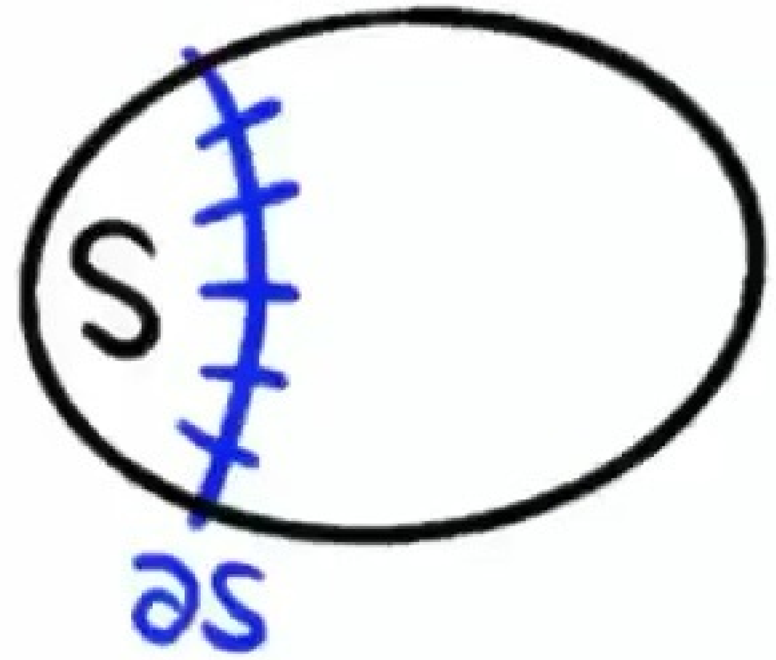
Expanders: Cut Definition

G is a ϕ-expander if for all $S \subseteq V$,

$|\partial S| >= \phi \min\{vol(S), vol(\bar{S})\}$

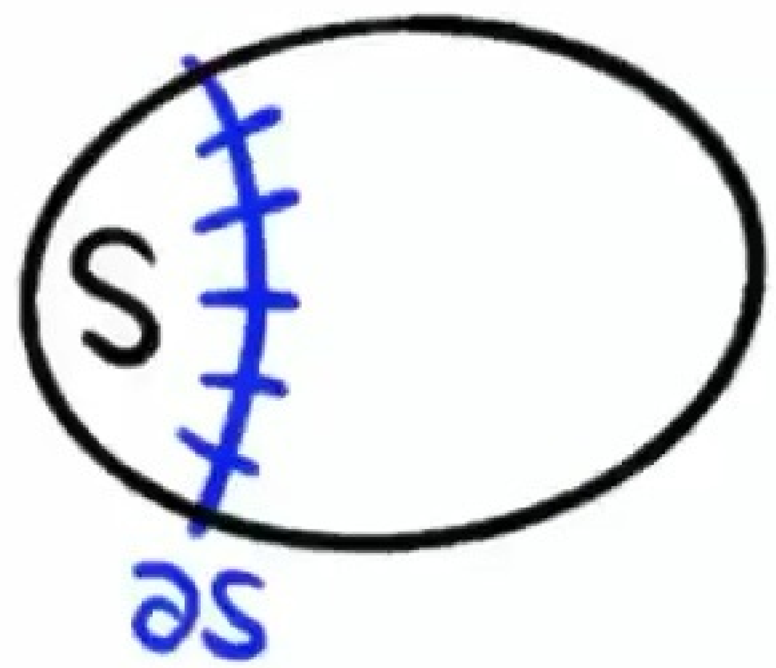


$vol(S) = \sum_{v \in S} deg(v) \approx$ # edges incident to S

Expanders: Cut Definition

G is a ϕ-expander if for all $S \subseteq V$,

$|\partial S| >= \phi \min\{vol(S), vol(\bar{S})\}$



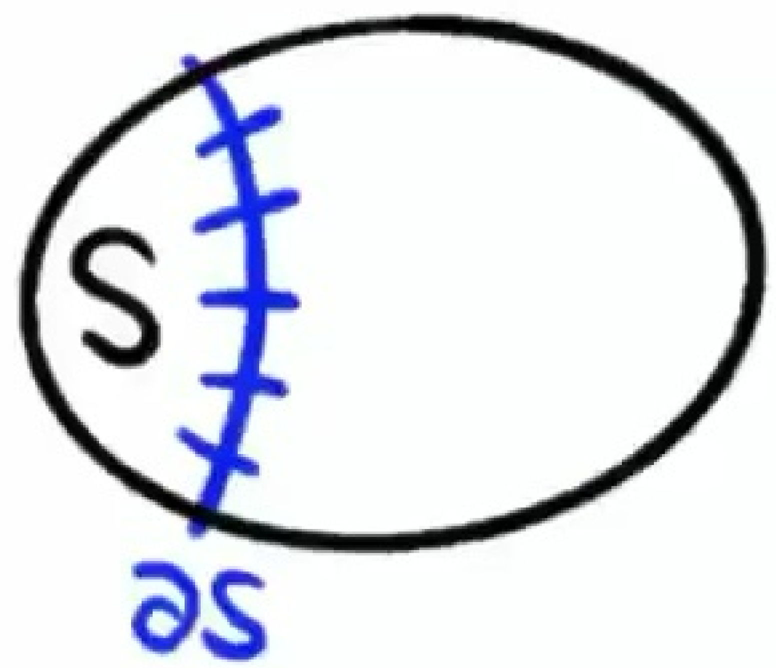
$vol(S) = \sum_{v \in S} deg(v) \approx$ # edges incident to S

$|\partial S| <= vol(S)$, so "all cuts are within $1/\phi$ of largest possible"

Expanders: Cut Definition

G is a ϕ-expander if for all $S \subseteq V$,

$|\partial S| >= \phi \min\{\mathrm{vol}(S), \mathrm{vol}(\bar{S})\}$



$\mathrm{vol}(S) = \sum_{v \in S} \deg(v) \approx$ # edges incident to S

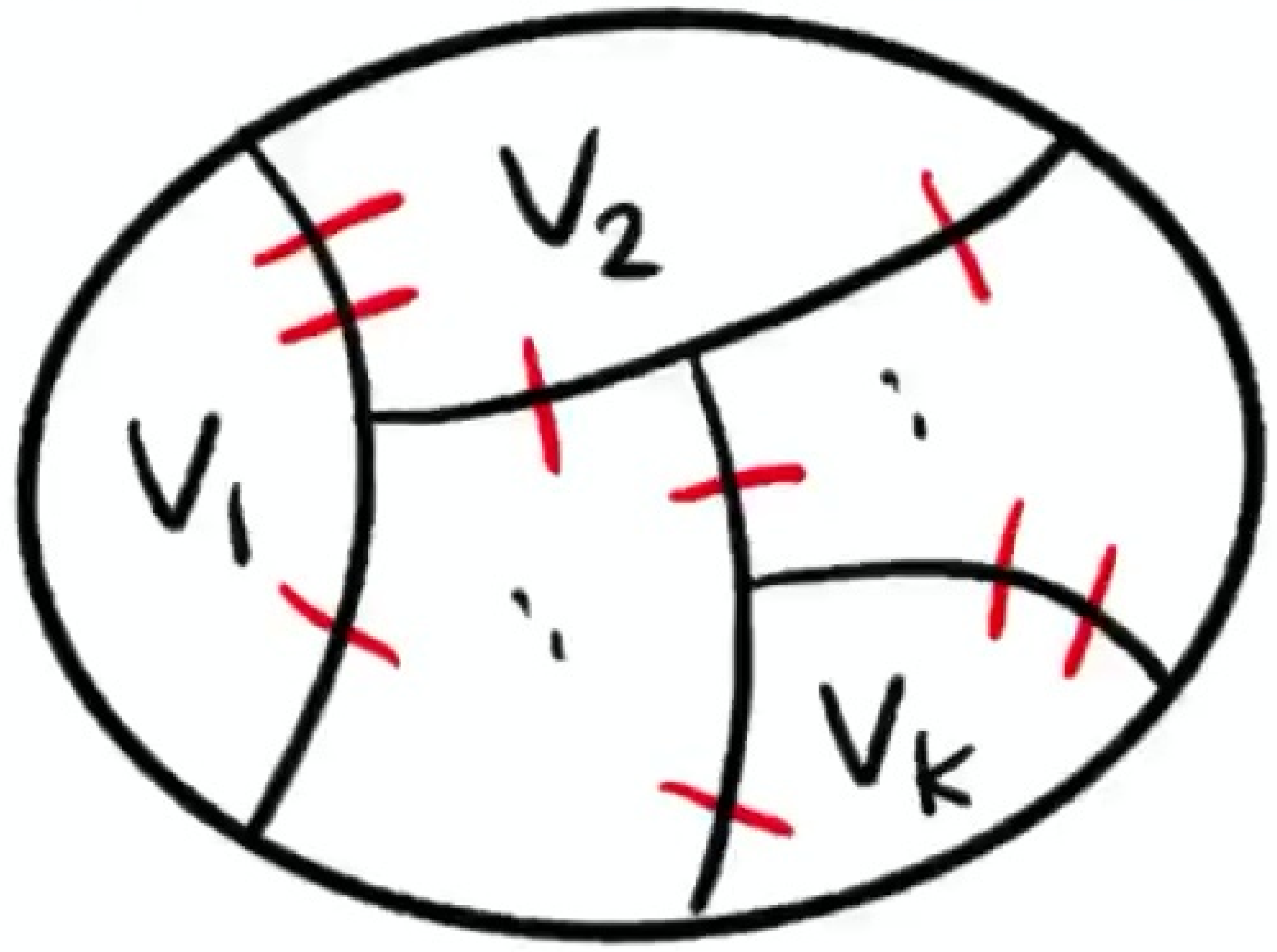
$|\partial S| <= \mathrm{vol}(S)$, so "all cuts are within $1/\phi$ of largest possible"

"At least ϕ fraction of edges escape the set"

Expander Decomposition

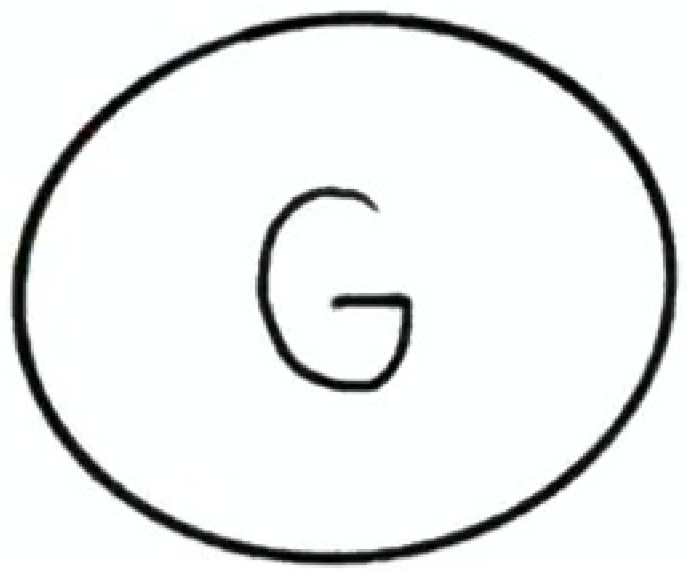
For any undirected graph $G=(V,E)$, can decompose V into $V_1, \ldots, V_k$ s.t.

- each induced graph $G[V_i]$ is a ϕ-expander
- at most ϕ fraction of edges are inter-cluster



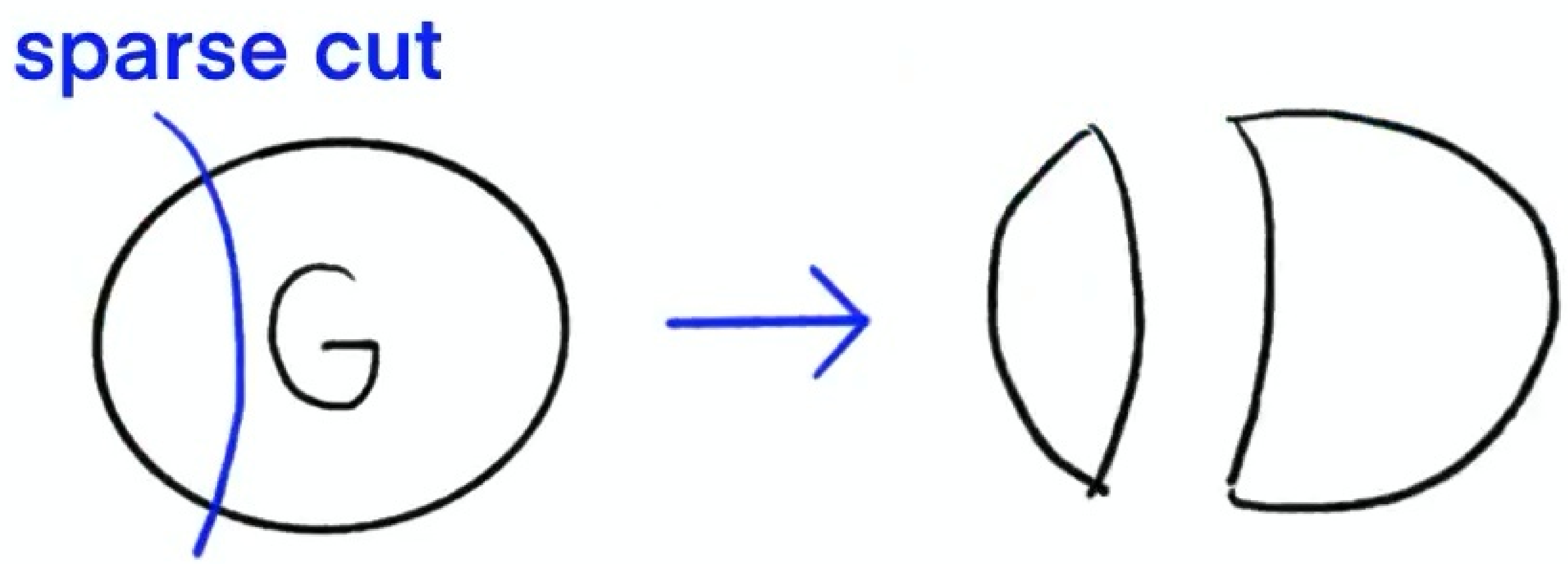
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \, \mathrm{vol}(S)$



