A Simple and Fast Reduction from Gomory-Hu Trees to Polylog Maxflows

Maximilian Probst Gutenberg

Rasmus Kyng

Weixuan Yuan

Wuwei Yuan

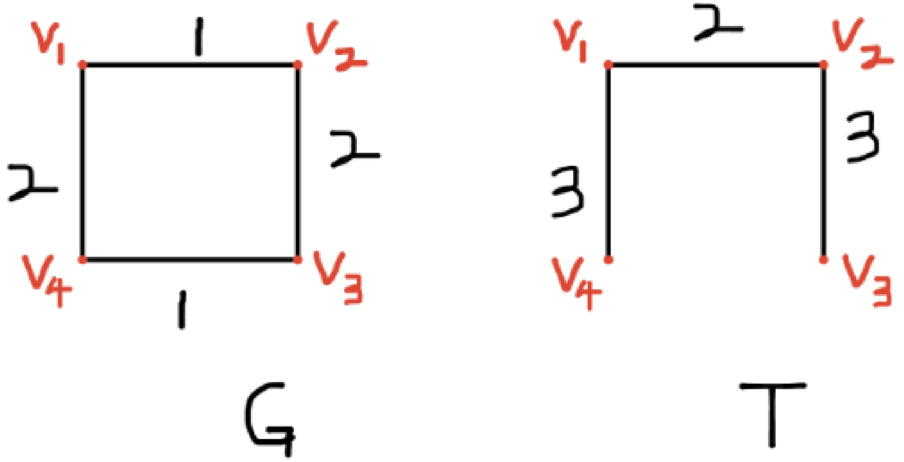
ETH Zurich

SODA'2026

Gomory-Hu Tree

- Let $G = (V, E, w)$ be an undirected weighted graph with n vertices and m edges.
- A Gomory-Hu tree of G is a tree defined on V such that

$$\forall s, t \in V, \mathrm{mincut}_T(s, t) = \mathrm{mincut}_G(s, t)$$



Prior Works

- Max-flow takes rand. [CKLPPS'22] / deter. [BCKLPPSS'23][BCKLMGS'24] $m^{1+o(1)}$ time.
-

Reference	Rand./Deter.	Time
Gomory-Hu'61	Deter.	$nm^{1+o(1)}$
AKLPST'21	Rand.	$\tilde{O}(n^{2.875})$
Zhang'21	Rand.	$\tilde{O}(n^2)$
AKLPST'21, ALPS'23	Rand.	$m^{1+o(1)}$
AGKLPS**YY**'25	Deter.	$m^{1+o(1)}$

$\tilde{O}(\,)$ hides polylog factors.

What is Missing?

- 20+ pages subroutine SSMC (single source mincut)
- Fastest algorithm : $\tilde{O}(T(n,m)) + n^{1+o(1)}$ still has a $n^{1+o(1)}$ overhead

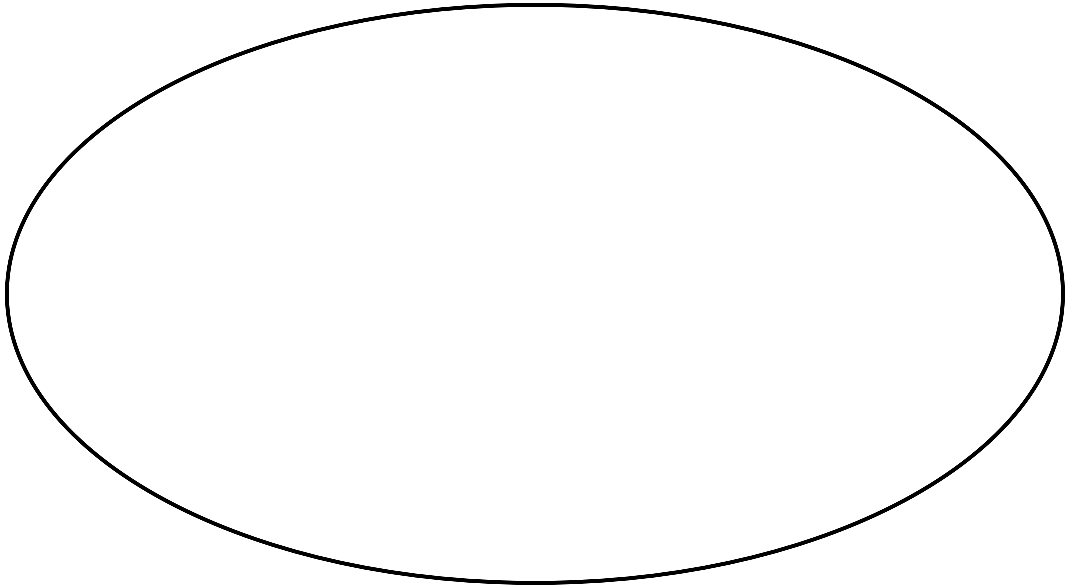
Our Contribution

- An optimal algorithm up to $\mathrm{polylog}(n)$ factors for <u>unweighted</u> graphs, i.e., a running time $O(T(n, m) \log^6 n)$

- Very simple algorithm without SSMC

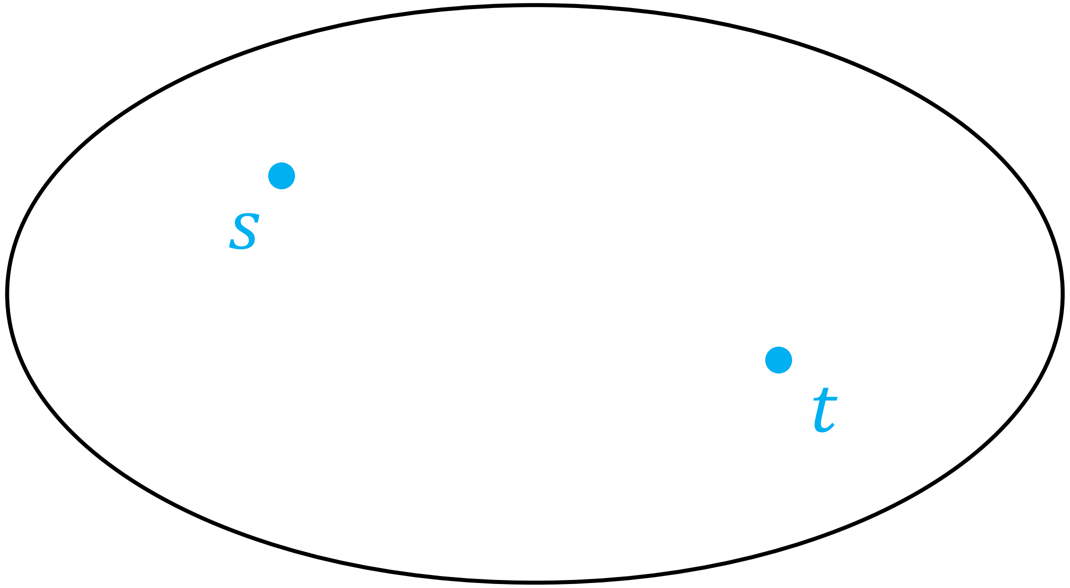
- Follow-up work [P**Y**'25]: a de-randomization of this work

Gomory and Hu's Algorithm [GH'61]



