Faster Exact and Approximate k-cut

Jason Li

Joint with Anupam Gupta (CMU), Euiwoong Lee (NYU)

FOCS 2018

10/7/2018

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction:	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
• Faster exact?	

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction:	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$

Fine-grained complexity?

- Faster exact?

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction:	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$

Fine-grained complexity?

- Faster exact?

Approx

- [SV'95] Greedy 2-apx
- [Man'17] $(2-\varepsilon)$-apx is NP-hard, assuming Small Set Expansion
- [GLL'18] 1.9997-apx in FPT time $f(k)\,\mathrm{poly}(n)$
- Better apx?

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
• Faster exact?	
• This work: exact	$O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Approx

- [SV'95] Greedy 2-apx
- [Man'17] $(2-\varepsilon)$-apx is NP-hard, assuming Small Set Expansion
- [GLL'18] 1.9997-apx in FPT time $f(k)\,\mathrm{poly}(n)$
- Better apx?

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
• Faster exact?	
• This work: exact	$O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Approx

- [SV'95] Greedy 2-apx
- [Man'17] $(2-\varepsilon)$-apx is NP-hard, assuming Small Set Expansion
- [GLL'18] 1.9997-apx in FPT time $f(k)\,\mathrm{poly}(n)$
- Better apx?
- This work: $(1+\varepsilon)$-apx in $f(k,\varepsilon)\,n^{k+o(1)}$

Min k-cut Problem

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time	Approx
• Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$	• [SV'95] Greedy 2-apx
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$	• [Man'17] $(2-\varepsilon)$-apx is NP-hard, assuming Small Set Expansion
• Faster exact?		• [GLL'18] 1.9997-apx in FPT time $f(k)\,\mathrm{poly}(n)$
• This work: exact	$O(n^{(2\omega/3+o(1))k})$ $\approx n^{1.6k}$	• Better apx?
		• This work: $(1+\varepsilon)$-apx in $f(k,\varepsilon)\,n^{k+o(1)}$
		• This work: 1.81-apx in FPT time

Techniques

Techniques

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
• Faster exact?	
• This work: exact	$O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Techniques

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
Faster exact?	
This work: exact	$O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Idea: connect k-cut with k-clique

Techniques

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
• Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
• Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
• Faster exact?	
• This work: exact	$O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Idea: connect k-cut with k-clique

- $n^{(\omega/3)k}$ – time k-clique algo using matrix multiplication

Techniques

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact	Time
Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
Harder than k-clique:	$\Omega(n^{(\omega/3)k})$
Faster exact?	
This work: exact	$O(n^{(2\omega/3+o(1))k})$ $\approx n^{1.6k}$

Idea: connect k-cut with k-clique

- $n^{(\omega/3)k}$-time k-clique algo using matrix multiplication
- This talk: $n^{(1+\omega/3)k}$ algo $\approx n^{1.8k}$

Techniques

- Given graph G, delete min weight edges to cut graph into $\geq k$ components

Exact

Exact	Time
Karger's random edge contraction: Fine-grained complexity?	$O(n^{2k})$
Harder than k-clique:	$\Omega(n^{(\omega/3)k})$

- Faster exact?
- This work: exact $O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Idea: connect k-cut with k-clique

- $n^{(\omega/3)k}$-time k-clique algo using matrix multiplication
- This talk: $n^{(1+\omega/3)k}$ algo $\approx n^{1.8k}$
  - Idea 1: Thorup's tree packing
  - Idea 2: Reduction to $(k-1)$-respecting
  - Idea 3: k-clique-like mtx. mult.

Techniques

- Given graph G, delete min weight edges to cut $\geq k$ components

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Exact	Time
Karger's random edge contraction:	$O(n^{2k})$
Harder than k-clique:	$\Omega(n^{(\omega/3)k})$

Fine-grained complexity?

- Faster exact?
- This work: exact $O(n^{(2\omega/3+o(1))k}) \approx n^{1.6k}$

Idea: connect k-cut with k-clique

- $n^{(\omega/3)k}$-time k-clique algo using matrix multiplication
- This talk: $n^{(1+\omega/3)k}$ algo $\approx n^{1.8k}$
  - Idea 1: Thorup's tree packing
  - Idea 2: Reduction to (k-1)-respecting
  - Idea 3: k-clique-like mtx.mult.

Hardness from k-clique

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

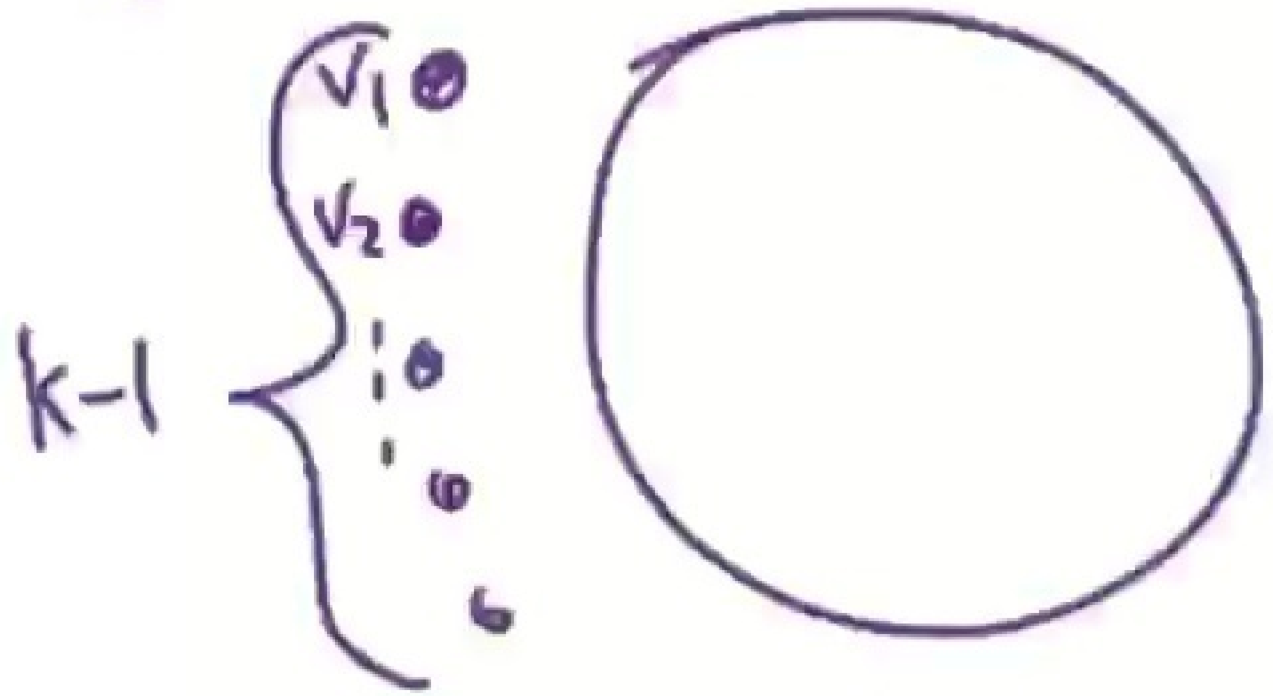
Reduction to (k-1)-resp.

k-clique-like mtx.mult.

→ Hardness $n^{(\omega/3)k}$

Hardness from k-clique

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph



Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
→ Hardness $n^{(\omega/3)k}$

Hardness from k-clique

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

→ Hardness $n^{(\omega/3)k}$

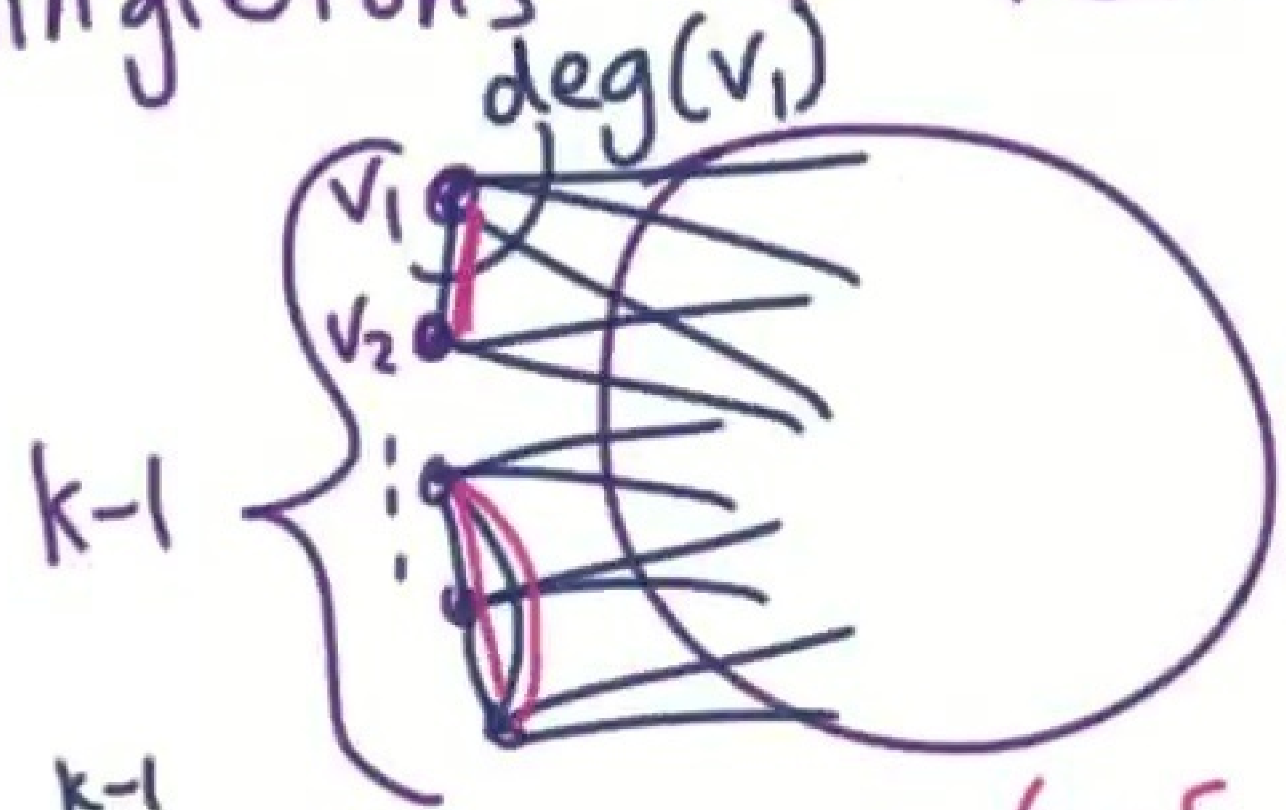
Idea: suppose optimal k-cut is (k-1) singletons + rest of graph

• $\text{k-cut} = \sum_{i=1}^{k-1} \deg(v_i)$

Hardness from k-clique

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph



- $k\text{-cut} = \sum_{i=1}^{k-1} \deg(v_i) - w(E[v_1, v_2, \ldots, v_{k-1}])$

Hardness from k-clique

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph

• $\text{k-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{w(E[v_1, v_2, \dots, v_{k-1}])}_{\text{is maximized}}$

• If G is regular, minimized when

• maximum when k-clique!

Hardness from k-clique

Exact $n^{(1+\omega/3)k}$

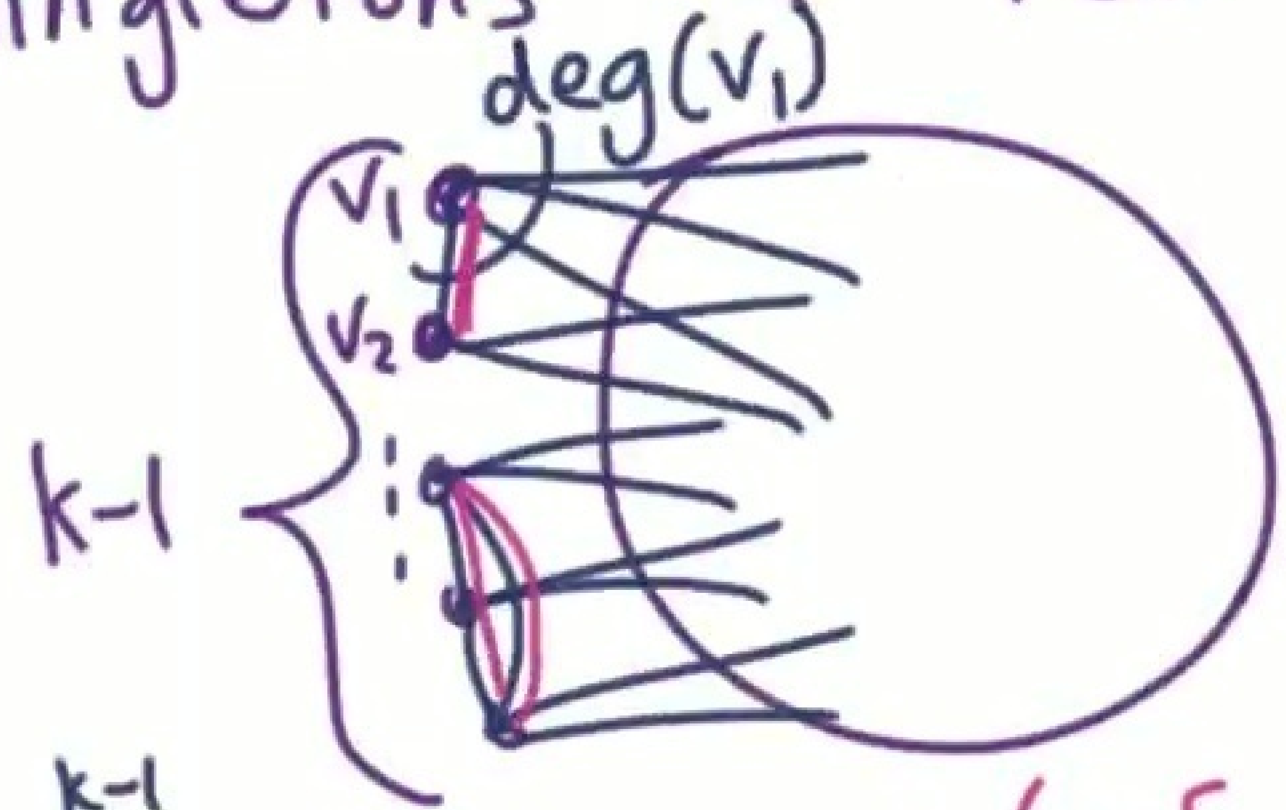
Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph



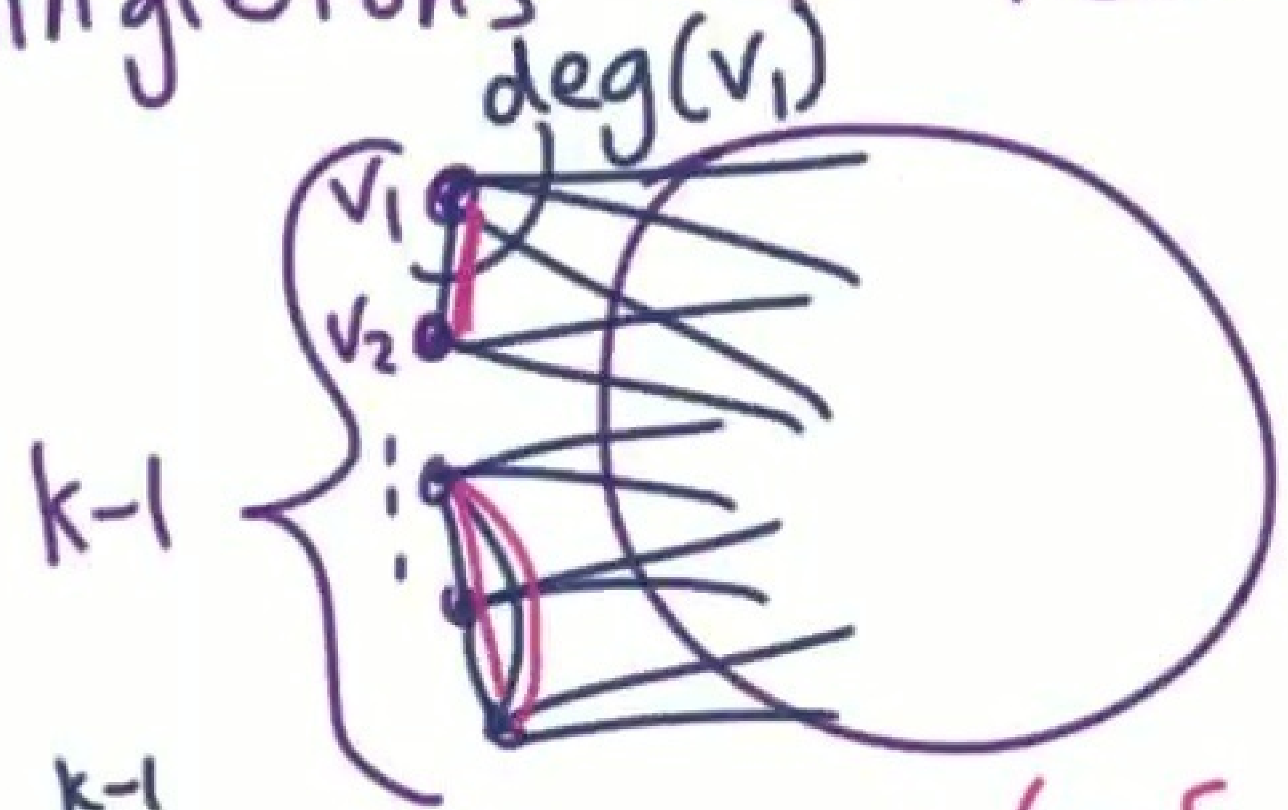
- $k\text{-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{w(E[v_1, v_2, \ldots, v_{k-1}])}_{\text{is maximized}}$
- If G is regular, minimized when is maximized
  - maximum when k-clique!
- Add large weights to rule out non-singleton cuts

Hardness from k-clique

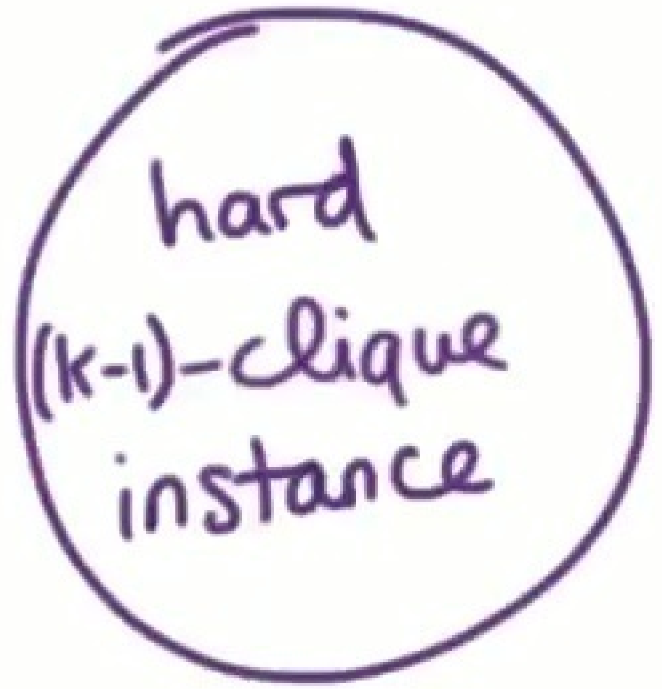
Exact $n^{(1+\omega/3)k}$
- Thorup's tree packing
- Reduction to (k-1)-resp.
- k-clique-like mtx.mult.

→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph



- $\text{k-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{w(E[v_1, v_2, \dots, v_{k-1}])}_{\text{is maximized}}$
- If G is regular, minimized when is maximized
  - maximum when k-clique!
- Add large weights to rule out non-singleton cuts



Hardness from k-clique

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph

• k-cut $= \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{w(E[v_1, v_2, \ldots, v_{k-1}])}_{\text{is maximized}}$

• If G is regular, minimized when is maximized

• maximum when k-clique!