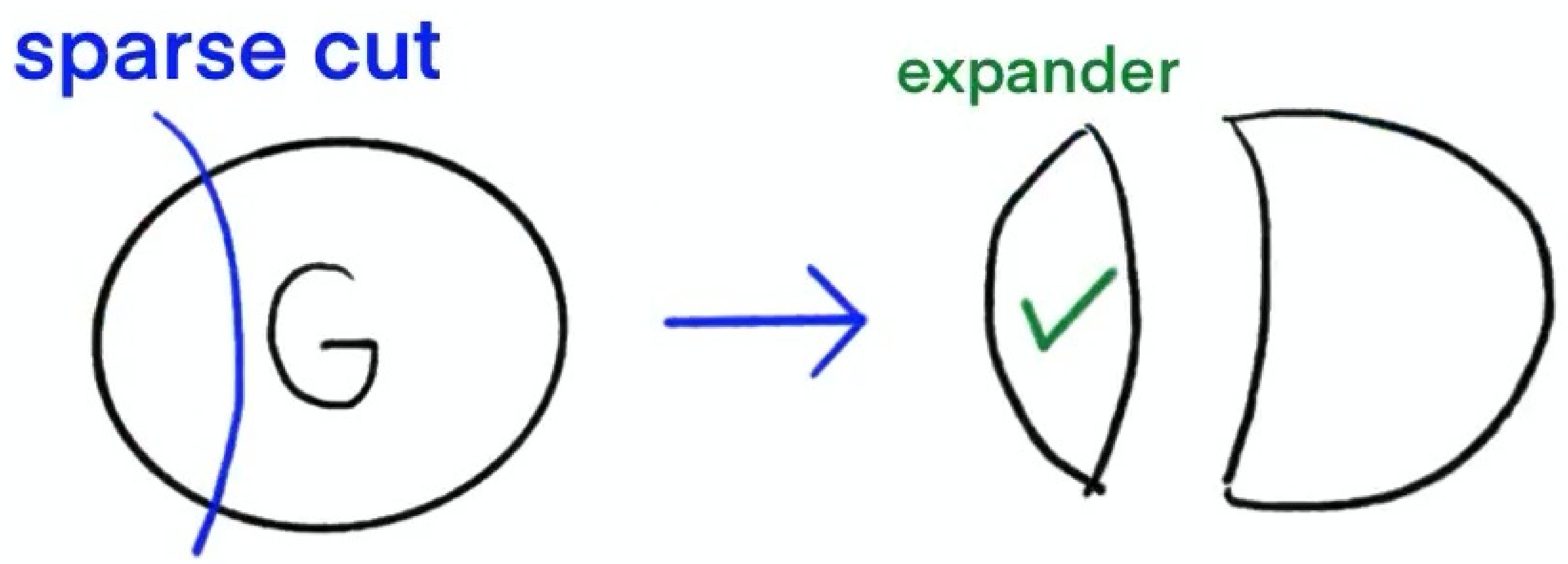
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi\, \mathrm{vol}(S)$



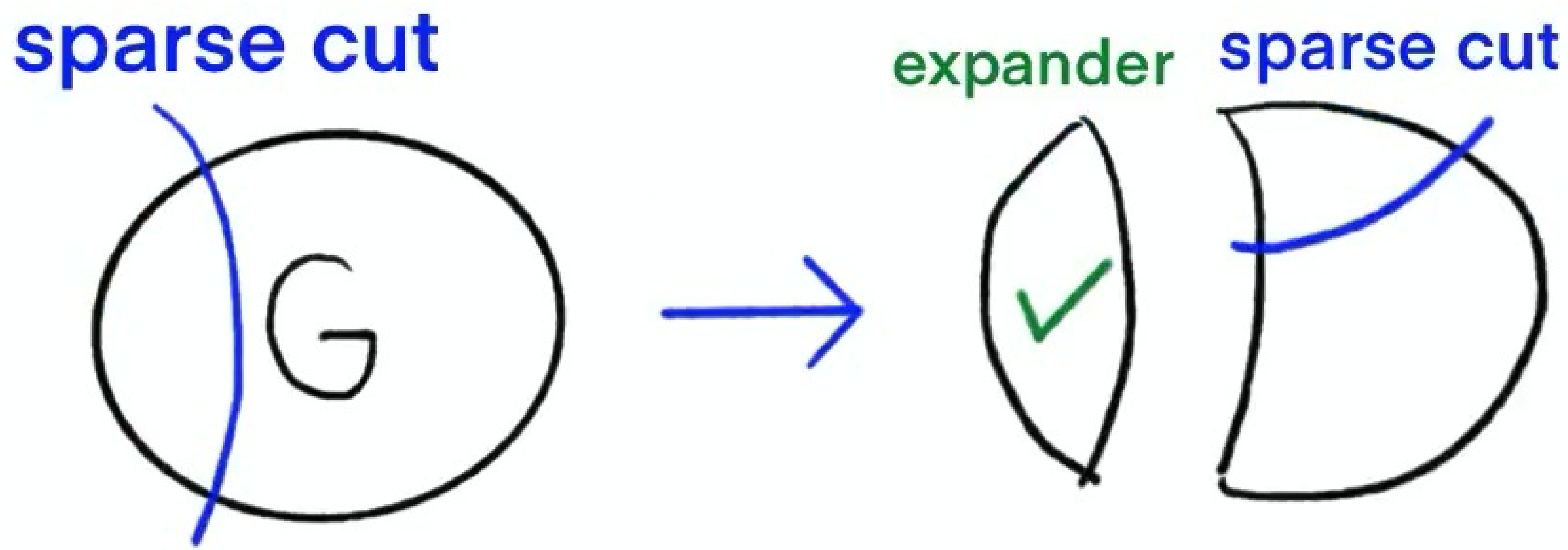
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \, \mathrm{vol}(S)$



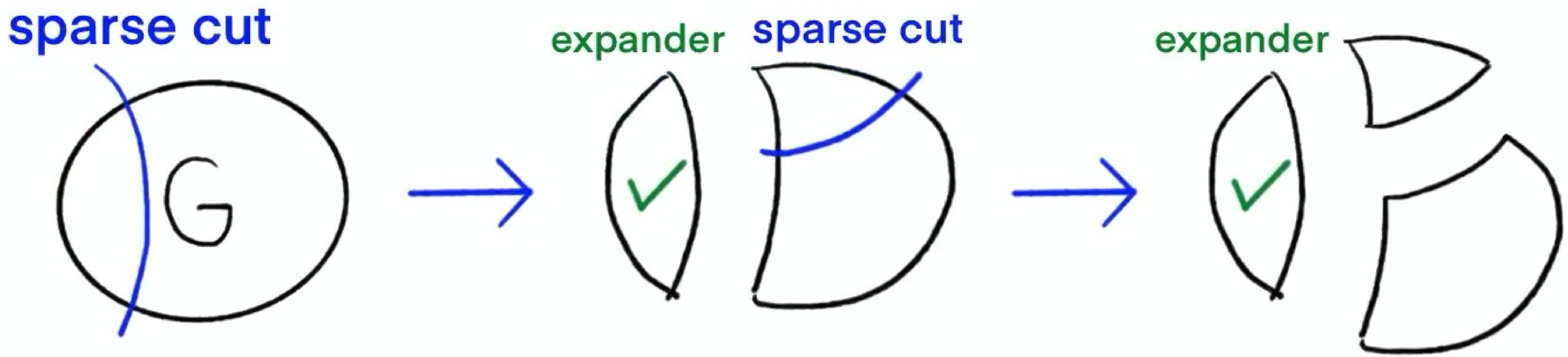
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \operatorname{vol}(S)$



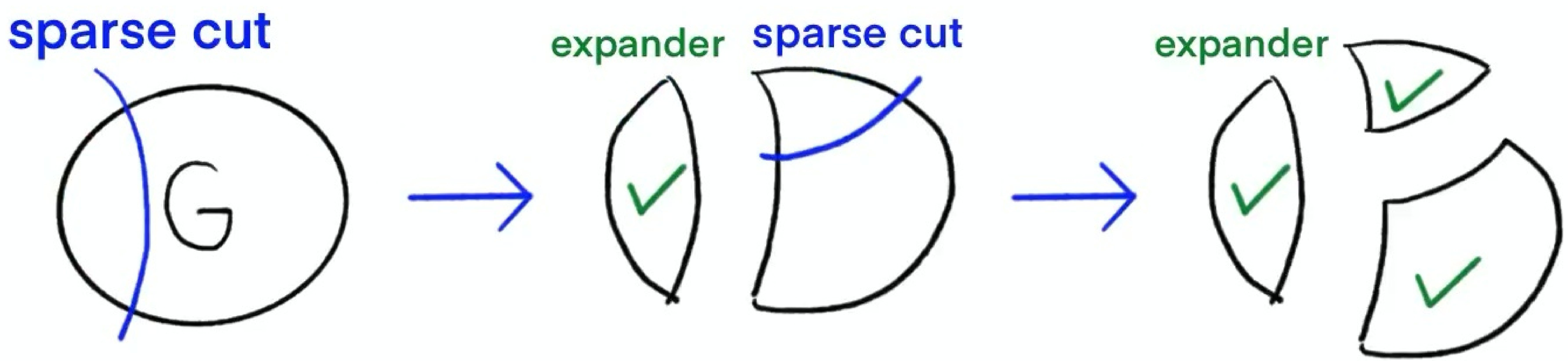
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \operatorname{vol}(S)$



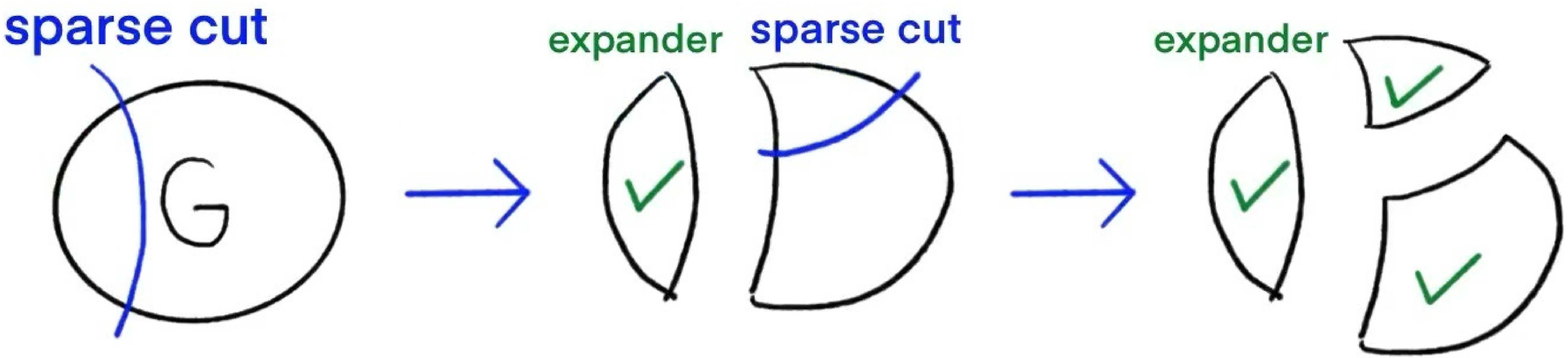
Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \operatorname{vol}(S)$



Expander Decomposition: Existence

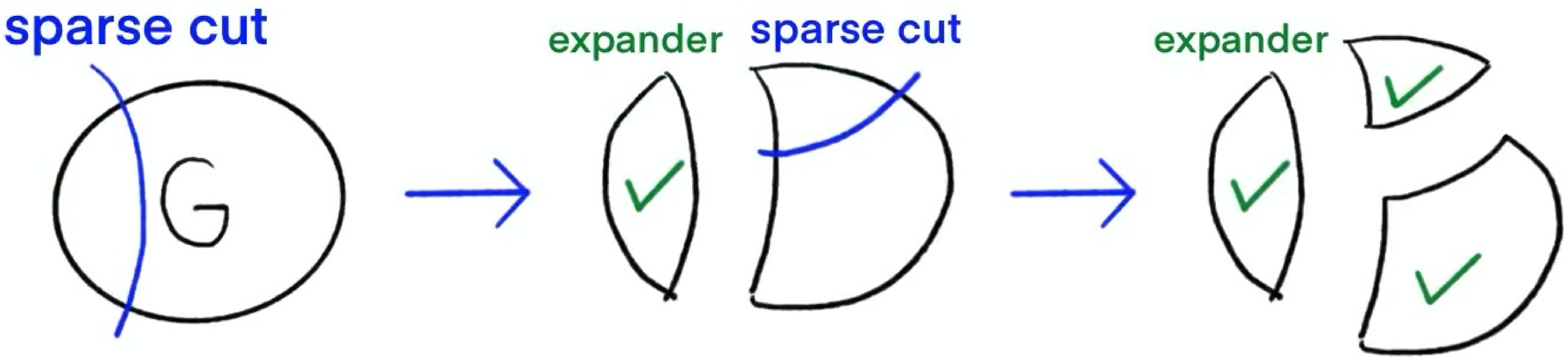
Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \operatorname{vol}(S)$



Claim: # edges cut $<= O(\phi\, m \log m)$

Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \, \mathrm{vol}(S)$

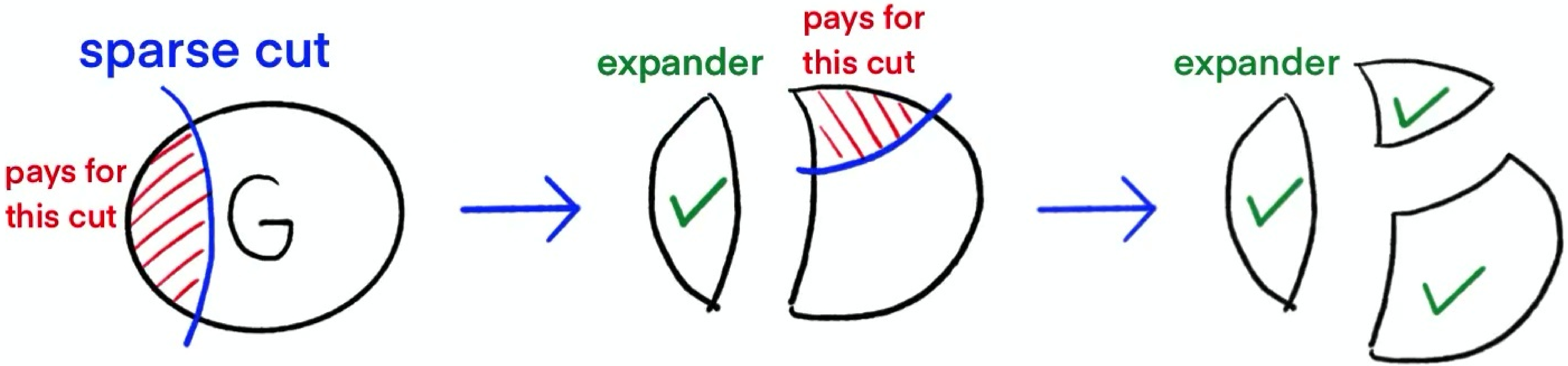


Claim: # edges cut <= $O(\phi \, m \log m)$

Proof: "charge each cut to vertices on the smaller side"