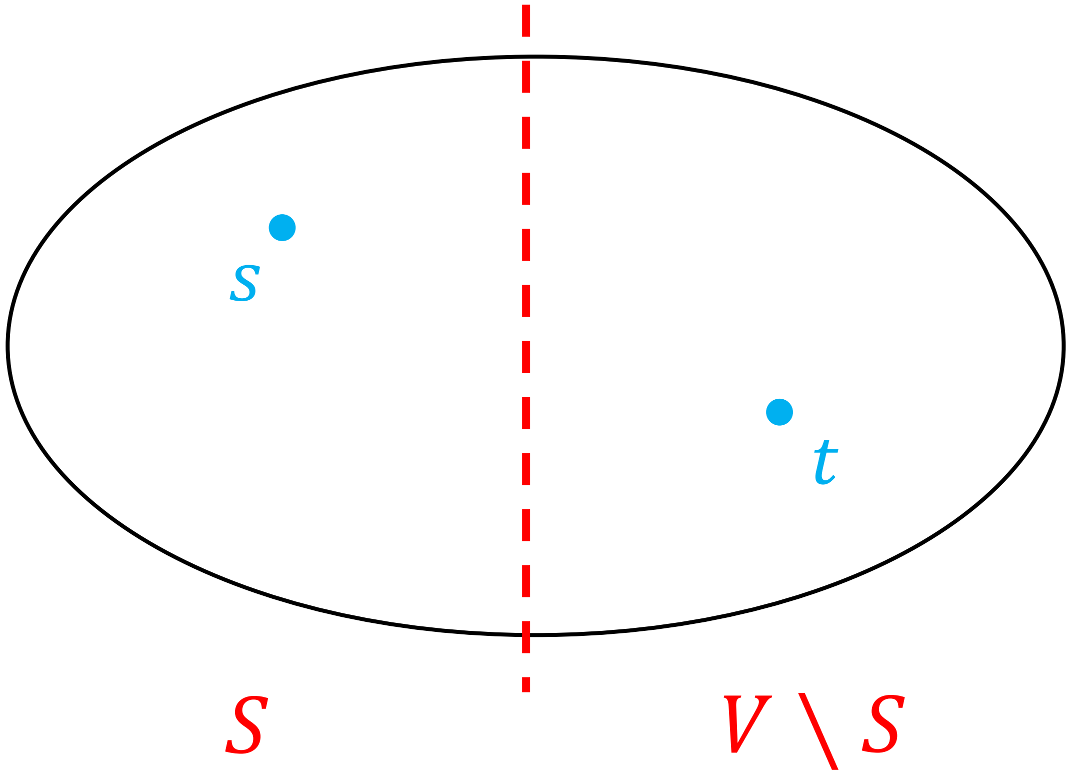
Gomory and Hu's Algorithm [GH'61]

- Pick arbitrary $s, t \in V$.



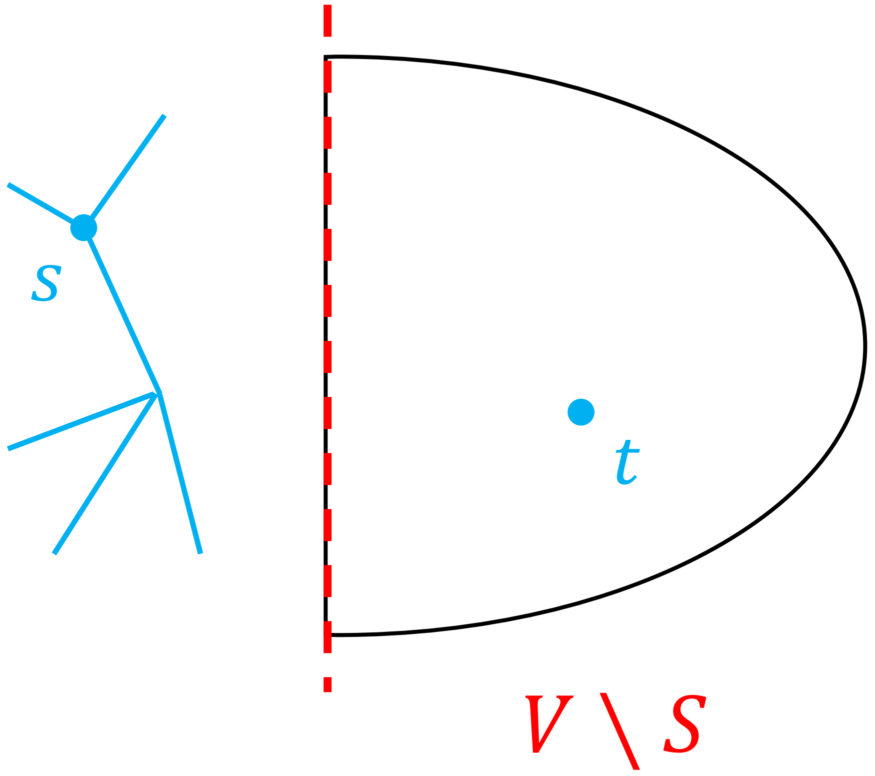
Gomory and Hu's Algorithm [GH'61]

- Pick arbitrary $s, t \in V$.
- Find (s, t)-mincut $(S, V \setminus S)$.



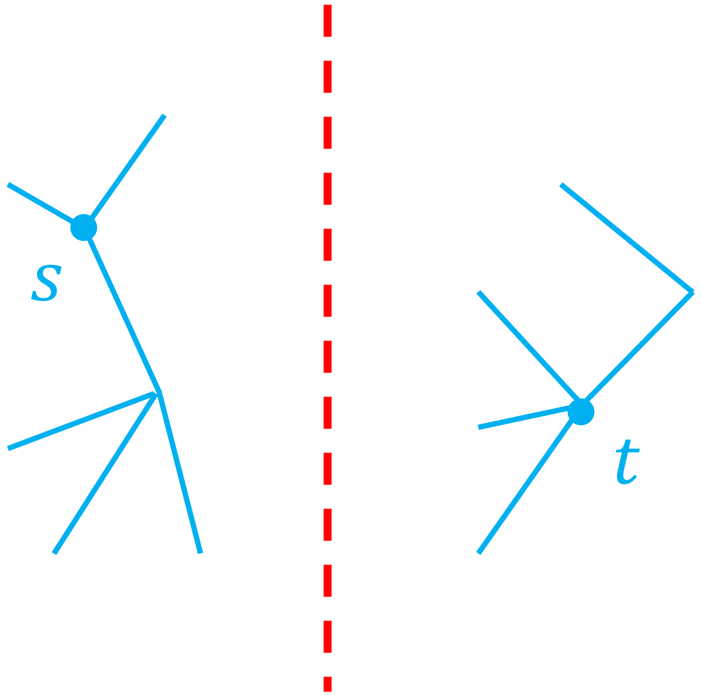
Gomory and Hu's Algorithm [GH'61]

- Pick arbitrary $s, t \in V$.
- Find (s, t)-mincut $(S, V \setminus S)$.
- Recurse on each side.



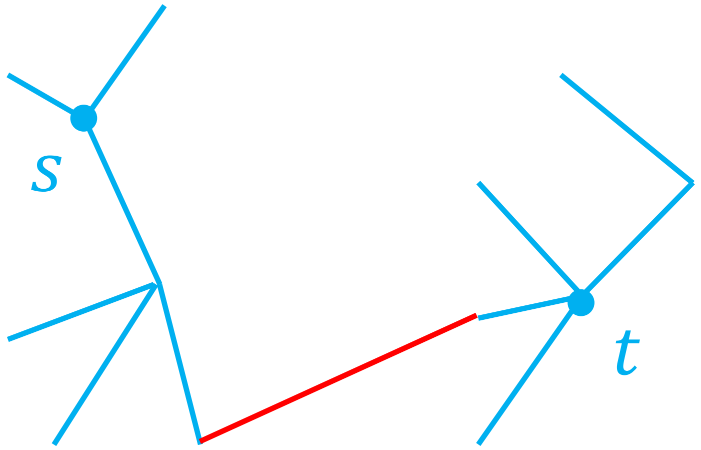
Gomory and Hu's Algorithm [GH'61]

- Pick arbitrary $s, t \in V$.
- Find (s, t)-mincut $(S, V \setminus S)$.
- Recurse on each side.



