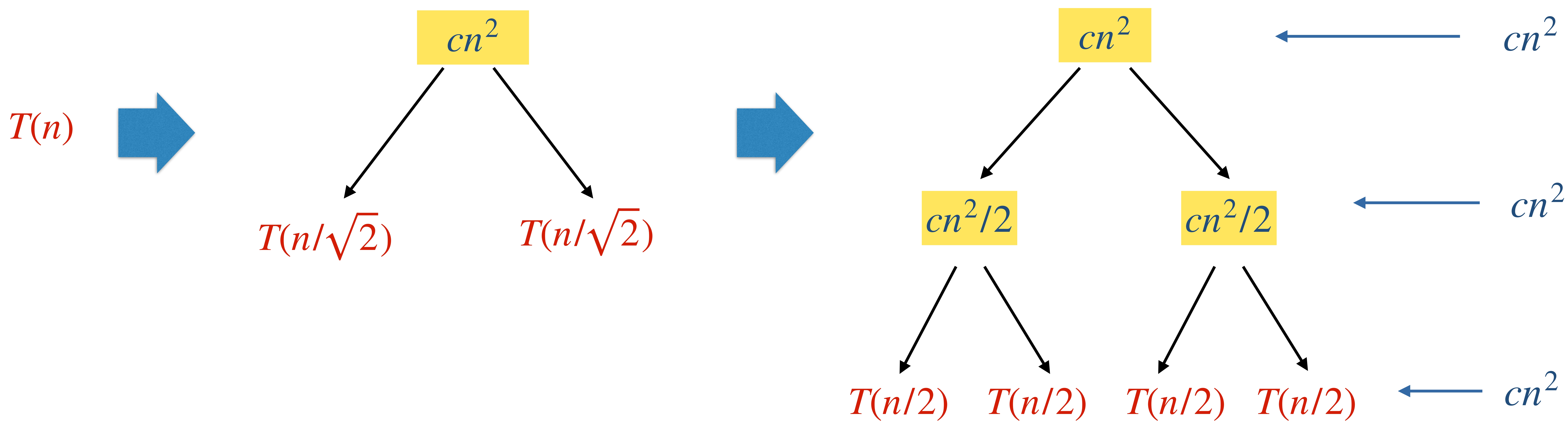
$$4 \cdot \left(\frac{1}{2}\right)^2 = 1$$

$$8 \cdot \left(\frac{1}{2\sqrt{2}}\right)^2 = 1$$

Time Complexity Analysis

Time complexity satisfies the relation:

$$T(n) = cn^2 + 2T(n/\sqrt{2})$$



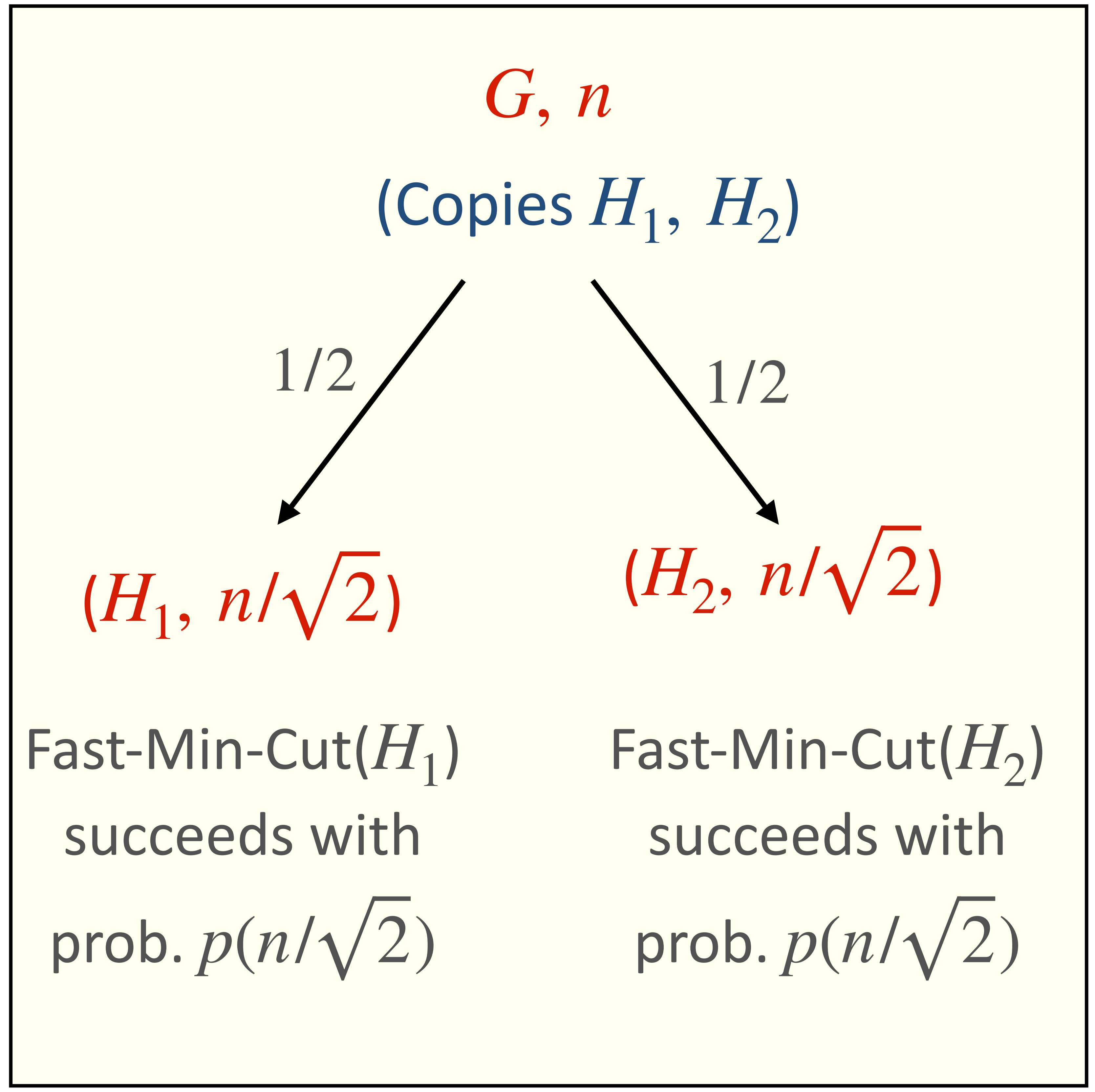
Number of levels = $\log_{\sqrt{2}} n$. So, time complexity is $O(n^2 \log n)$

Success Probability Analysis

Suppose $p(n)$ is success probability of Fast-Min-Cut on an input graph with n vertices.

Then,

$$1 - p(n) \;=\; \Bigg(1 - \underbrace{p\big(n/\sqrt{2}\big)\cdot\frac{1}{2}}_{\text{success prob of first chain}}\Bigg)\Bigg(1 - \underbrace{p\big(n/\sqrt{2}\big)\cdot\frac{1}{2}}_{\text{success prob of second chain}}\Bigg)$$



Thus,

$$p(n) \;=\; 1 - \left(1 - \frac{p(n/\sqrt{2})}{2}\right)^2$$

i.e., $$\boxed{p(n) \;=\; p(n/\sqrt{2}) \;-\; \frac{p\big(n/\sqrt{2}\big)^2}{4}}$$

To solve this, we define $q(k) := p(2^k)$. So, the recursion is $$\boxed{q(k) \;=\; q(k-1) \;-\; \frac{q(k-1)^2}{4}}$$

We need to solve the recurrence $q(k) = q(k-1) - \dfrac{q(k-1)^2}{4}$.