Expander Decomposition: Existence

Greedy algorithm: while a cluster is not an expander, take a ϕ-sparse cut: $|\partial S| < \phi \, \mathrm{vol}(S)$



Claim: # edges cut <= $O(\phi \, m \log m)$

Proof: "charge each cut to vertices on the smaller side"

Expander Decomposition: Existence

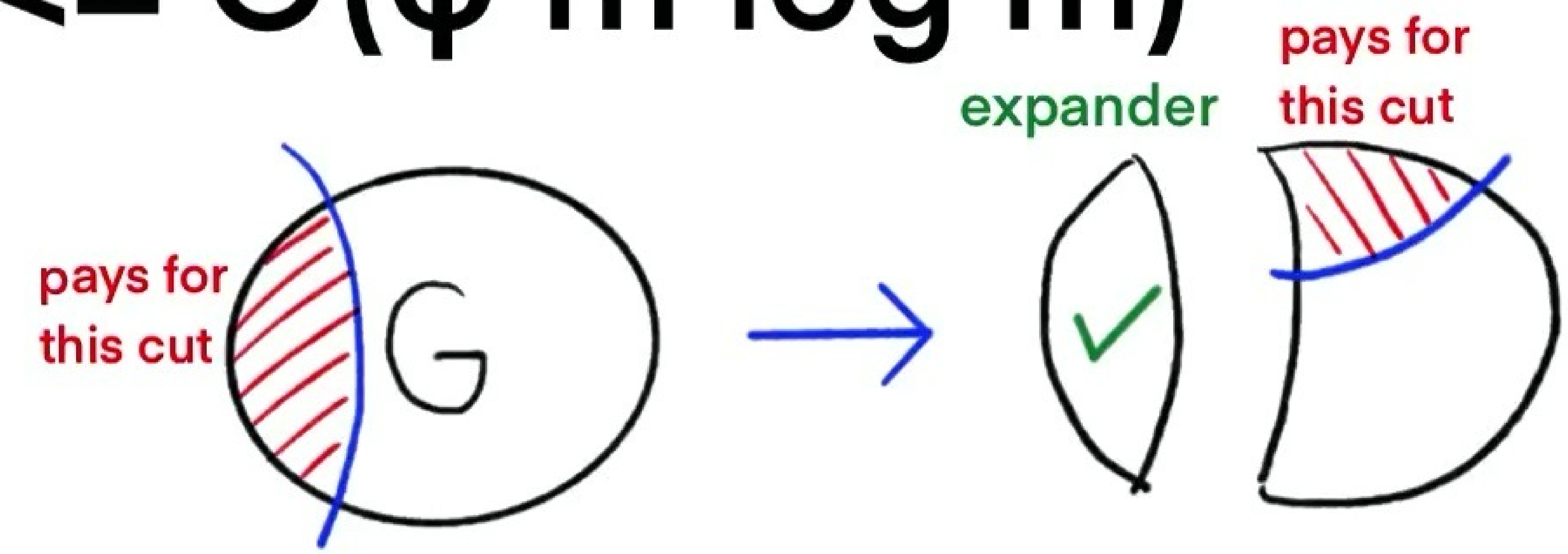
Claim: # edges cut <= $O(\phi\ m \log m)$

Expander Decomposition: Existence

Claim: # edges cut <= O(ϕ m log m)

Proof:

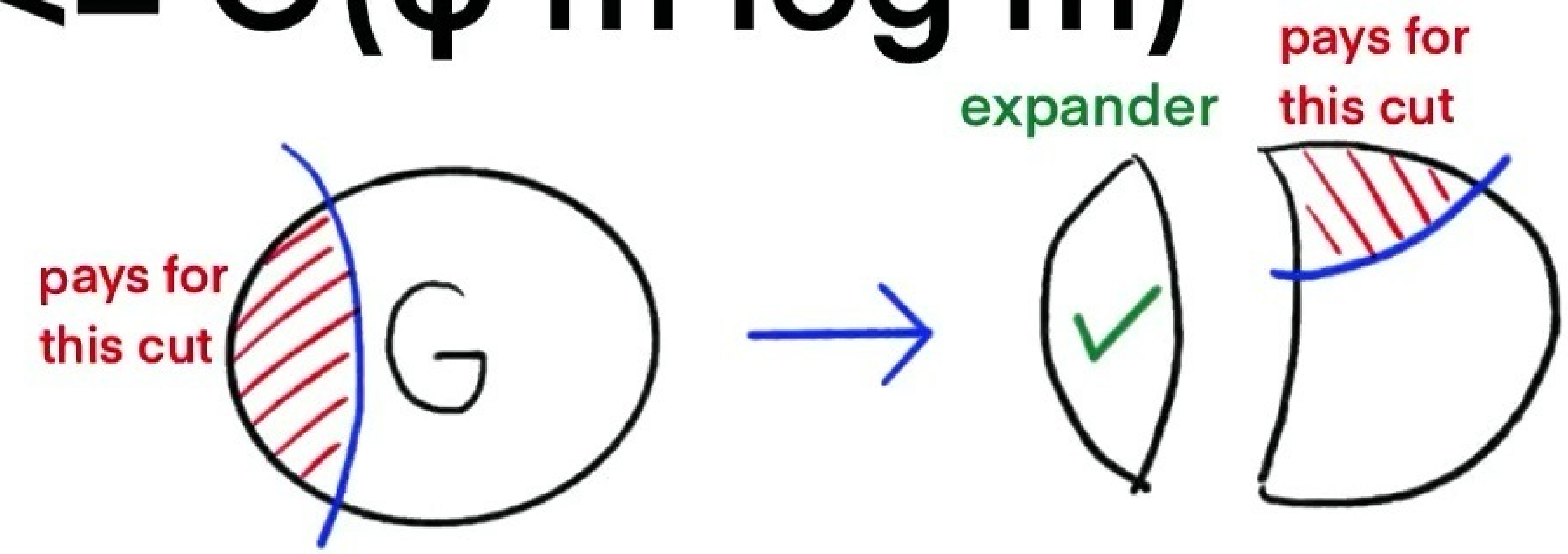
On each cut, each vertex on smaller side pays ϕ*deg(v).



Expander Decomposition: Existence

Claim: # edges cut <= O(ϕ m log m)

Proof:



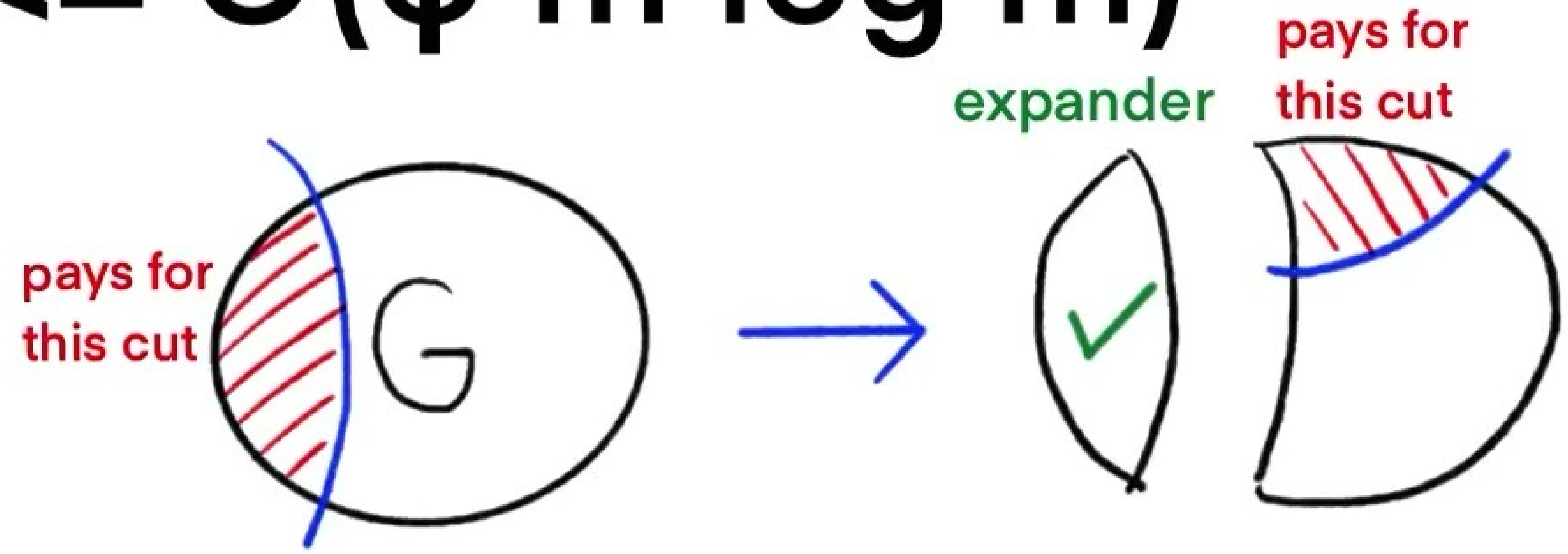
On each cut, each vertex on smaller side pays ϕ*deg(v).

Total of ϕ*vol(S) >= l∂Sl paid, covers the cut

Expander Decomposition: Existence

Claim: # edges cut <= O(ϕ m log m)

Proof:



On each cut, each vertex on smaller side pays ϕ*deg(v).

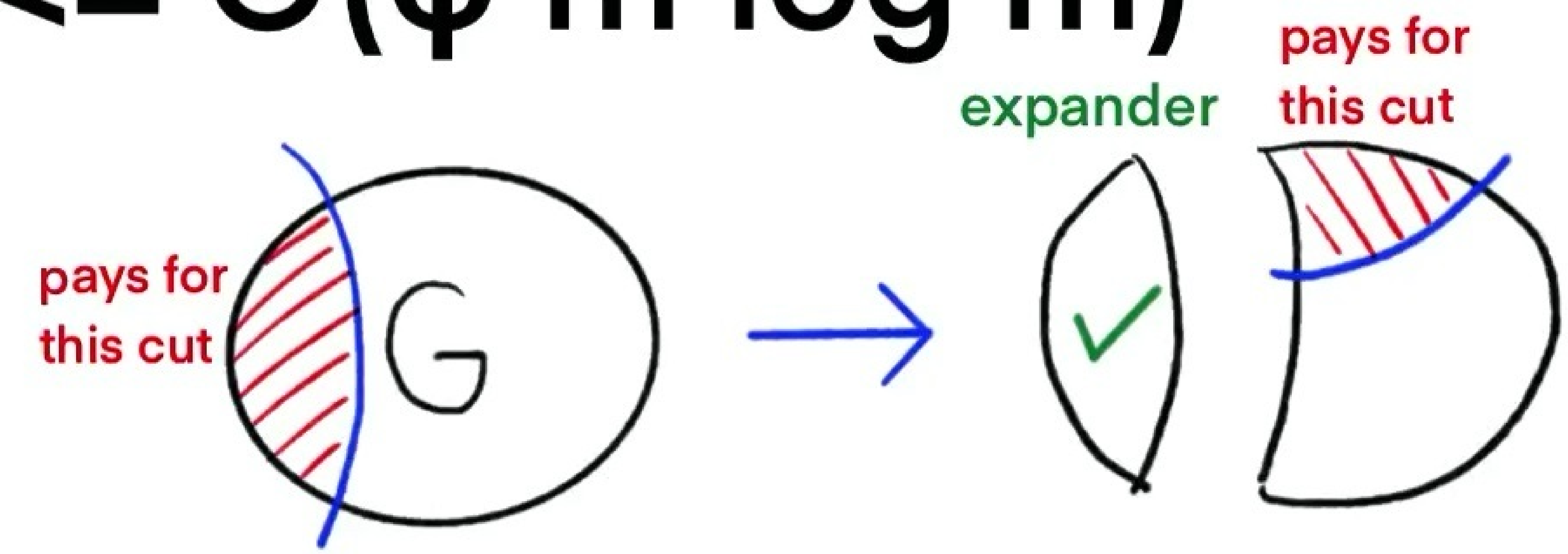
Total of ϕ*vol(S) >= |∂S| paid, covers the cut

Each time a vertex pays, its cluster halves in size (smaller side), so pays $\log_2 m$ times

Expander Decomposition: Existence

Claim: # edges cut <= $O(\phi\ m \log m)$

Proof:



On each cut, each vertex on smaller side pays ϕ*deg(v).

Total of ϕ*vol(S) >= $|\partial S|$ paid, covers the cut

Each time a vertex pays, its cluster halves in size (smaller side), so pays $\log_2 m$ times

Total <= $\Sigma_v\ \phi*\deg(v)*\log_2 m = \phi*2m*\log_2 m$

Expander Decomposition: Algorithm

Sparsest cut is NP-hard!

Expander Decomposition: Algorithm

Sparsest cut is NP-hard!

α-approx -> $O(\alpha\phi m \log n)$ inter-cluster edges

Expander Decomposition: Algorithm

Sparsest cut is NP-hard!

α-approx -> $O(\alpha\phi m \log n)$ inter-cluster edges

Theme: approx is OK for exact algorithms!

Pay approx factor in the running time.

Expander Decomposition: Algorithm

Sparsest cut is NP-hard!

α-approx -> $O(\alpha\phi m \log n)$ inter-cluster edges

Theme: approx is OK for exact algorithms!

Pay approx factor in the running time.

Cut-matching game [KRV'07]: $\alpha=O(\log^2 n)$ approx in near-linear time

Expander Decomposition: Algorithm

Sparsest cut is NP-hard!