• Add <u>large weights</u> to rule out non-singleton cuts

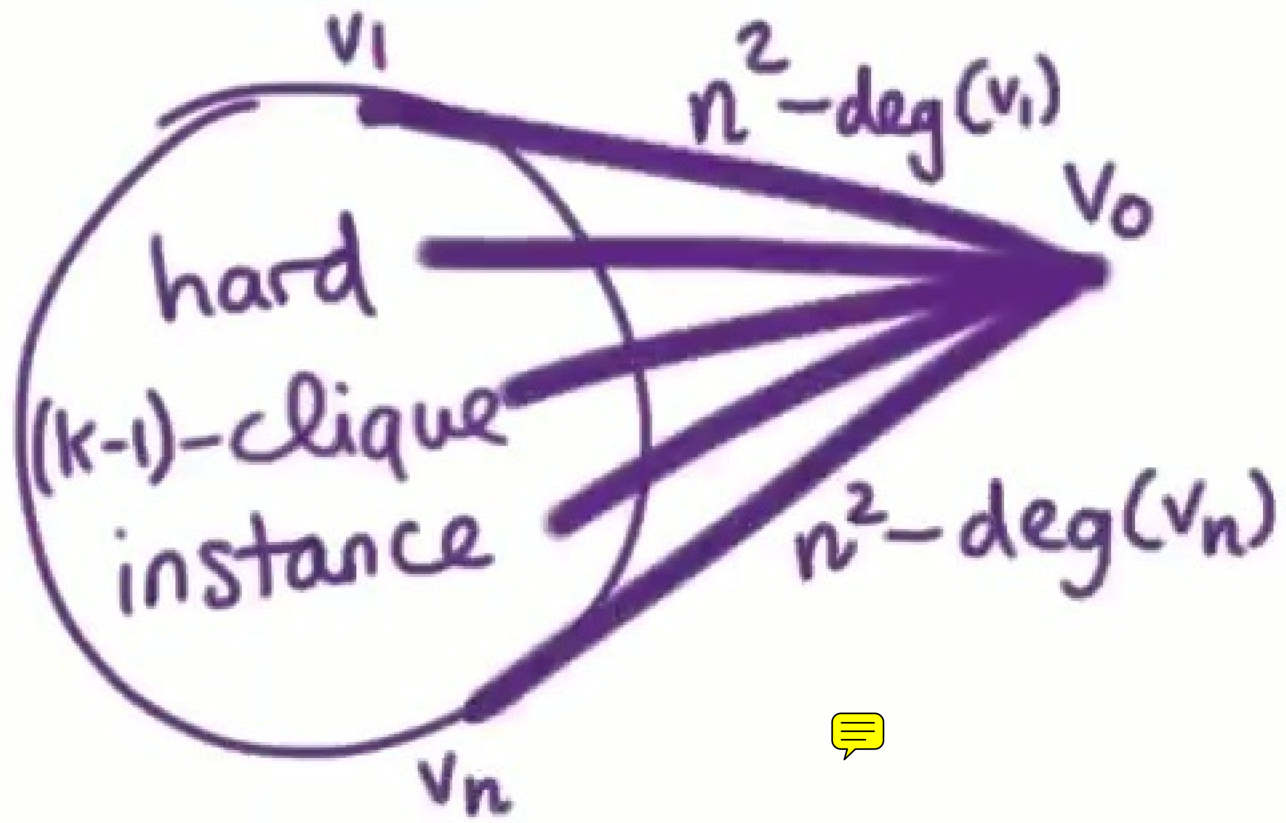
Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

→ Hardness $n^{(\omega/3)k}$

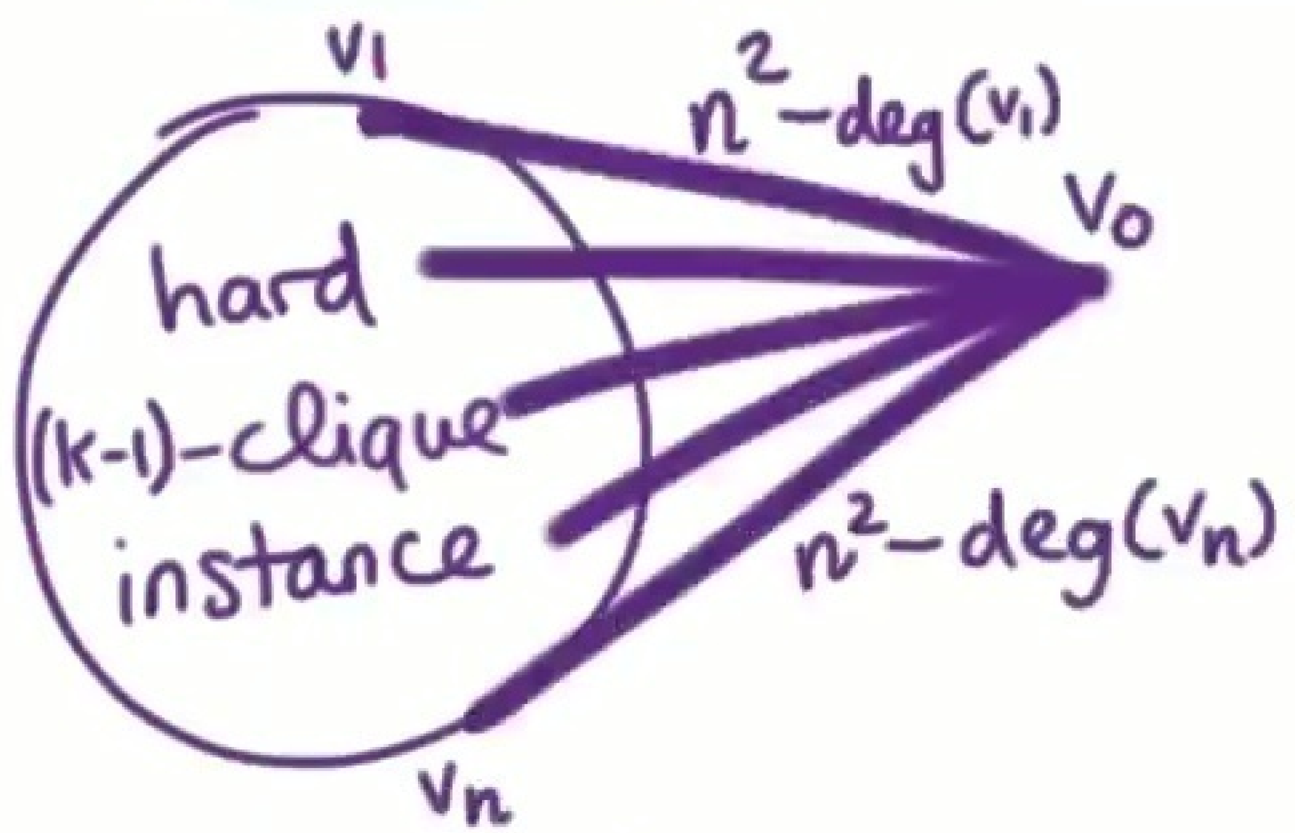


Hardness from k-clique

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph

- $k\text{-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{\omega(E[v_1, v_2, \ldots, v_{k-1}])}_{\text{is maximized}}$
- If G is regular, minimized when is maximized
  - maximum when k-clique!
- Add large weights to rule out non-singleton cuts

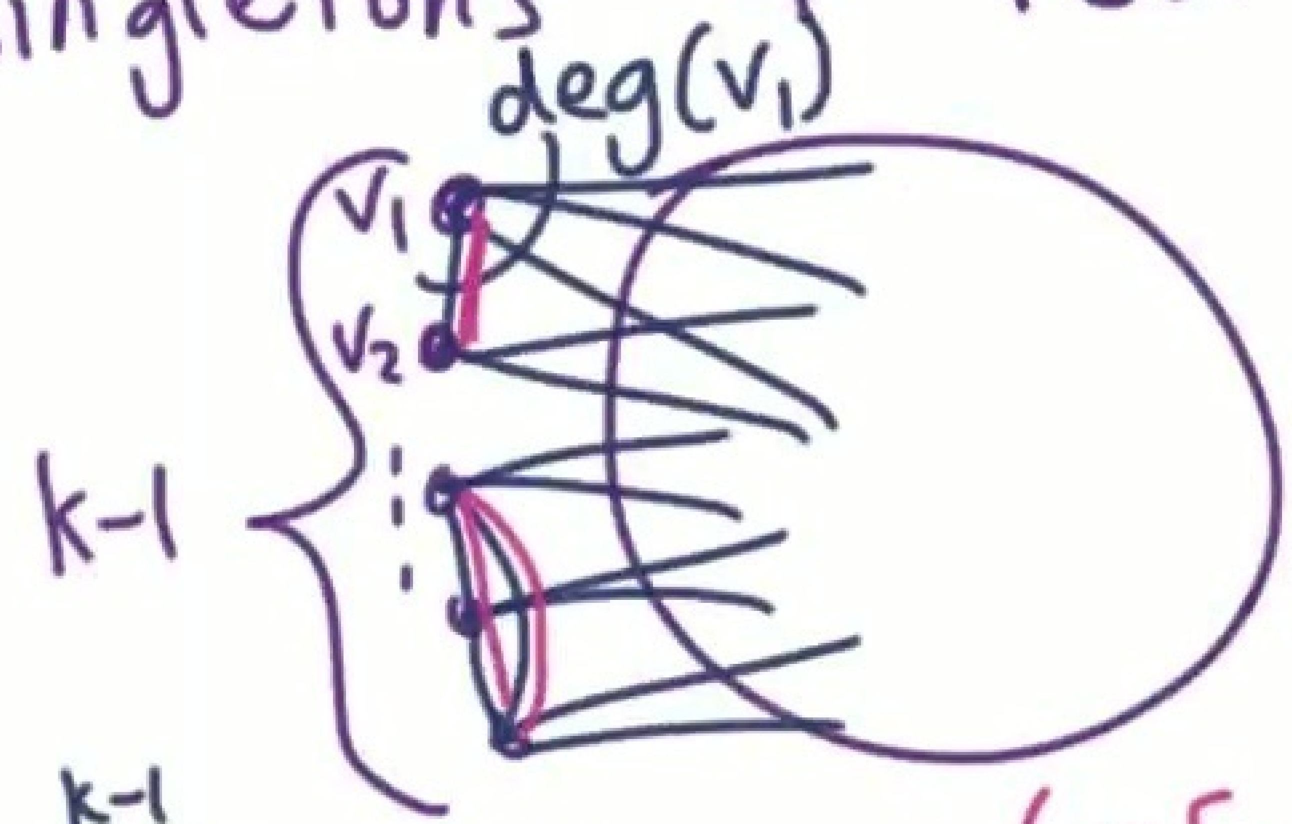


- $\deg(v) = n^2 \quad \forall v$
- min k-cut should only cut k-1 heavy edges ($(k-1)n^2$)

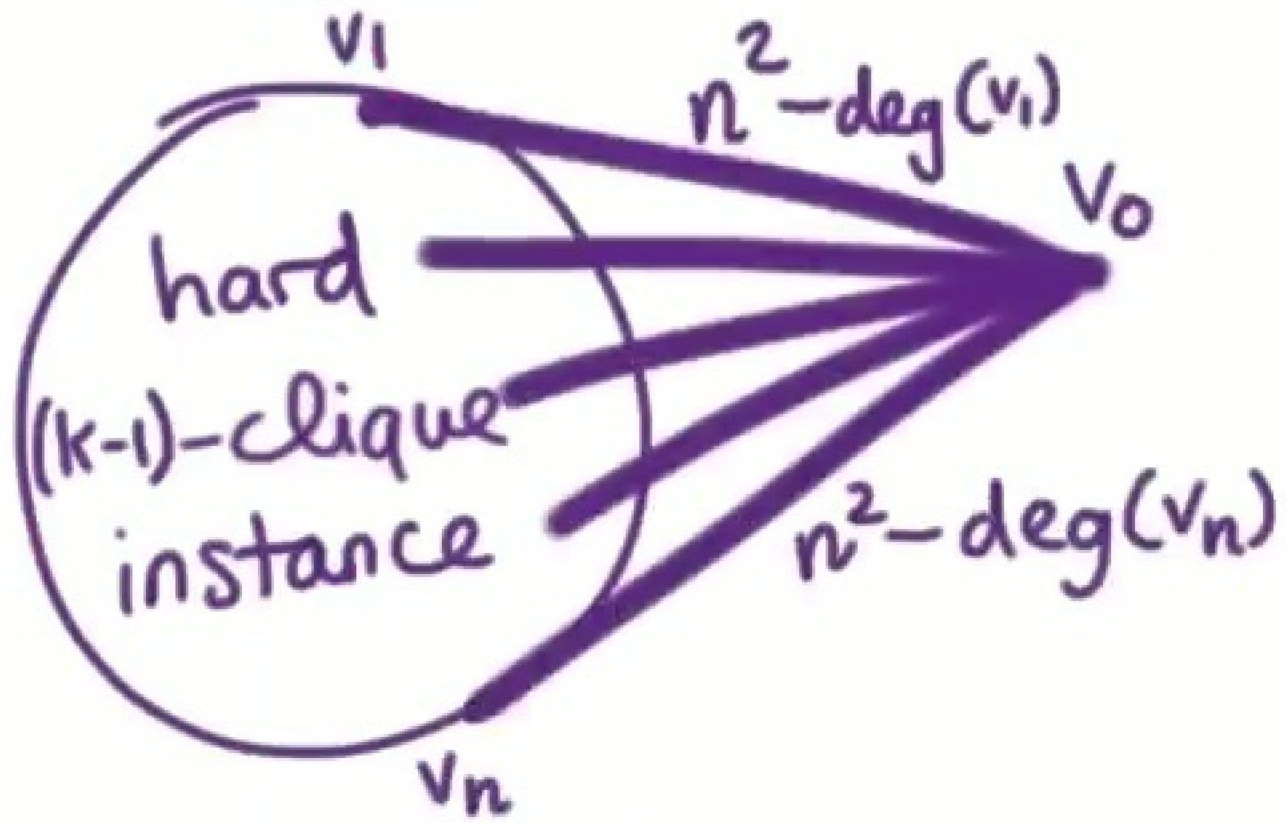
Hardness from k-clique

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx. mult.
→ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is (k-1) singletons + rest of graph



- $\text{k-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{\omega(E[v_1, v_2, \ldots, v_{k-1}])}_{\text{is maximized}}$
- If G is regular, minimized when is maximized
  - maximum when k-clique!
- Add <u>large weights</u> to rule out non-singleton cuts

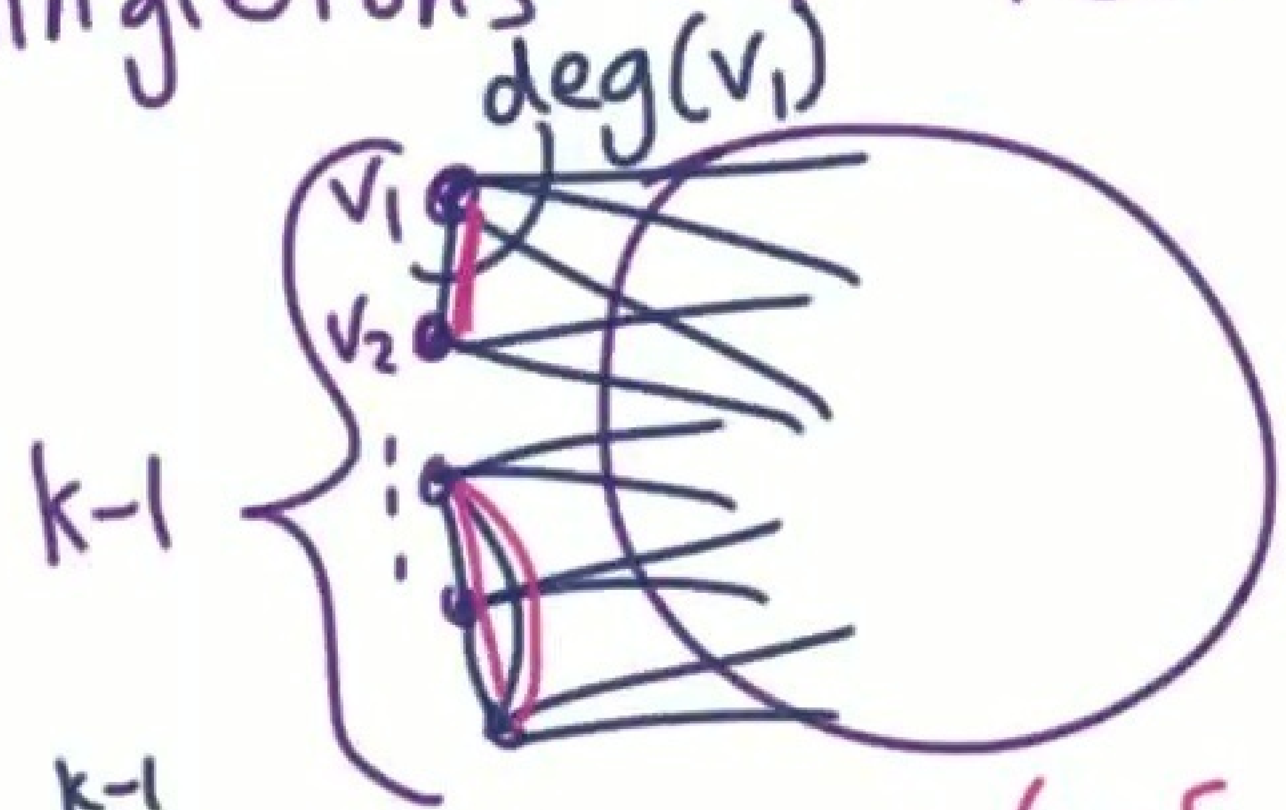


- $\deg(v) = n^2 \quad \forall v$
- min k-cut should only cut k-1 heavy edges $((k-1)n^2)$
  $\Rightarrow$ (k-1) singletons

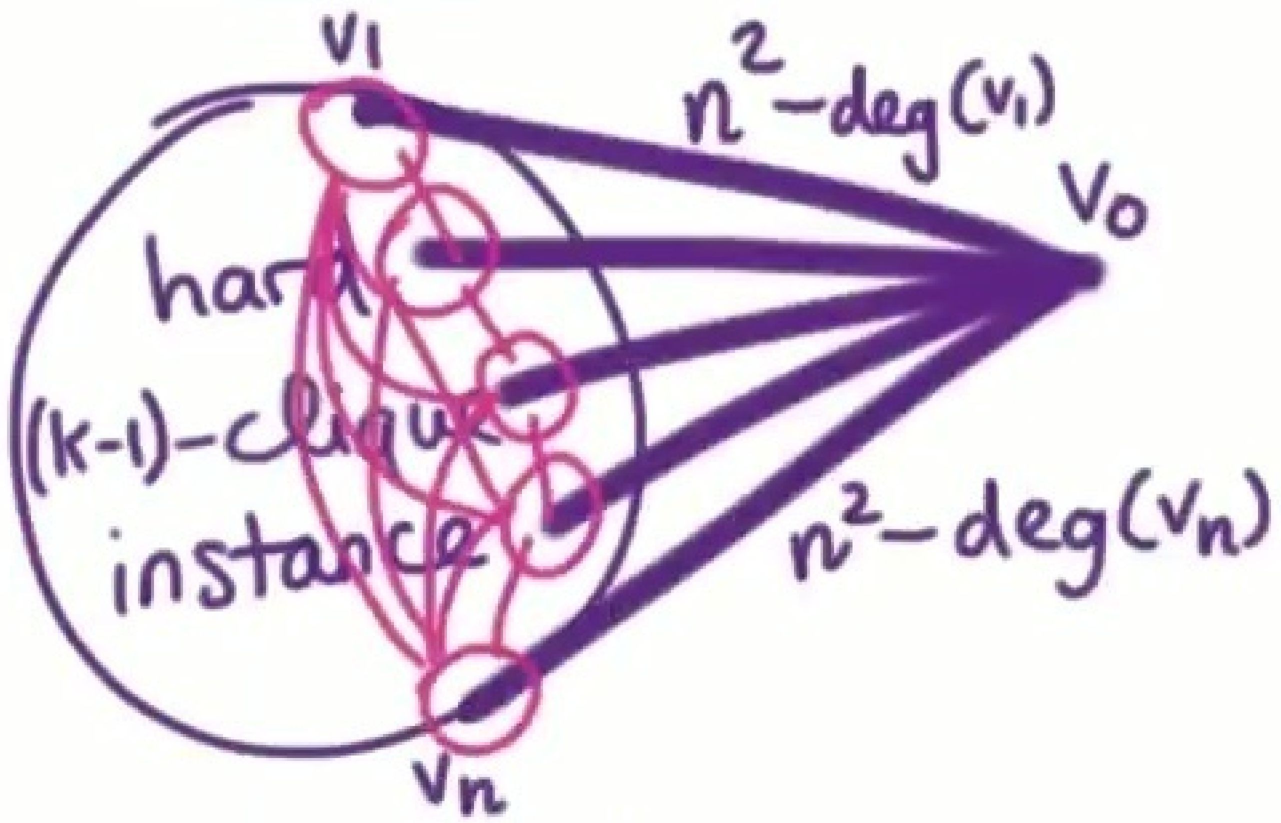
Hardness from k-clique

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
$\rightarrow$ Hardness $n^{(\omega/3)k}$

Idea: suppose optimal k-cut is $(k-1)$ singletons + rest of graph



- $k\text{-cut} = \sum_{i=1}^{k-1} \deg(v_i) - \underbrace{\omega(E[v_1, v_2, \dots, v_{k-1}])}_{\text{is maximized}}$
- If G is regular, minimized when is maximized
  - maximum when k-clique!
- Add <u>large weights</u> to rule out non-singleton cuts



- $\deg(v) = n^2 \ \forall v$
- min k-cut should only cut k-1 heavy edges $((k-1)n^2)$
  $\Longrightarrow (k-1)$ singletons
- $(k-1)$-clique $\Longleftrightarrow$ min k-cut is $(k-1)n^2 - \binom{k}{2}$

Thorup's Tree Packing

Exact $n^{(1+\omega/3)k}$

→ Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Thorup's Tree Packing

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

For any min k-cut, $\exists$ tree s.t. the k-cut cuts $\leq 2k-2$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \leq 2k-2$

Thorup's Tree Packing

- Thm [Thorup '08]: Can find poly(n) spanning trees of G with the following property:

  For any min k-cut, ∃ tree s.t. the k-cut cuts $\underbrace{\le 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \le 2k-2$

  k-cut $\le 2k-2$-respects the tree

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.