Gomory and Hu's Algorithm [GH'61]

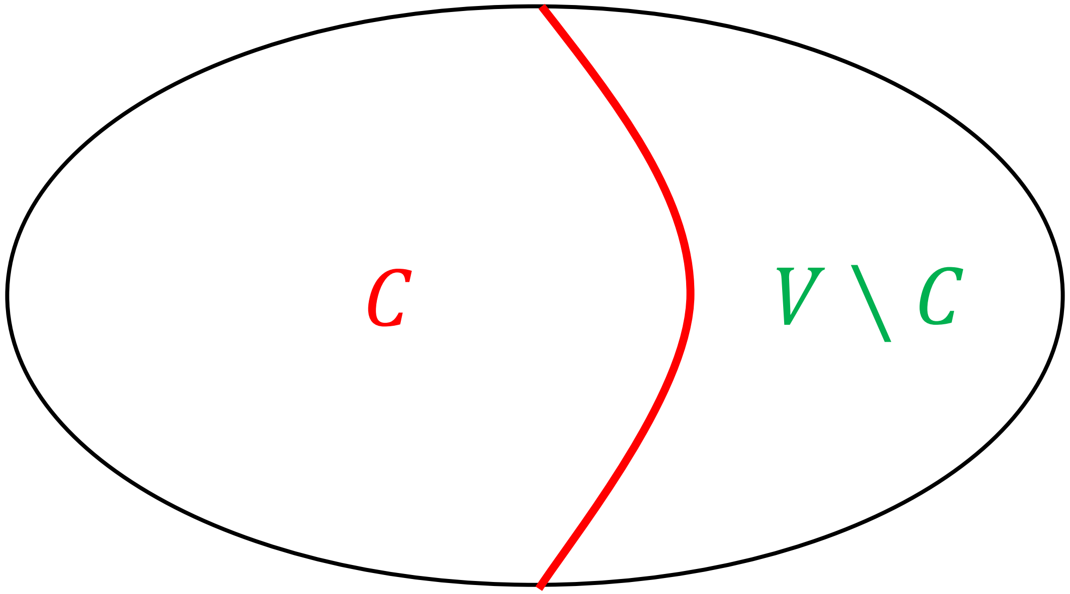
- Pick arbitrary $s, t \in V$.
- Find (s, t)-mincut $(S, V \setminus S)$.
- Recurse on each side.
- Merge two parts together.



- Each level: max-flow takes $\Omega(m)$ time.
- #levels can be $\Omega(n)$.
- Total time: $\Omega(nm)$. Too slow!

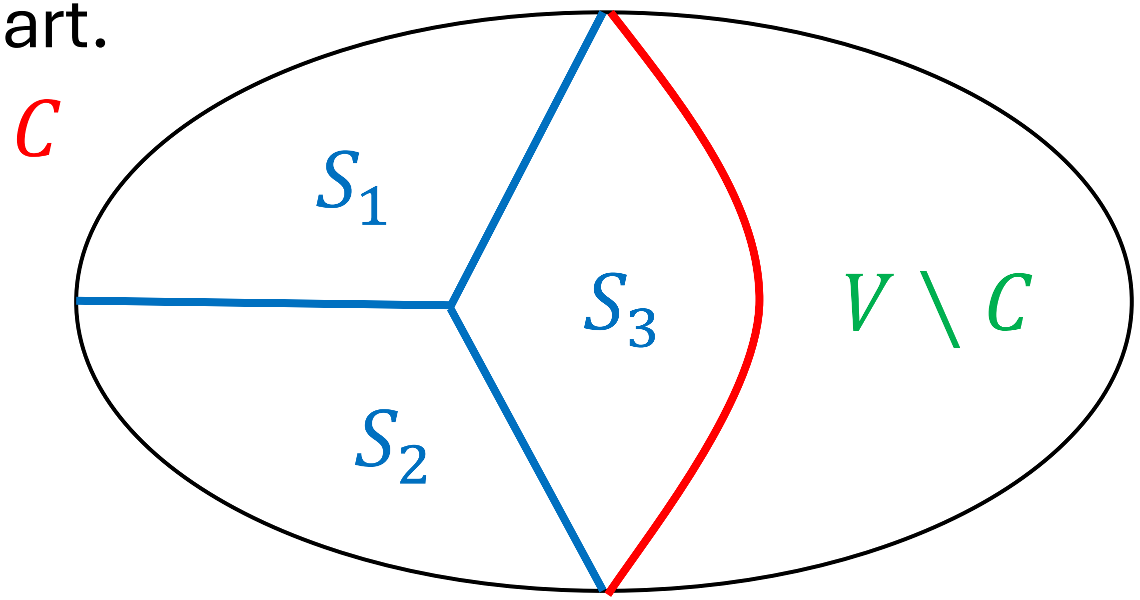
Core Idea: Find Balanced Decomposition

- Find threshold τ s.t.:
- Largest τ-connected component C contains $\geq |V|/2$ vertices, and



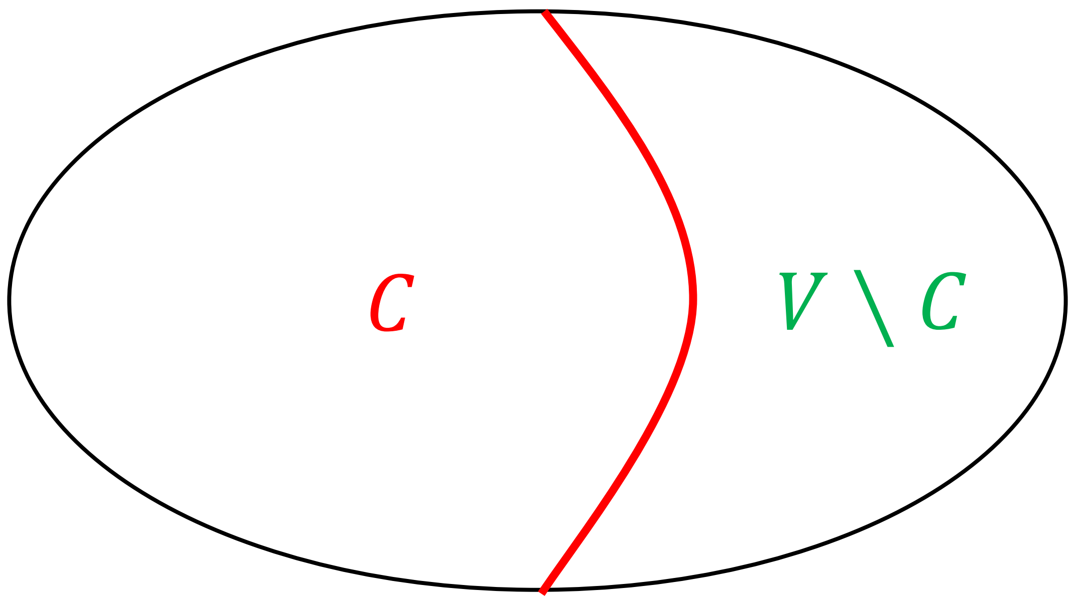
Core Idea: Find Balanced Decomposition

- Find threshold τ s.t.: $\Rightarrow |V \setminus C| \leq |V|/2$
- Largest τ-connected component C contains $\geq |V|/2$ vertices, and
- Every $(\tau + 1)$-connected component S_i contains $< |V|/2$ vertices.
- Computing GH-tree on each part.
- Merge them together.



