Advanced Kernelization

Parameterized Complexity Summer School

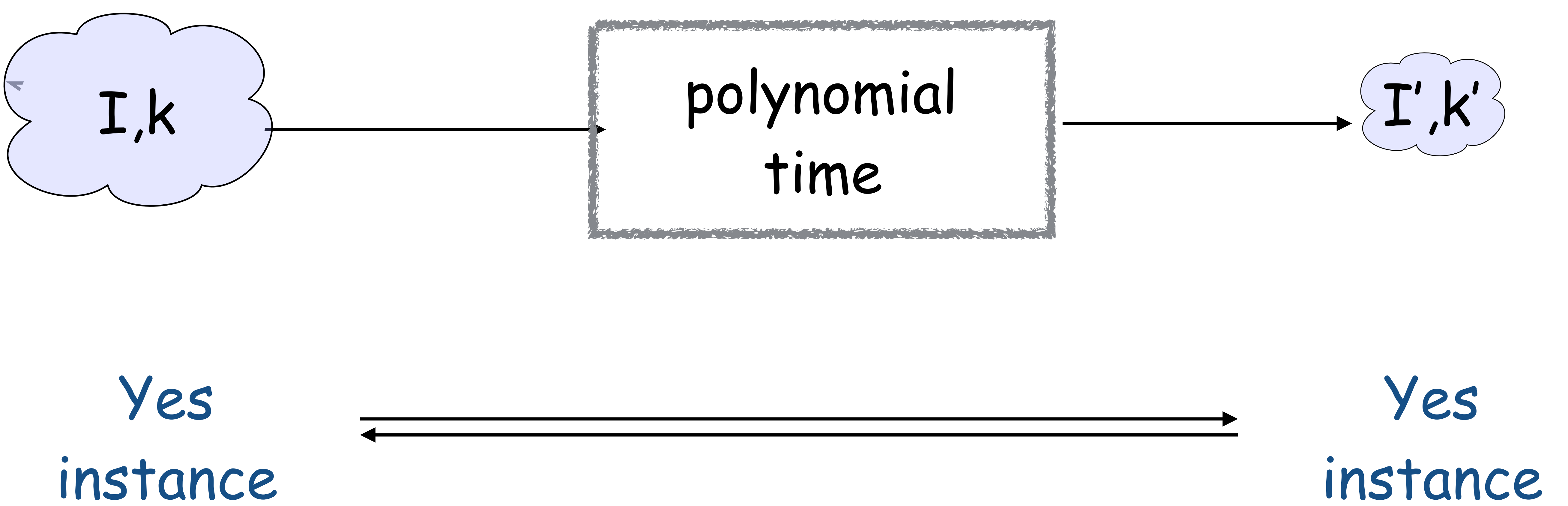
ALGO 2017

Vienna

M. S. Ramanujan

TU Wien

Kernelization



Polynomial Kernel if $|I'| + k' < poly(k)$

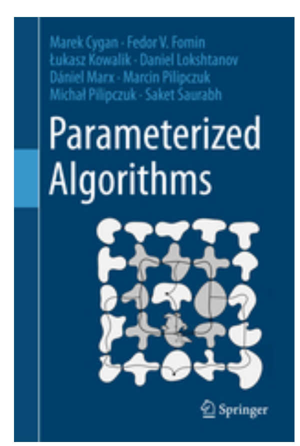
Feedback Vertex Set

Input: Graph G, integer k

Question: Is there a feedback vertex set of size at most k in G?

A subset of vertices S such that G-S is a forest (acyclic)

CHAPTER 9 of



Free copy available online. Really!

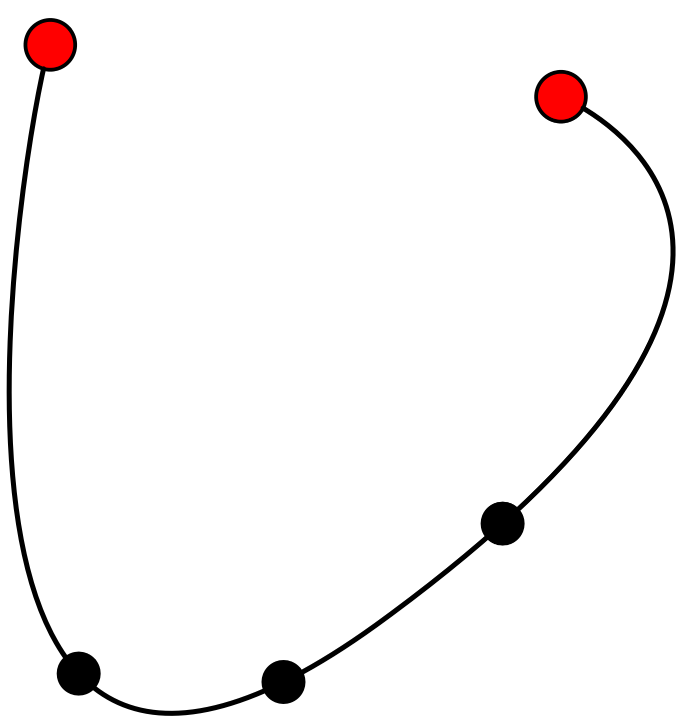
Feedback Vertex Set

Input: Graph G, integer k

Question: Is there a feedback vertex set of size at most k in G?

1. Delete a vertex of degree 1 because it is in no cycle

2. If



then

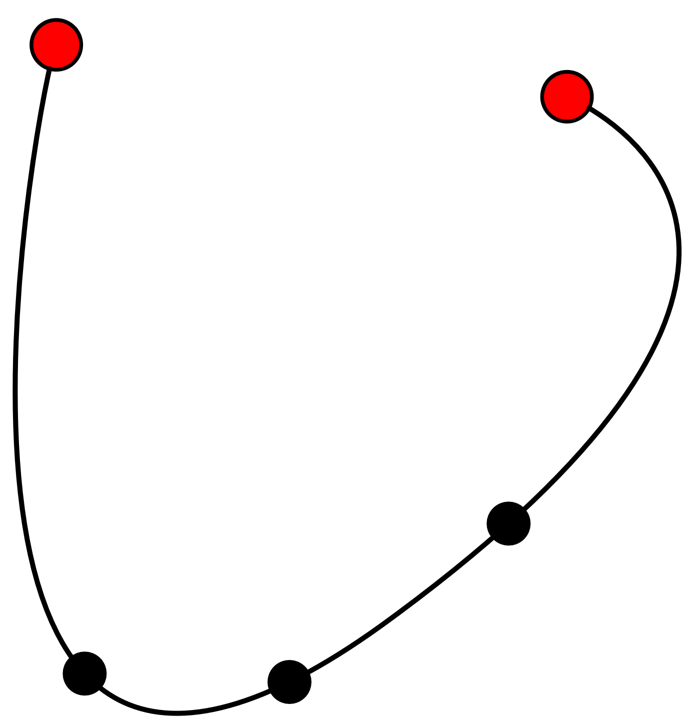
Feedback Vertex Set

Input: Graph G, integer k

Question: Is there a feedback vertex set of size at most k in G?

1. Delete a vertex of degree 1 because it is in no cycle

2. If



then

Now, Minimum Degree of graph is >=3

Feedback Vertex Set

Feedback Vertex Set

G,k

G',k'

G' has max degree d

Yes instance → Yes instance

Yes instance ← Yes instance

EXERCISE:

Since G' has min degree at least 3 and max degree at most d, $V(G') <= 2(dk + (k-1) + k^2)$

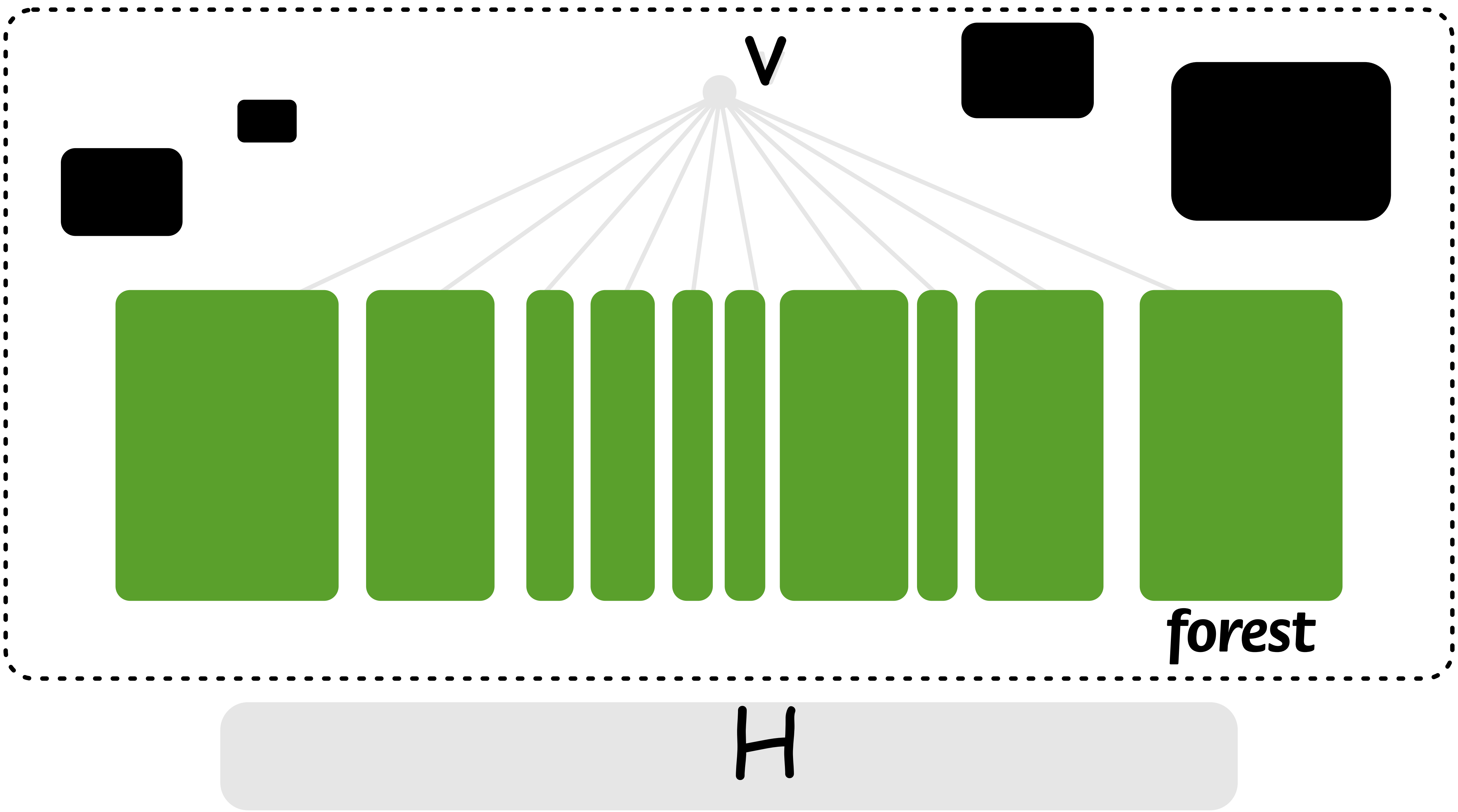
Feedback Vertex Set

If $d=O(k)$, then $V(G')=O(k^2)$

EXERCISE:

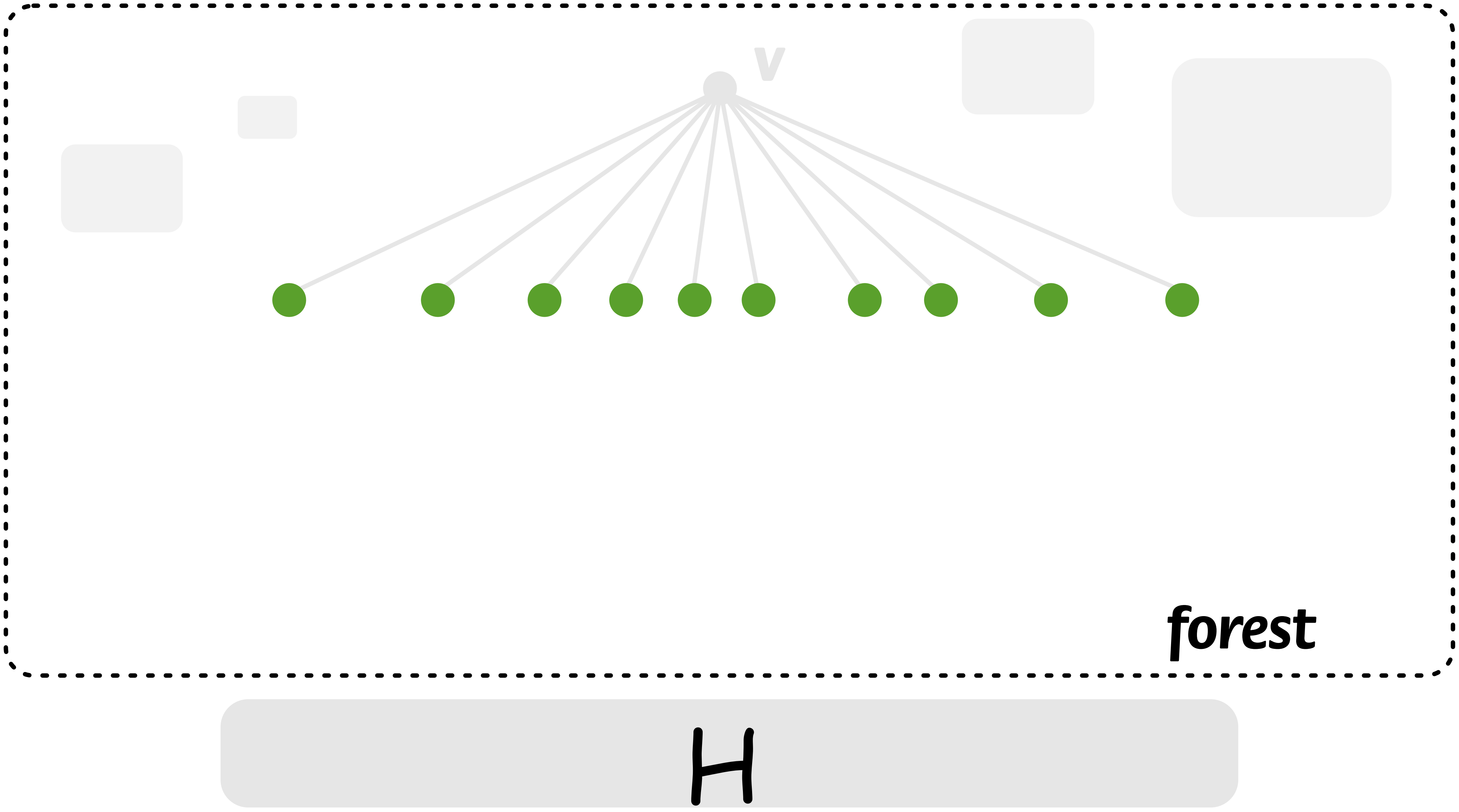
Since G' has min degree at least 3 and max degree at most d, $V(G') \le 2(dk + (k-1) + k^2)$

We can reach here using Gallai's theorem [see Chapter 9]