Claim: $\dfrac{-q(k-1)}{4} \geq \dfrac{-q(k)}{3}$

Proof: $q(k) = q(k-1) - \dfrac{q(k-1)^2}{4} \geq q(k-1) - \dfrac{q(k-1)}{4} = \dfrac{3q(k-1)}{4}$

(Here, the last inequality follows from the fact that $q()$ lies in range $[0,1]$).

Now,

$$q(k) = q(k-1)\left[1 - \frac{q(k-1)}{4}\right]$$

$$\Rightarrow q(k) \geq q(k-1)\left[1 - \frac{q(k)}{3}\right] \quad \text{(by above claim)}$$

$$\Rightarrow \frac{1}{q(k-1)} \geq \frac{1}{q(k)} - \frac{1}{3}$$

$$\Rightarrow \frac{1}{q(k)} \leq \frac{1}{q(k-1)} + \frac{1}{3}$$

On solving this, we get $q(k) \geq \dfrac{1}{\Theta(k)}$ or $p(n) \geq \dfrac{1}{\Theta(\log n)}$

Faster Min-Cut Algorithm

Fast-Min-Cut(G)

Initialise H_1 and H_2 to G;

Repeat $(n - n/\sqrt{2})$ **times:**

Pick a random edge e in H_1 and Contract(e, H_1);

Pick a random edge e in H_2 and Contract(e, H_2);

$(C_1, \bar{C}_1) \leftarrow$ Fast-Min-Cut(H_1);

$(C_2, \bar{C}_2) \leftarrow$ Fast-Min-Cut(H_2);

Return Smaller of the two cuts $(C_1, \bar{C}_1)$ and $(C_2, \bar{C}_2)$;

To get success probability of 1 - inverse polynomial, we need to repeat the algorithm $O(\log^2 n)$ times.

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

Success Probability : $\geq \dfrac{1}{\Theta(\log n)}$

Probabilistic Method:
Bound on
the number of distinct Global Min-Cuts

To Show: In any undirected graph G, the number of distinct global min-cuts is at most $\dfrac{n(n-1)}{2}$

Proof: Suppose $(A_1, \bar{A}_1),\ (A_2, \bar{A}_2),\ \ldots, (A_k, \bar{A}_k)$ are k different global min-cuts in G.

For $i \in [1,\ k]$, define random variable

$$X_i = \begin{cases} 1 & \text{If after } (n-2) \text{ edge contractions in } G, \text{ min-cut } (A_i, \bar{A}_i) \text{ is preserved .} \\ 0 & \text{otherwise .} \end{cases}$$

Then, $E(X_i) = Prob\Big(\text{Min-cut } (A_i, \bar{A}_i) \text{ is preserved after } (n-2) \text{ random edge contractions}\Big) \geq \dfrac{2}{n(n-1)}$

Now, define $X = X_1 + \cdots + X_k$.

Remark: $X = 1$ *iff* after $(n-2)$ edge contractions in G some global min-cut is preserved, and 0 otherwise.

Thus, $E(X) = \displaystyle\sum_{i=1}^{k} E(X_i) \geq k\frac{2}{n(n-1)}$, and also $E[X] \leq 1$. These together implies that $k \leq \dfrac{n(n-1)}{2}$.

Question:

Can we use Fast-Min-Cut algorithm to claim that total number of distinct min-cuts is $O(\log n)$??

No!!

Following is Incorrect claim:

If $(A_1, \bar{A}_1),\ (A_2, \bar{A}_2),\ \ldots, (A_k, \bar{A}_k)$ are k different global min-cuts in G, then for each $i \in [1,\ k]$

$$Prob\Big((A_i, \bar{A}_i) \text{ is preserved by Fast-Min-Cut algorithm}\Big) \geq \frac{1}{\Theta(\log n)}$$

Correct claim is:

$$Prob\Big(\text{Some global min-cut is preserved by Fast-Min-Cut algorithm}\Big) \geq \frac{1}{\Theta(\log n)}$$