α-approx -> $O(\alpha\phi m \log n)$ inter-cluster edges

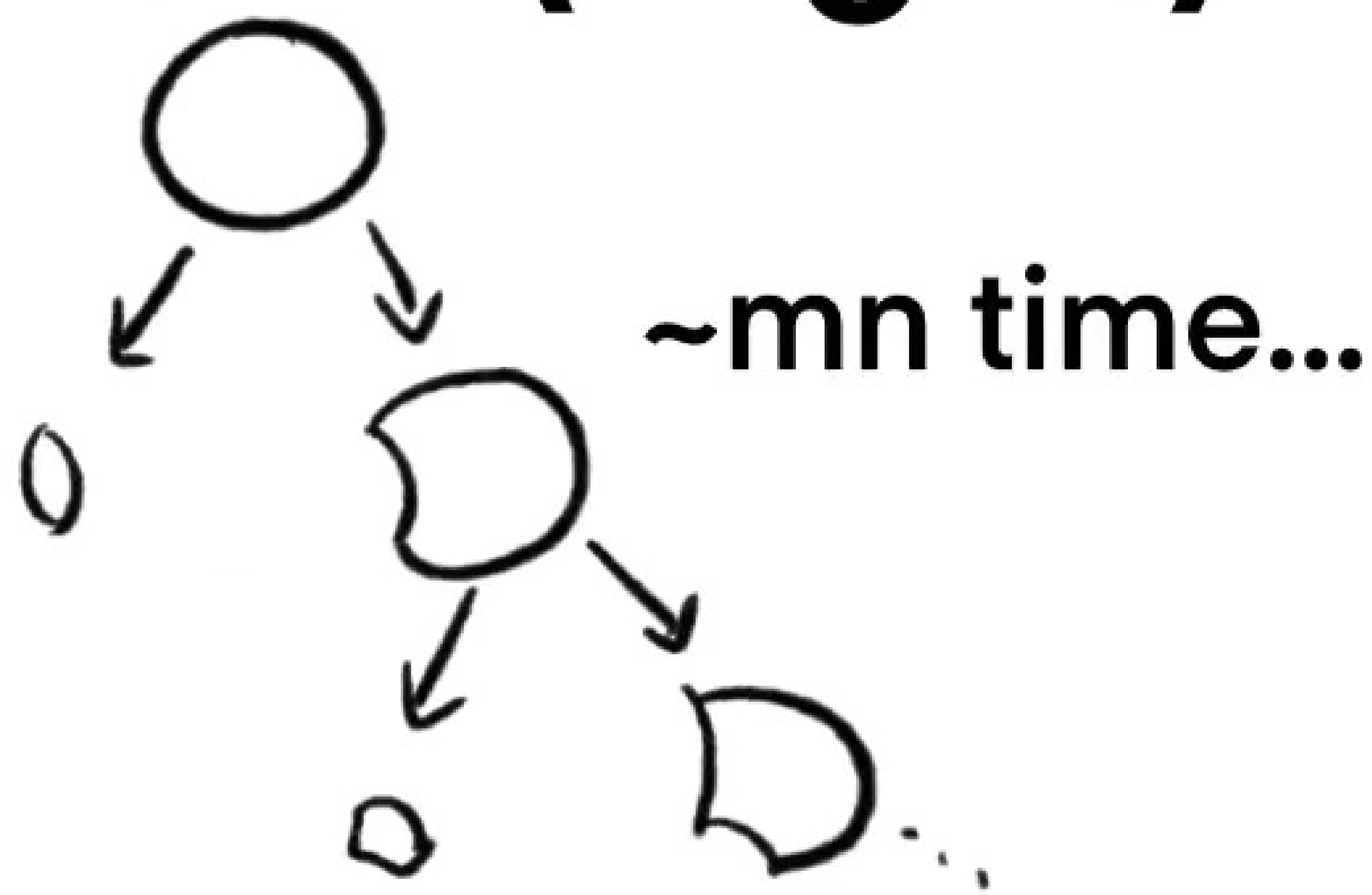
Theme: approx is OK for exact algorithms!

Pay approx factor in the running time.

Cut-matching game [KRV'07]: $\alpha = O(\log^2 n)$

approx in near-linear time

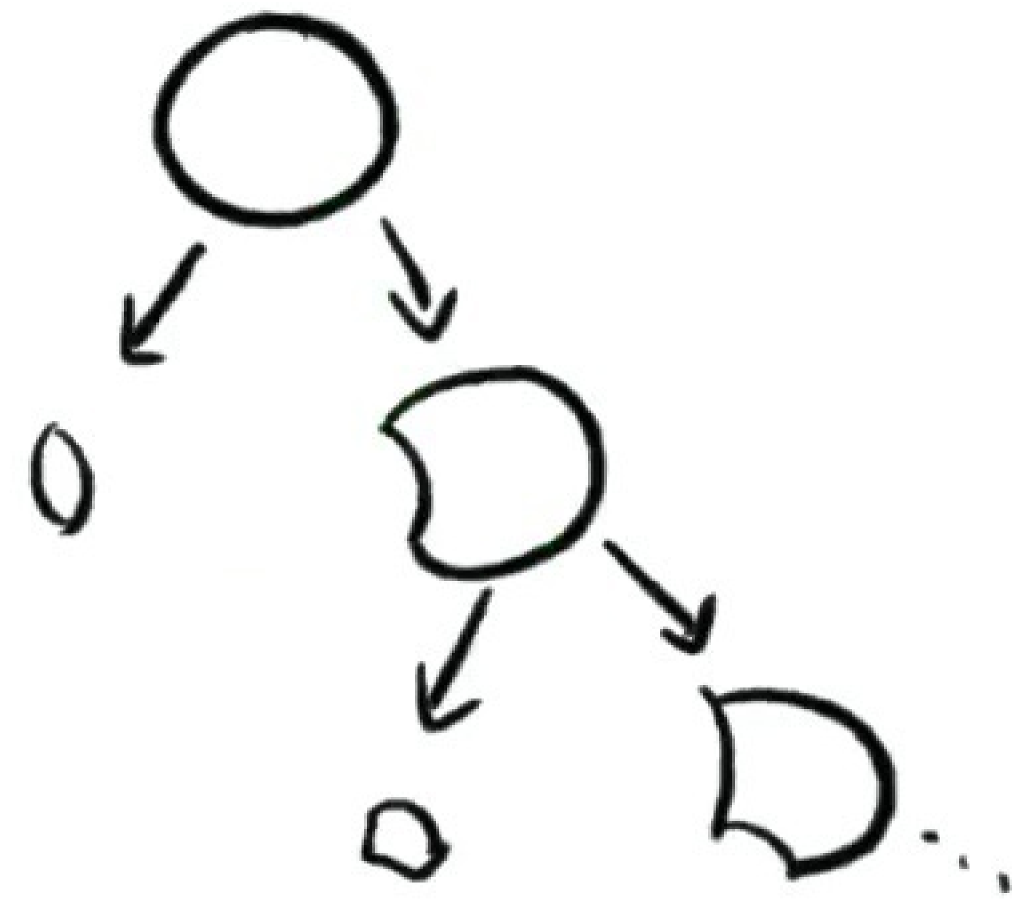
Recursion may be lopsided...



Expander Decomposition: Algorithm

Near-linear time (Saranurak-Wang 2019)

If lopsided recursion, certify large side is an expander

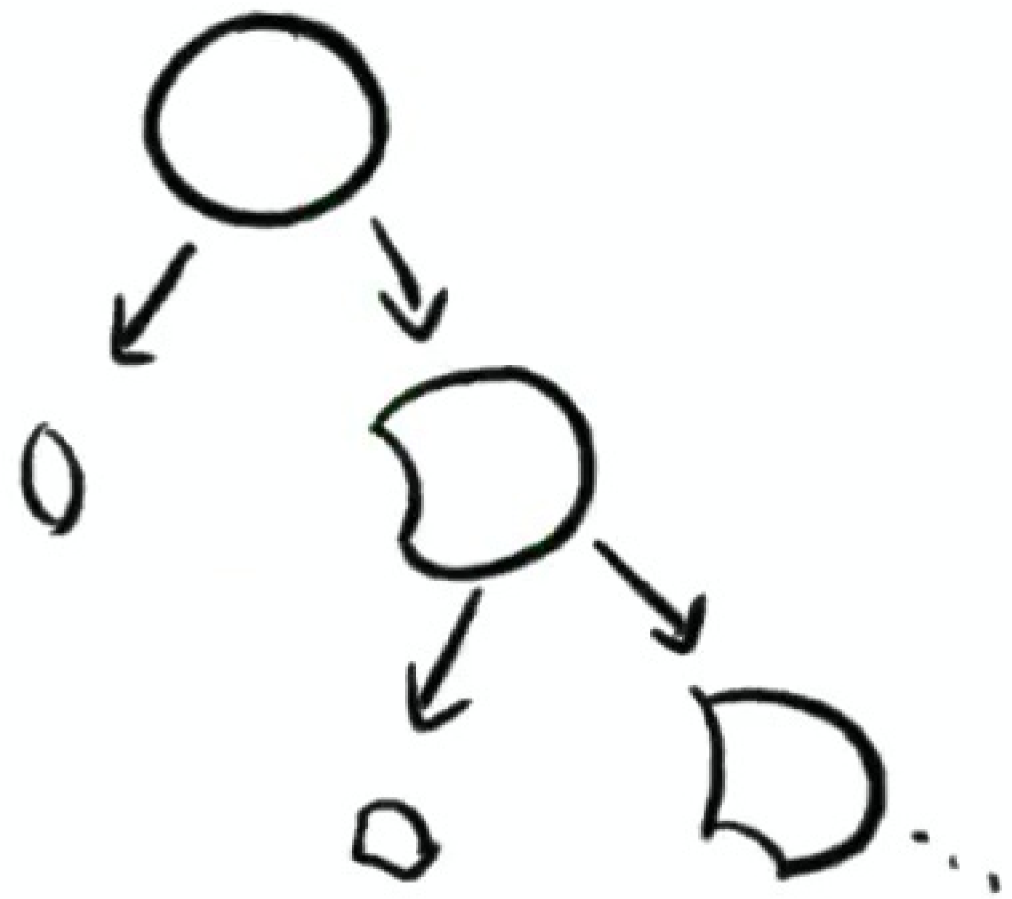


Expander Decomposition: Algorithm

Near-linear time (Saranurak-Wang 2019)

If lopsided recursion, certify large side is an expander

Uses non-stop cut-matching game [RST'14]

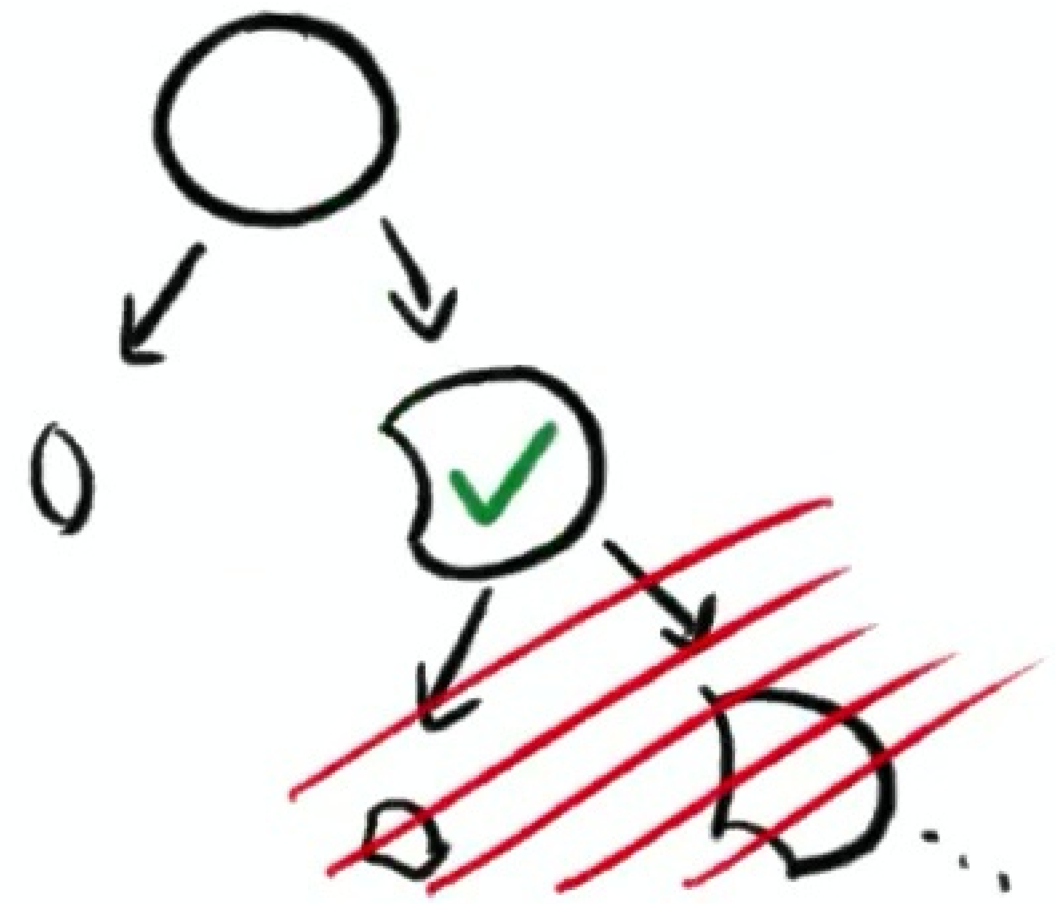


Expander Decomposition: Algorithm

Near-linear time (Saranurak-Wang 2019)

If lopsided recursion, certify large side is an expander

Uses non-stop cut-matching game [RST'14]