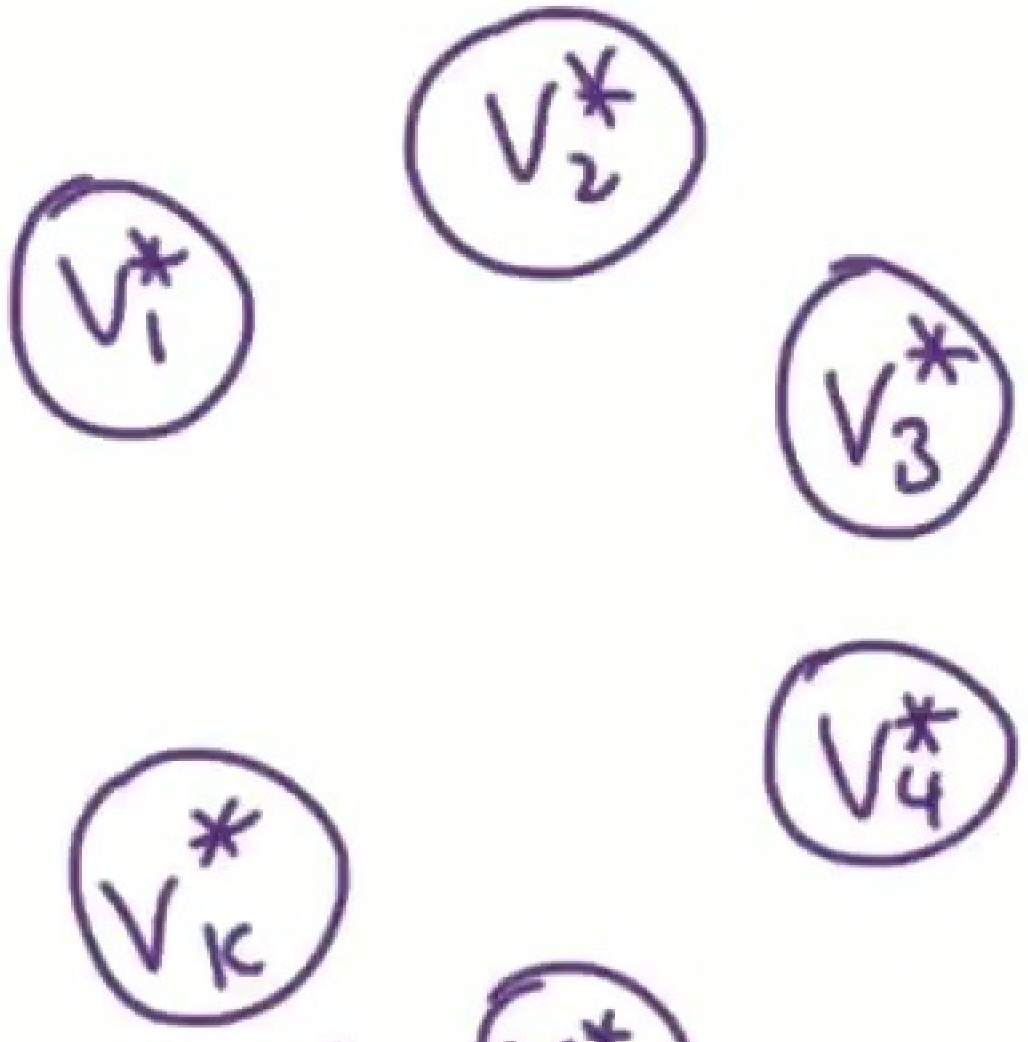
Hardness $n^{(\omega/3)k}$

Thorup's Tree Packing

- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

For any min k-cut, ∃tree s.t. the k-cut cuts $\underbrace{\leq 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \leq 2k-2$

k-cut $\leq 2k-2$-<u>respects</u> the tree



(k=6)

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.

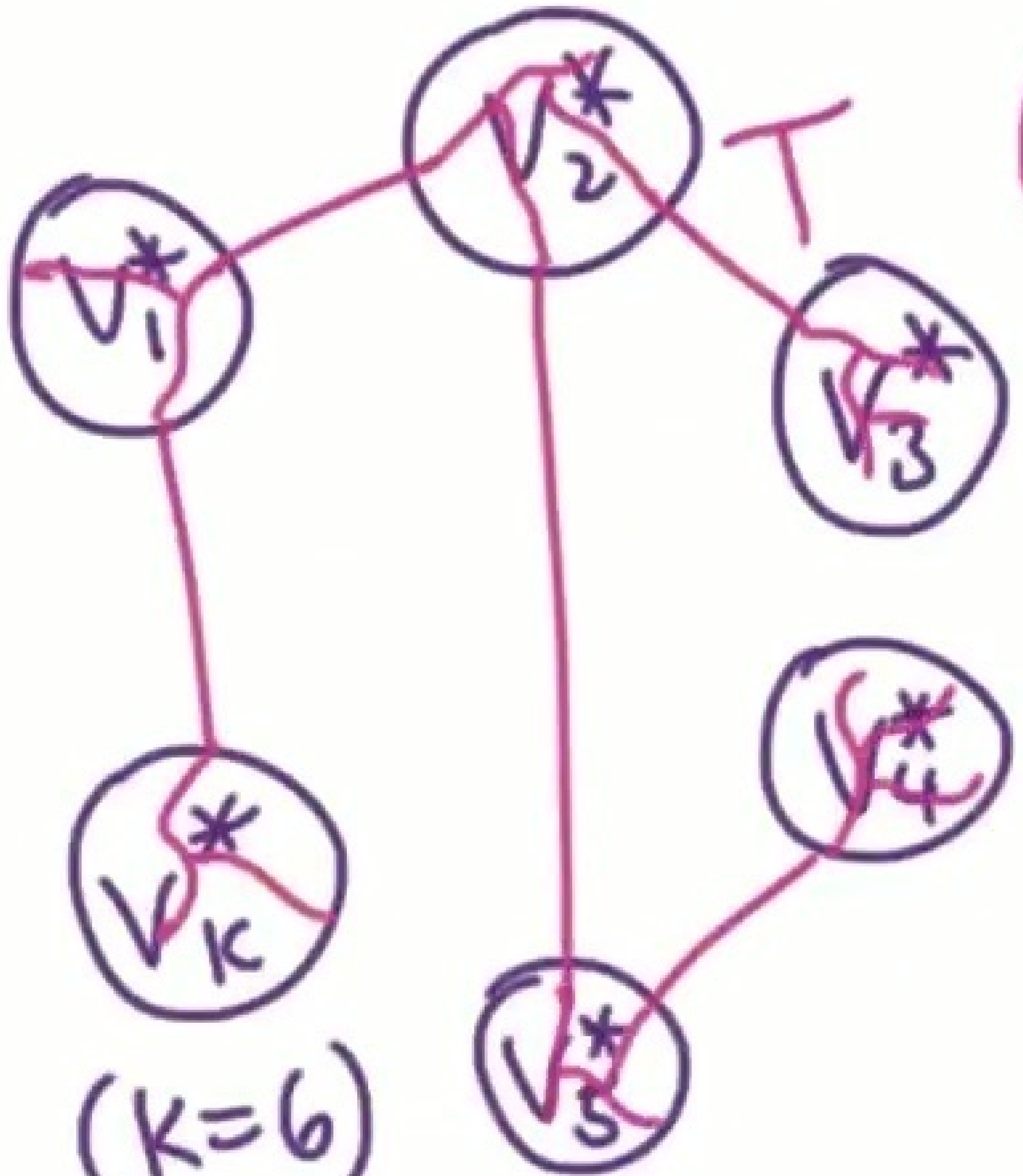
Hardness $n^{(\omega/3)k}$

Thorup's Tree Packing

- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

For any min k-cut, $\exists$ tree s.t. the k-cut cuts $\underbrace{\le 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \le 2k-2$

k-cut $\le 2k-2$-respects the tree



(k=6)

(k-1)-respects

(best possible)

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Thorup's Tree Packing

- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

For any min k-cut, ∃tree s.t. the k-cut cuts $\underbrace{\leq 2k-2}$ edges of the tree: $|E_T[V_1^*, \ldots, V_k^*]| \leq 2k-2$

k-cut $\leq 2k-2$-respects the tree

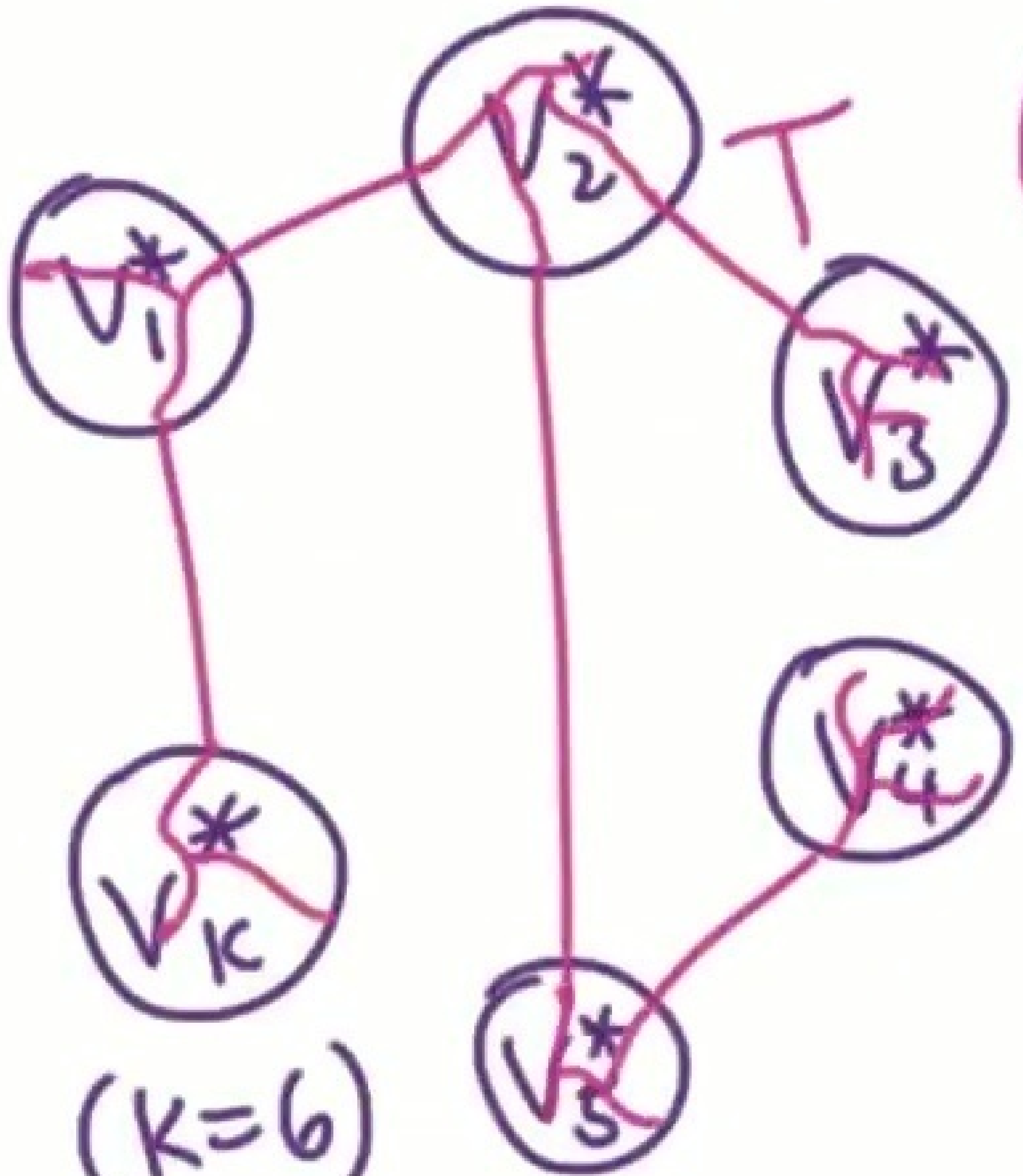
Exact $n^{(1+\omega/3)k}$

→ Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

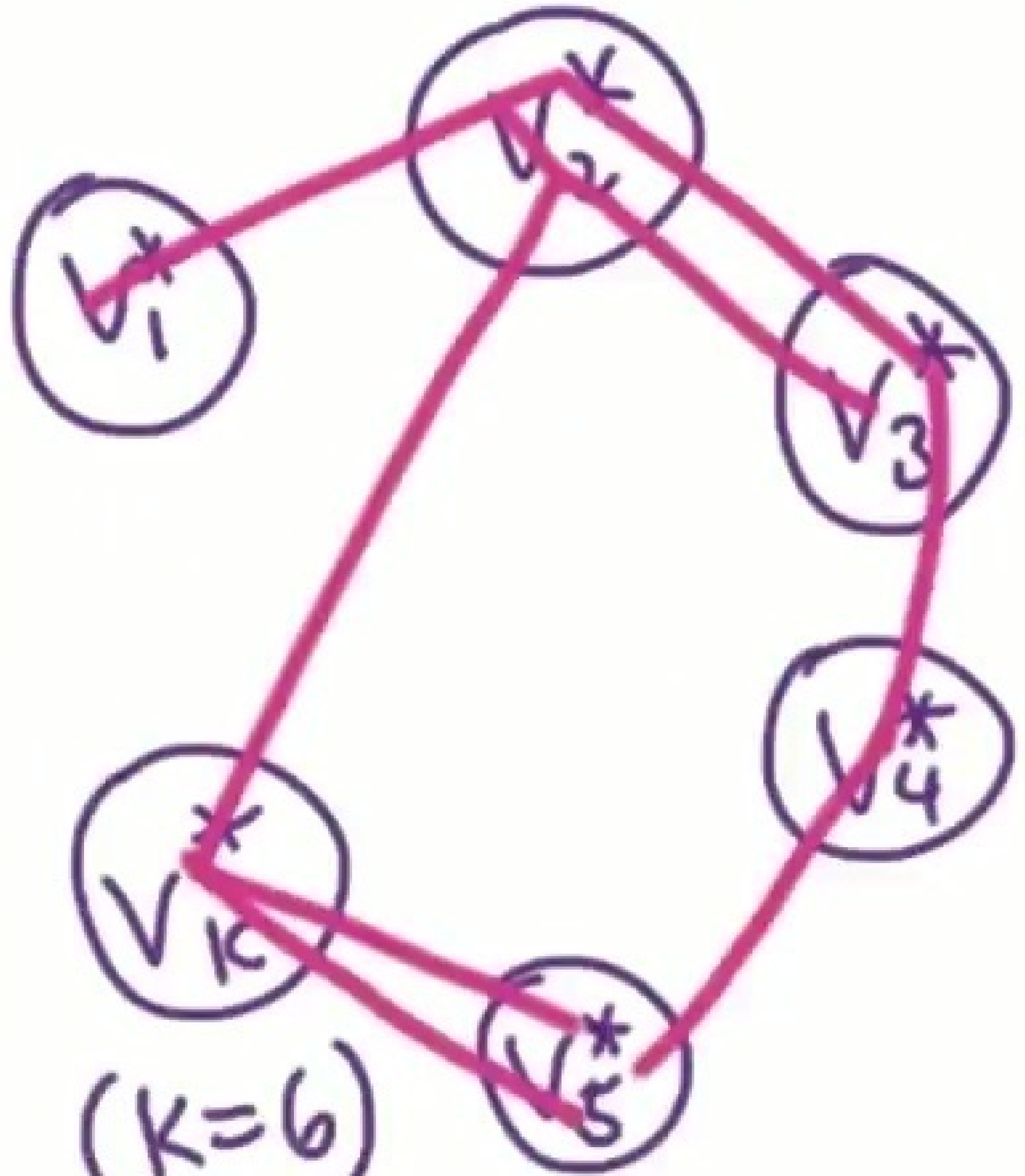
Hardness $n^{(\omega/3)k}$



(k=6)

(k-1)-respects

(best possible)



(k=6)

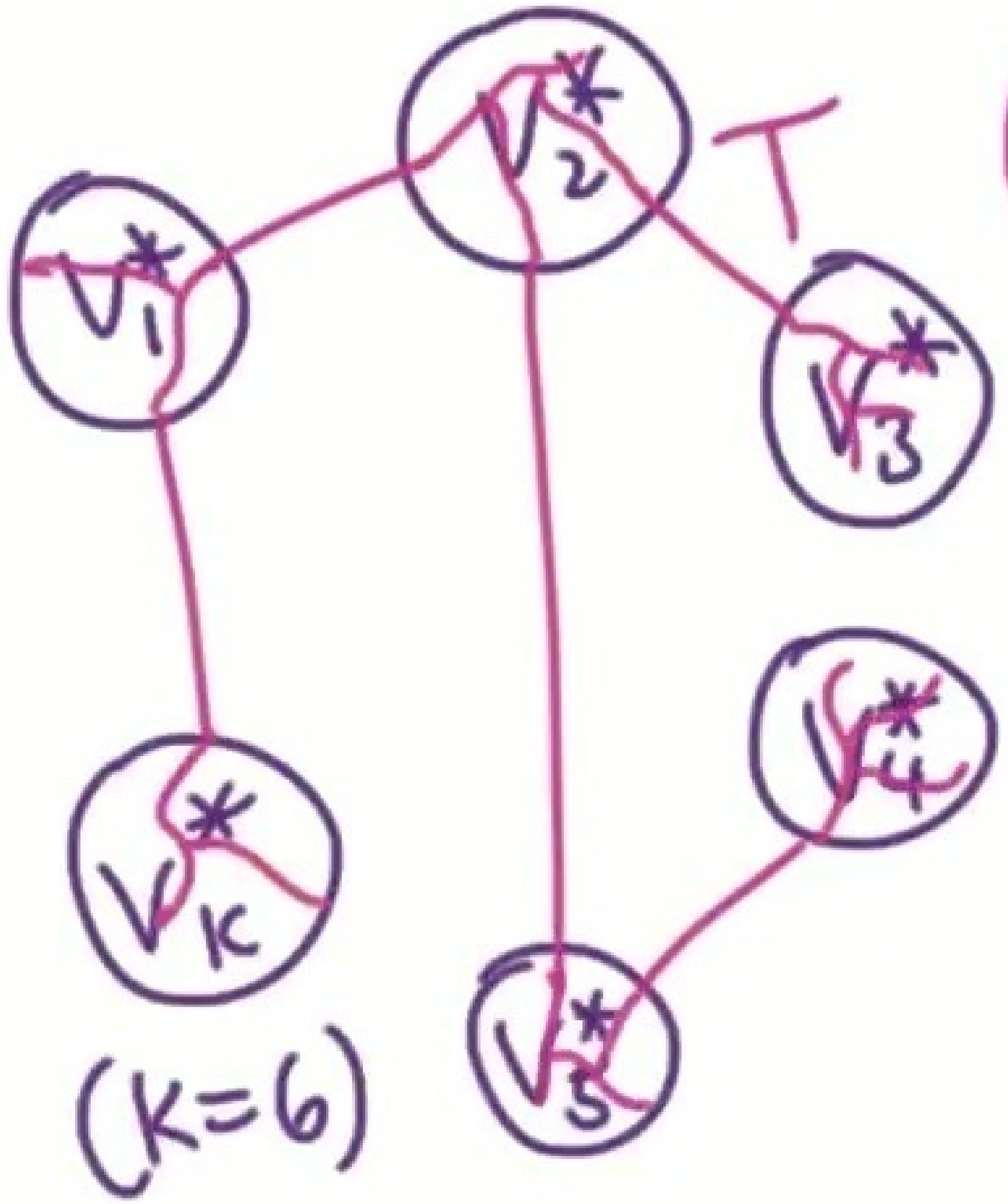
Thorup's Tree Packing

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

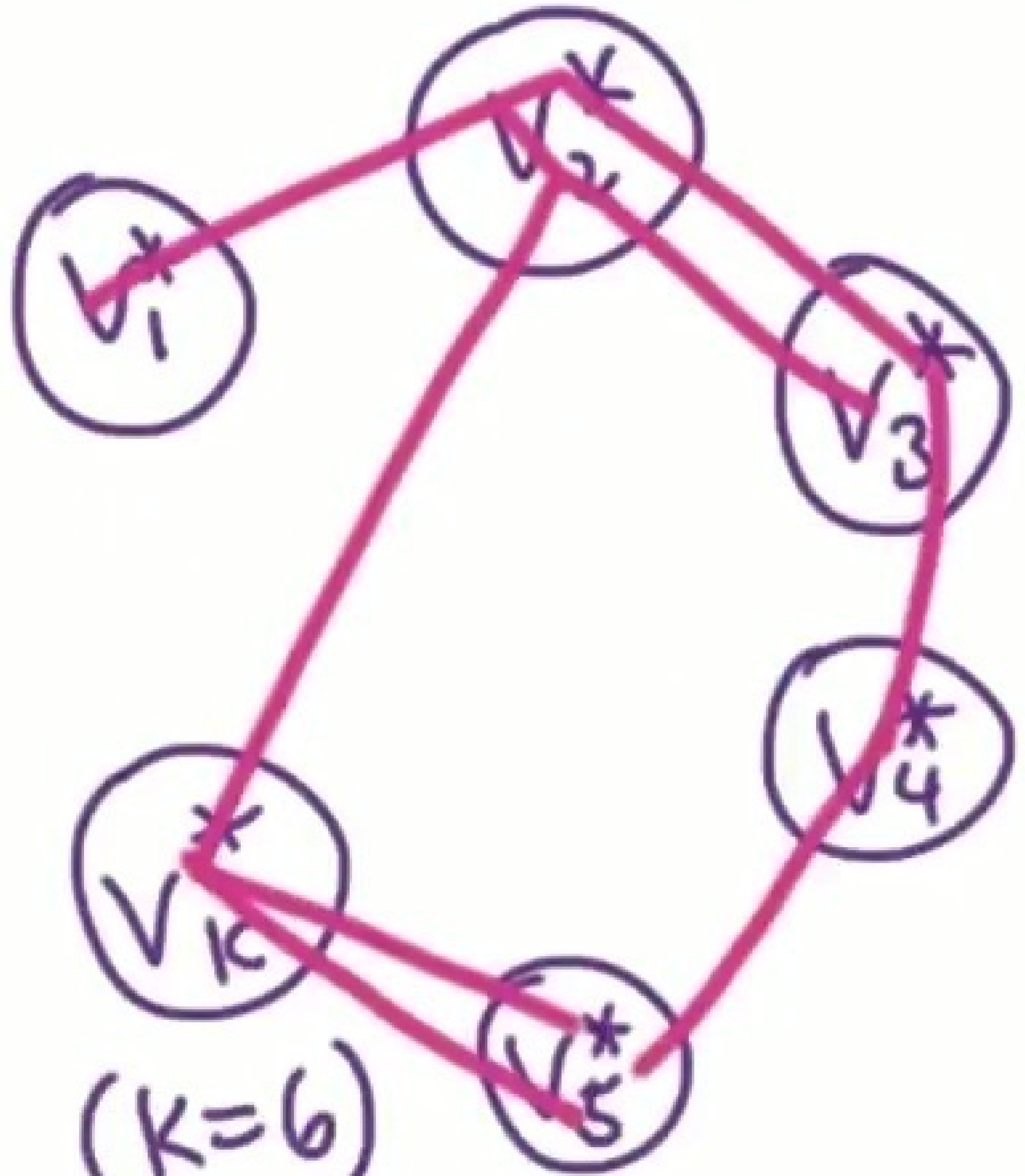
For any min k-cut, ∃tree s.t. the k-cut cuts $\underbrace{\leq 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \leq 2k-2$