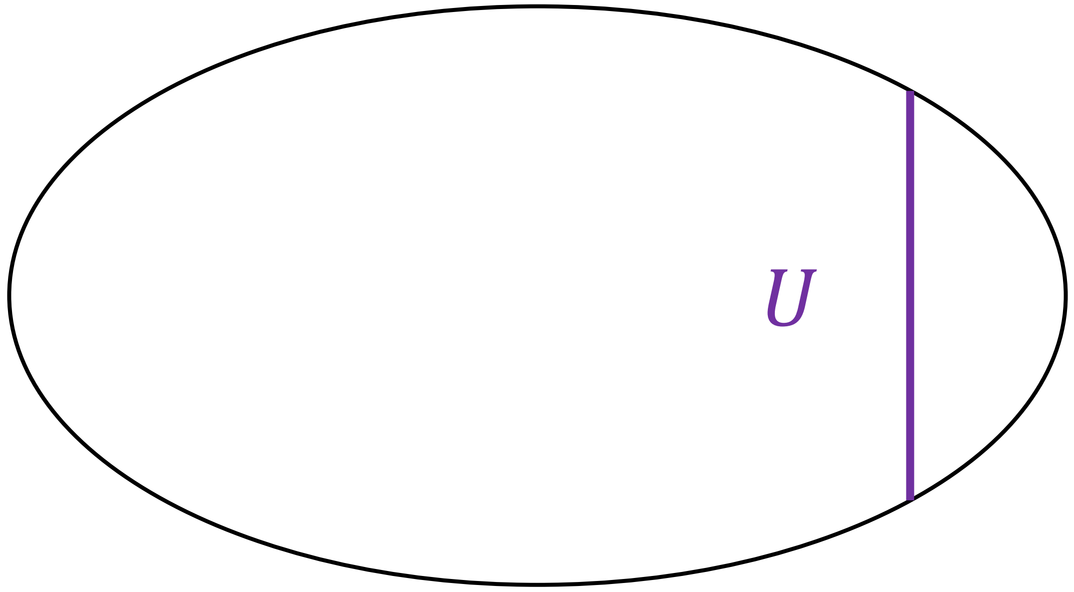
Find τ and Largest τ-Connected Component C

- Do binary search on τ.
- Assume largest τ-connected component C contains $\geq |V|/2$ vertices.



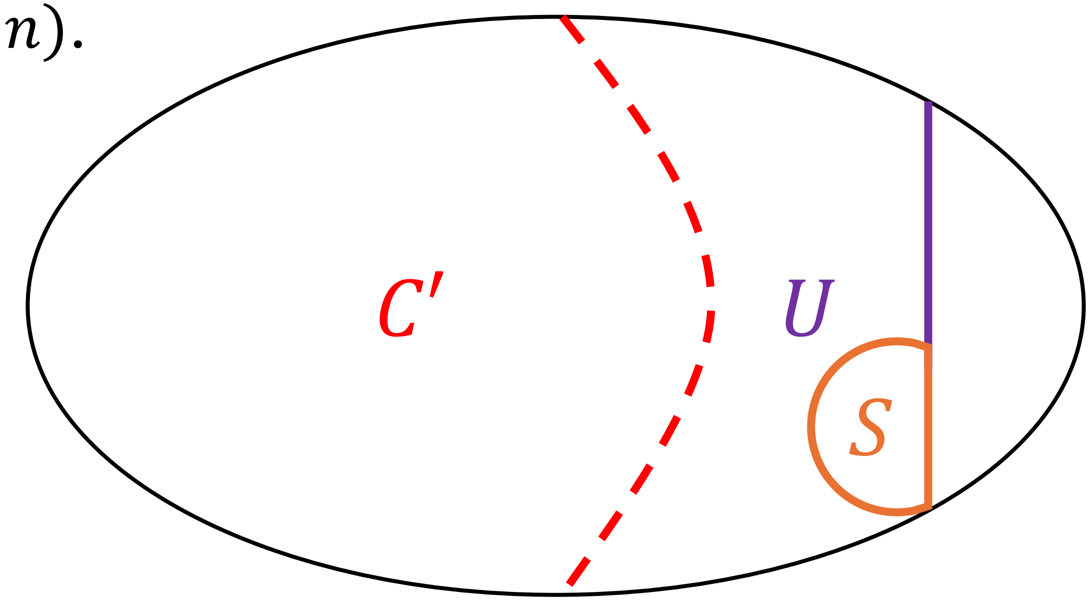
Balanced Partition Lemma

- Lemma: given τ' and terminals $U \subseteq V$,



Balanced Partition Lemma

- Lemma: given τ' and terminals $U \subseteq V$, there is algorithm finds collection $\mathcal{S}$ of disjoint vertex sets s.t.
  - Each vertex set $S \in \mathcal{S}$ satisfies $|E(S, V \setminus S)| < \tau'$ and $|S \cap U| \leq |U|/2$,
  - Let C' be largest τ'-connected component in G w.r.t. U, we have $\mathbb{E}(|\bigcup_{S \in \mathcal{S}} S \cap U|) \geq \Omega(|U \setminus C'|/\log n)$.
- Via isolating mincut [LP'20].

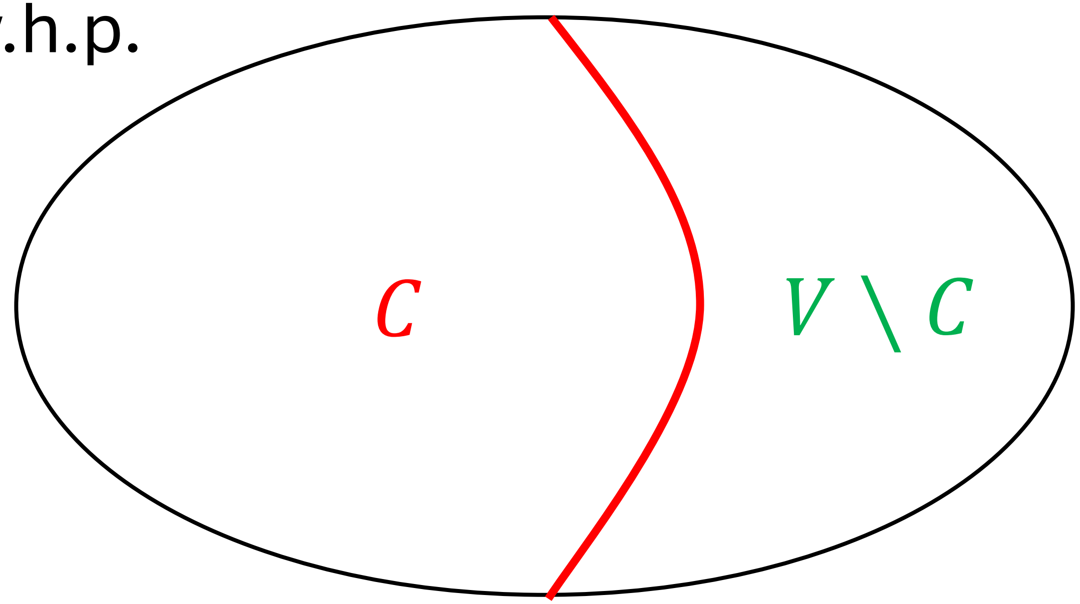


Find τ and C (Cont.)