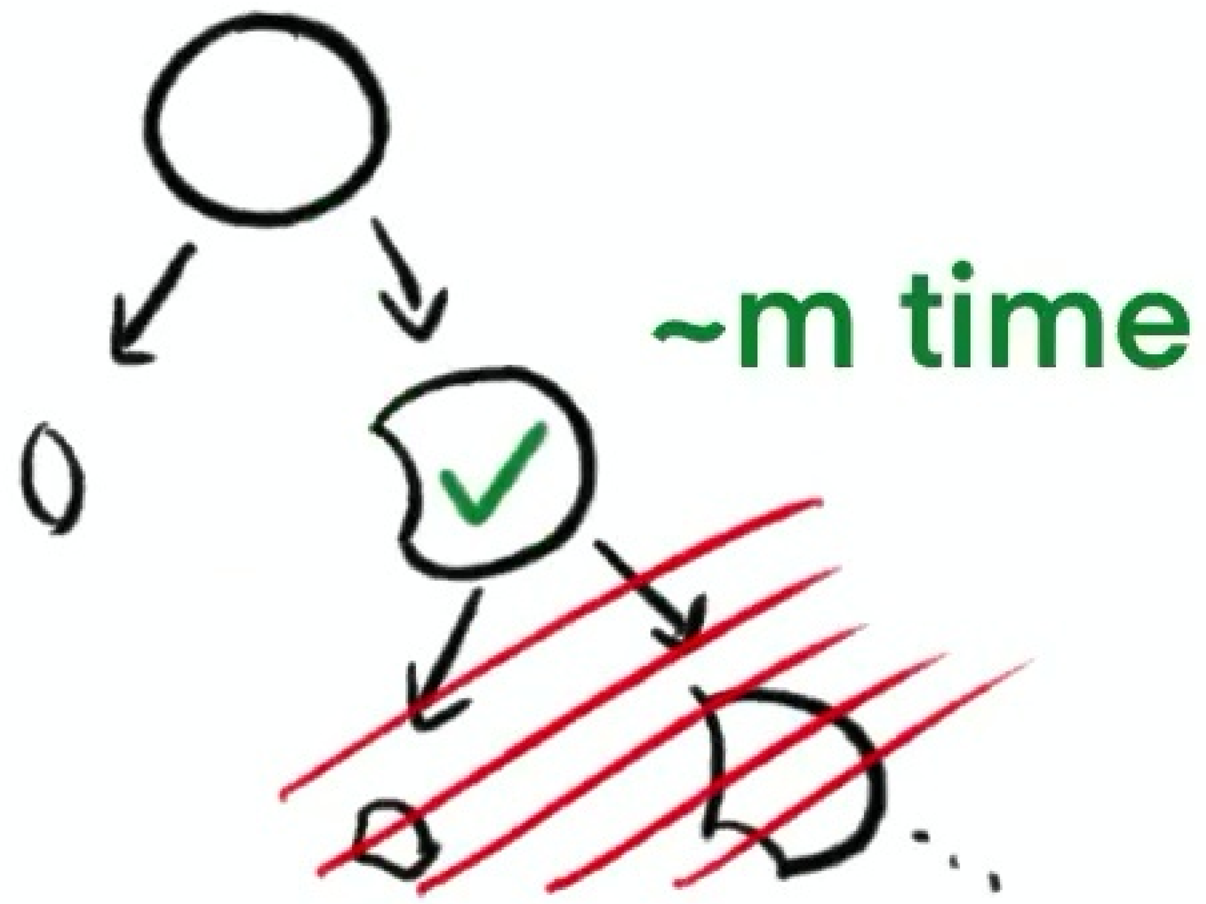


Expander Decomposition: Algorithm

Near-linear time (Saranurak-Wang 2019)

If lopsided recursion, certify large side is an expander

Uses non-stop cut-matching game [RST'14]



Using Expander Decomposition

High-level template:

- Decompose graph into expanders
- Solve on each expander
- Stitch solutions together
- Recursively solve inter-cluster edges

Using Expander Decomposition

High-level template:

- Decompose graph into expanders
- Solve on each expander
- Stitch solutions together
- Recursively solve inter-cluster edges

Ideal guarantee: if S_1 solution to instance G_1 and S_2 solution to instance G_2, then S_1+S_2 solution to instance G_1+G_2

Example: Graph Sparsification

H is a $(1+\varepsilon)$-approx cut sparsifier of G if

for all $S \subseteq V$, $(1-\varepsilon)\partial_G S <= \partial_H S <= (1+\varepsilon)\partial_G S$

Example: Graph Sparsification

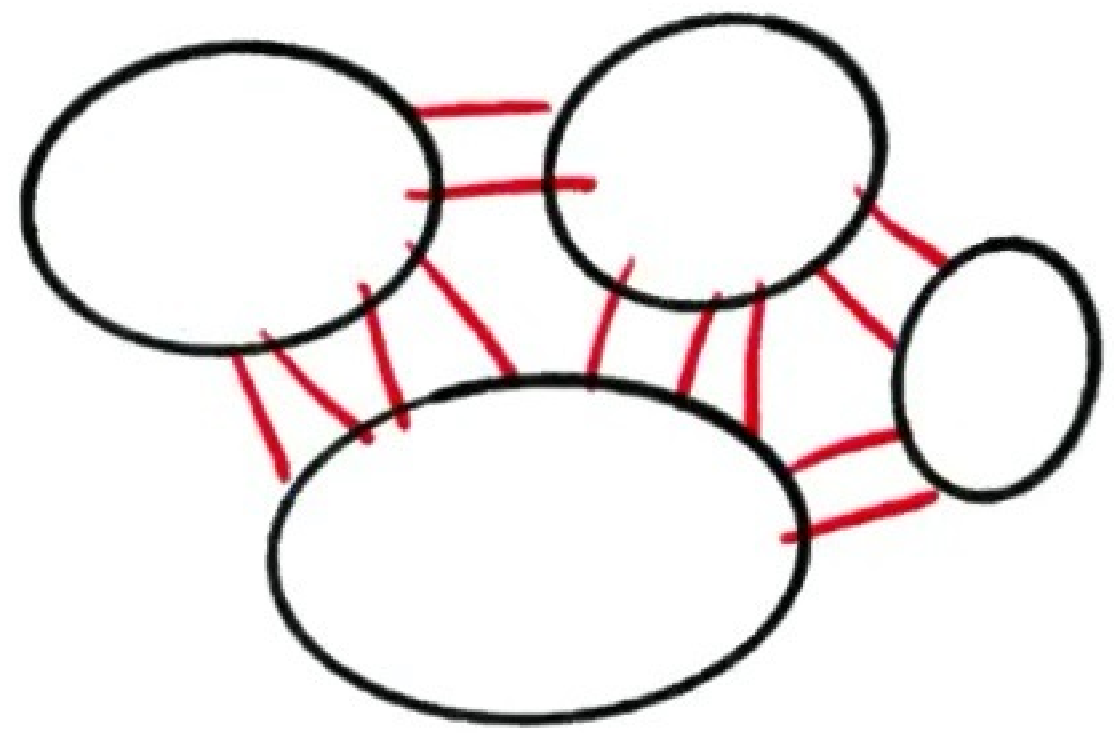
H is a $(1+\varepsilon)$-approx cut sparsifier of G if for all $S \subseteq V$, $(1-\varepsilon)\partial_G S <= \partial_H S <= (1+\varepsilon)\partial_G S$

Ideal guarantee: if H_1 is $(1+\varepsilon)$-sparsifier of G_1 and H_2 is $(1+\varepsilon)$-sparsifier of G_2, then H_1+H_2 is $(1+\varepsilon)$-sparsifier of G_1+G_2

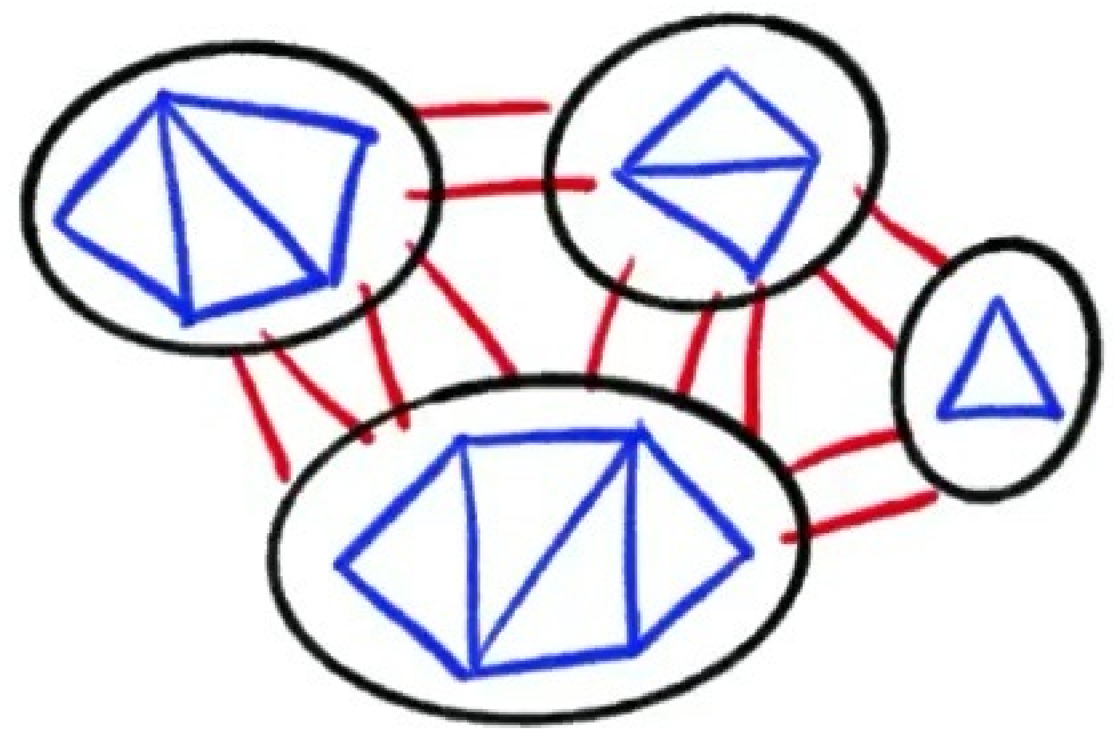
Example: Graph Sparsification

Algorithm:

- Run expander decomposition



Example: Graph Sparsification



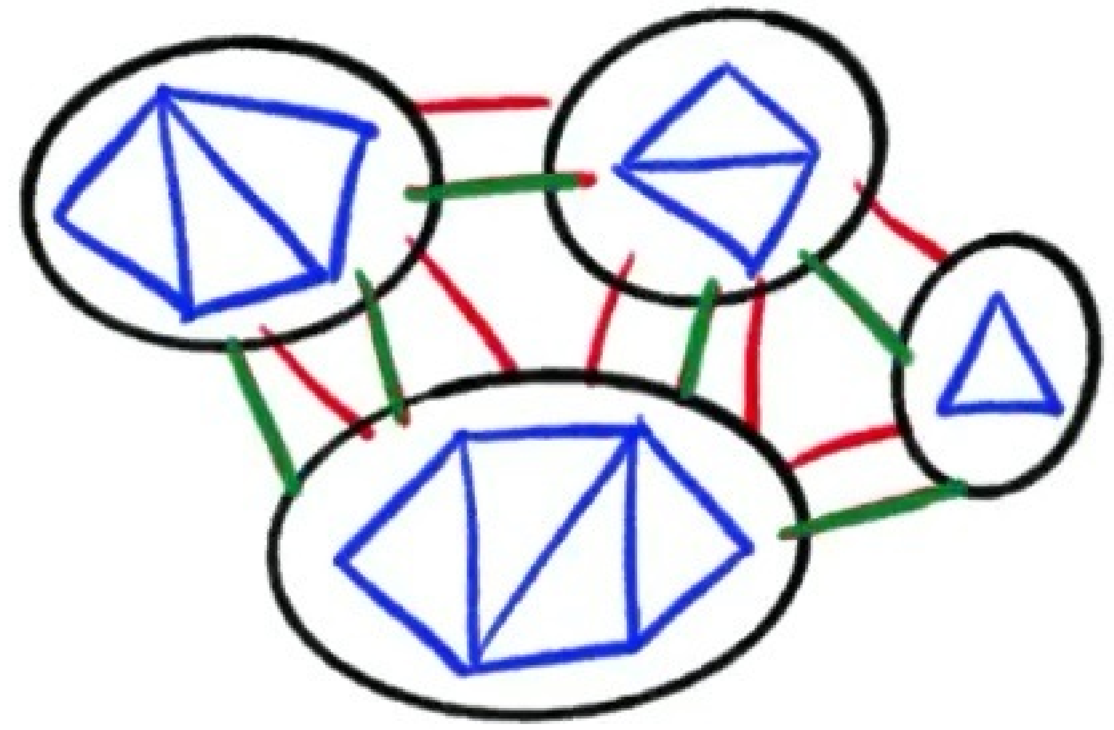
Algorithm:

- Run expander decomposition
- Find sparsifier H_i of expander $G[V_i]$

Example: Graph Sparsification

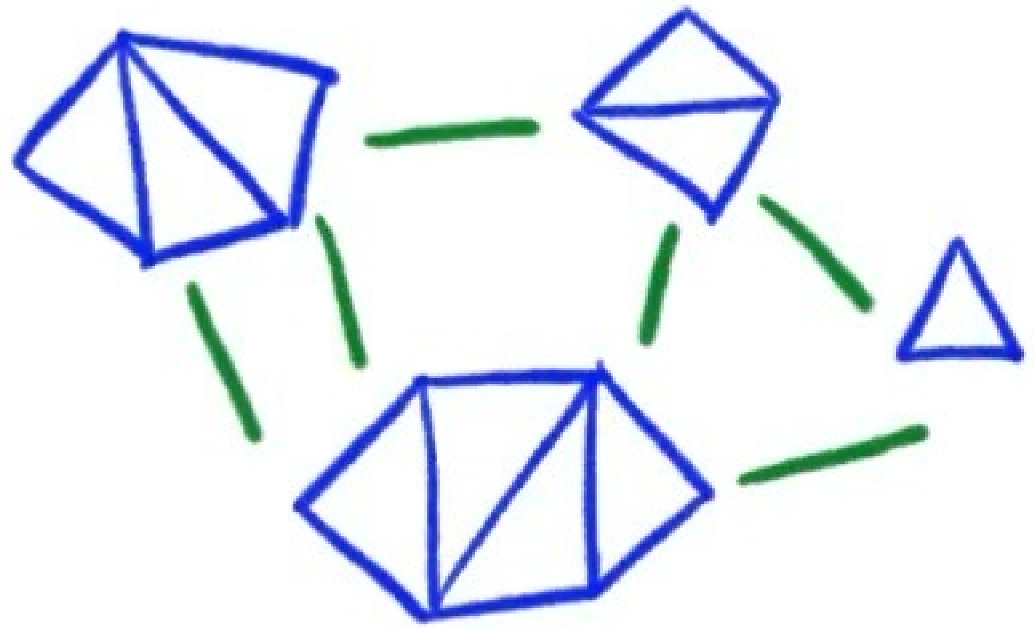
Algorithm:

- Run expander decomposition
- Find sparsifier H_i of expander $G[V_i]$
- Recursively sparsify subgraph of inter-cluster edges, call the sparsifier H'



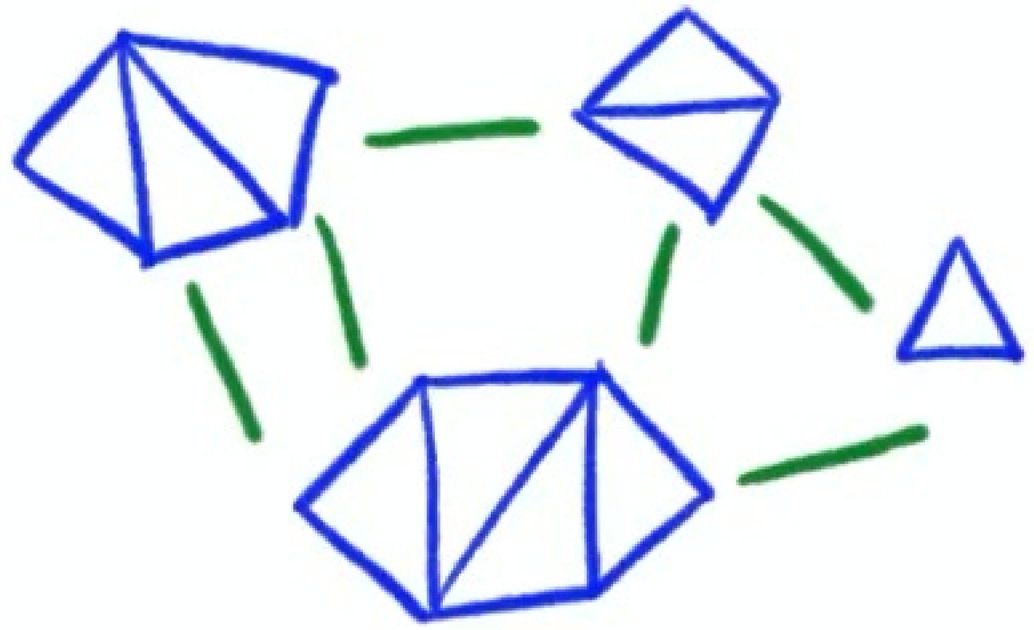
Example: Graph Sparsification

Algorithm:



- Run expander decomposition
- Find sparsifier H_i of expander $G[V_i]$
- Recursively sparsify subgraph of inter-cluster edges, call the sparsifier H'
- Return $H_1+\ldots+H_k+H'$

Example: Graph Sparsification



Algorithm:

- Run expander decomposition
- Find sparsifier H_i of expander $G[V_i]$
- Recursively sparsify subgraph of inter-cluster edges, call the sparsifier H'
- Return $H_1+\ldots+H_k+H'$
- State-of-the-art for "for-each" sparsifier

Why are expanders useful?