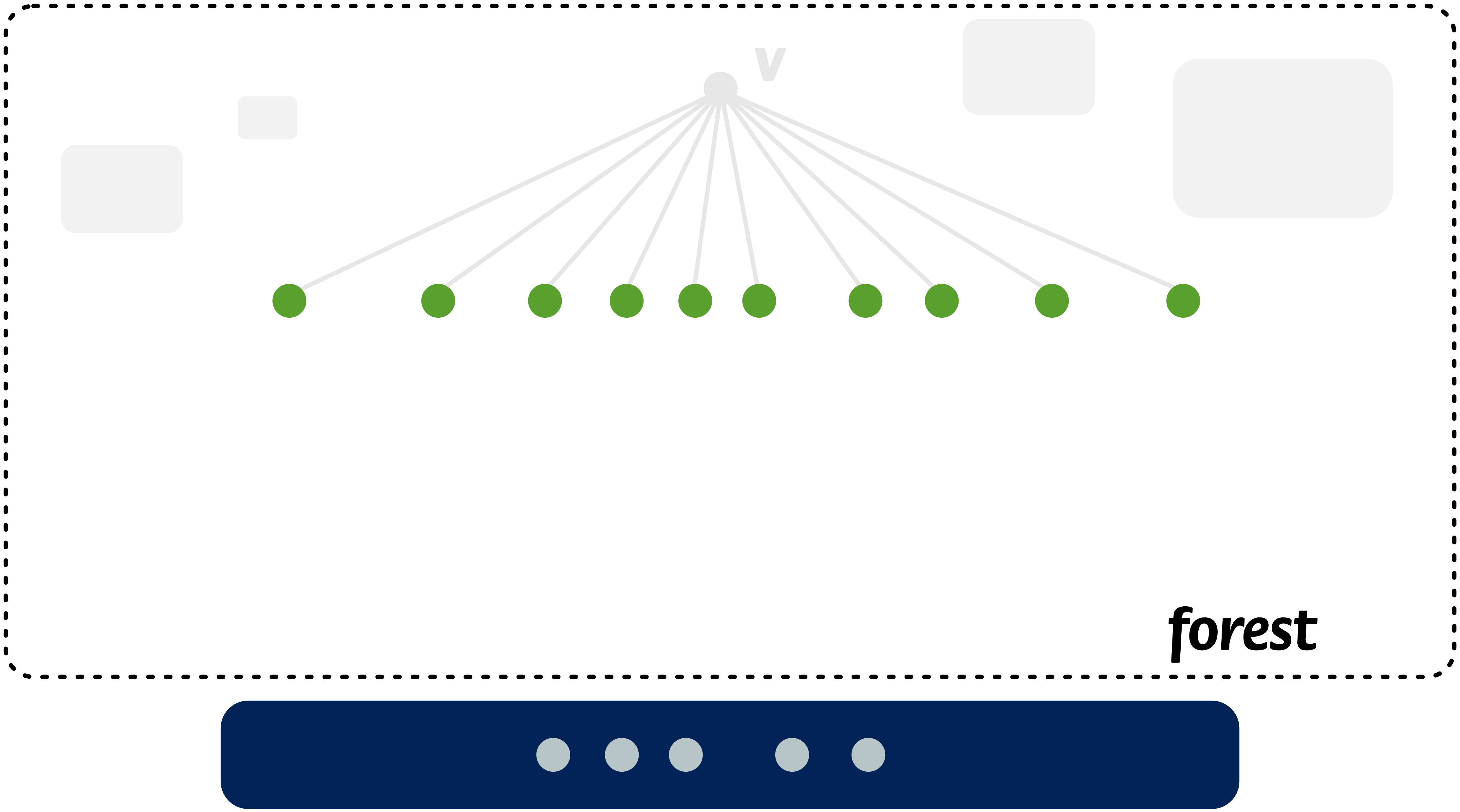


set H of size 2k which intersects all cycles through v

assuming v is not the centre of a k+1-flower

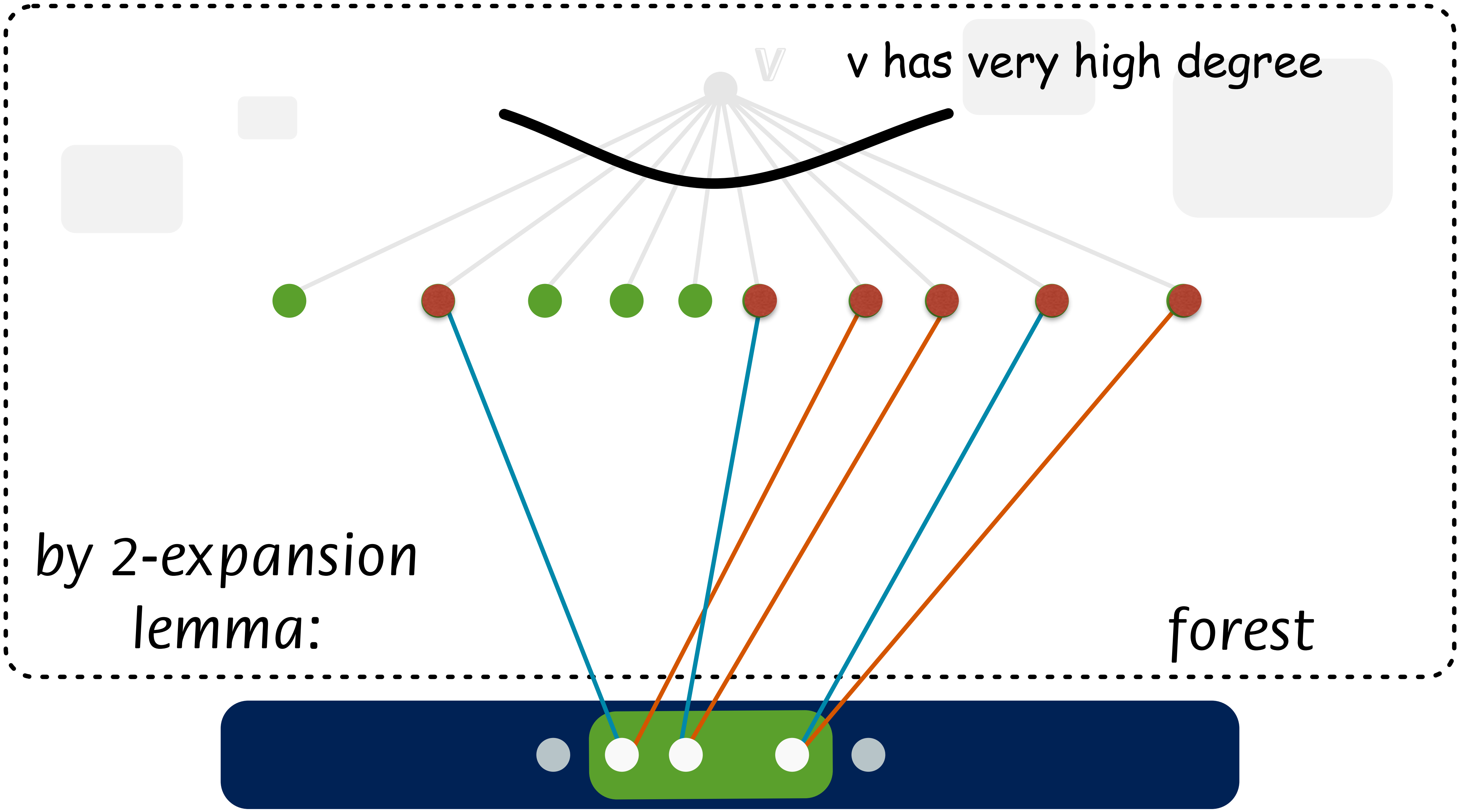


set H of size 2k which intersects all cycles through v



set H of size 2k which intersects all cycles through v

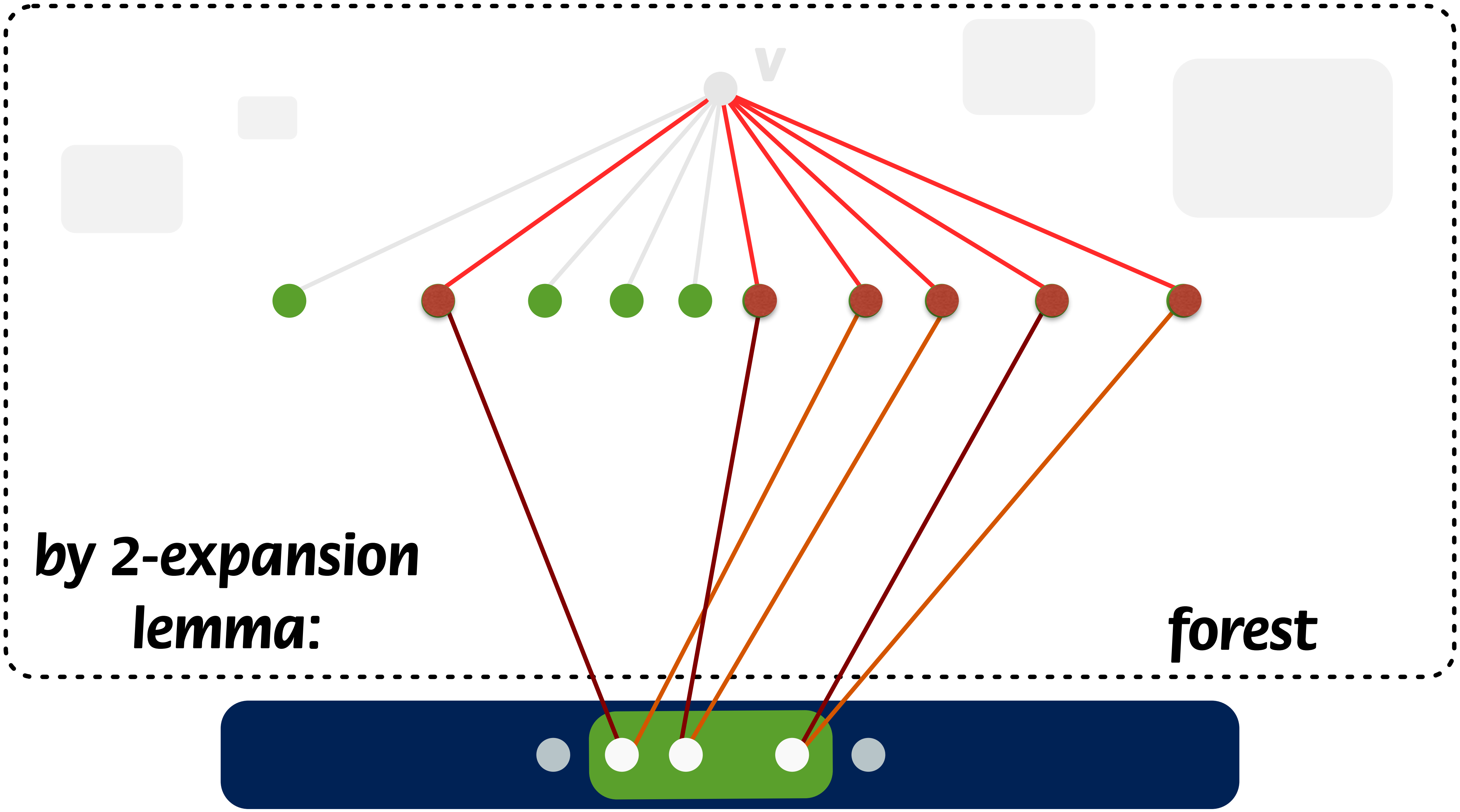
2-expansion lemma —> fancy crown decompositions



set H of size 2k which intersects all cycles through v

2-matching saturating the lower green vertices

the red vertices have ALL neighbors in the lower green vertices



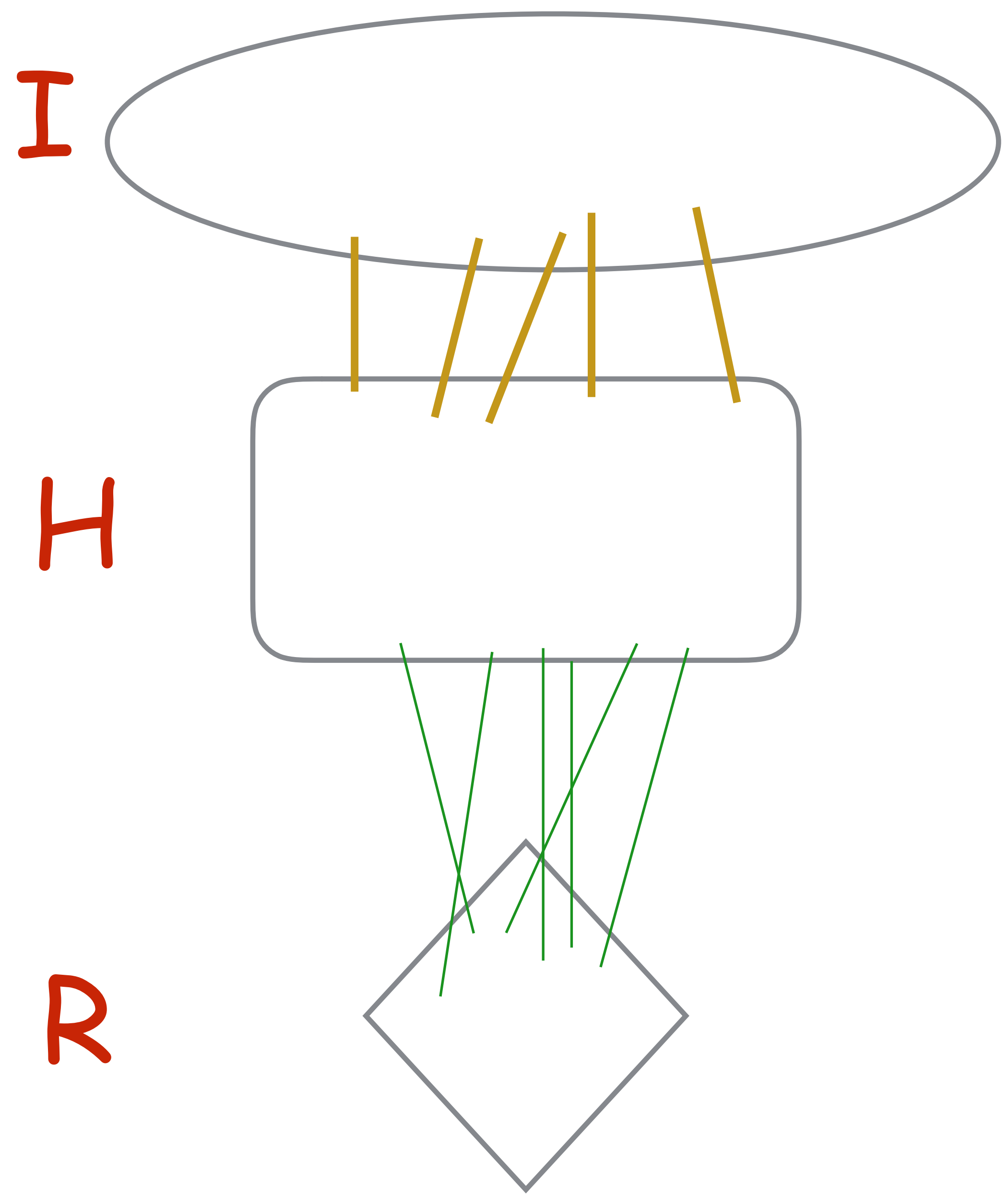
set H of size 2k which intersects all cycles through v

2-matching saturating the lower green vertices

the red vertices have ALL neighbors in the lower green vertices

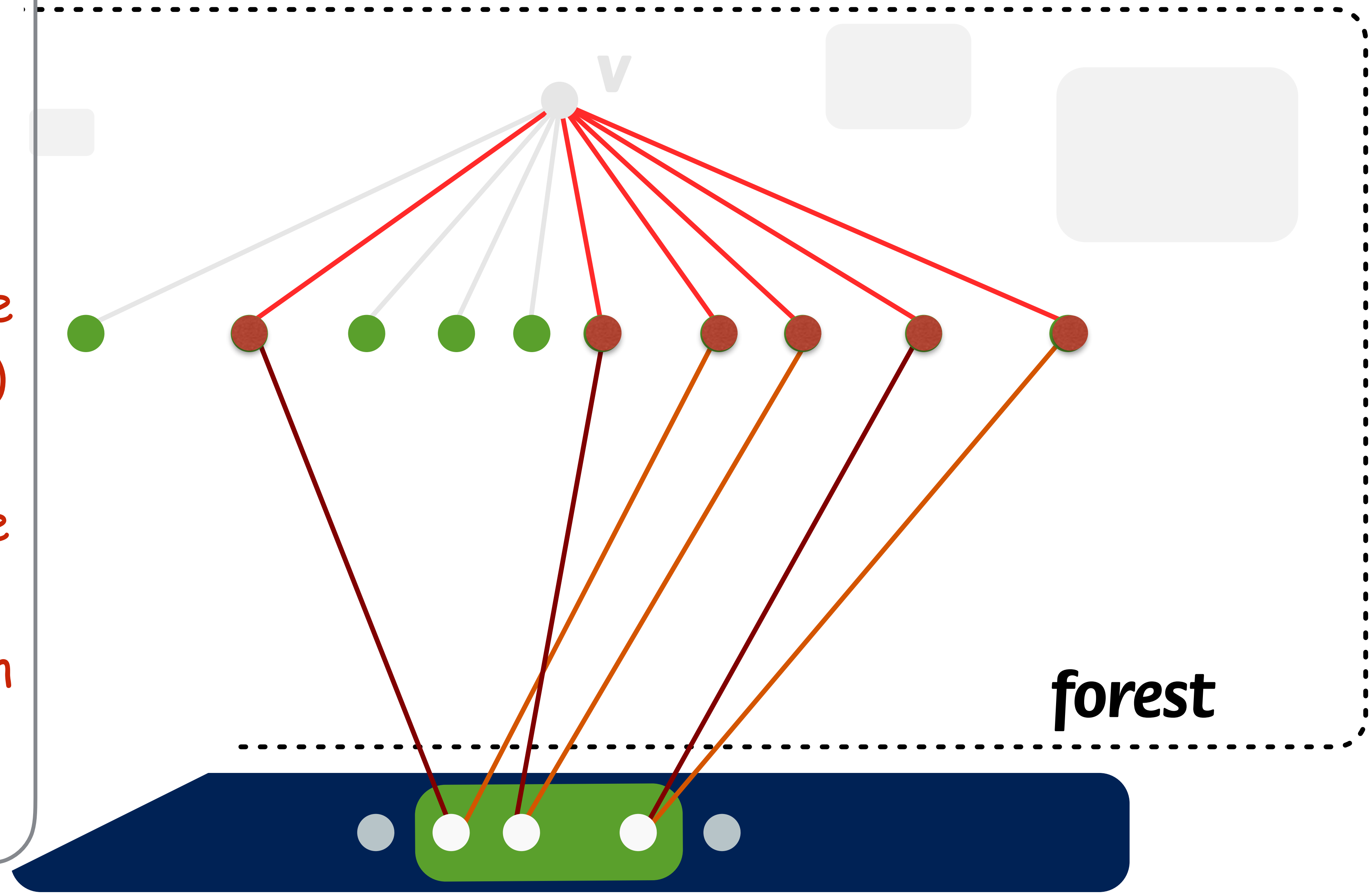
Recap: crown reductions

Here is a local part of the graph which must contribute at least 5 vertices to the solution and exactly these 5 vertices in the head suffice to cover ALL edges incident on I