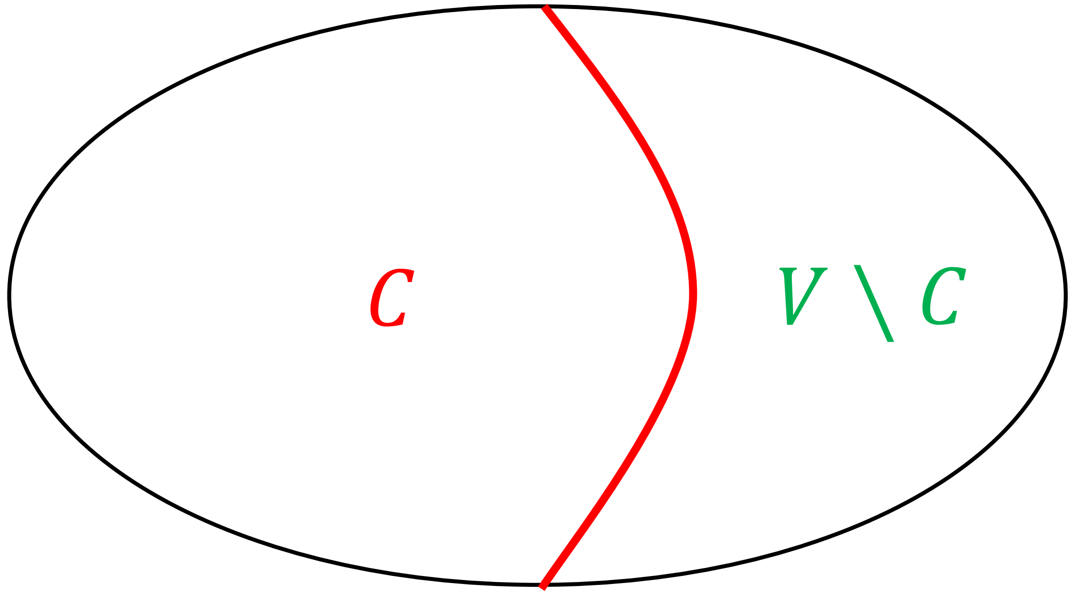
Balanced partition lemma: given τ' and terminals $U \subseteq V$, there is algorithm finds collection $\mathcal{S}$ of disjoint vertex sets s.t.
- Each $S \in \mathcal{S}$ satisfies $|E(S, V \setminus S)| < \tau'$ and $|S \cap U| \leq |U|/2$,
- Let C' be largest τ'-connected component in G w.r.t. U, we have $\mathbb{E}(|\bigcup_{S \in \mathcal{S}} S \cap U|) \geq \Omega(|U \setminus C'|/\log n)$.

Set $\tau' = \tau$ and $U = V$.

- Do binary search on τ.
- Assume largest τ-connected component C contains $\geq |V|/2$ vertices.
- In each round, remove $\Omega(1/\log n)$ fraction of vertices from $U \setminus C$.
- After $\mathcal{O}(\log^2 n)$ rounds, $U = C$ w.h.p.

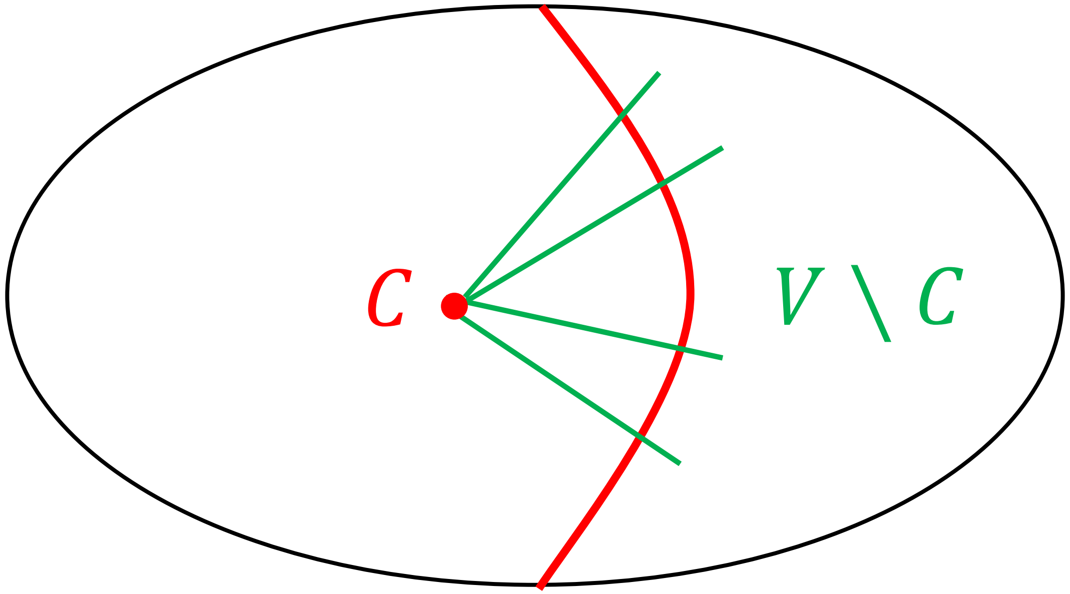


Gomory-Hu Tree on $V \setminus C$



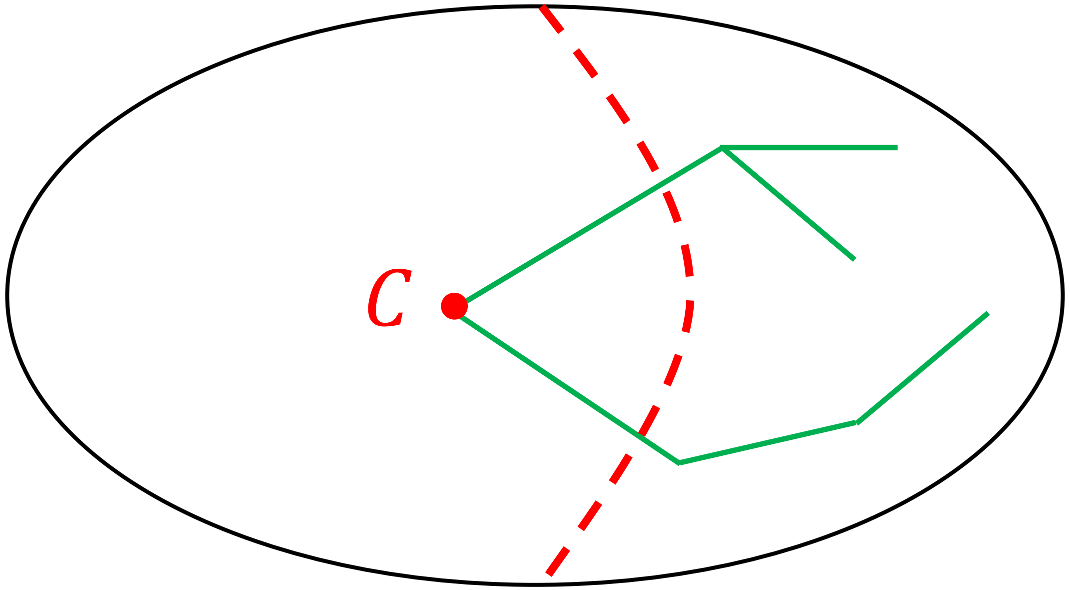
Gomory-Hu Tree on $V \setminus C$

- Contract C into one vertex.