k-cut $\leq 2k-2$-respects the tree

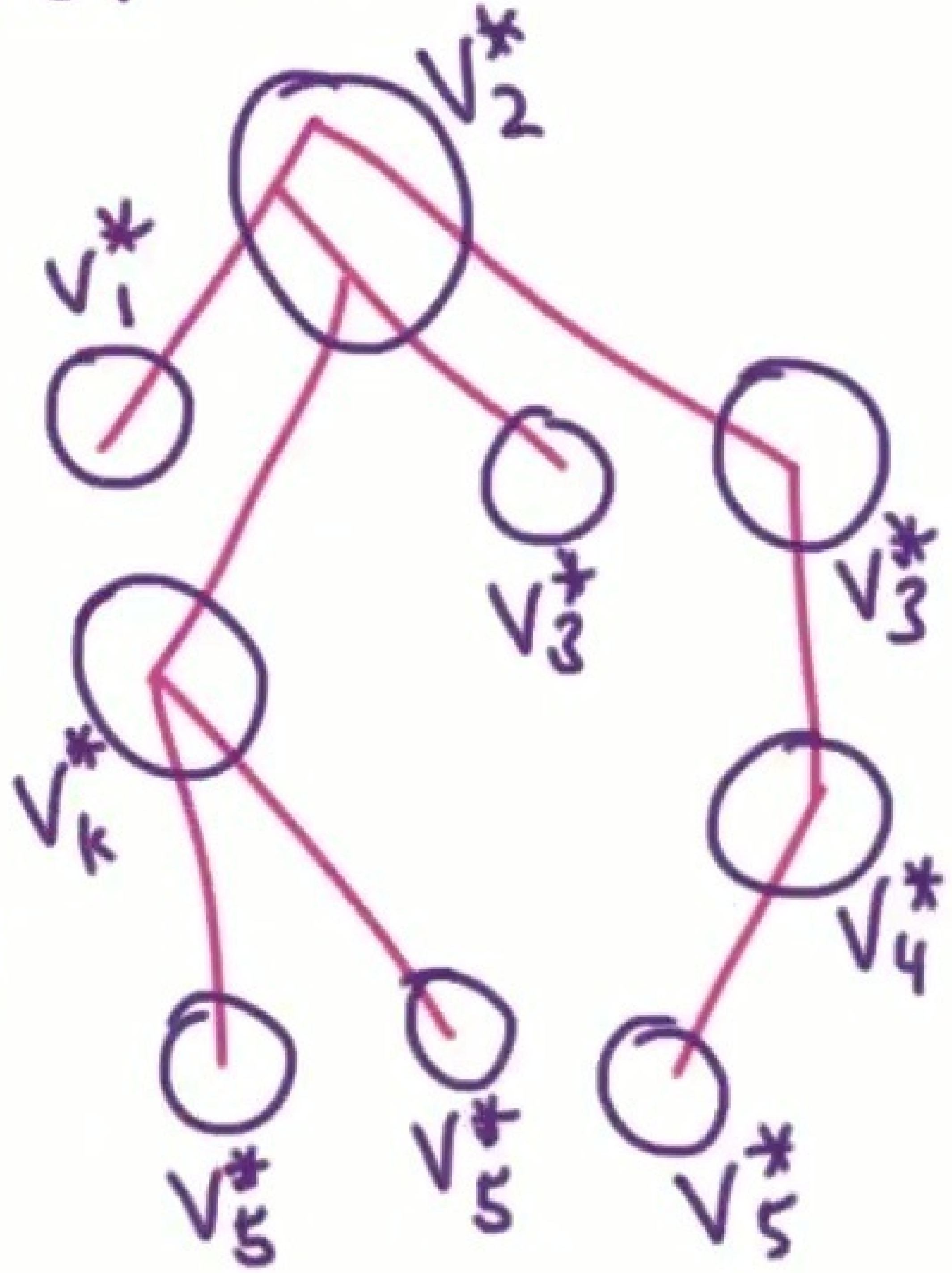


(k=6)

(k-1)-respects
(best possible)



(k=6)



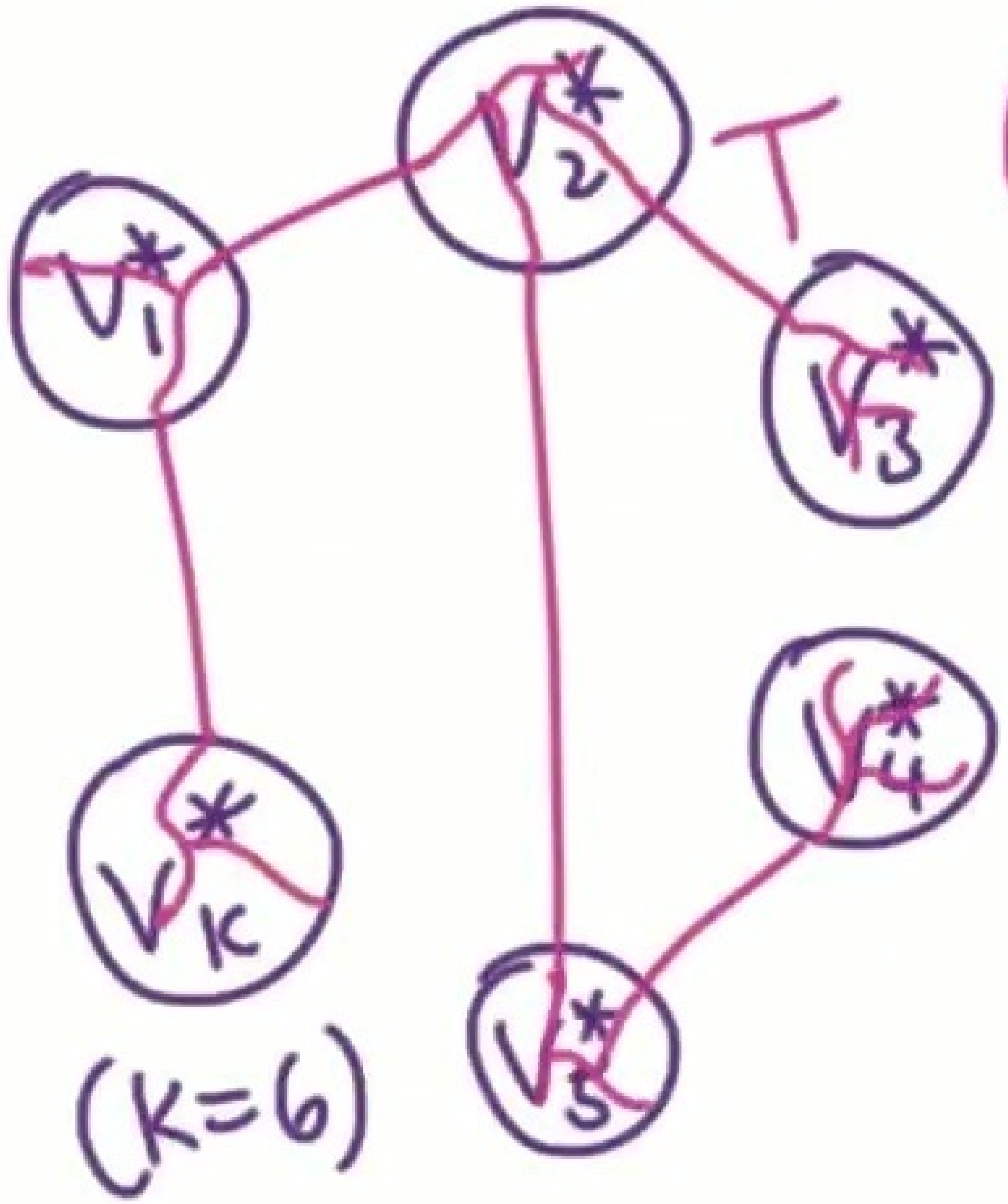
Thorup's Tree Packing

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

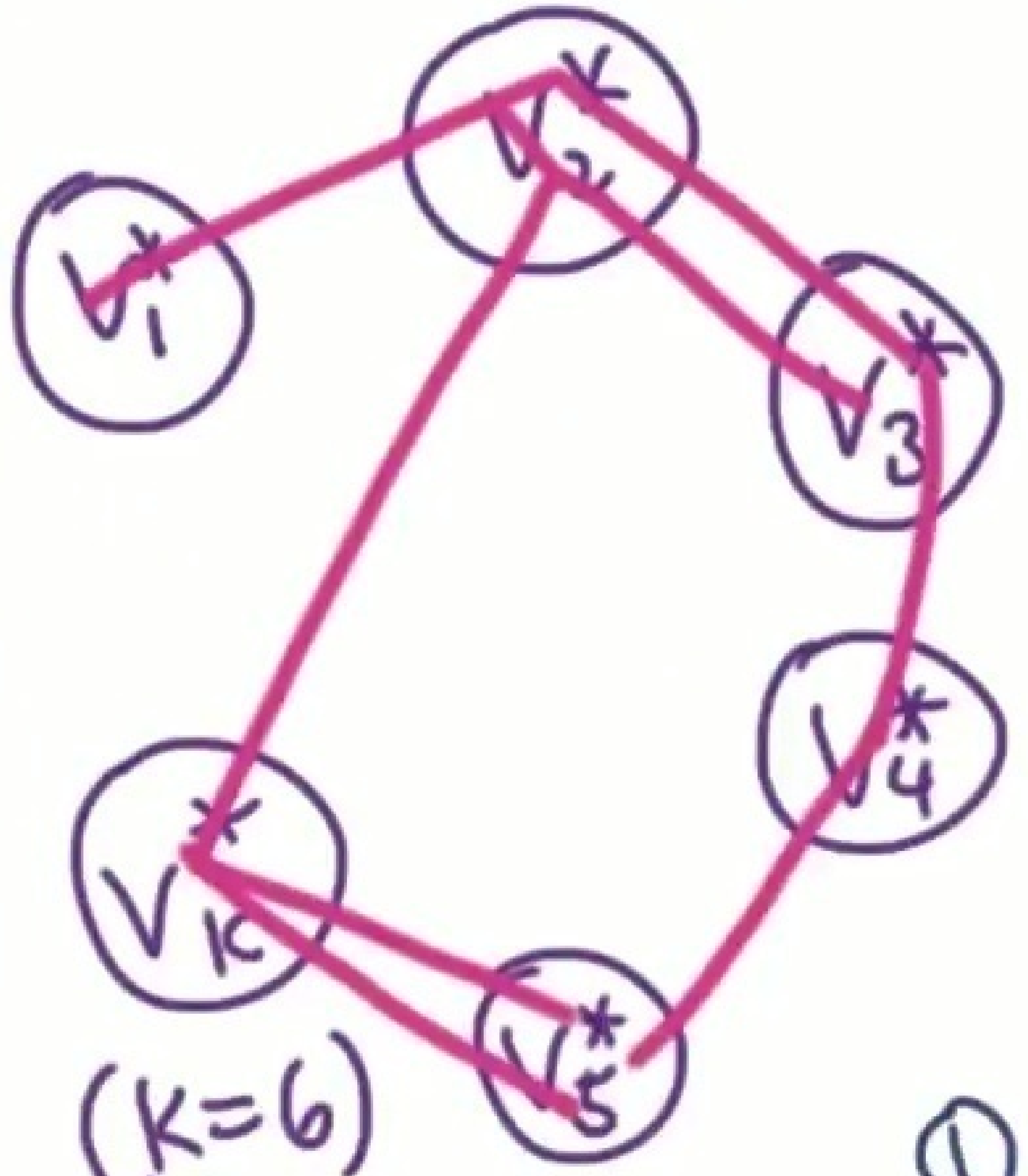
- Thm [Thorup'08]: Can find poly(n) spanning trees of G with the following property:

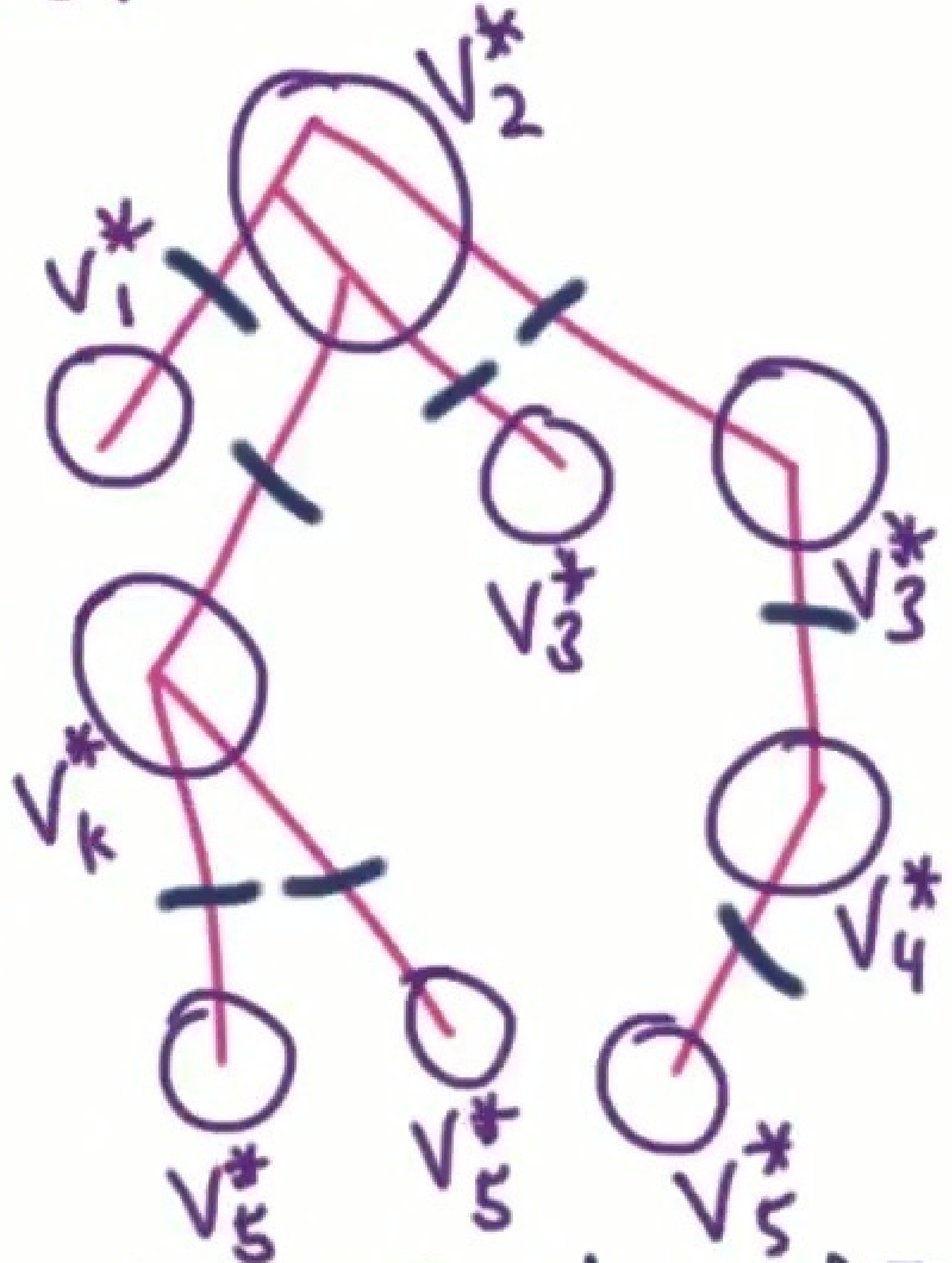
For any min k-cut, ∃tree s.t. the k-cut cuts $\underbrace{\le 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \le 2k-2$

k-cut $\le 2k-2$-respects the tree



(k-1)-respects
(best possible)





① cut $\le 2k-2$ edges of T

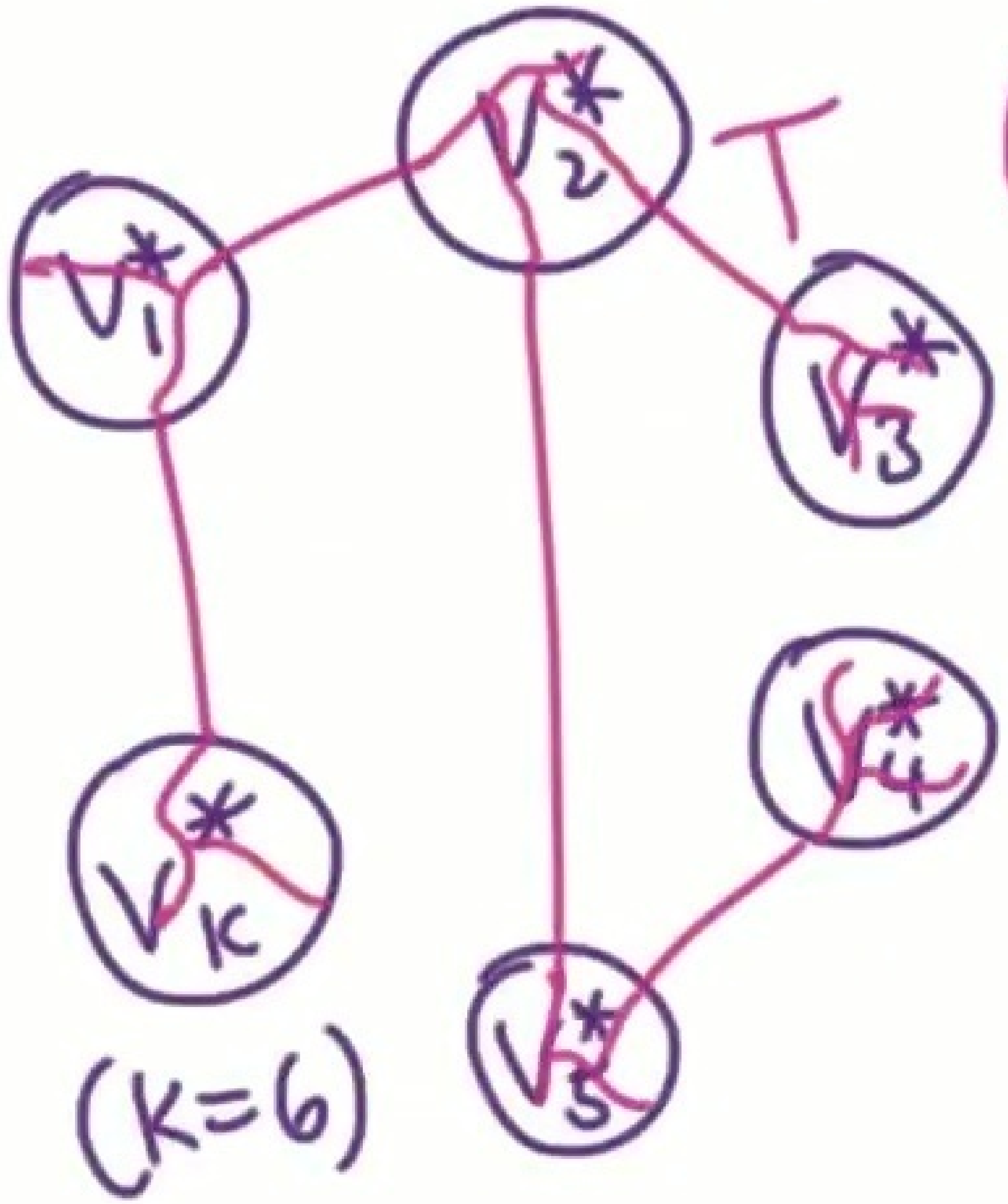
Thorup's Tree Packing

Exact $n^{(1+\omega/3)k}$
→ Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

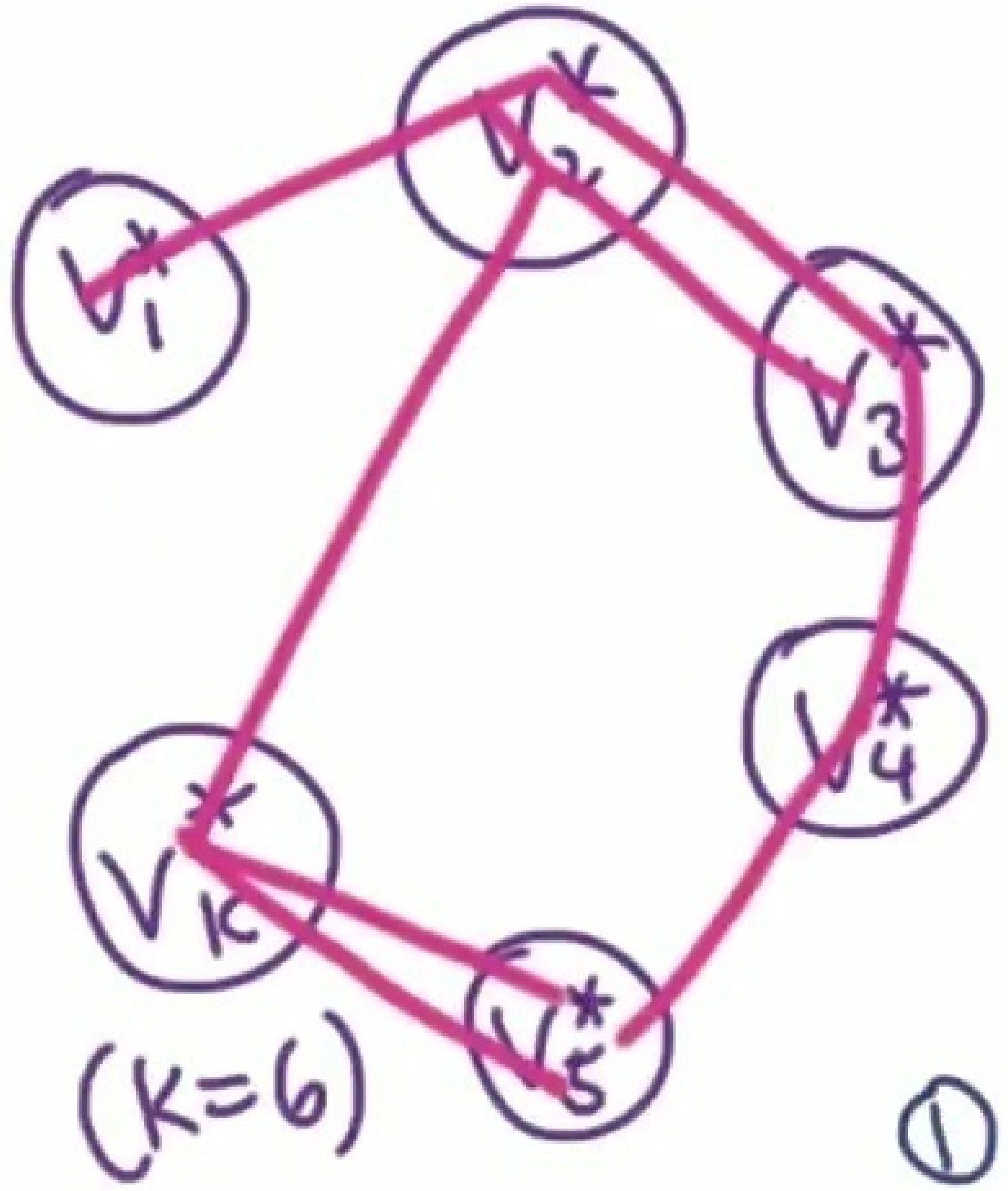
- Thm [Thorup '08]: Can find poly(n) spanning trees of G with the following property:

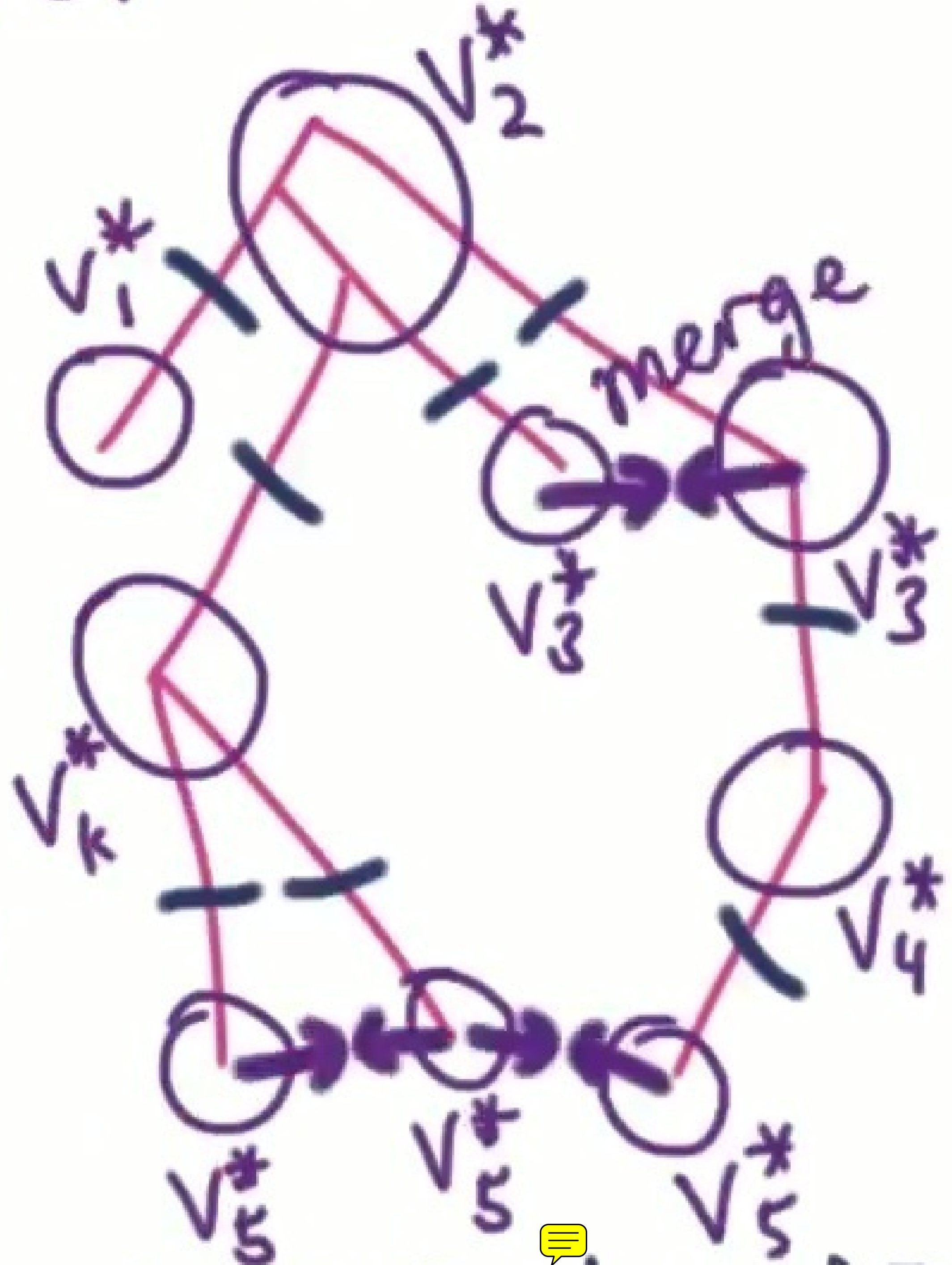
For any min k-cut, ∃tree s.t. the k-cut cuts $\underbrace{\le 2k-2}$ edges of the tree: $|E_T[V_1^*, \dots, V_k^*]| \le 2k-2$

k-cut $\le 2k-2$-respects the tree



(k-1)-respects
(best possible)



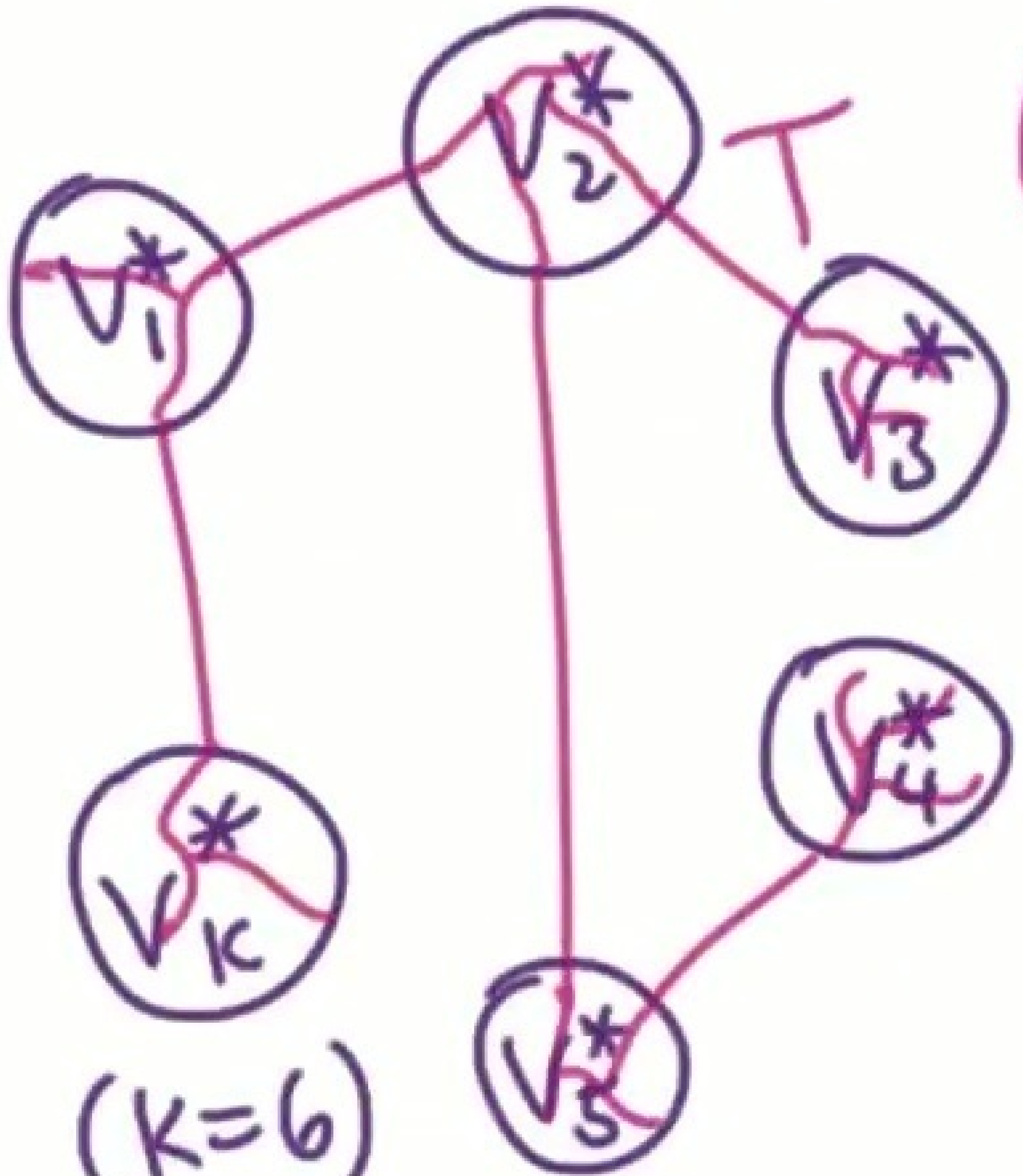


① cut $\le 2k-2$ edges of T
② merge comps

Matrix Multiplication

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Easy case: ① (k-1)-respects tree

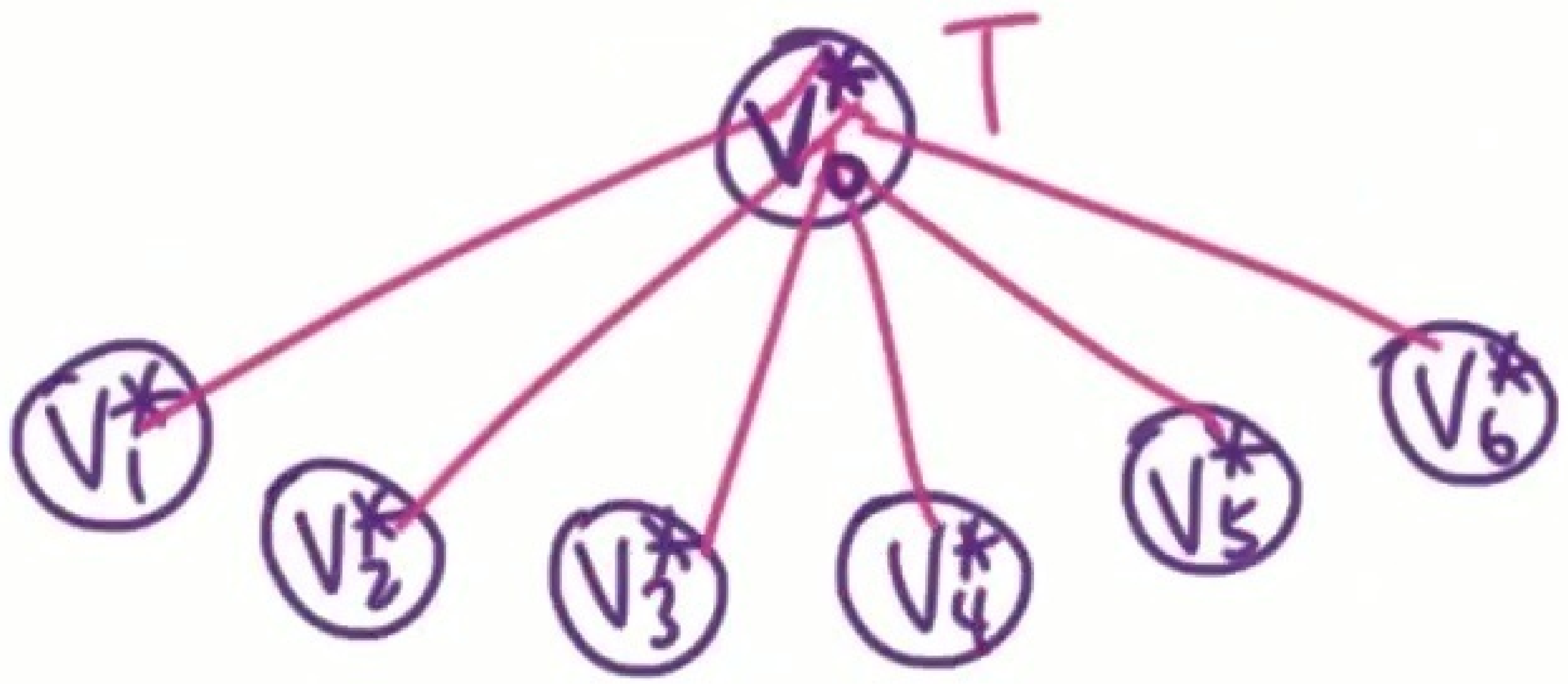


(k-1)-respects
(best possible)

Matrix Multiplication

Easy case: ① $(k-1)$-respects tree

② T connects V_i^* in a star



Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to $(k-1)$-resp.

$\rightarrow$ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Matrix Multiplication

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

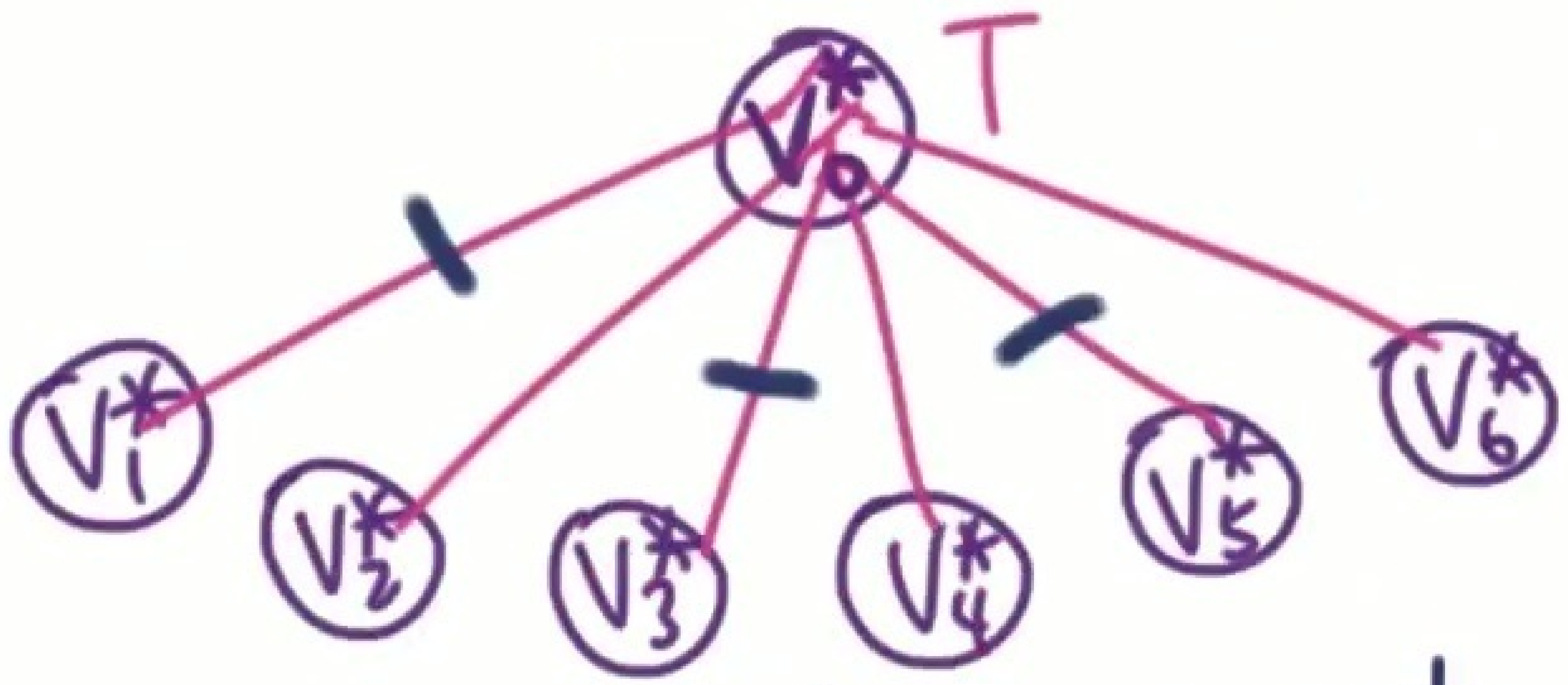
Reduction to (k-1)-resp.

→ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Easy case: ① (k-1)-respects tree

② T connects V_i^* in a star



k-cut: isolate (k-1) components V_i^* to minimize k-cut between $V_{i_1}^*, V_{i_2}^*, \ldots, V_{i_{k-1}}^*$ and $\overline{V_{i_1}^* \cup \cdots \cup V_{i_{k-1}}^*}$

Matrix Multiplication

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

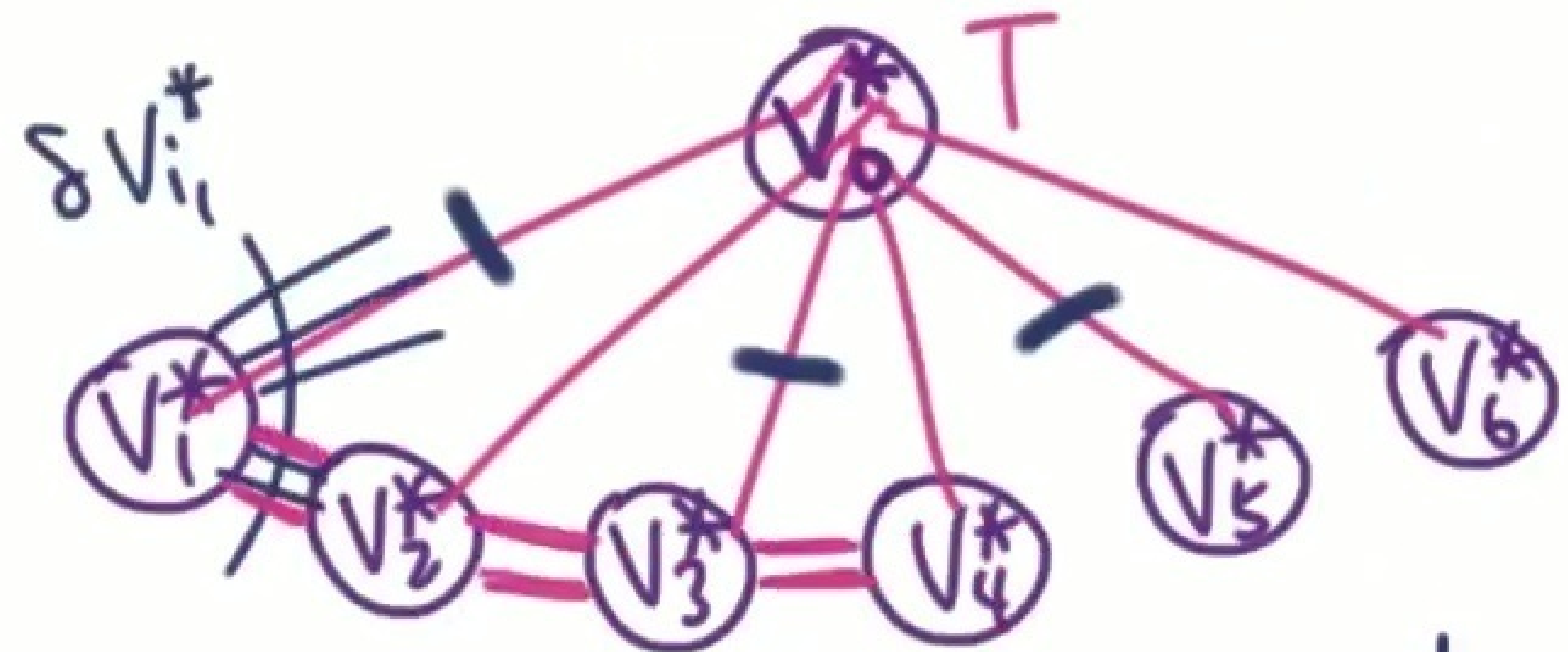
Reduction to $(k-1)$-resp.

$\to$ k-clique-like mtx. mult.