Graph sparsification: can sample each edge (u,v) with probability ~1/min{deg(u),deg(v)}

Why are expanders useful?

Graph sparsification: can sample each edge (u,v) with probability ~1/min{deg(u),deg(v)}

Global minimum cut: smaller side has <=1/ϕ vertices

Why are expanders useful?

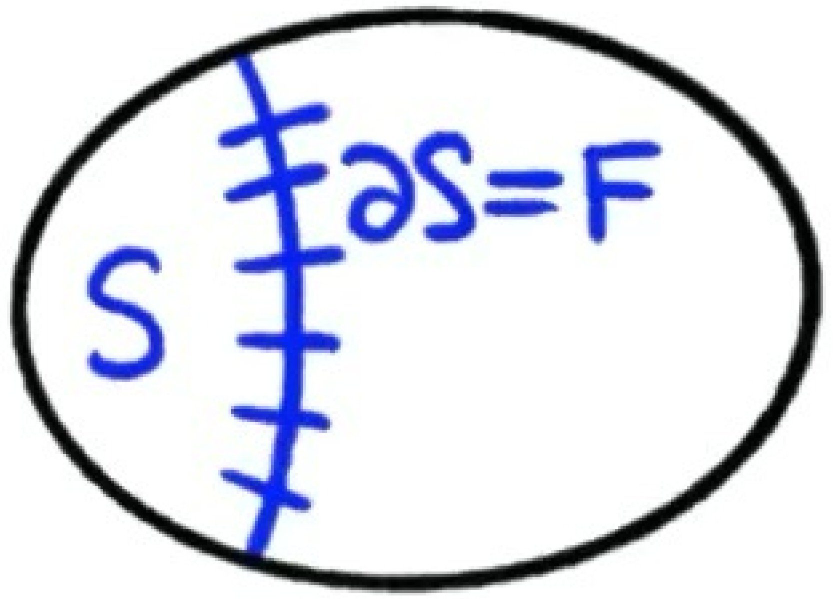
Graph sparsification: can sample each edge (u,v) with probability ~1/min{deg(u),deg(v)}

Global minimum cut: smaller side has <=1/ϕ vertices

Dynamic graph connectivity [NSW'17]: if delete $D \subseteq E$ and query s,t, can check whether s,t connected in $O(|D|/\phi)$ time

Global Minimum Cut

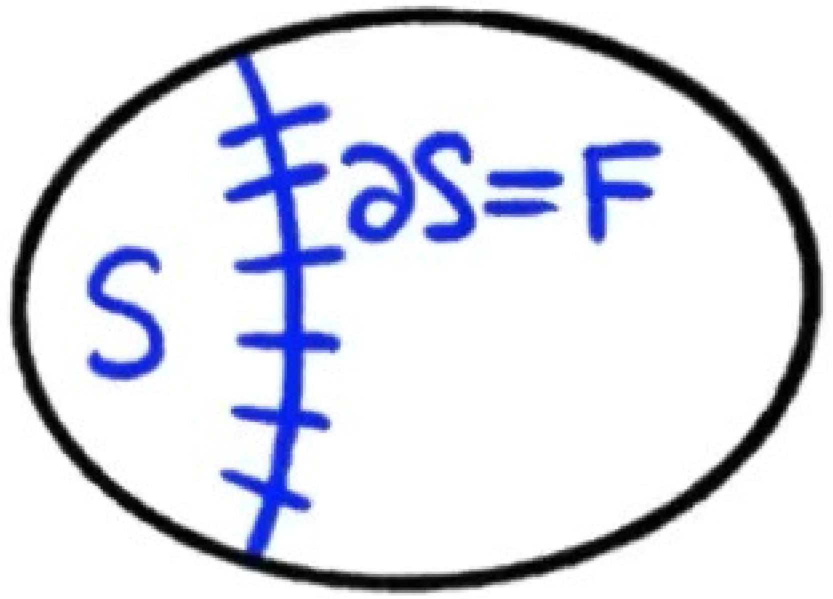
The (global) minimum cut of G is the



minimum value of ∂S over nontrivial S⊆V

= minimum size F⊆E s.t. G-F is disconnected

Global Minimum Cut

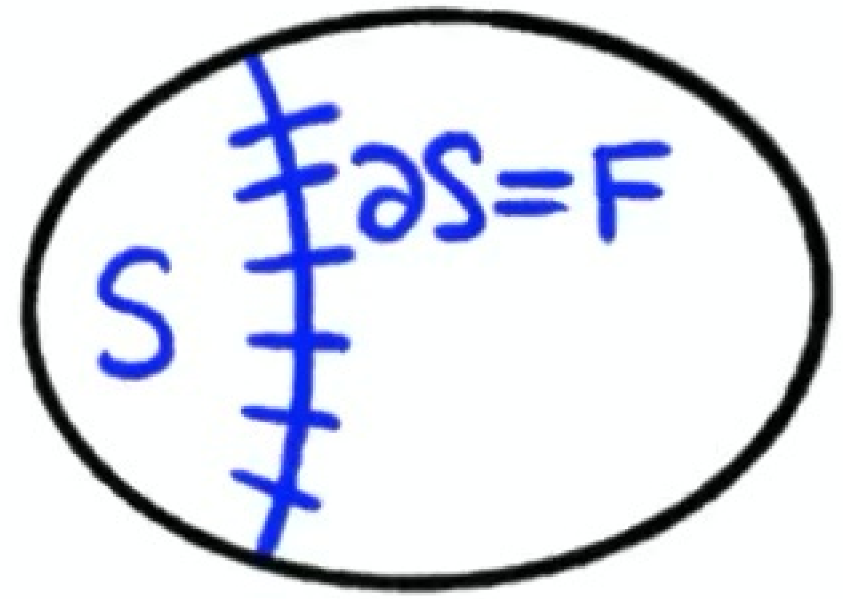


The (global) minimum cut of G is the minimum value of ∂S over nontrivial $S \subseteq V$

= minimum size $F \subseteq E$ s.t. G-F is disconnected

Lemma: if G is a ϕ-expander and S is smaller side of the minimum cut, then $|S| <= 1/\phi$

Global Minimum Cut



The (global) minimum cut of G is the minimum value of ∂S over nontrivial S⊆V

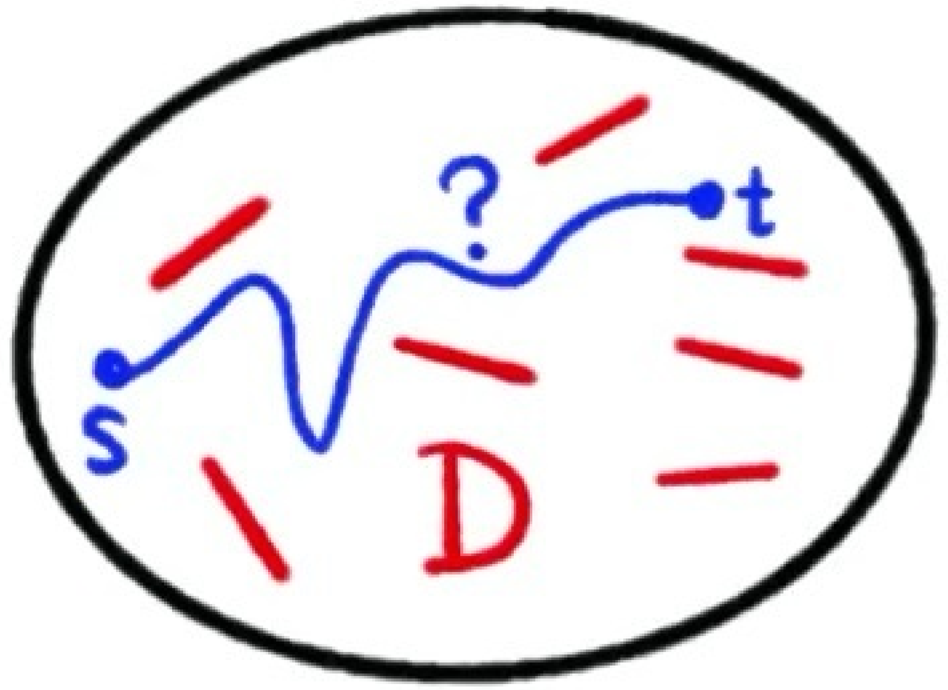
= minimum size F⊆E s.t. G-F is disconnected

Lemma: if G is a ϕ-expander and S is smaller side of the minimum cut, then |S|<=1/ϕ

Proof: deg(v) >= |∂S| since ∂{v} is valid cut, so |∂S| >= ϕ*vol(S) >= ϕ*|S|*|∂S|

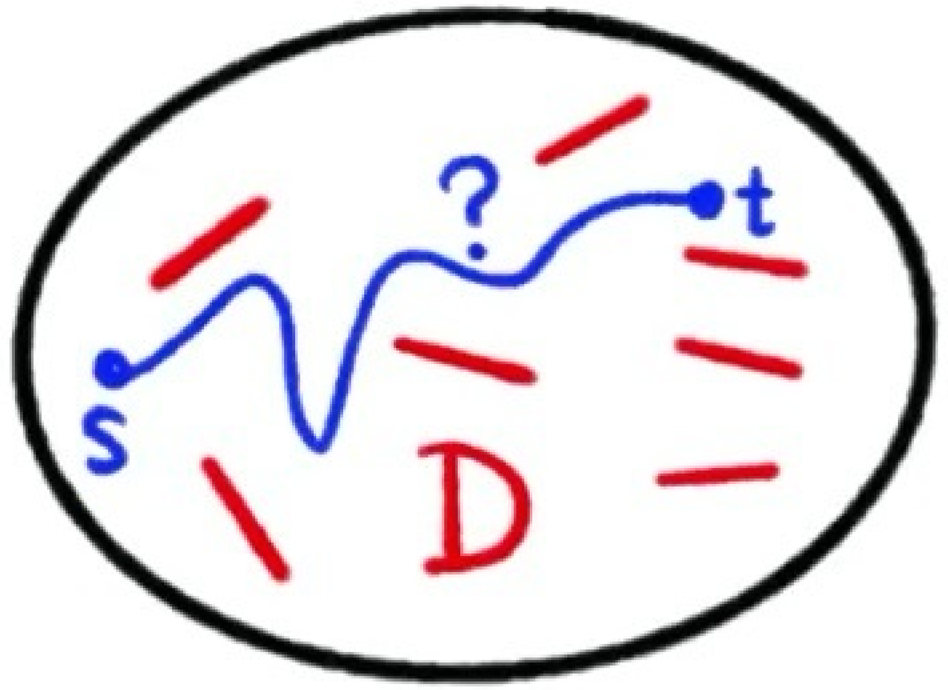
Dynamic Connectivity

Lemma: if G is a ϕ-expander, then on query $F \subseteq E$ and s,t, can check whether s,t connected in $O(|D|/\phi)$ time



Dynamic Connectivity

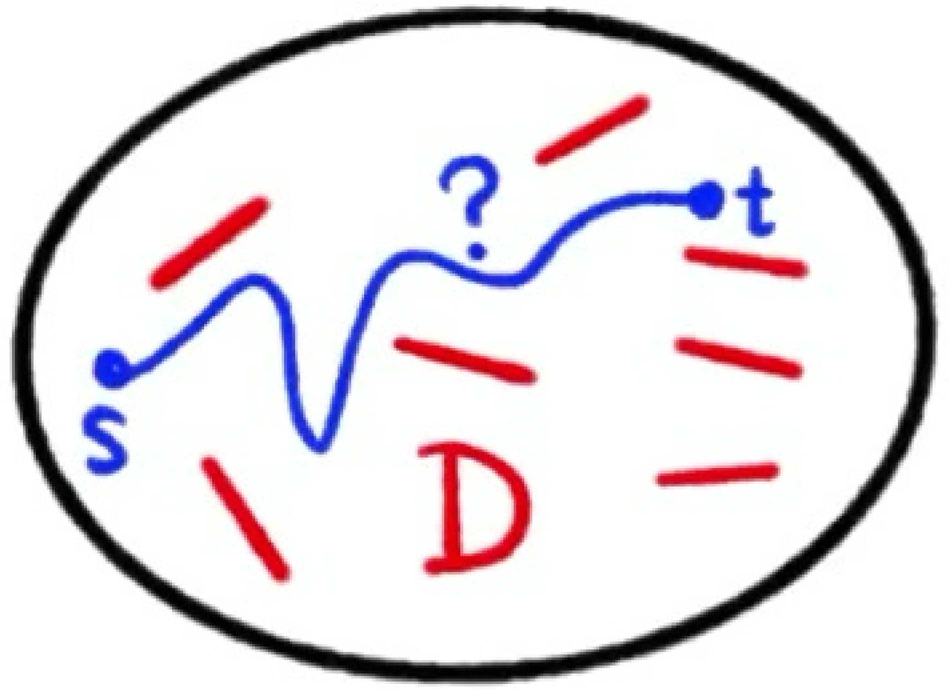
Lemma: if G is a ϕ-expander, then on query $F \subseteq E$ and s,t, can check whether s,t connected in $O(|D|/\phi)$ time