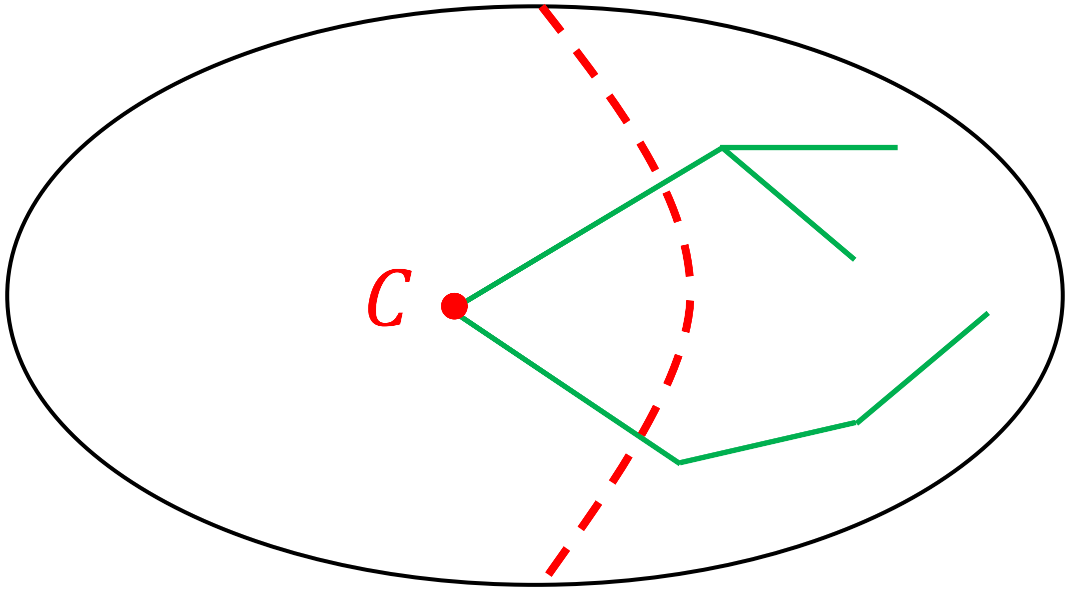
Gomory-Hu Tree on $V \setminus C$

- Contract C into one vertex. $\Rightarrow |V(G/C)| \leq |V|/2 + 1$
- Recursively compute GH-Tree on G/C. $\Rightarrow T_{small}$



Find S_i

$(\tau + 1)$-connected component

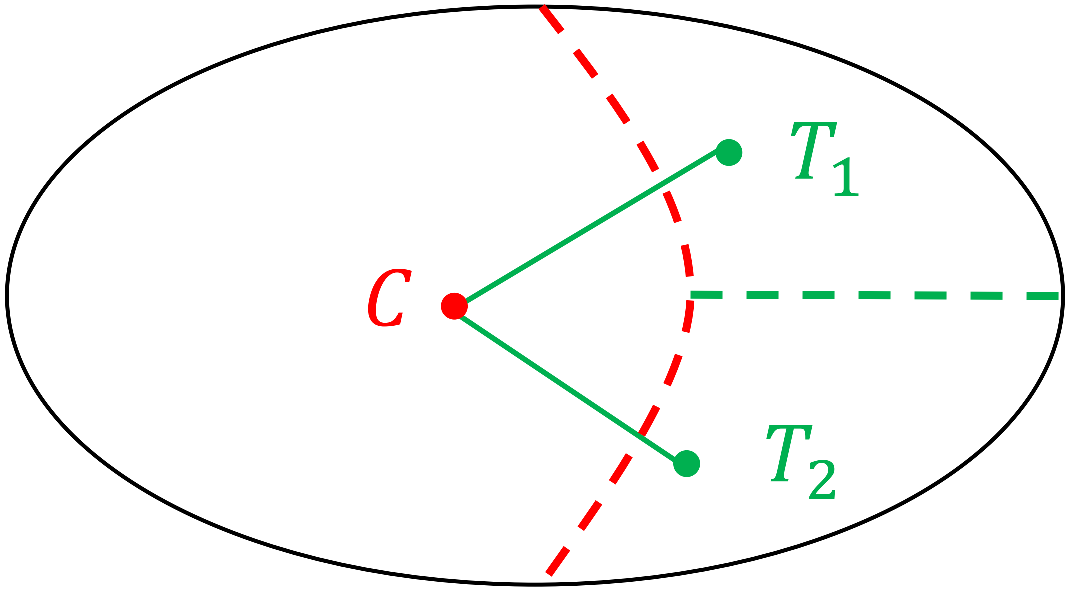


Find S_i

$(\tau + 1)$-connected component

Balanced partition lemma: given τ' and terminals $U \subseteq V$, there is algorithm finds collection $\mathcal{S}$ of disjoint vertex sets s.t.
- Each $S \in \mathcal{S}$ satisfies $|E(S, V \setminus S)| < \tau'$ and $|S \cap U| \leq |U|/2$,
- Let C' be largest τ'-connected component in G w.r.t. U, we have $\mathbb{E}(|\bigcup_{S \in \mathcal{S}} S \cap U|) \geq \Omega(|U \setminus C'|/\log n)$.

- Contract each "subtree" in T_{small}.
- Use balanced partition lemma with $\tau' = \tau + 1$ and $U = C$.