Hardness $n^{(\omega/3)k}$

Easy case: ① $(k-1)$-respects tree

② T connects V_i^* in a star



k-cut: isolate $(k-1)$ components V_i^* to minimize k-cut between $V_{i_1}^*, V_{i_2}^*, \ldots, V_{i_{k-1}}^*$ and $\overline{V_{i_1}^* \cup \cdots \cup V_{i_{k-1}}^*}$

$$= \sum w(\delta V_{i_j}^*) - w(E[V_{i_1}^*, \ldots, V_{i_{k-1}}^*])$$

Matrix Multiplication

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

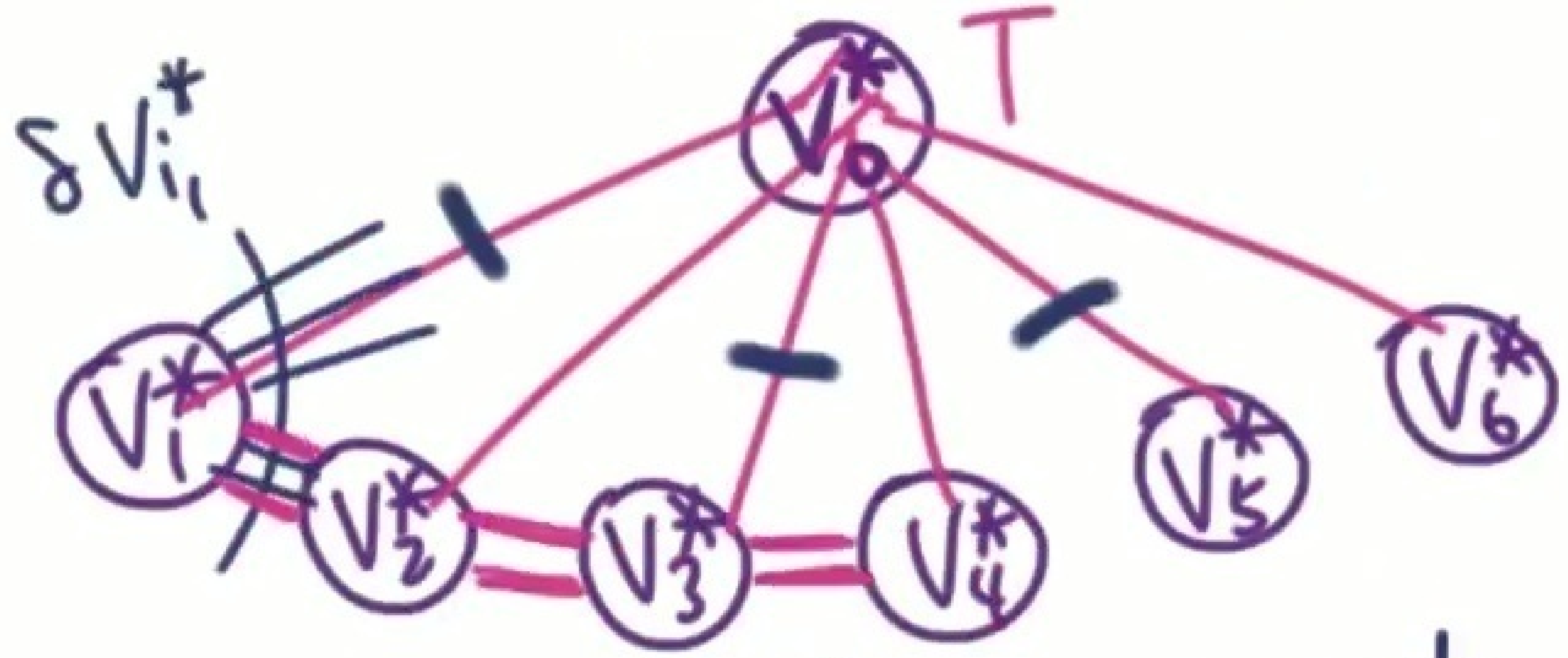
Reduction to $(k-1)$-resp.

$\rightarrow$ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Easy case: ① $(k-1)$-respects tree

② T connects V_i^* in a <u>star</u>



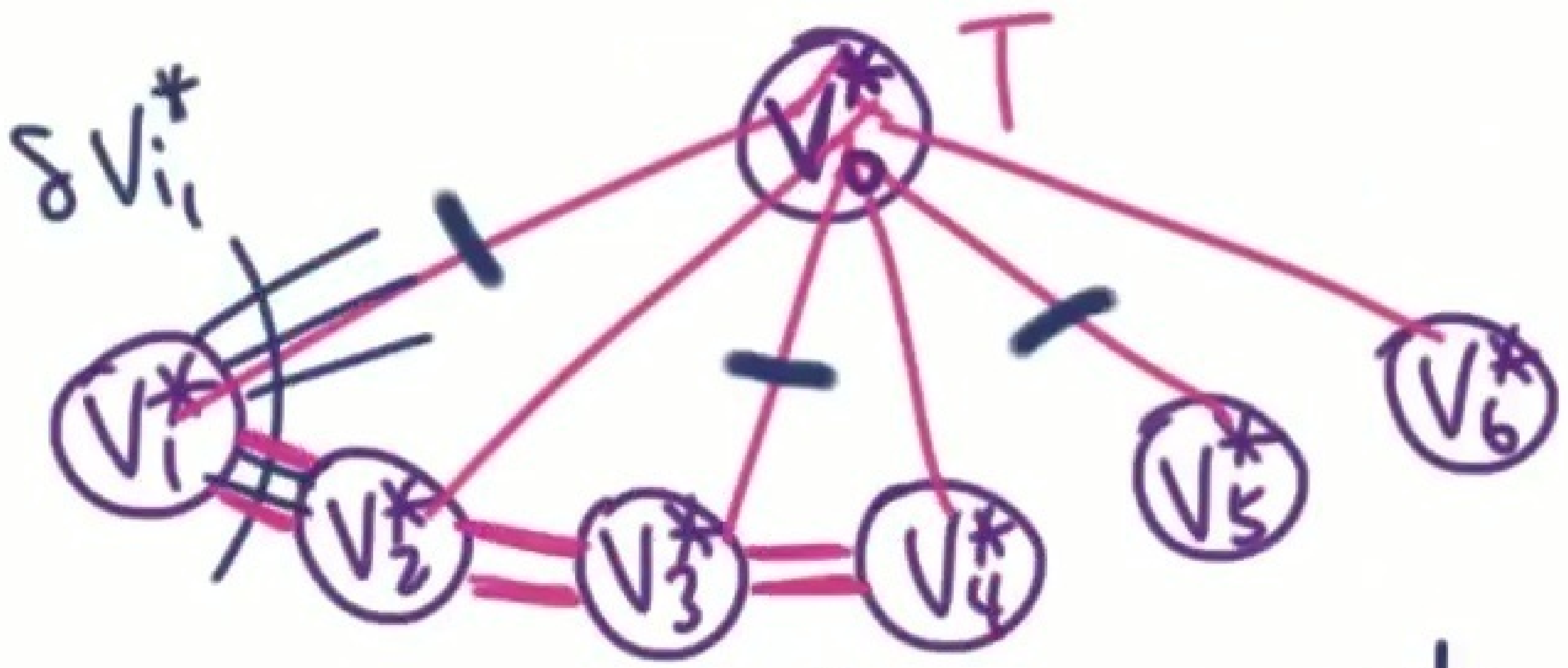
k-cut: isolate $(k-1)$ components V_i^* to minimize k-cut between $V_{i_1}^*, V_{i_2}^*, \dots, V_{i_{k-1}}^*$ and $\overline{V_{i_1}^* \cup \cdots \cup V_{i_{k-1}}^*}$

$$= \underbrace{\sum w(\delta V_{i_j}^*)}_{\text{node weights}} - \underbrace{w(E[V_{i_1}^*, \dots, V_{i_{k-1}}^*])}_{\text{"weight" of a certain k-clique (negative)}}$$

Matrix Multiplication

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Easy case: ① (k-1)-respects tree
② T connects V_i^* in a star



k-cut: isolate (k-1) components V_i^* to minimize k-cut between $V_{i_1}^*, V_{i_2}^*, \ldots, V_{i_{k-1}}^*$ and $\overline{V_{i_1}^* \cup \cdots \cup V_{i_{k-1}}^*}$

$$= \underbrace{\sum w(\delta V_{i_j}^*)}_{\text{node weights}} \underbrace{- w(E[V_{i_1}^*, \ldots, V_{i_{k-1}}^*])}_{\text{"weight" of a certain k-clique (negative)}}$$

- Unweighted graph: node weights $w(\delta V_i^*) \in [n^2]$, edge weights $-w(E[V_i^*, V_j^*]) \in [-n^2]$

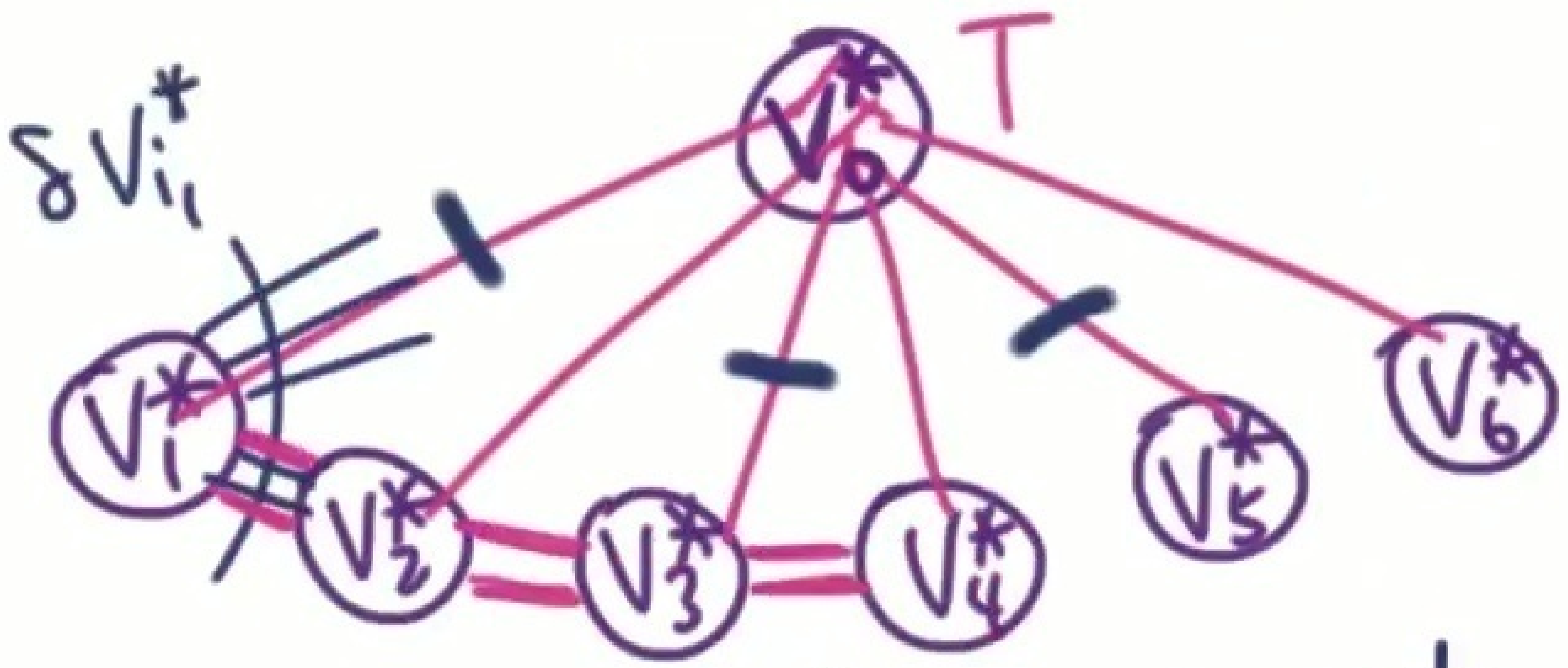
Matrix Multiplication

Exact $n^{(1+\omega/3)k}$
- Thorup's tree packing
- Reduction to $(k-1)$-resp.

$\rightarrow$ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Easy case: ① $(k-1)$-respects tree
② T connects V_i^* in a <u>star</u>



k-cut: isolate $(k-1)$ components V_i^* to minimize k-cut between $V_{i_1}^*, V_{i_2}^*, \ldots, V_{i_{k-1}}^*$ and $\overline{V_{i_1}^* \cup \cdots \cup V_{i_{k-1}}^*}$

$$= \underbrace{\sum w(\delta V_{i_j}^*)}_{\text{node weights}} - \underbrace{w(E[V_{i_1}^*, \ldots, V_{i_{k-1}}^*])}_{\text{"weight" of a certain k-clique (negative)}}$$

- Unweighted graph: node weights $w(\delta V_i^*) \in [n^2]$, edge weights $-w(E[V_i^*, V_j^*]) \in [-n^2]$
- Can solve node+edge weighted k-clique, integer weights in $[-W, W]$, in $O(W \cdot n^{(\omega/3)k})$

(k-1)-respecting tree

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

→ k-clique-like mtx.mult.

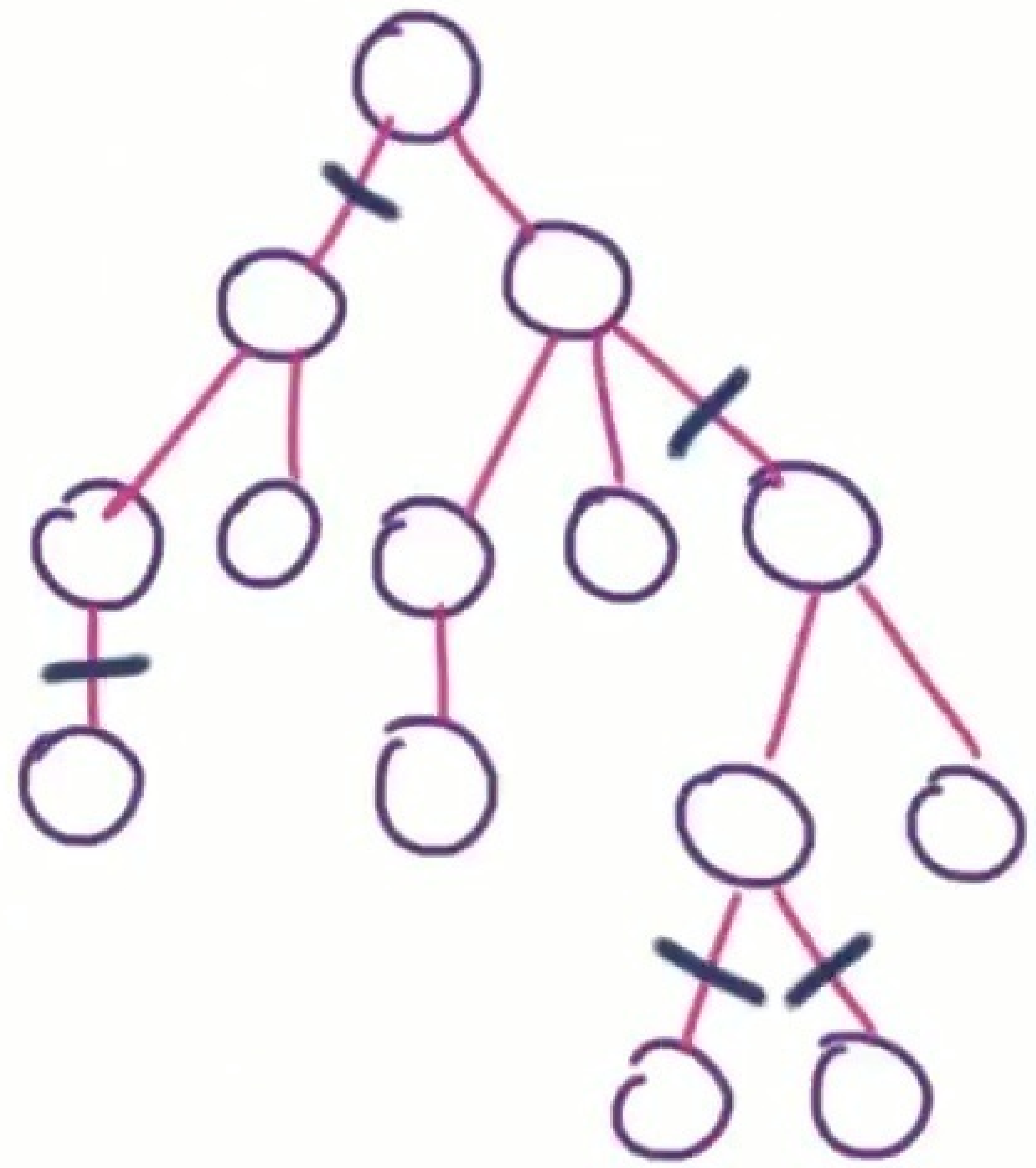
Hardness $n^{(\omega/3)k}$

$(k-1)$-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to $(k-1)$-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • $(k-1)$-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut $(k-1)$ edges of tree to minimize k-cut in G of the k connected components



(k-1)-respecting tree

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

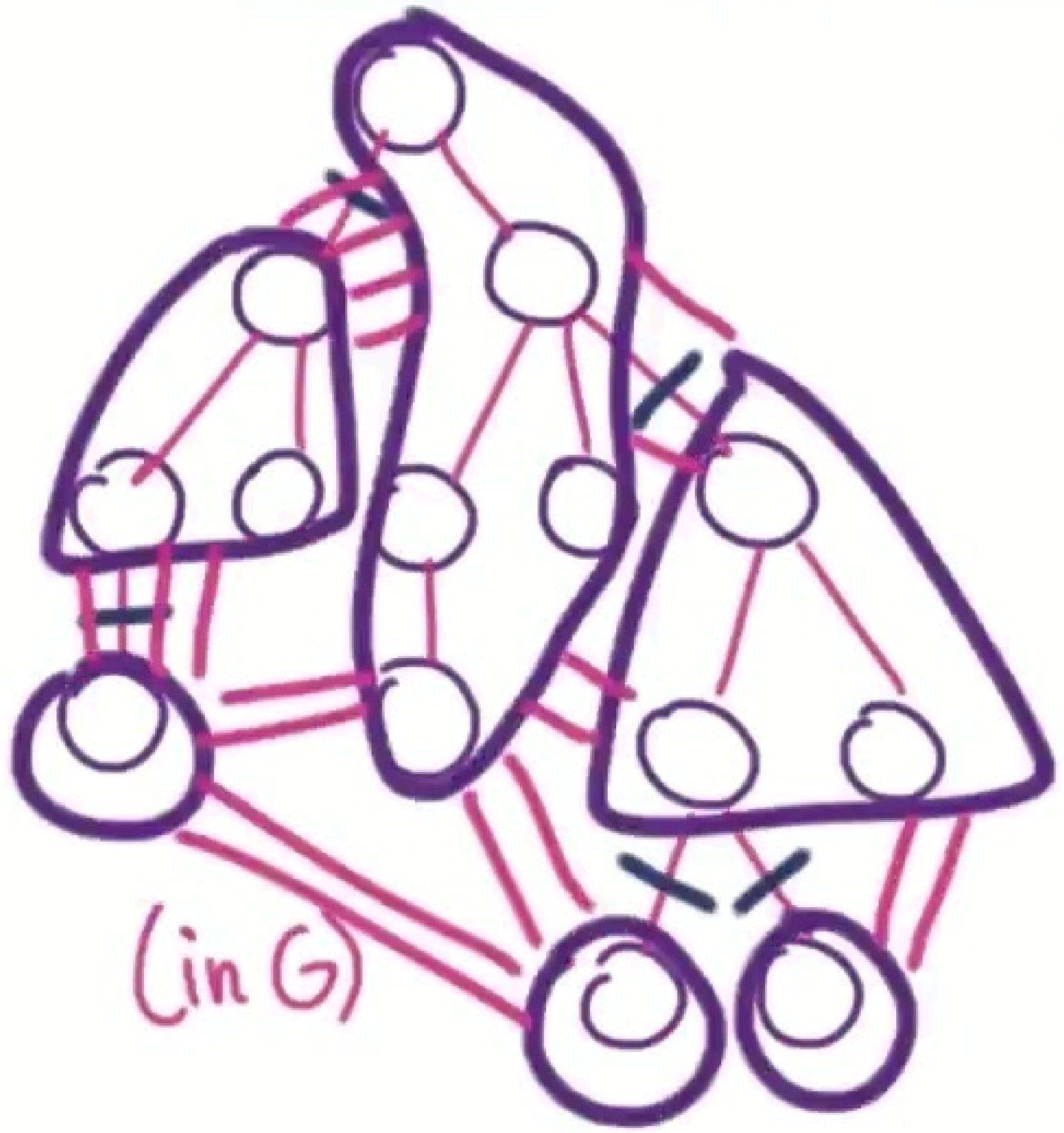
Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

→ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$



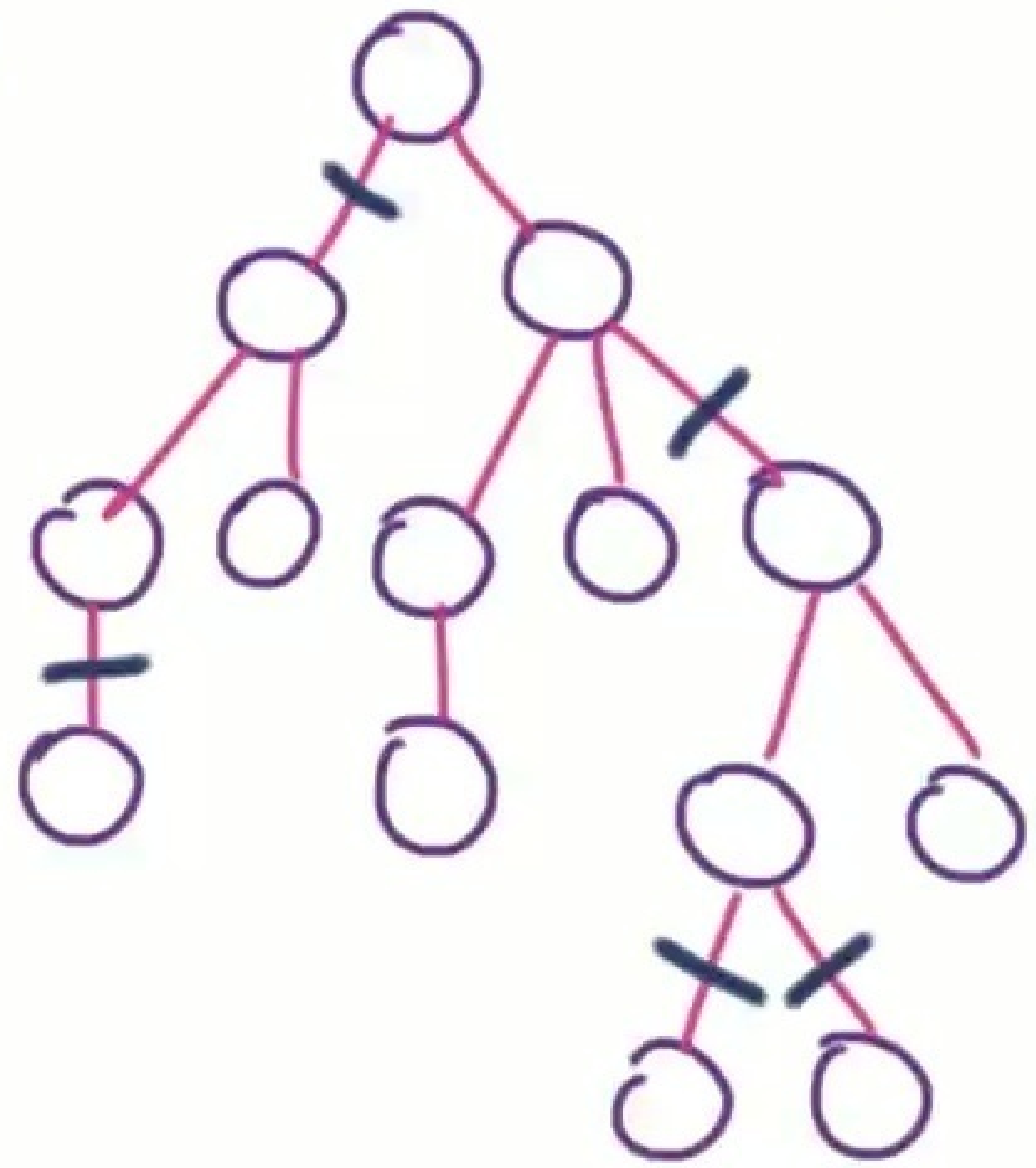
(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \leq k-1)$



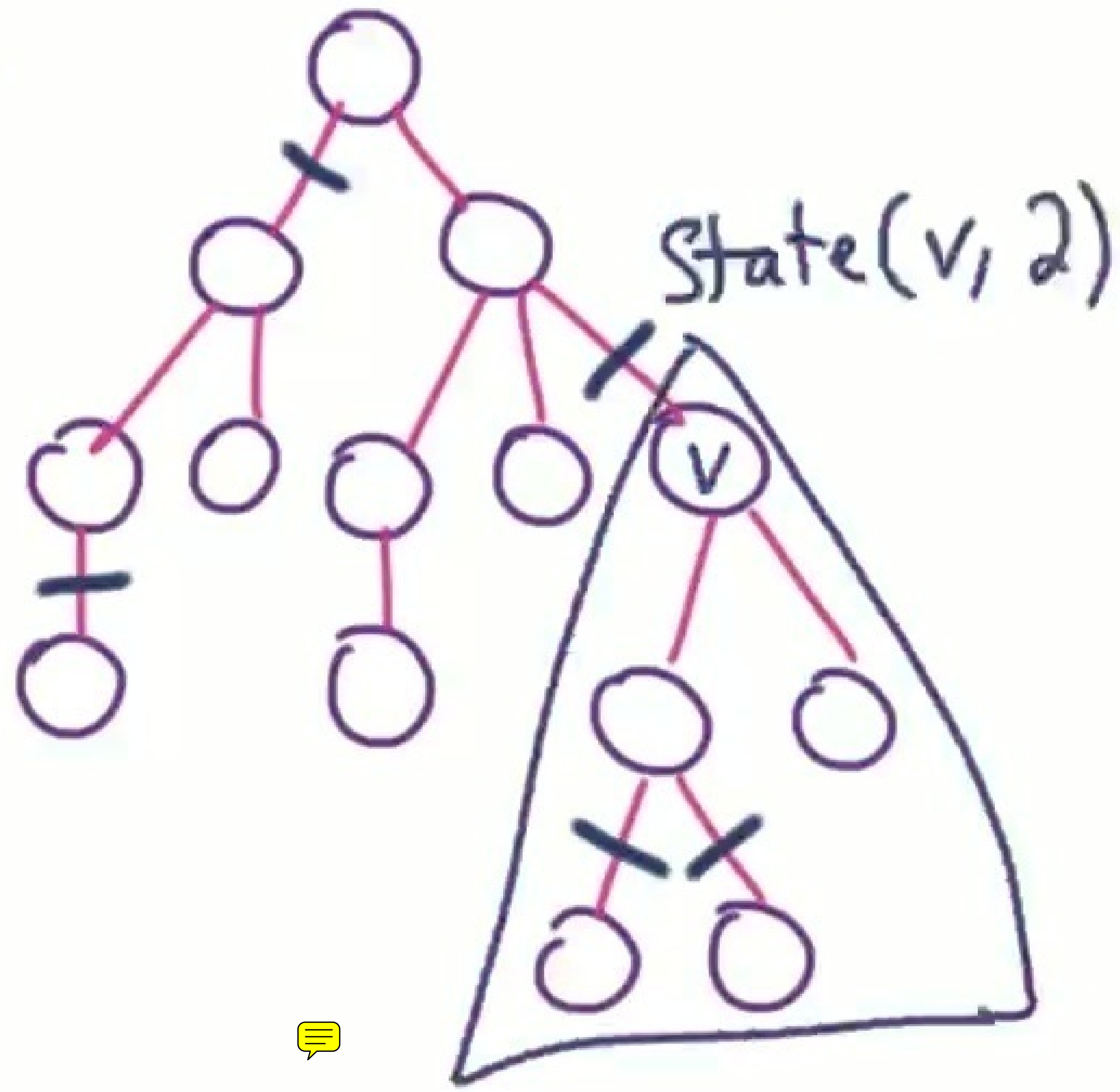
(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\to$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \le k-1)$



$(k-1)$-respecting tree

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to $(k-1)$-resp.

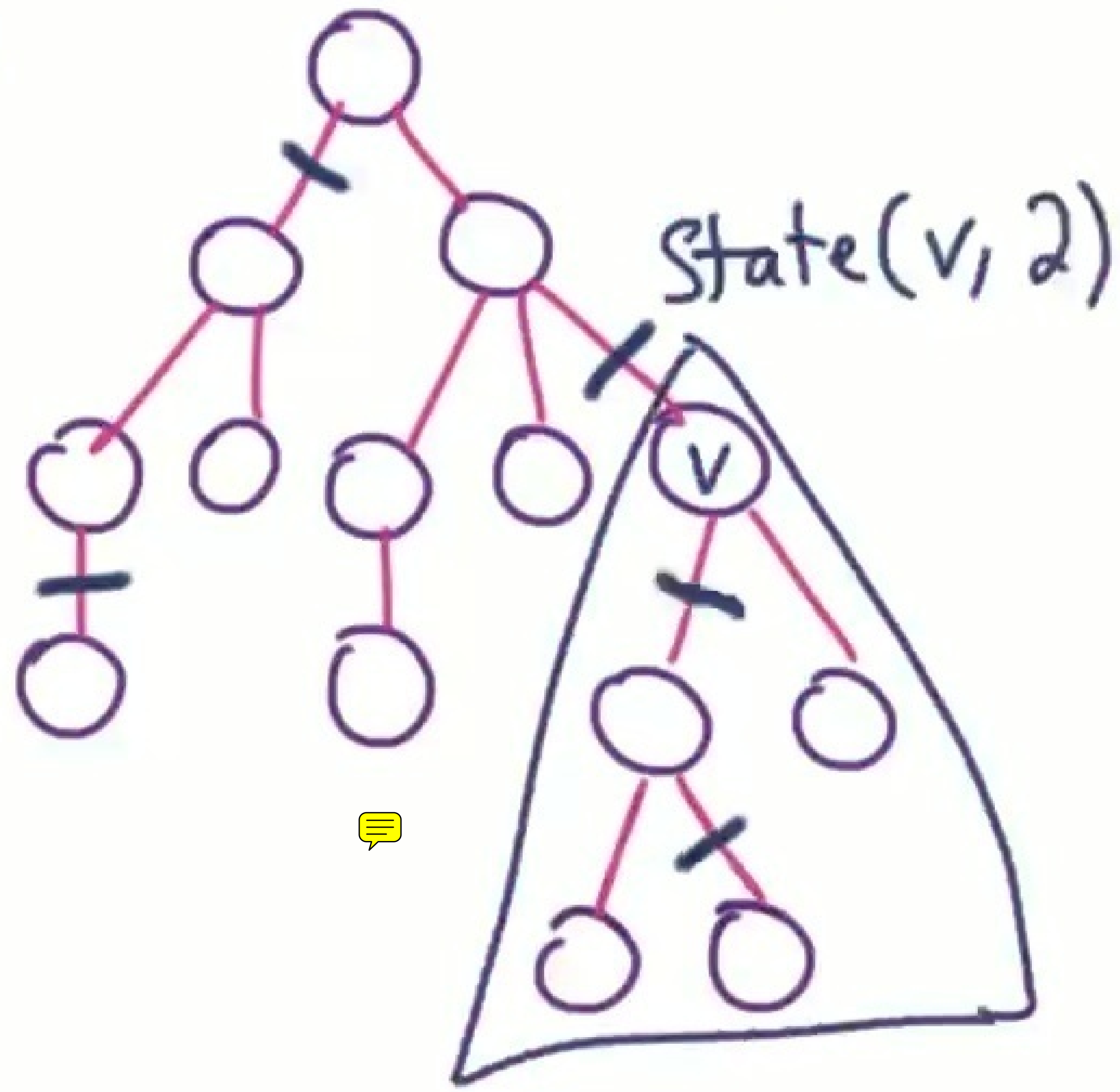
$\rightarrow$ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Medium case: • $(k-1)$-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut $(k-1)$ edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with $(s-1)$ edges in v's subtree $(s \leq k-1)$



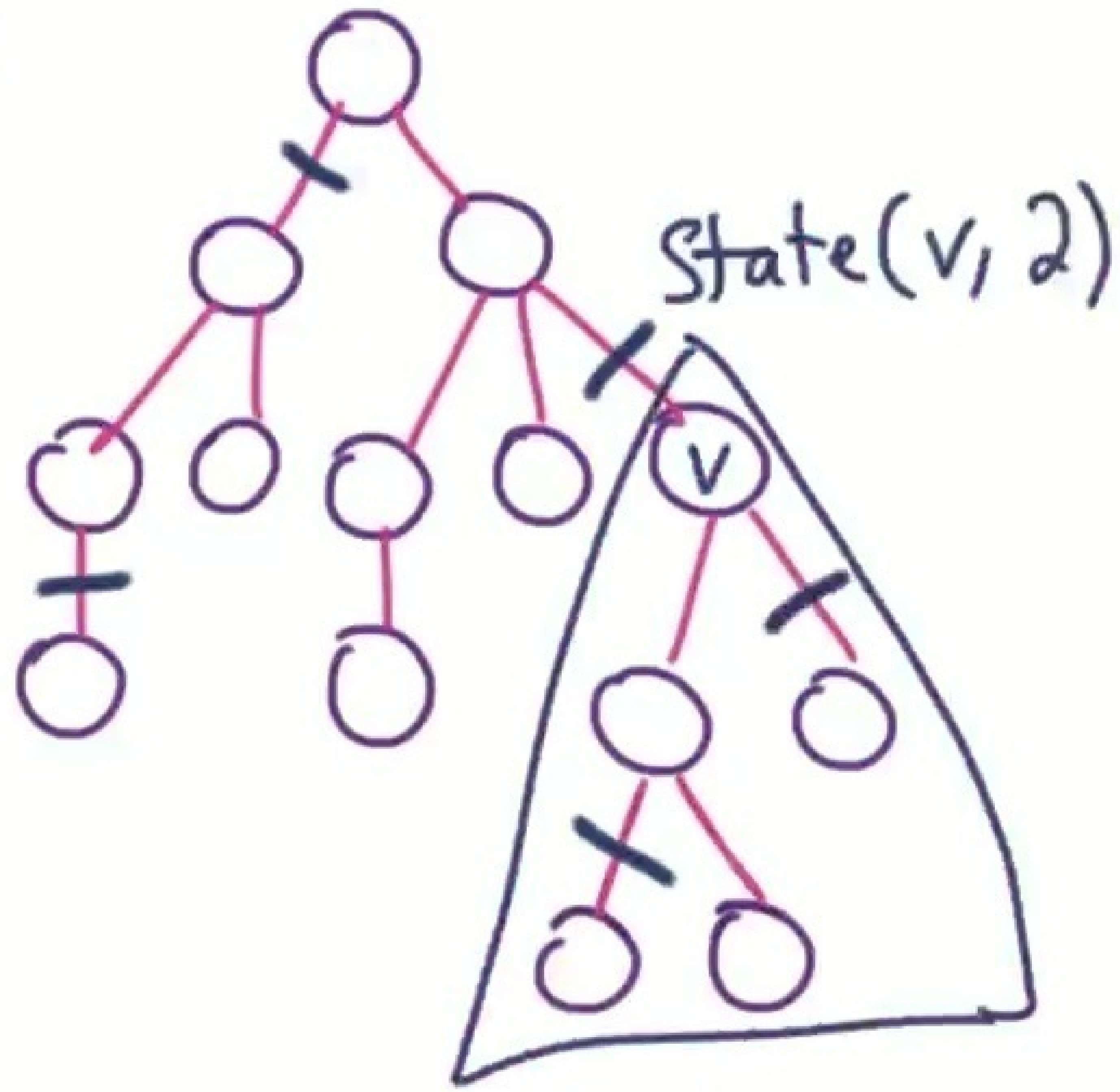
(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \leq k-1)$



(k-1)-respecting tree

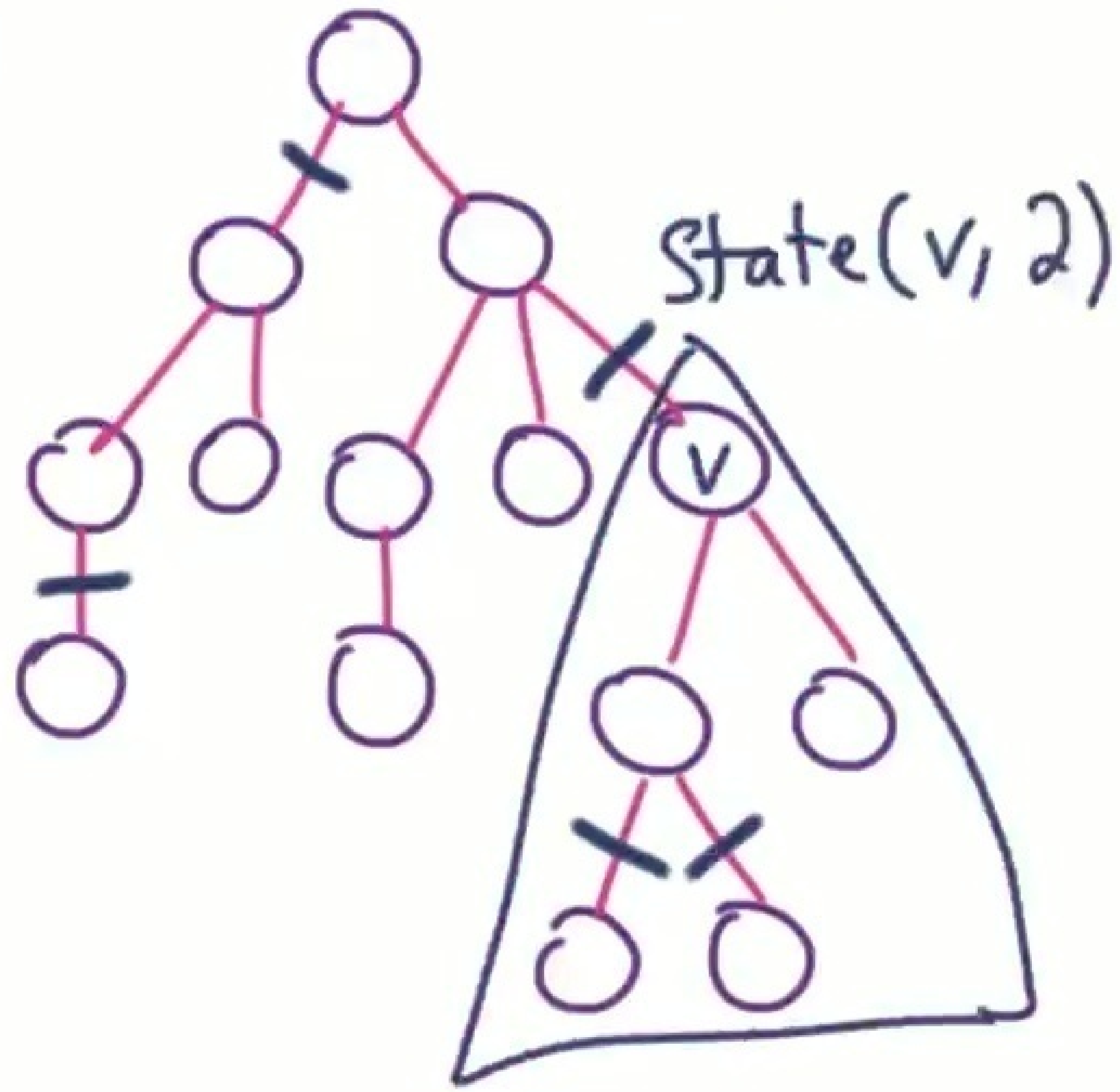
Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
→ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \le k-1)$

Intuition: if delete v's parent, then v's subtree is independent instance



(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

$\to$ k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

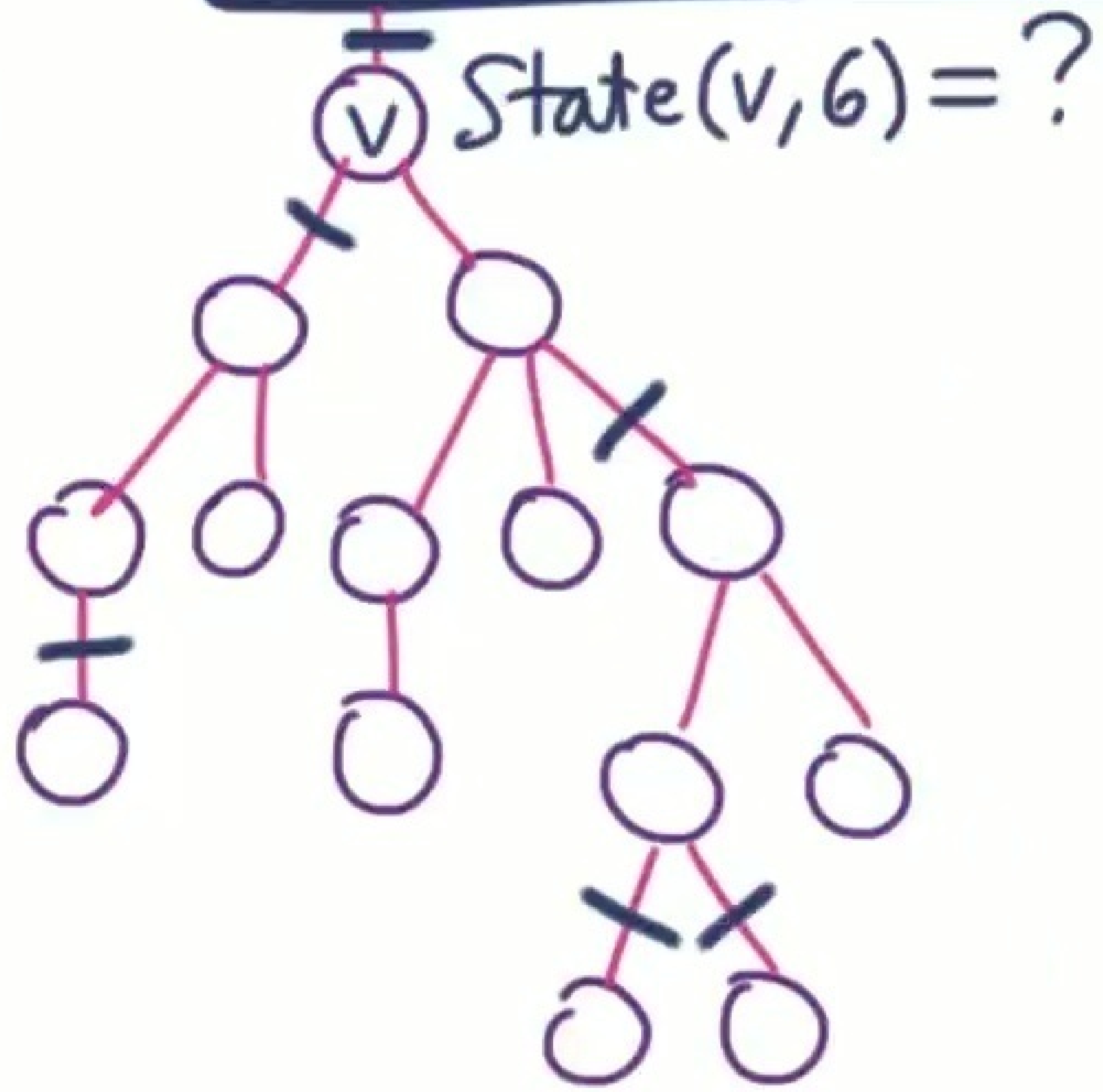
Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree ($s \le k-1$)