Message of Crown Reduction for vertex cover

Here is a local part of the graph which must contribute at least 3 vertices to the solution and exactly these 3 vertices (OR v) in the lower green set suffice to cover ALL cycles incident on the upper green set

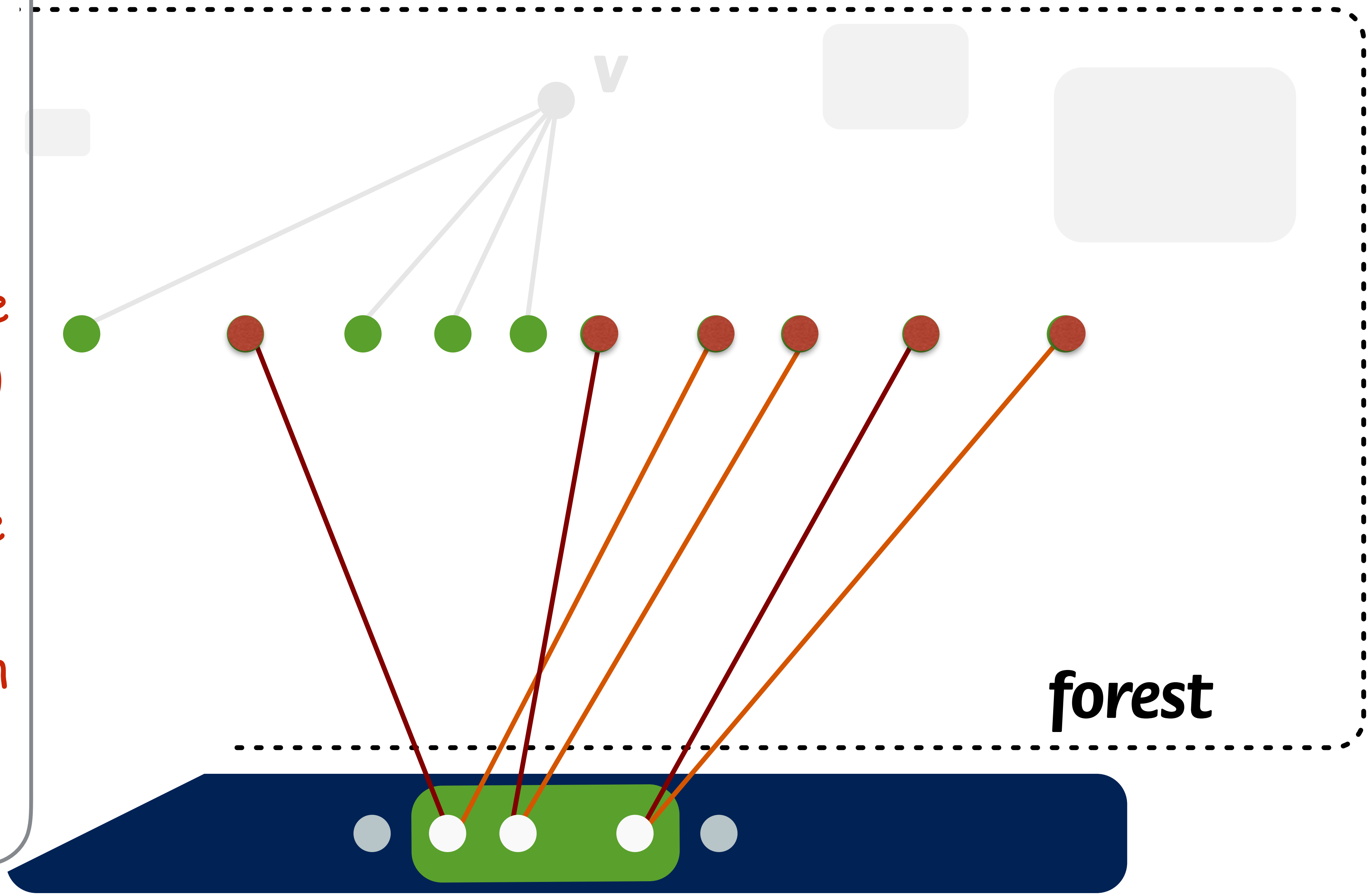


set H of size 2k which intersects all cycles through v

2-matching saturating the lower green vertices

the red vertices have ALL neighbors in the lower green vertices

Here is a local part of the graph which must contribute at least 3 vertices to the solution and exactly these 3 vertices (OR v) in the lower green set suffice to cover ALL cycles incident on the upper green set

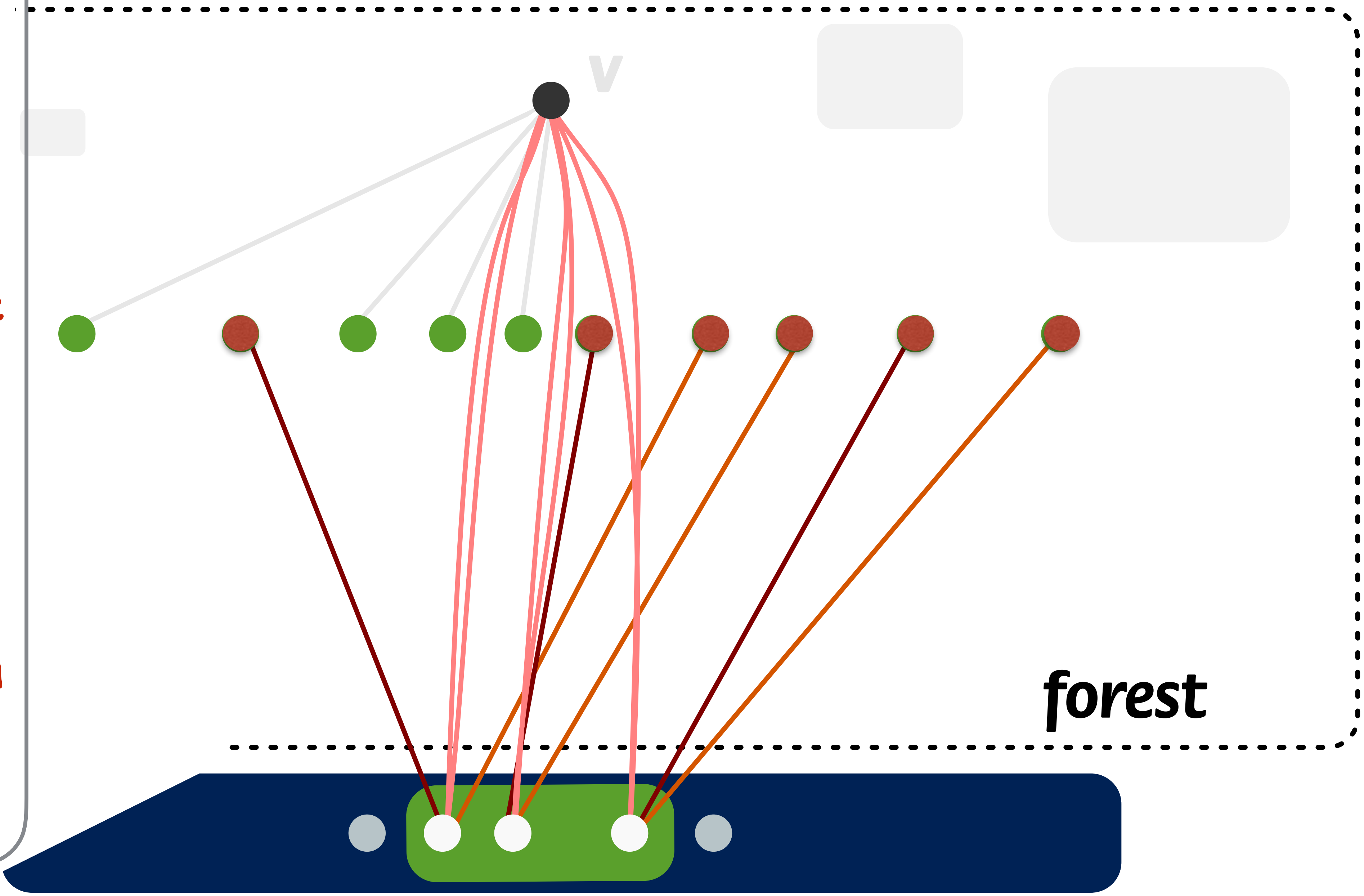


set H of size 2k which intersects all cycles through v

2-matching saturating the lower green vertices

the red vertices have ALL neighbors in the lower green vertices

Here is a local part of the graph which must contribute at least 3 vertices to the solution and exactly these 3 vertices (OR v) in the lower green set suffice to cover ALL cycles incident on the upper green set



set H of size 2k which intersects all cycles through v

2-matching saturating the lower green vertices

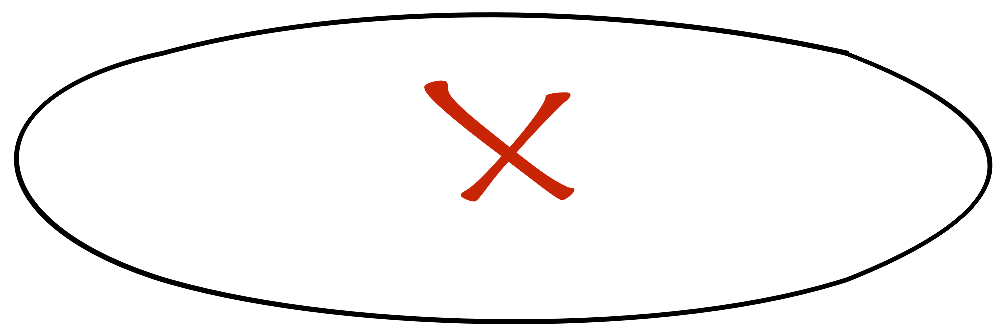
the red vertices have ALL neighbors in the lower green vertices

Feedback Vertex Set

We just took a quick peek at the black box which guarantees that the reduced instance has max degree d=O(k)