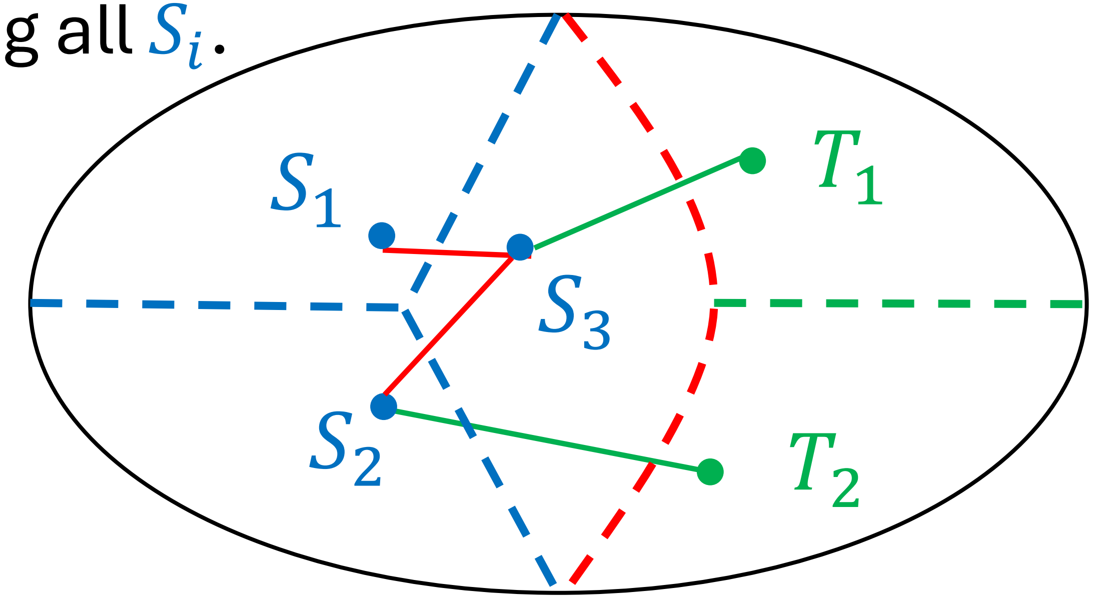


Find S_i

$(\tau + 1)$-connected component

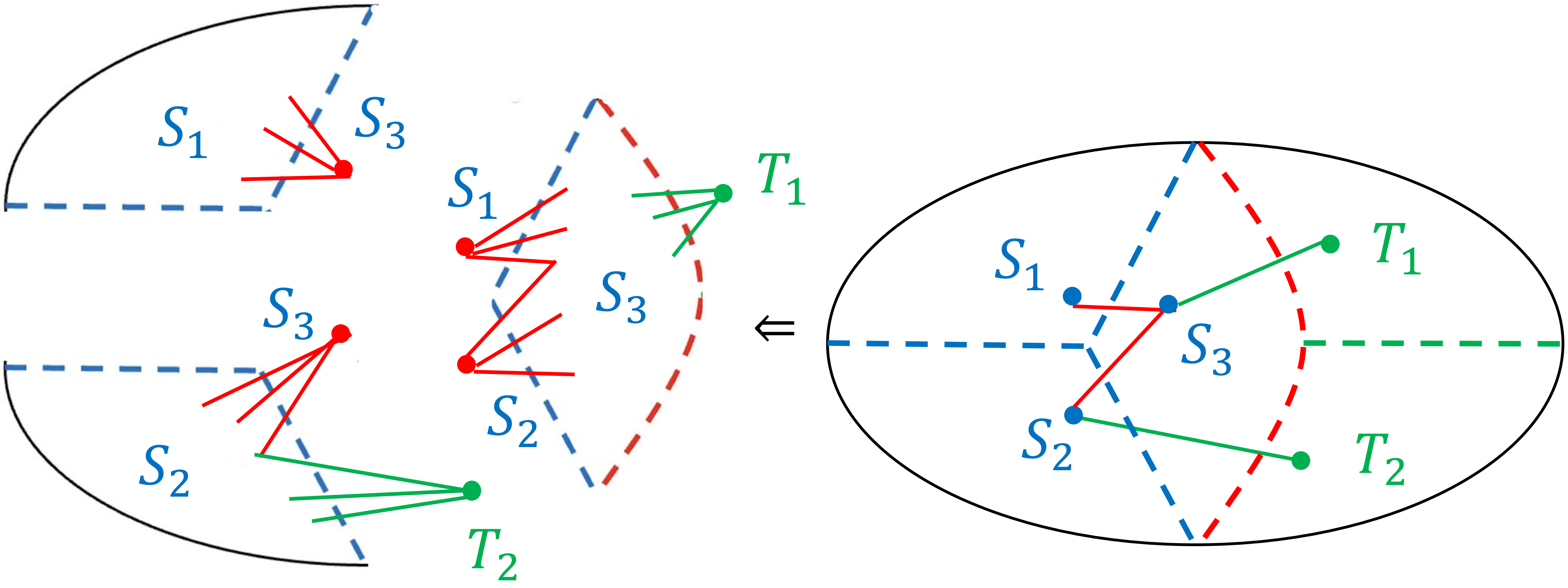
Balanced partition lemma: given τ' and terminals $U \subseteq V$, there is algorithm finds collection $\mathcal{S}$ of disjoint vertex sets s.t.
- Each $S \in \mathcal{S}$ satisfies $|E(S, V \setminus S)| < \tau'$ and $|S \cap U| \leq |U|/2$,
- Let C' be largest τ'-connected component in G w.r.t. U, we have $\mathbb{E}(|\bigcup_{S \in \mathcal{S}} S \cap U|) \geq \Omega(|U \setminus C'|/\log n)$.

- Contract each ”subtree” in T_{small}.
- Use balanced partition lemma with $\tau' = \tau + 1$ and $U = C$.
- Recurse on $U \setminus \bigcup_{S \in \mathcal{S}} S$ and each S, until reaching $\mathcal{O}(\log^2 n)$ levels.
- Also find ”GH-Tree” T_{large} linking all S_i.



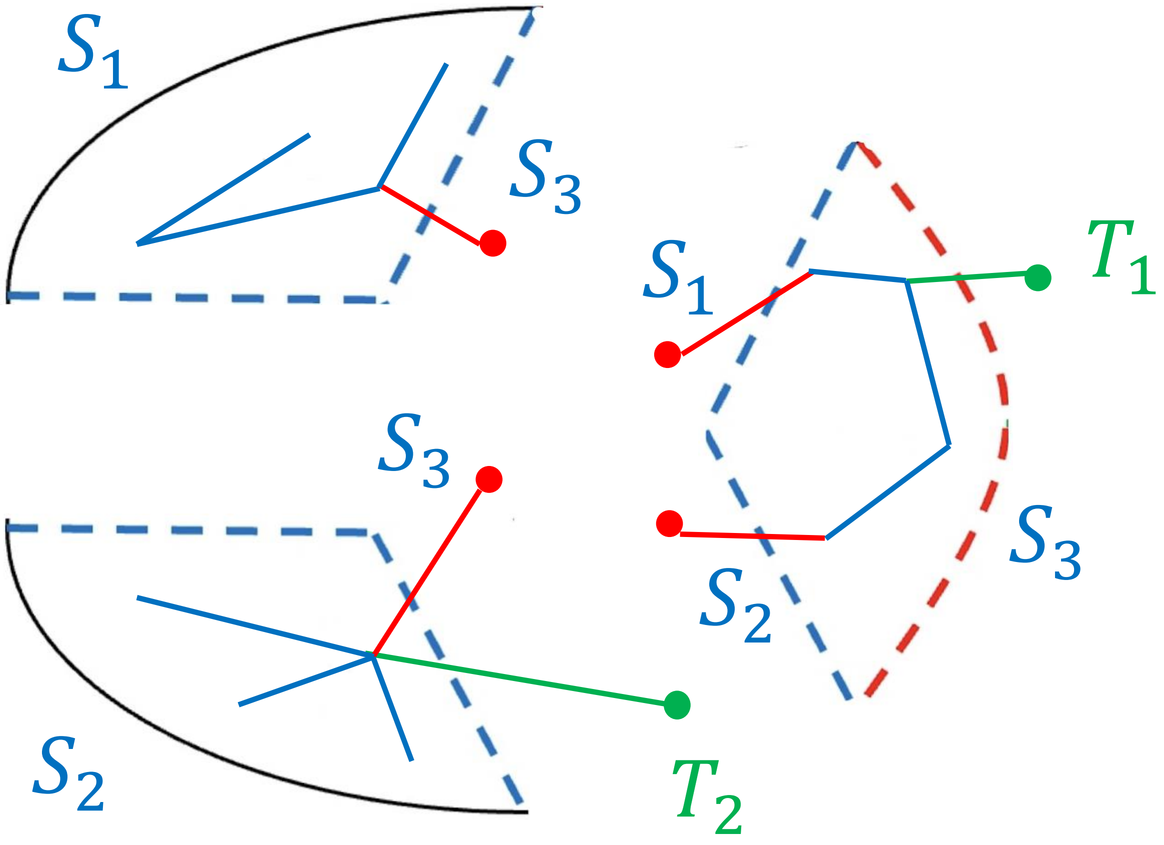
Gomory-Hu Tree on S_i

- For each S_i , contract every "subtree" in T_{large}.