Proof: if s,t disconnected by cut S, then $\partial S \subseteq D$ but also $|\partial S| >= \phi * vol(S)$, so $vol(S) <= \frac{D}{\phi}$

Dynamic Connectivity

Lemma: if G is a ϕ-expander, then on query $F \subseteq E$ and s,t, can check whether s,t connected in $O(|D|/\phi)$ time

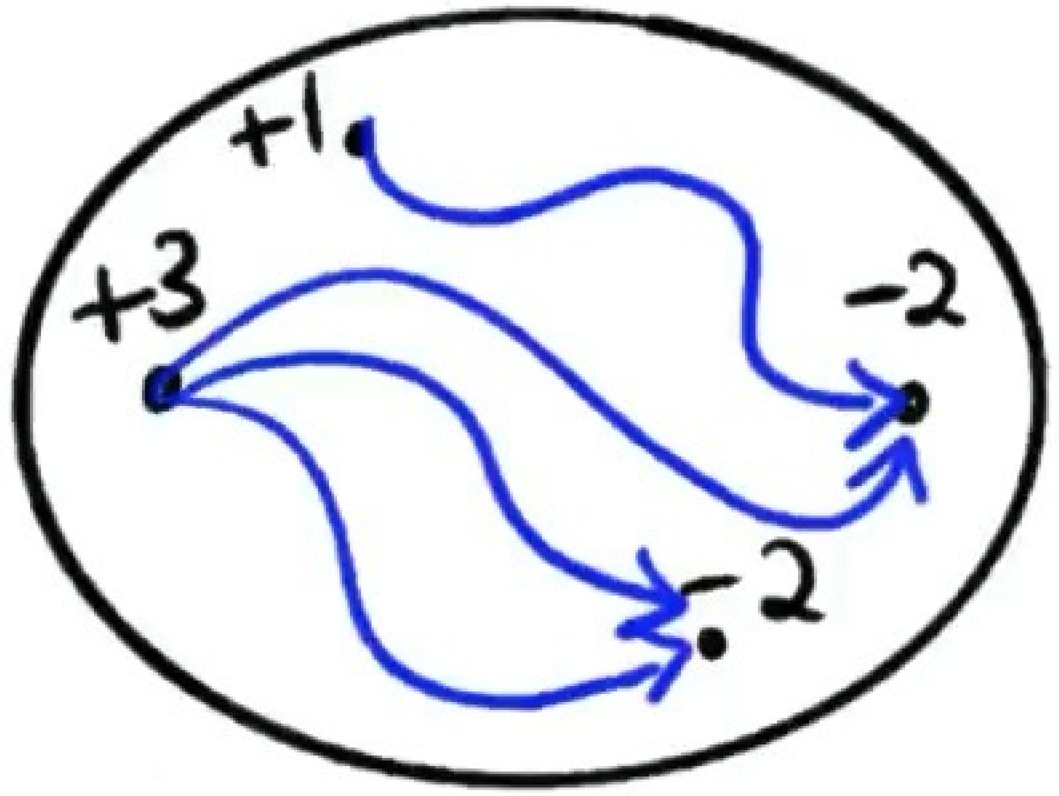


Proof: if s,t disconnected by cut S, then $\partial S \subseteq D$ but also $|\partial S| >= \phi^*\text{vol}(S)$, so $\text{vol}(S) <= \frac{D}{\phi}$

The algorithm runs DFS/BFS from s but terminates after reaching total volume D/ϕ, repeats from t in case t is smaller side

Expanders: Flow Definition

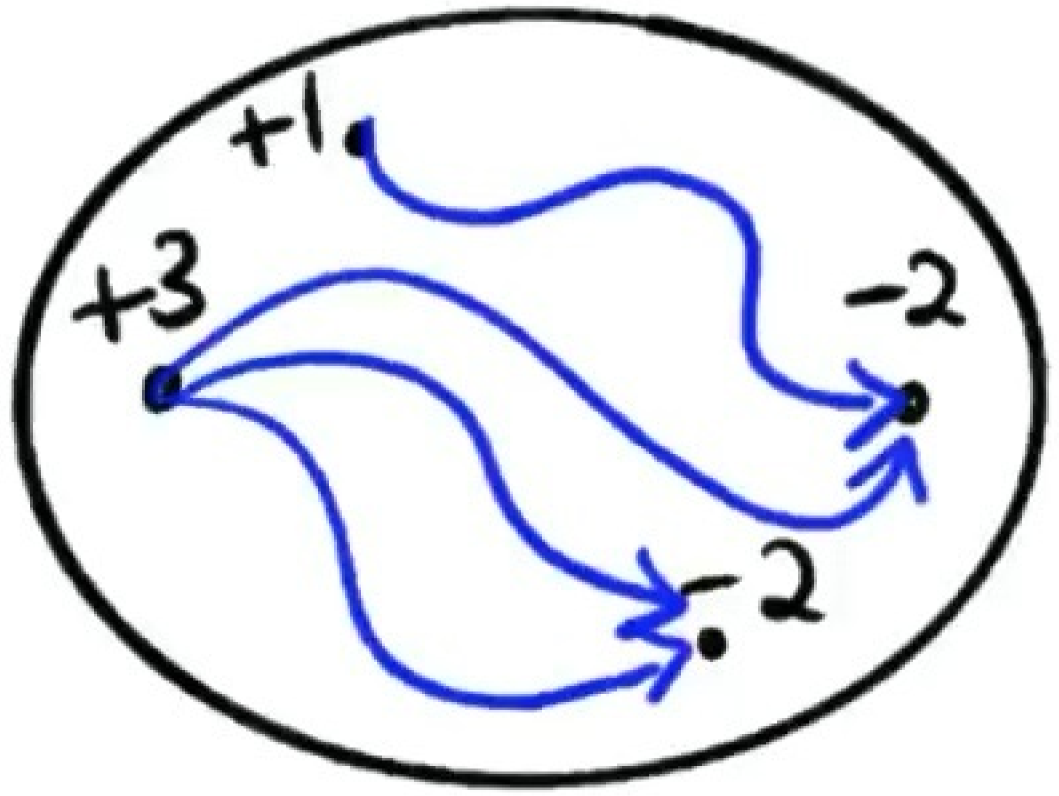
A vertex demand $b \in \mathbb{R}^V$ represents excesses/ deficits at vertices that need to be routed



Expanders: Flow Definition

A vertex demand $b \in \mathbb{R}^V$ represents excesses/ deficits at vertices that need to be routed

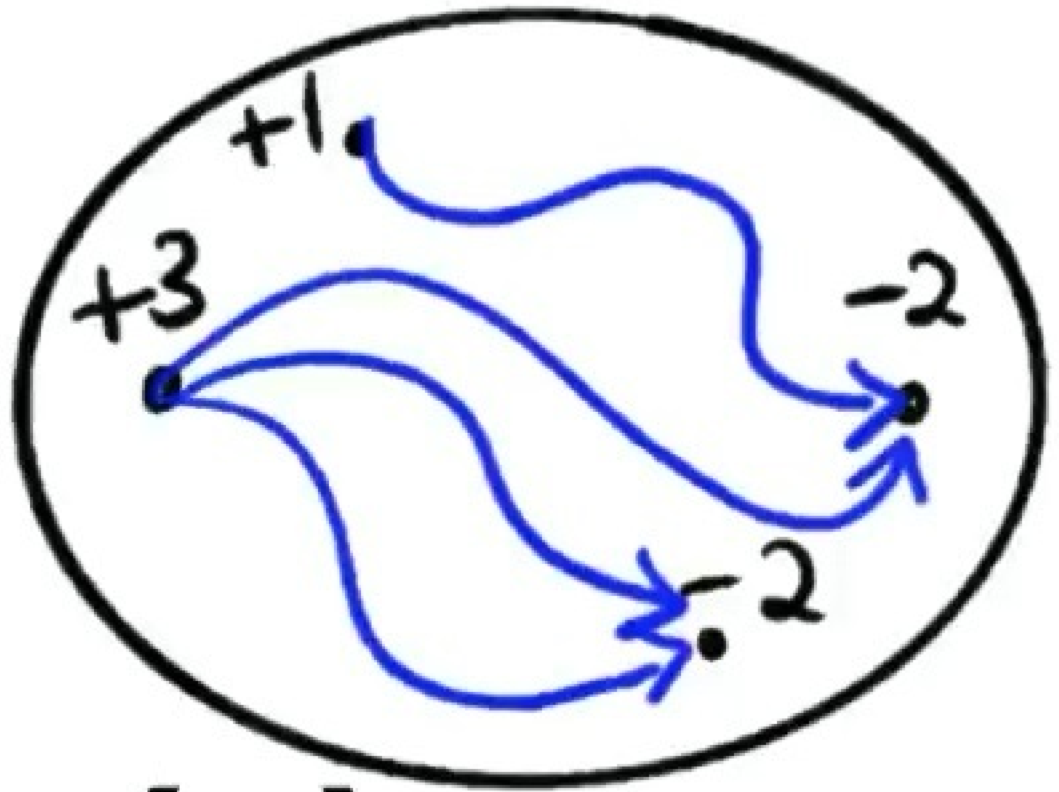
Example: s-t flow: $b = 1_s - 1_t$



Expanders: Flow Definition

A vertex demand $b \in \mathbb{R}^V$ represents excesses/ deficits at vertices that need to be routed

Example: s-t flow: $b = 1_s - 1_t$

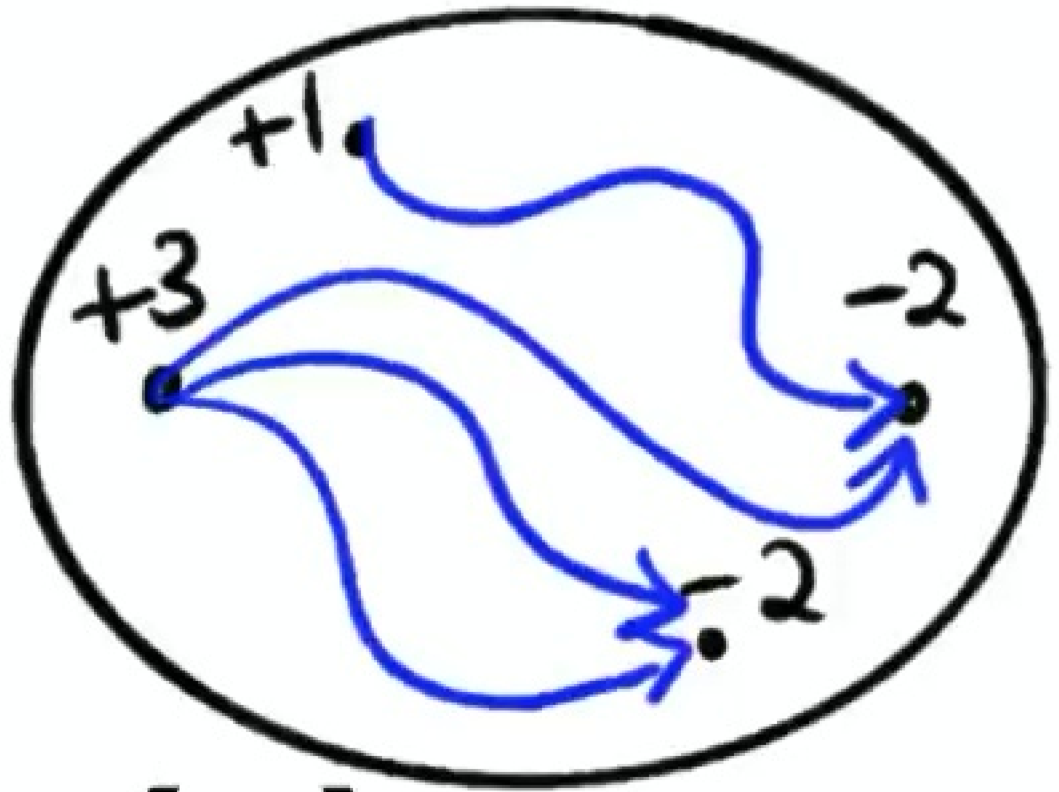


Lemma: G is a ϕ-expander iff any vertex demand b with $|b(v)| <= deg(v)$ can be routed with congestion $1/\phi$

Expanders: Flow Definition

A vertex demand $b \in \mathbb{R}^V$ represents excesses/ deficits at vertices that need to be routed

Example: s-t flow: $b = 1_s - 1_t$



Lemma: G is a ϕ-expander iff any vertex demand b with $|b(v)| <= deg(v)$ can be routed with congestion $1/\phi$

Proof: max-flow/min-cut theorem

Max-flow on Expanders

$\max_v |b(v)|/\deg(v)$ already $1/\phi$-approximates flow congestion

Max-flow on Expanders

$\max_v |b(v)|/\deg(v)$ already $1/\phi$-approximates flow congestion

Many flow algorithms run fast on an expander: blocking flow, push relabel, multiplicative weights

Max-flow on Expanders

$\max_v |b(v)|/\deg(v)$ already $1/\phi$-approximates flow congestion

Many flow algorithms run fast on an expander: blocking flow, push relabel, multiplicative weights

Simple $(1+\varepsilon)$-approx max-flow on expanders (exact is still hard!)

Cut-Matching Game

Tuesday, April 15, 2025 2:34 AM

Recall the Sparsest Cut problem: minimize $\dfrac{|\partial S| \leftarrow \#\text{edges between } S, \bar{S}}{\min\{vol(S), vol(\bar{S})\}}$

$vol(S) = \sum_{v \in S} deg(v)$

Recall Cheeger rounding: find smallest nontrivial eigenvector of Laplacian,
try all possible "threshold cuts"
best one has $\Phi(S) \le 2\sqrt{\lambda} \le O(\sqrt{\Phi_G})$
"$\sqrt{\ }$ – approximation"

Very good if Φ_G is constant, but what if Φ_G is very small?