Completes sketch of $O(k^2)$ kernel for FVS

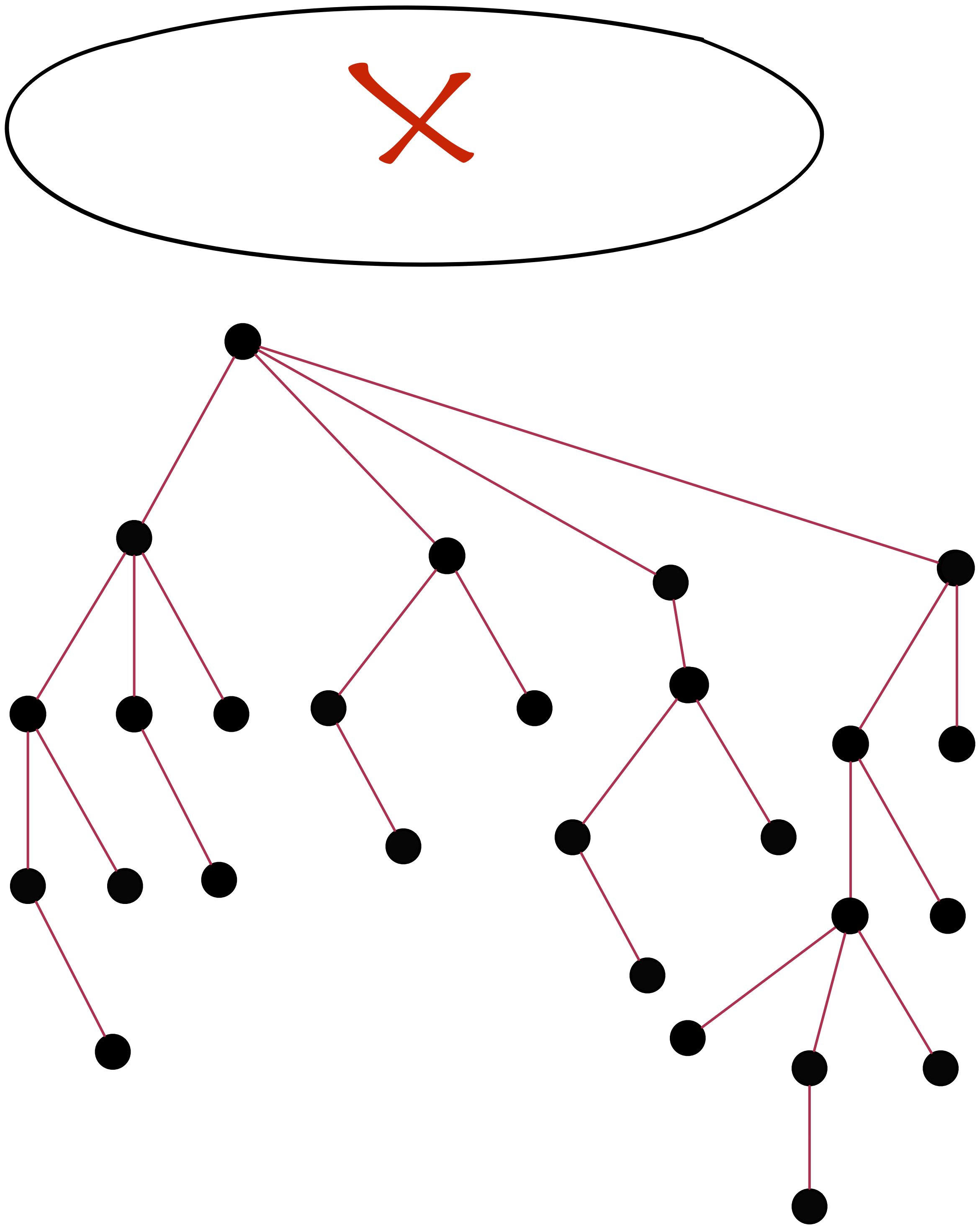
Feedback Vertex Set



Let us get a different perspective on the reduction rules that we have applied.

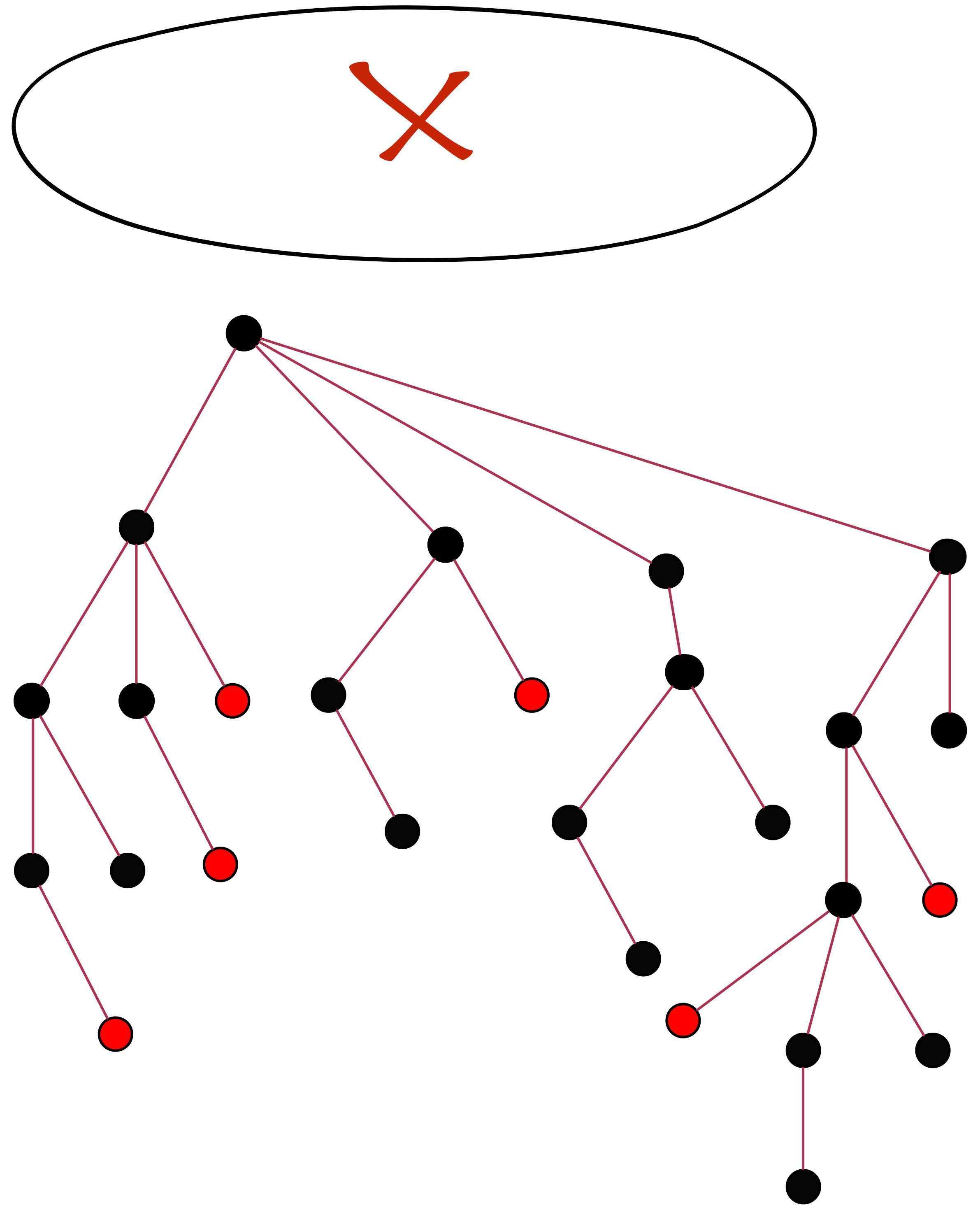
X is a hypothetical feedback vertex set of size at most k. We don't need to compute it, just need it for our arguments.

Feedback Vertex Set

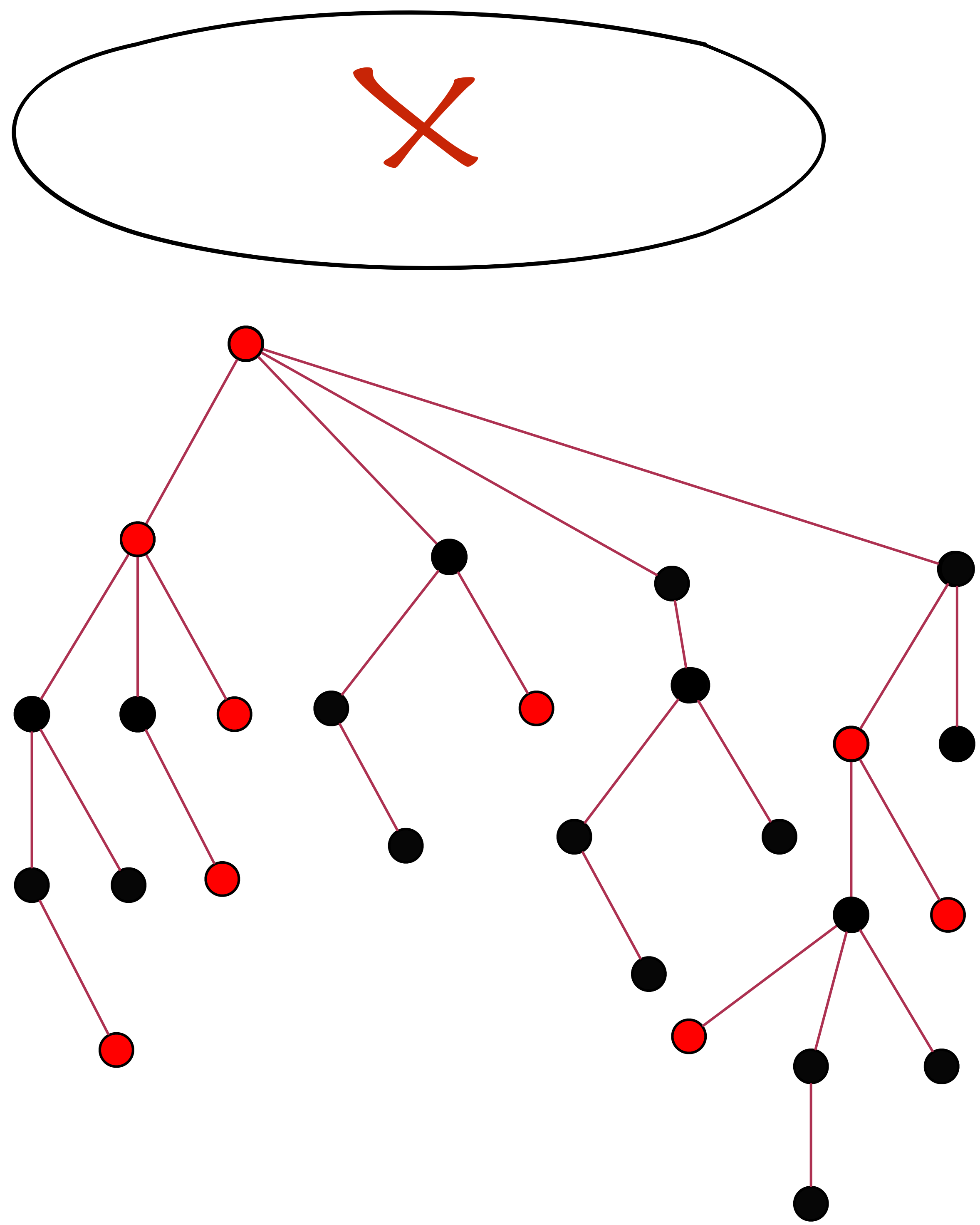


Feedback Vertex Set

Red vertices = N(X)