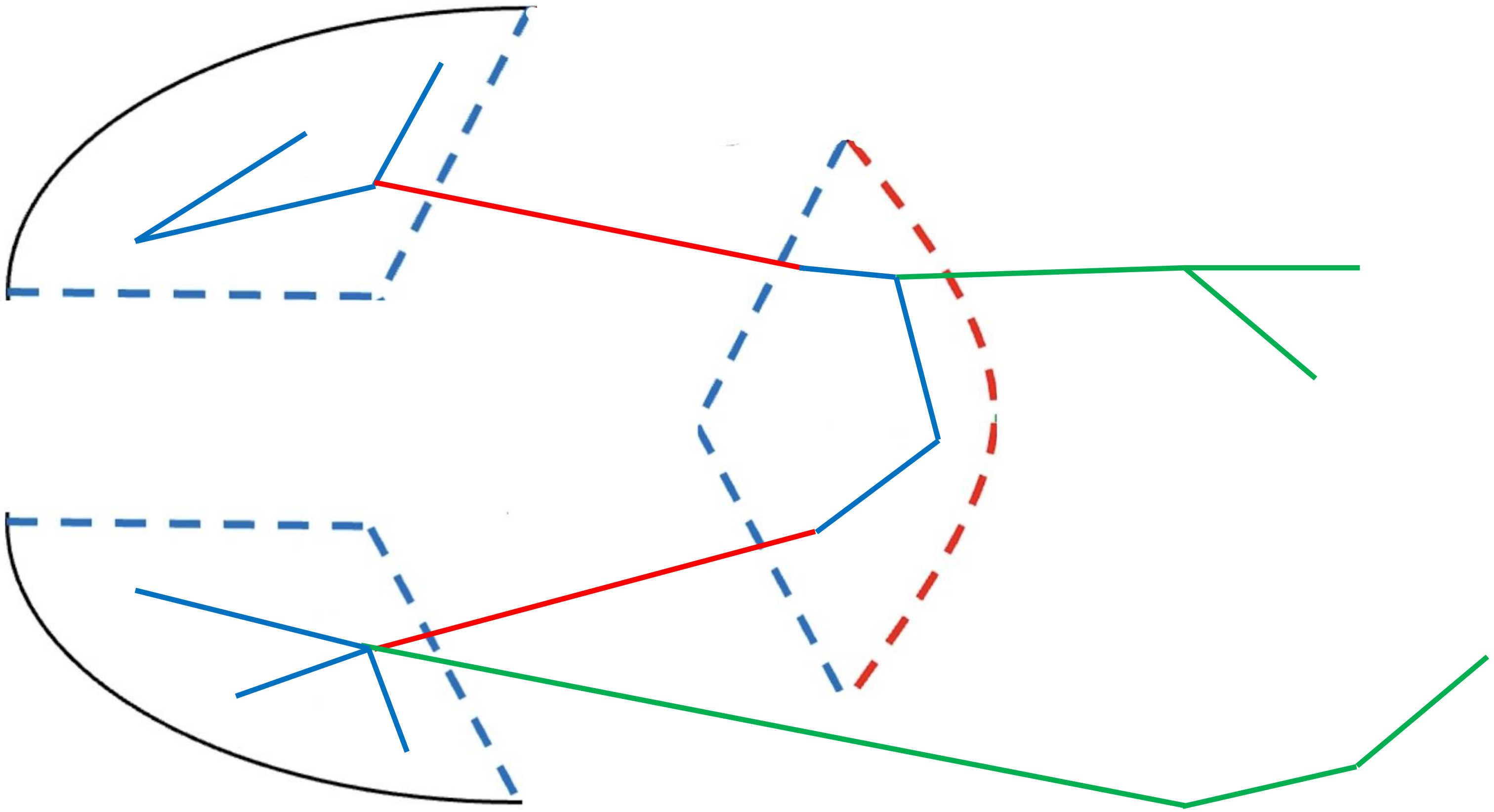
Gomory-Hu Tree on S_i

- For each S_i , contract every “subtree” in T_{large}.

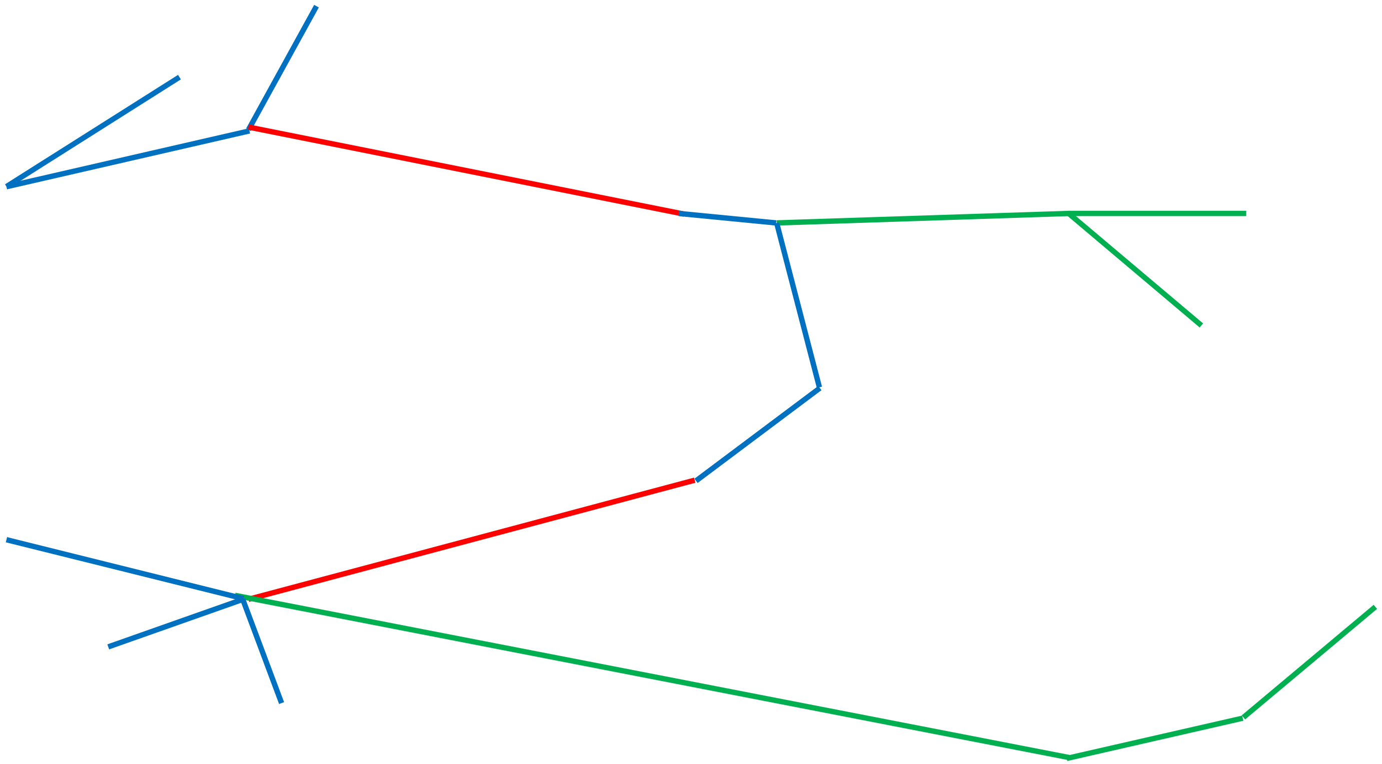


Recurse!

Merge All Together



Merge All Together



Summary

- (Randomized) reduction from Gomory-Hu tree to $\mathcal{O}(\log^6 n)$ maxflows on unweighted graphs.
- Simple algorithm. Also works for hypergraphs.
- See also: [P**Y**'25] deter. reduction to maxflows and expander decompositions with total size $\tilde{\mathcal{O}}(m)$ on unweighted graphs.
- Open problems:
  - Weighted graphs?
  - Element connectivity?
  - Maxflow in $\tilde{\mathcal{O}}(m)$ time?

Thanks!