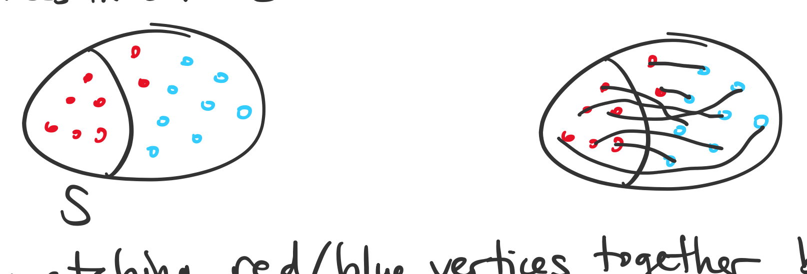
Cut-Matching Game [Khandekar–Rao–Vazirani '07]

$O(\log^2 n)$–approx multiplicative, i.e. find S: $\Phi(S) \le O(\Phi_G \log^2 n)$

This lecture: minimize $\Psi(S) = \dfrac{|\partial S|}{\min\{|S|, |\bar{S}|\}}$ instead (equivalent when G is regular)

Setting: Guess ψ. Either find S: $\Psi(S) < \psi$, or certify $\Psi_G \ge \Omega(\psi/\log^2 n)$.

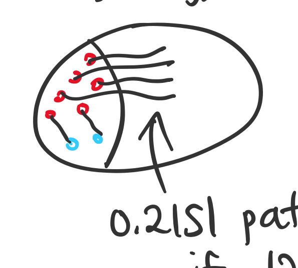
Idea: Suppose I color half the vertices red, other half blue, s.t. all vertices in some S are red:



Consider matching red/blue vertices together by paths:
find paths between red/blue vertices s.t.

- each red/blue vertex is start/end of exactly one path,
- each edge belongs to $\le 1/\psi$ paths (for some guess ψ)

If $\Psi(S) < \psi$, then S makes this problem infeasible.

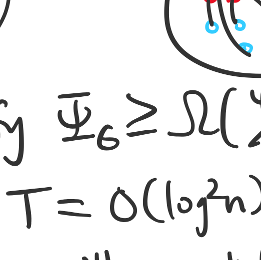


$|S|$ paths exit S

$|\partial S|$ edges, each supports $\le \psi$ paths,
but $|\partial S| = |S| \cdot \Psi(S) < |S| \cdot \psi$,
so cannot support all $|S|$ paths

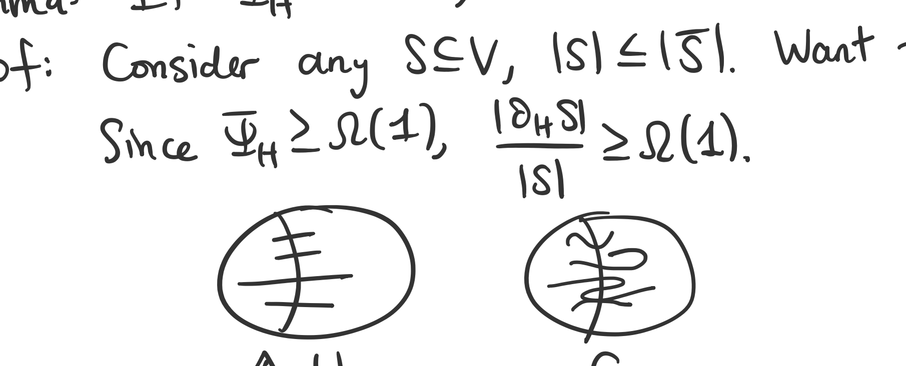
Max-flow/Min-cut theorem: either successfully match red/blue, or find violating S with $|\partial S| < |S| \cdot \psi \iff \Psi(S) < \psi$.

In fact, just need, say, 60% of vertices in S red:



$0.2|S|$ paths need to cross,
so if $|\partial S| < 0.2|S| \cdot \psi \iff \Psi(S) < 0.2\psi$ "5-approx"

Goal is to guess colorings, hoping to color most of some sparse cut red (or blue). If can't find good coloring, certify $\Psi_G \ge \Omega(\psi/\log^2 n)$.

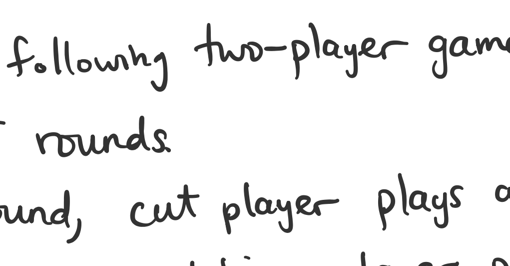


How to certify $\Psi_G \ge \Omega(\psi/\log^2 n)$?

Suppose find $T = O(\log^2 n)$ cut-matchings. Let H = union of all matchings. We will certify $\Psi_H \ge \Omega(1)$.

Lemma: If $\Psi_H \ge \Omega(1)$, then $\Psi_G \ge \Omega(\psi/T)$.

Proof: Consider any $S \subseteq V$, $|S| \le |\bar{S}|$. Want to show $\dfrac{|\partial_G S|}{|S|} \ge \Omega(\frac{\psi}{T})$.
Since $\Psi_H \ge \Omega(1)$, $\dfrac{|\partial_H S|}{|S|} \ge \Omega(1)$.



$\partial_H S \ge \Omega(|S|)$ matched edges

each corresponds to a path (with same endpoints) in G

Each edge in G belongs to $\le T/\psi$ paths, so
$\Omega(|S|) \le \partial_H S = \#\text{paths crossing} \le T/\psi \cdot \partial_G S \Rightarrow \dfrac{|\partial_G S|}{|S|} \ge \Omega(\frac{T}{\psi})$

Consider the following two-player game on H:

There are T rounds

On each round, cut player plays a balanced red/blue coloring ("cut")
matching player plays perfect red-blue matching.

At the end, let H be union of all T matchings.

Theorem [KRV'07]: there is a cut player strategy for $T = O(\log^2 n)$ rounds s.t. for any matching player strategy, $\Psi_H \ge 1/2$.
In fact, a certain random walk will mix in H.

KRV Cut Player

Let M_t be the lazy random walk matrix for t^{th} matching:

$$M_t = \begin{bmatrix} \frac{1}{2} & & \frac{1}{2} \\ & \frac{1}{2} & \\ & & \ddots \\ \frac{1}{2} & & & \frac{1}{2} \end{bmatrix} \leftarrow \text{if } (i,j) \text{ matched}$$

On each iteration $t+1$, choose random unit vector $r_{t+1} \perp \mathbb{1}$ and compute $u_{t+1} = M_t M_{t-1} \cdots M_1 r_{t+1}$.

Color smallest $\frac{n}{2}$ coordinates of u_{t+1} red, largest $\frac{n}{2}$ blue.

Matching player plays M_{t+1}

Remark: on first iteration, $u_1 = r_1$ so coloring is random.

Analysis

Consider $P_t = M_t M_{t-1} \cdots M_1$, the "round-robin" walk matrix

Want to show that $P_T \approx$ all-$\frac{1}{n}$'s-matrix: random walk mixes

Consider potential function $\Phi_t = \sum_{ij} (P_T(i,j) - \frac{1}{n})^2 = \sum_i \| P_T(i) - \frac{\mathbb{1}}{n} \|_2^2$. i^{th} row of P_T

Lemma: if $\Phi_T \le \frac{1}{4n^2}$, then $\Psi_H \ge \frac{1}{2}$.

Proof: P_t is the transition matrix for the round-robin walk, i.e. $P_t(u,v)$ is probability of finishing at u if starting at v.

Consider round-robin walk starting at random vertex.
After each step, still uniformly random vertex.

$$\Pr[\text{start in } S, \text{ end in } \bar{S}] = \sum_{\substack{v \in S \\ u \in \bar{S}}} \frac{1}{n} \cdot P_t(u,v) \ge \sum_{\substack{v \in S \\ u \in \bar{S}}} \frac{1}{n} \cdot \frac{1}{2n} = \frac{|S||\bar{S}|}{2n^2}.$$

For a vertex to start in S and end in $\bar{S}$, it must cross $S \to \bar{S}$ on some step. Union bound over all steps:

$\Pr[\text{start in } S, \text{ end in } \bar{S}]$

$$\le \sum_t \Pr[\text{cross } S \to \bar{S} \text{ on step } t]$$

$$= \sum_t \frac{\#\text{edges } S - \bar{S} \text{ in } M_t}{2n}$$ [at random vertex before step t, so $\frac{1}{2n}$ probability to cross each edge]

$$= \frac{\partial_H S}{2n}.$$

Therefore, $\dfrac{\partial_H S}{2n} \ge \dfrac{|S||\bar{S}|}{2n^2} \Rightarrow \dfrac{\partial_H S}{|S|} \ge \dfrac{|\bar{S}|}{n} \ge \dfrac{1}{2}$.

Lemma: Potential decrease $\Phi_t - \Phi_{t+1}$ from matching M_{t+1} is at least
$\frac{1}{2} \sum_{(i,j) \in M_{t+1}} \| \vec{P_t}(i) - \vec{P_t}(j) \|_2^2$. i^{th} row of P_t

Proof: By definition,

$$P_{t+1} = \begin{bmatrix} -\vec{P}_{t+1}(i)- \\ \vdots \end{bmatrix} = \begin{bmatrix} M_{t+1} \end{bmatrix} \begin{bmatrix} -\vec{P}_t(i)- \\ \vdots \end{bmatrix}$$

So for each edge $(i,j) \in M_{t+1}$, $\vec{P}_{t+1}(i) = \vec{P}_{t+1}(j) = \frac{1}{2}(\vec{P}_t(i) + \vec{P}_t(j))$

Consider difference of contributions of i,j in $\Phi_t - \Phi_{t+1}$:

$$\underbrace{\left(\|\vec{P}_{t+1}(i) - \tfrac{\mathbb{1}}{n}\|_2^2 + \|\vec{P}_{t+1}(j) - \tfrac{\mathbb{1}}{n}\|_2^2\right)} - \left(\|\vec{P}_t(i) - \tfrac{\mathbb{1}}{n}\|_2^2 + \|\vec{P}_t(j) - \tfrac{\mathbb{1}}{n}\|_2^2\right)$$

$$= 2 \left\| \tfrac{1}{2}(\vec{P}_t(i) + \vec{P}_t(j)) - \tfrac{\mathbb{1}}{n} \right\|_2^2$$

Set $u = \vec{P}_t(i) - \frac{\mathbb{1}}{n}$, $v = \vec{P}_t(j) - \frac{\mathbb{1}}{n}$:

$$= \tfrac{1}{2}\|u+v\|_2^2 - \|u\|_2^2 - \|v\|_2^2$$

$$= \tfrac{1}{2}\|u-v\|_2^2$$

$$= \tfrac{1}{2}\|\vec{P}_t(i) - \vec{P}_t(j)\|_2^2.$$

Sum up the contributions of each $(i,j) \in M_{t+1}$.

Recall that $u_{t+1} = P_t r_{t+1}$ for random unit $r_{t+1} \perp \mathbb{1}$, and for median η, all coordinates $\le \eta$ are red, $\ge \eta$ are blue.

Lemma: $\sum_{(i,j) \in M} |u_{t+1}(i) - u_{t+1}(j)|^2 \ge \sum_i u_{t+1}(i)^2$.

Proof: $\sum_{(i,j) \in M} |u_{t+1}(i) - u_{t+1}(j)|^2 \ge \sum_{(i,j) \in M} (|u_{t+1}(i) - \eta|^2 + |u_{t+1}(j) - \eta|^2)$

$= \sum_i |u_{t+1}(i) - \eta|^2 \leftarrow$ minimized when η = average of $u_{t+1}(i)$'s $= 0$ since $u_{t+1} \perp \mathbb{1}$

$\ge \sum_i u_{t+1}(i)^2$.

Lemma [Gaussian Behavior of Projections]:
For any $v \in \mathbb{R}^d$ and random unit vector $r \in \mathbb{R}^d$,

① $\mathbb{E}[\langle v, r \rangle^2] = \frac{1}{d}\|v\|_2^2$

② $\Pr[\langle v, r \rangle^2 \ge \frac{x}{d}\|v\|_2^2] \le e^{-x/4}$ for any $x \le \frac{d}{16}$.

Proof: By rotational symmetry and scaling, can assume $v = (1, 0, \dots, 0)$.
Then it's behavior of a single Gaussian.

Lemma [Expected Potential Decrease]: $\mathbb{E}[\Phi_t - \Phi_{t+1}] \ge \Omega\left(\frac{\Phi_t}{\log n}\right)$.

Proof:
Applying lemma to u_{t+1} (and $(n-1)$-dimensional subspace $\perp \mathbb{1}$)

- $\mathbb{E}[\underbrace{u_{t+1}(i)}^2] = \frac{1}{n-1}\|P_t(i) - \frac{\mathbb{1}}{n}\|_2^2$
  $= \langle 1_i, u_{t+1} \rangle = \langle 1_i, P_t r_{t+1} \rangle = \langle P_t(i), r_{t+1} \rangle = \langle \underbrace{P_t(i) - \tfrac{\mathbb{1}}{n}}_{\perp \mathbb{1}}, r_{t+1} \rangle$
- $\underbrace{|u_{t+1}(i) - u_{t+1}(j)|}^2 \le \frac{O(\log n)}{n-1} \|P_t(i) - P_t(j)\|_2^2$ with probability $1 - \frac{1}{poly(n)}$
  $= \langle 1_i - 1_j, u_{t+1} \rangle = \dots = \langle P_t(i) - P_t(j), r_{t+1} \rangle$

Putting everything together,

$$\mathbb{E}[\Phi_t - \Phi_{t+1}] = \frac{1}{2} \sum_{(i,j) \in M_{t+1}} \mathbb{E}\left[\|\vec{P}_t(i) - \vec{P}_t(j)\|_2^2\right]$$

$$\ge \Omega\left(\frac{n-1}{\log n}\right) \sum_{(i,j) \in M_{t+1}} \mathbb{E}\,|u_{t+1}(i) - u_{t+1}(j)|^2$$ [union bound over all i,j]

$$\ge \Omega\left(\frac{n-1}{\log n}\right) \sum_i \mathbb{E}[u_{t+1}(i)^2]$$

$$= \Omega\left(\frac{1}{\log n}\right) \sum_i \|P_t(i) - \tfrac{\mathbb{1}}{n}\|_2^2$$

$$= \Omega\left(\frac{\Phi_t}{\log n}\right).$$

Since $\Phi_0 = \sum_i \|I(i) - \frac{\mathbb{1}}{n}\|_2^2 \le poly(n)$ and each round decreases multiplicatively by $\Omega(\frac{1}{\log n})$ in expectation, we get $\Phi_T \le \frac{1}{4n^2}$ after $T = O(\log^2 n)$ rounds with probability $1 - \frac{1}{poly(n)}$.