Feedback Vertex Set

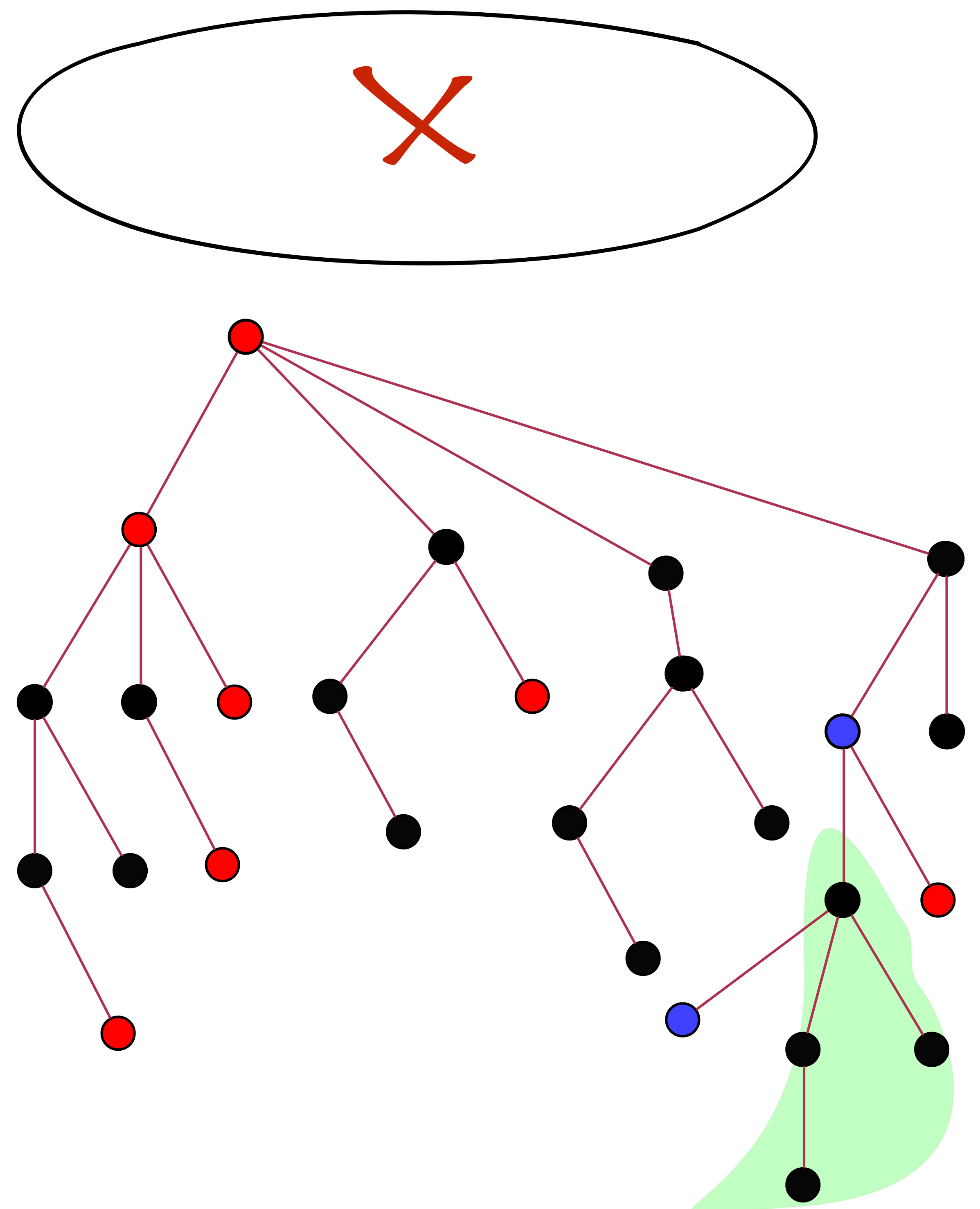


Red vertices = LCA(N(X))

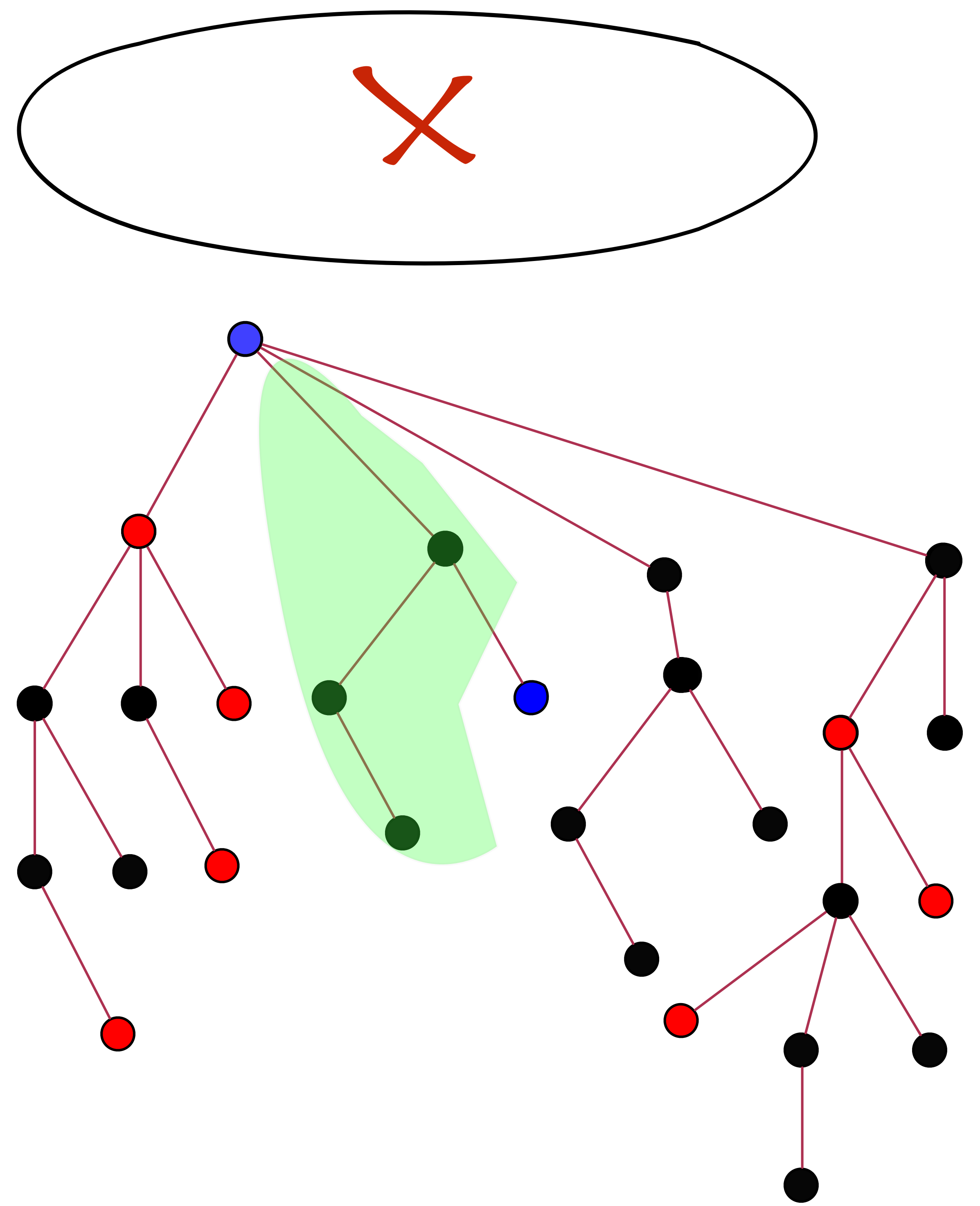
|LCA(Y)|<=2|Y|

Feedback Vertex Set

Red vertices = LCA(N(X))



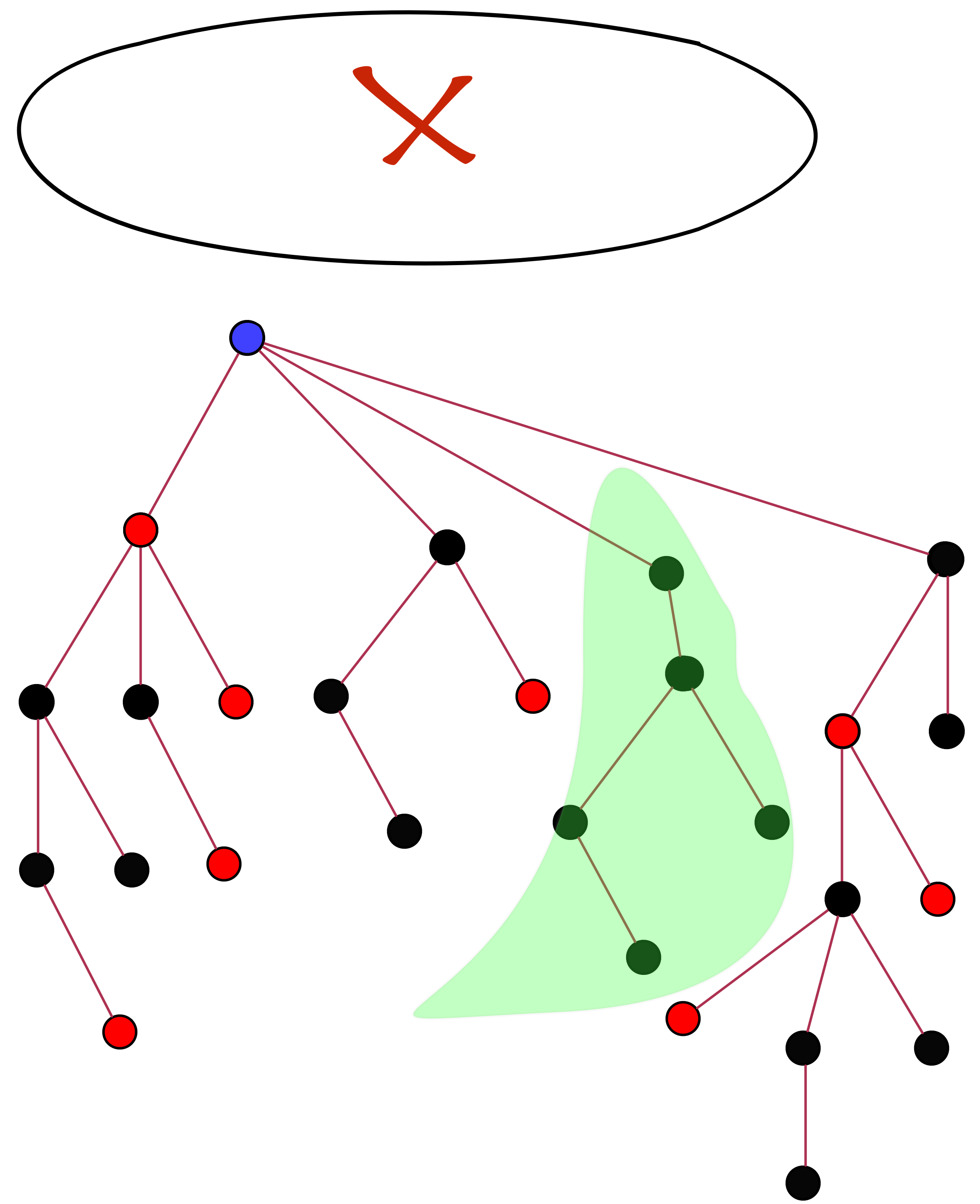
Feedback Vertex Set



Red vertices = LCA(N(X))

Feedback Vertex Set

Red vertices =
LCA(N(X))