Intuition: if delete v's parent, then v's subtree is independent instance

Computing State(v,s):



(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

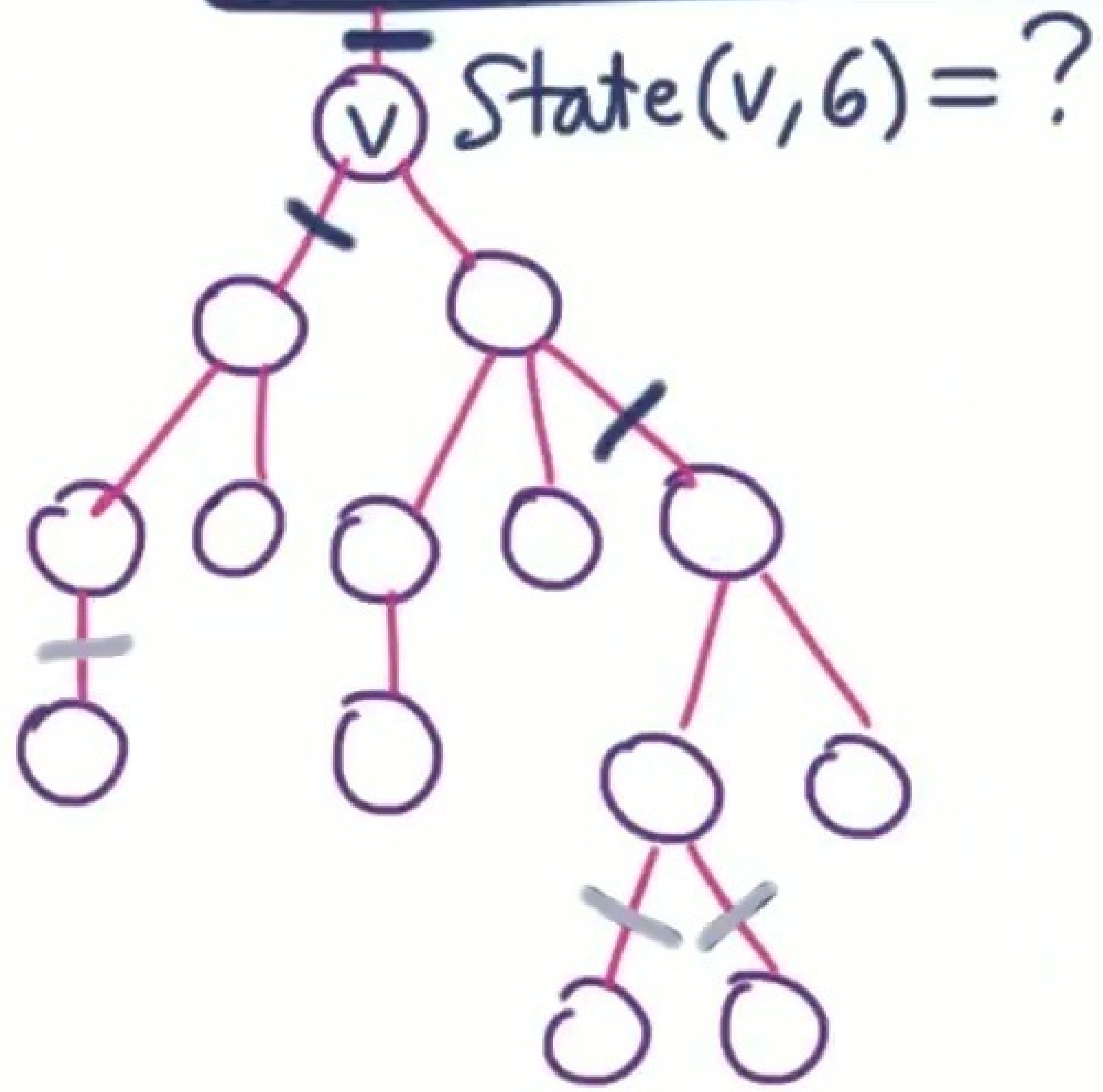
Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \leq k-1)$

Intuition: if delete v's parent, then v's subtree is independent instance

Computing State(v,s):
• focus on "maximal" edges



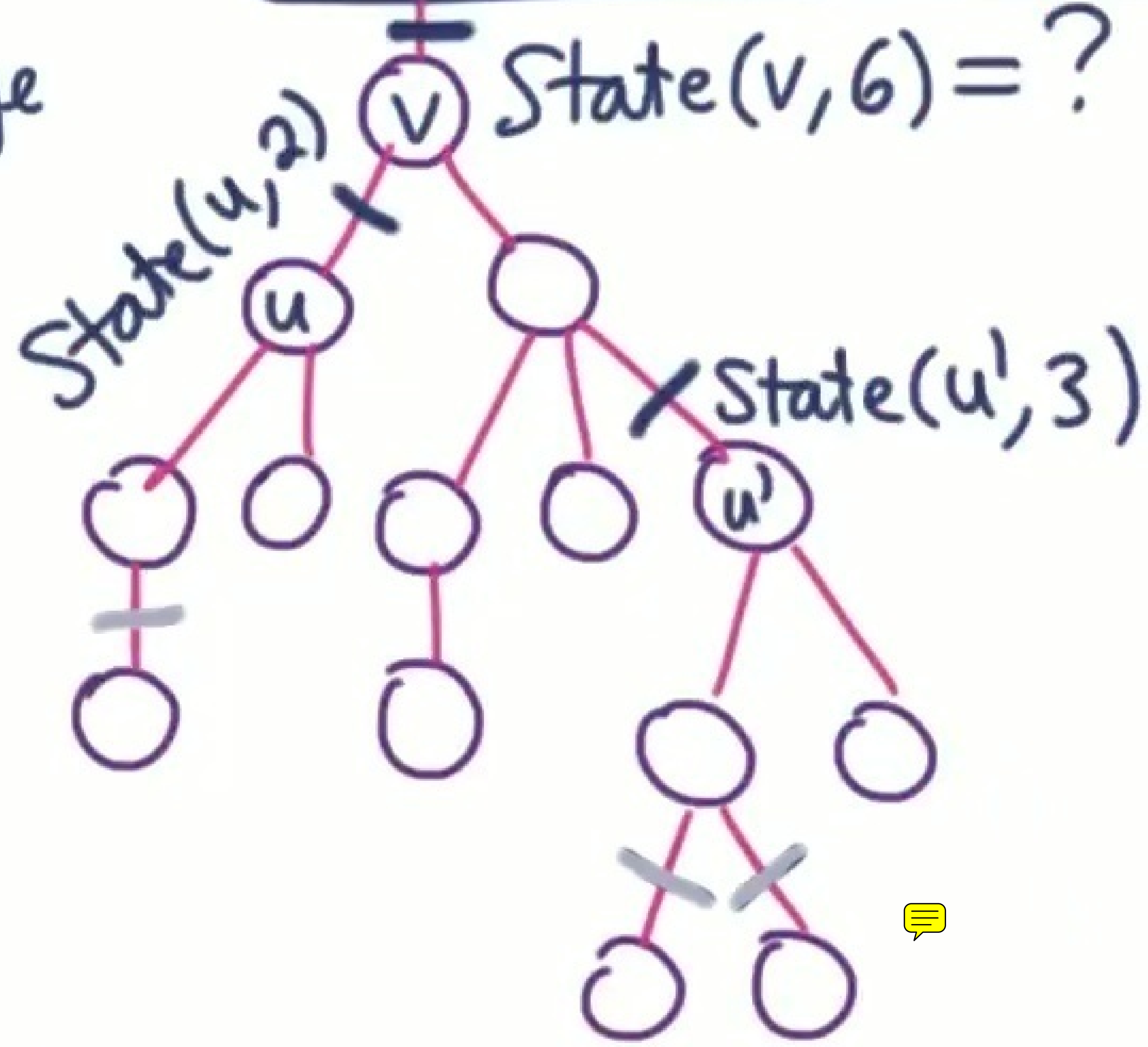
(k-1)-respecting tree

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
→ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

K-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \leq k-1)$



Intuition: if delete v's parent, then v's subtree is independent instance

Computing State(v,s):
- focus on "maximal" edges
- for each u, ℓ: node (u, ℓ) with node weight State(u, ℓ)

(k-1)-respecting tree

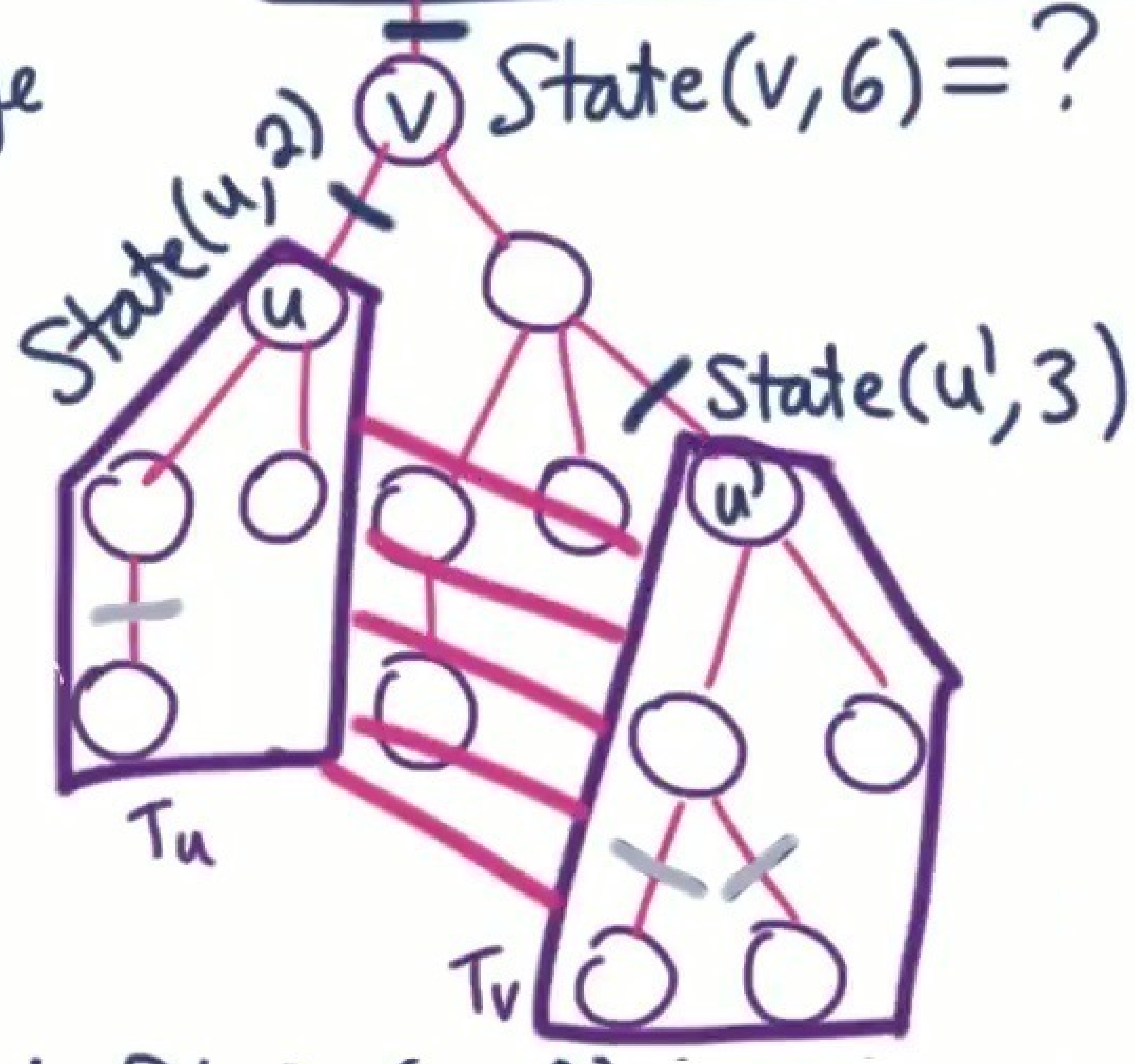
Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • (k-1)-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut (k-1) edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with (s-1) edges in v's subtree $(s \leq k-1)$

Intuition: if delete v's parent, then v's subtree is independent instance



Computing State(v,s):
- focus on "maximal" edges
- for each u, ℓ: node (u, ℓ) with node weight State(u, ℓ)
- Edge weight between (u, ℓ), (v, ℓ'): $-w(E[T_u, T_v])$

$(k-1)$-respecting tree

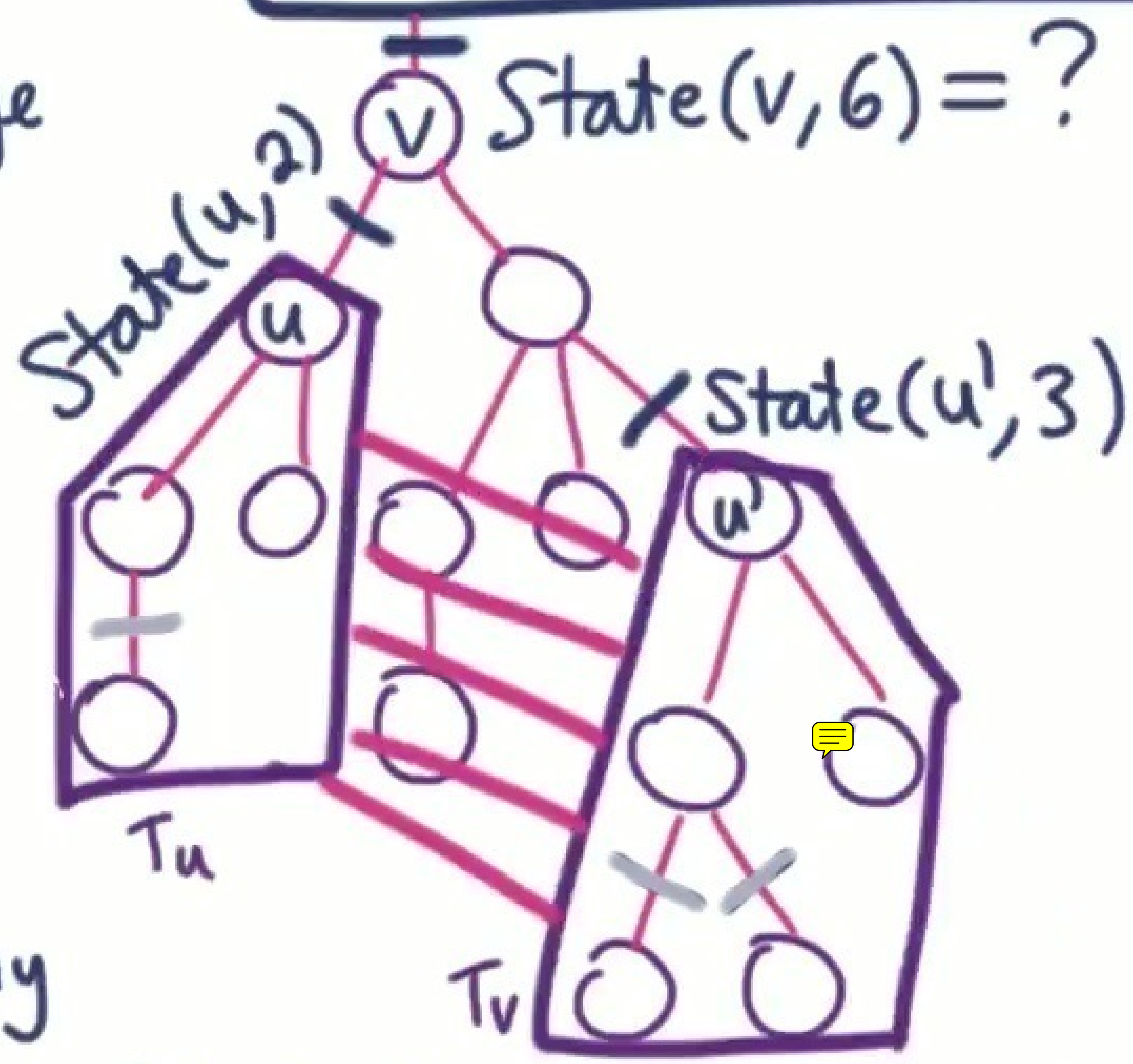
Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to $(k-1)$-resp.
$\rightarrow$ k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Medium case: • $(k-1)$-respects tree, still want $n^{(\omega/3)k}$

k-cut: cut $(k-1)$ edges of tree to minimize k-cut in G of the k connected components

DP: State(v, s): best way to delete parent edge of v, along with $(s-1)$ edges in v's subtree $(s \leq k-1)$

Intuition: if delete v's parent, then v's subtree is independent instance



Computing State(v,s):

- focus on "maximal" edges, suppose r many
- for each u, ℓ: node (u,ℓ) with node weight State(u, ℓ)
- Edge weight between (u,ℓ), (v,ℓ'): $-w(E[T_u, T_v])$
- Find min node+edge weight r-clique $(u_1,\ell_1)\ldots(u_r,\ell_r)$ s.t. $\begin{cases} u_i\text{'s are incomparable} \\ \sum \ell_i = s-1 \end{cases}$

$2k-2 \rightarrow (k-1)$-respecting

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

$\rightarrow$ Reduction to $(k-1)$-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Thm: Given a $(2k-2)$-respecting tree, can compute $f(k)n^k$ many trees s.t. one of them is 1-respecting, w.h.p.

$2k-2 \to (k-1)$-respecting

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

$\to$ Reduction to $(k-1)$-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Thm: Given a $(2k-2)$-respecting tree, can compute $f(k)n^k$ many trees s.t. one of them is 1-respecting, w.h.p.

- poly(n) trees, one of which is $(2k-2)$-resp.

$\Rightarrow f(k)n^{k+o(1)}$ trees, one of which is $(k-1)$-resp.

$2k-2 \rightarrow (k-1)$-respecting

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
$\rightarrow$ Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Thm: Given a $(2k-2)$-respecting tree, can compute $f(k)n^k$ many trees s.t. one of them is 1-respecting, w.h.p.

- poly(n) trees, one of which is $(2k-2)$-resp.
  $\Rightarrow f(k)n^{k+o(1)}$ trees, one of which is $(k-1)$-resp.
- $n^{(\omega/3)k}$ time per tree, $f(k)n^{(1+\omega/3)k+o(1)}$ total

$(1+\varepsilon)$-approx k-cut

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Thm: Given (k-1)-respecting tree, can find $(1+\varepsilon)$-apx k-cut in $f(k,\varepsilon)\,\mathrm{poly}(n)$ (FPT) time

$(1+\varepsilon)$-approx k-cut

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Thm: Given $(k-1)$-respecting tree, can find $(1+\varepsilon)$-apx k-cut in $f(k,\varepsilon)\,\text{poly}(n)$ (FPT) time

- Exact is impossible in FPT (W[1]-hard)

$(1+\varepsilon)$-approx k-cut

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to $(k-1)$-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

Thm: Given $(k-1)$-respecting tree, can find $(1+\varepsilon)$-apx k-cut in $f(k,\varepsilon)\,\mathrm{poly}(n)$ (FPT) time

- Exact is impossible in FPT (W[1]-hard)
- Uses approximation algo techniques (ε-nets) combined with FPT techniques (color-coding)

Apply thm on each of $f(k)n^{k+O(1)}$ trees

$\Longrightarrow$ $(1+\varepsilon)$-apx in $f(k,\varepsilon)\,n^{k+O(1)}$ time

Open problems

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Open problems

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

- Faster exact algo? $\begin{cases} \text{upper bound } n^{(2\omega/3)k} \\ \text{lower bound } n^{(\omega/3)k} \end{cases}$

  "fine-grained complexity" of k-cut?

Open problems

- Faster exact algo?
  - upper bound $n^{(2\omega/3)k}$
  - lower bound $n^{(\omega/3)k}$

  "fine-grained complexity" of k-cut?

- Faster combinatorial exact algo?

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

Open problems

Exact $n^{(1+\omega/3)k}$

Thorup's tree packing

Reduction to (k-1)-resp.

k-clique-like mtx.mult.

Hardness $n^{(\omega/3)k}$

- Faster exact algo? $\begin{cases} \text{upper bound } n^{(2\omega/3)k} \\ \text{lower bound } n^{(\omega/3)k} \end{cases}$

  "fine-grained complexity" of k-cut?

- Faster <u>combinatorial</u> exact algo?
  - [GLL'18, unpublished] $f(k)\, n^{1.99k}$ time

    $\Longrightarrow f(k)\, n^{1.99k}$ extremal # min k-cuts

- Lower bound n^k (combinatorial k-clique)

Open problems

Exact $n^{(1+\omega/3)k}$
Thorup's tree packing
Reduction to (k-1)-resp.
k-clique-like mtx.mult.
Hardness $n^{(\omega/3)k}$

- Faster exact algo? $\begin{cases} \text{upper bound } n^{(2\omega/3)k} \\ \text{lower bound } n^{(\omega/3)k} \end{cases}$

  "fine-grained complexity" of k-cut?

- Faster combinatorial exact algo?
  - [GLL'18, unpublished] $f(k)\,n^{1.99k}$ time

    $\Longrightarrow f(k)\,n^{1.99k}$ extremal # min k-cuts
  - Lower bound n^k (combinatorial k-clique)
- Better approximation?
  - $(1+\varepsilon)$-apx in $f(k,\varepsilon)\,\mathrm{poly}(n)$ time?