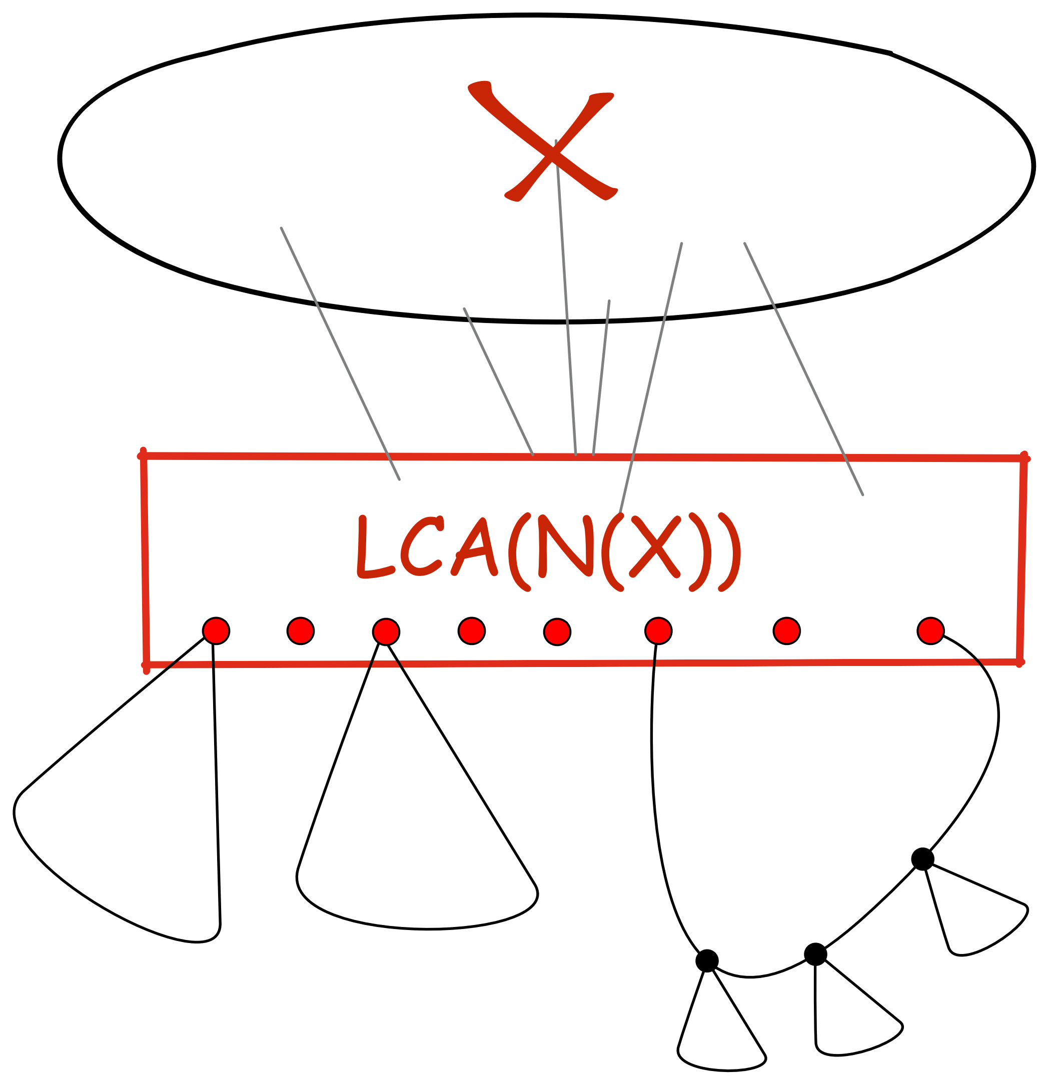


Feedback Vertex Set

Size of this set —>
$|X| = k$

max. degree (G) = d

Size of this set —>
$O(d\ |X|)=O(d\ k)$

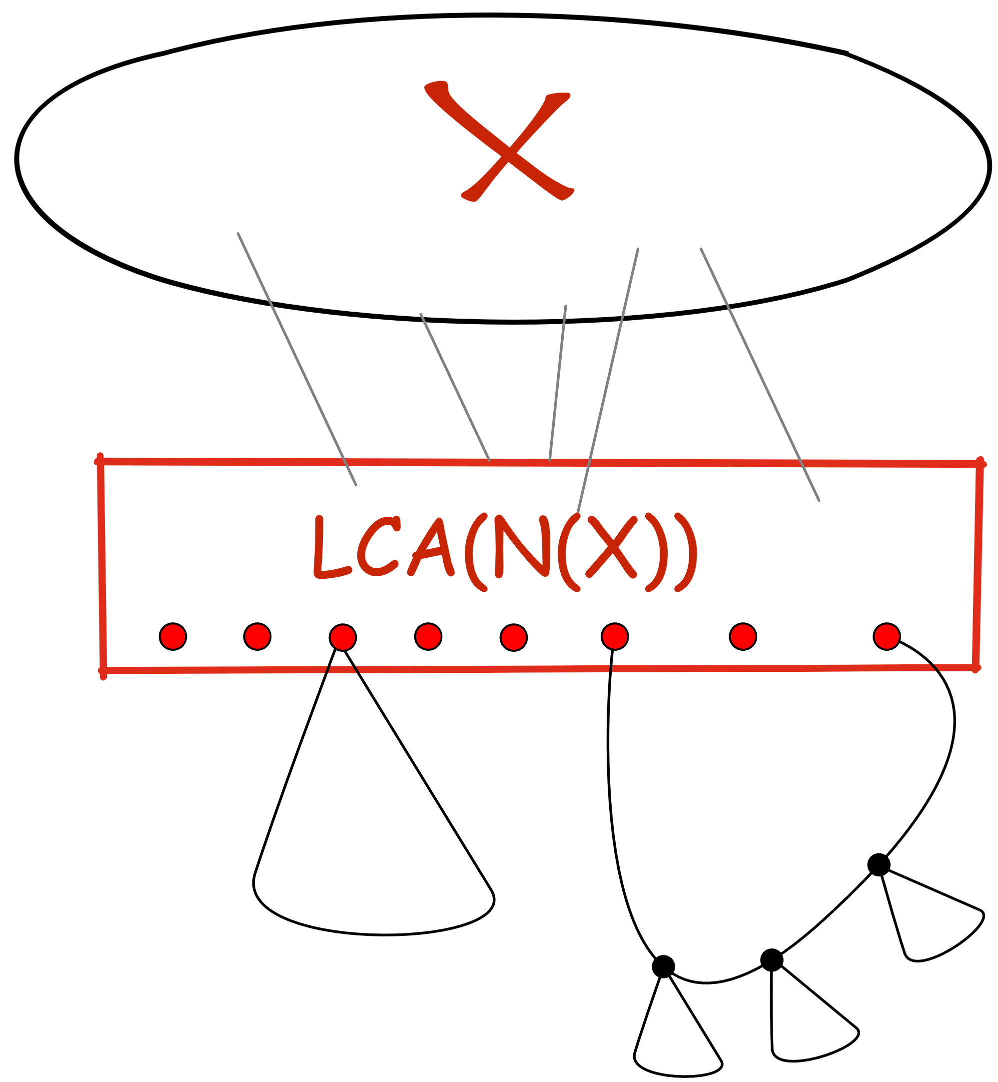


Feedback Vertex Set

Size of this set —> $|X| = k$

max. degree $(G) = d$

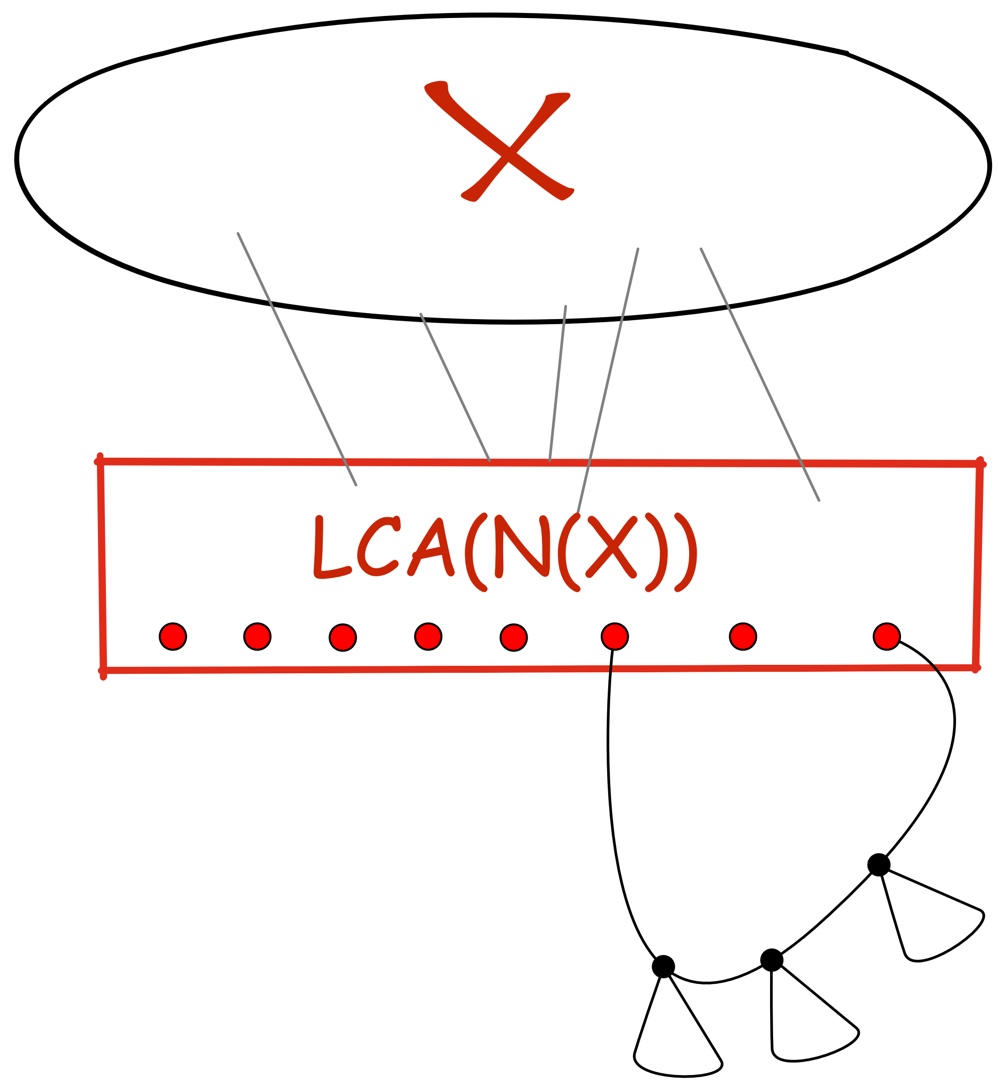
Size of this set —> $O(d\,|X|) = O(d\,k)$



Feedback Vertex Set

Size of this set —>
$|X| = k$

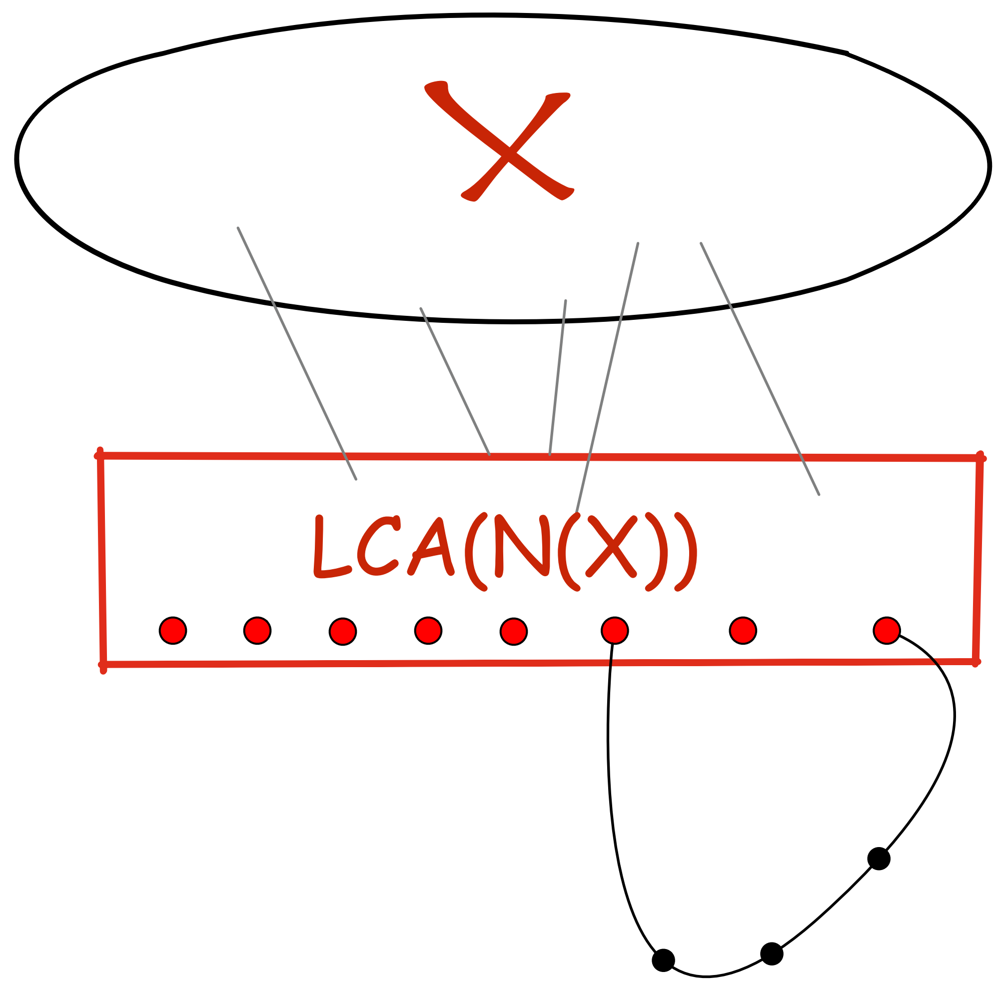
max. degree (G) = d



Size of this set —>
$O(d\,|X|)=O(d\,k)$

Feedback Vertex Set

Size of this set —>
$|X| = k$

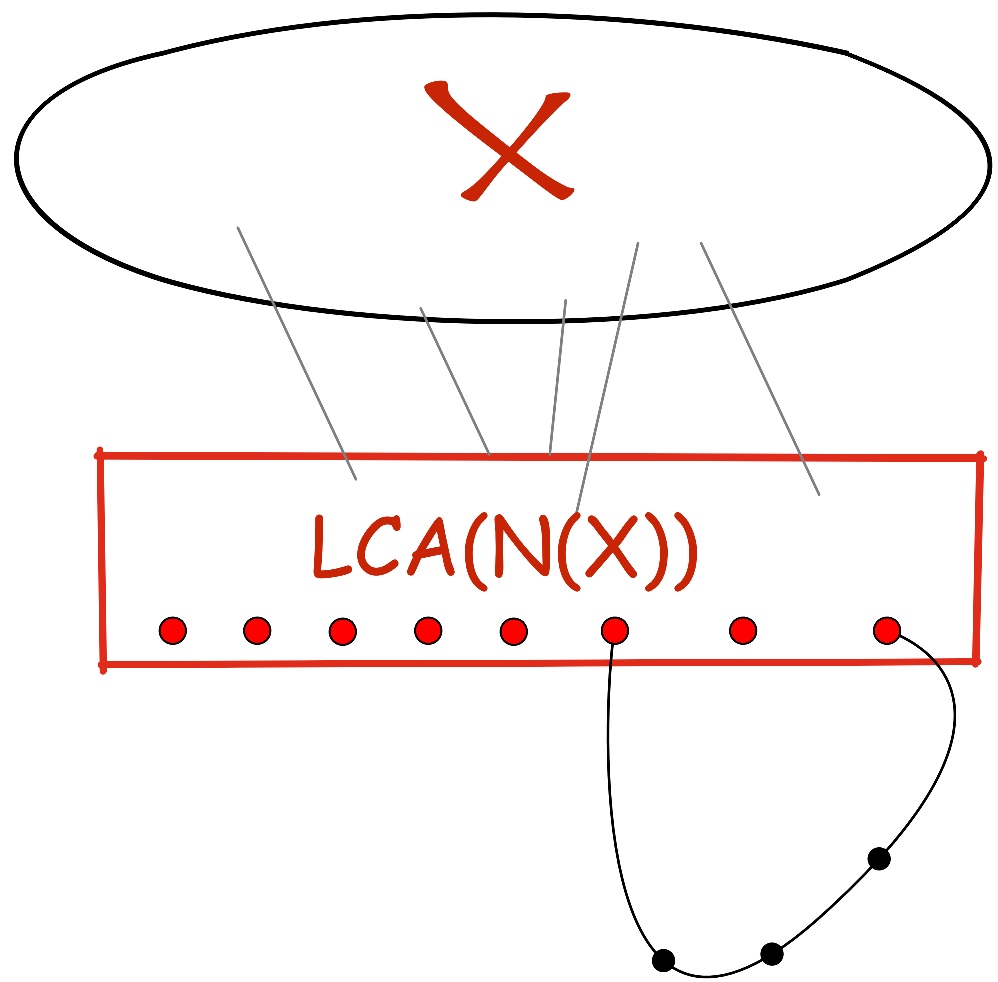


max. degree (G) = d

Size of this set —>
$O(d\ |X|) = O(d\ k)$

Feedback Vertex Set

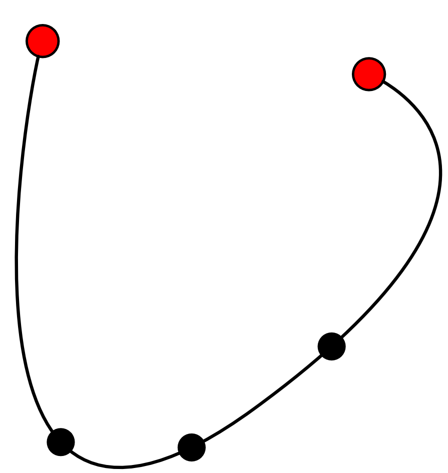
Size of this set —>
$|X| = k$



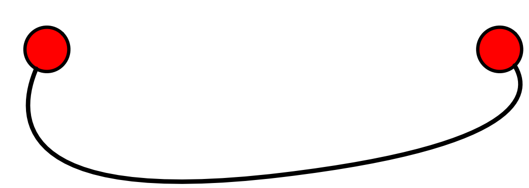
max. degree (G) = d

Size of this set —>
$O(d\ |X|) = O(d\ k)$

Rule 2. If



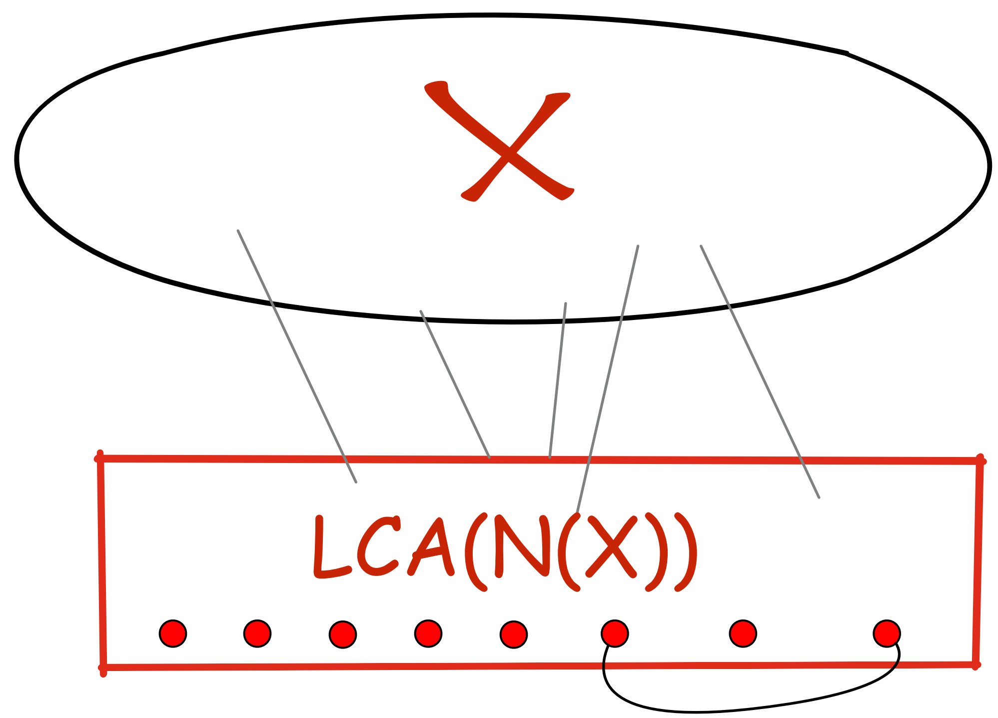
then



Feedback Vertex Set

Size of this set —>
$|X| = k$

Size of this set —>
$O(d\,|X|)=O(d\,k)$

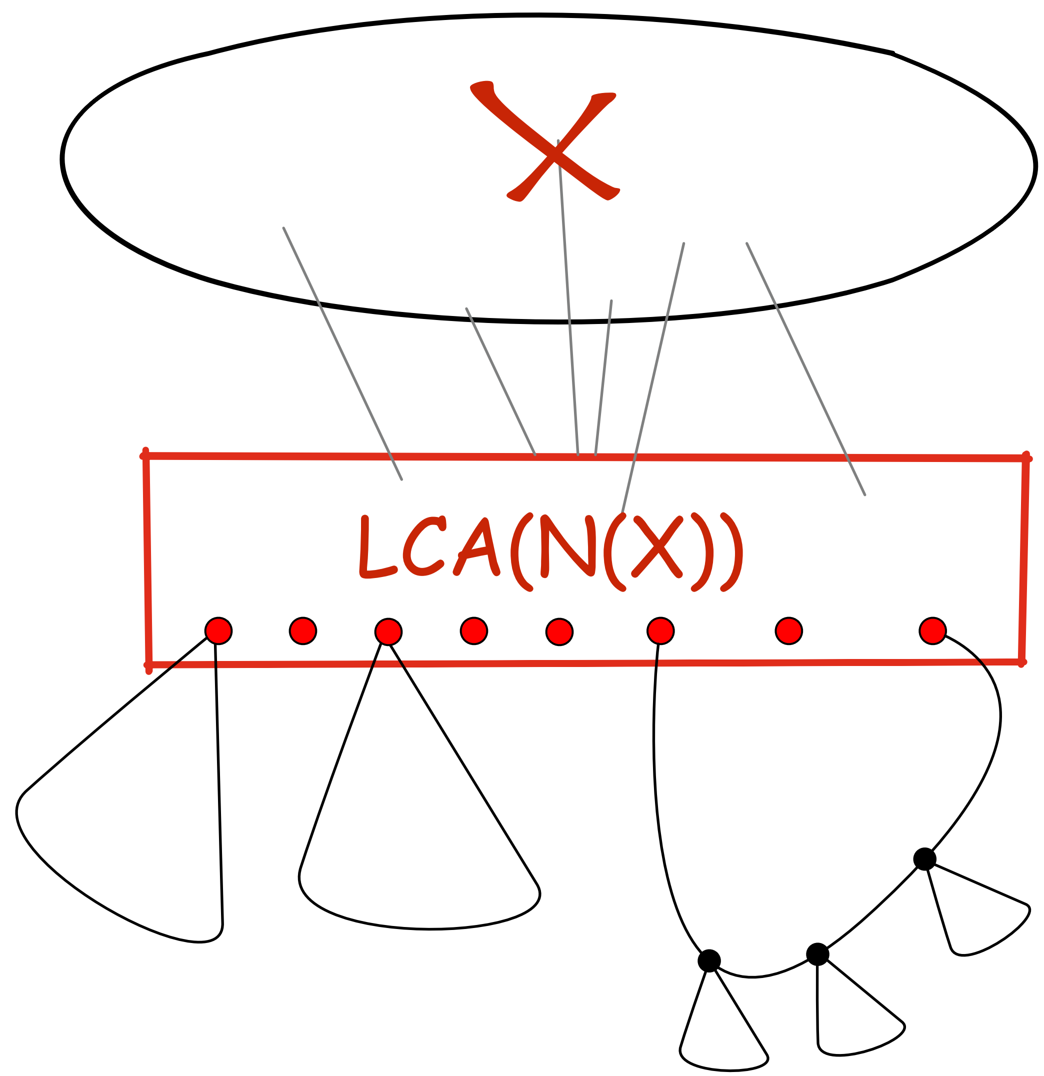


max. degree (G) = d

Feedback Vertex Set: A general perspective

Size of this set —>
$|X| = k$

max. degree (G) = d



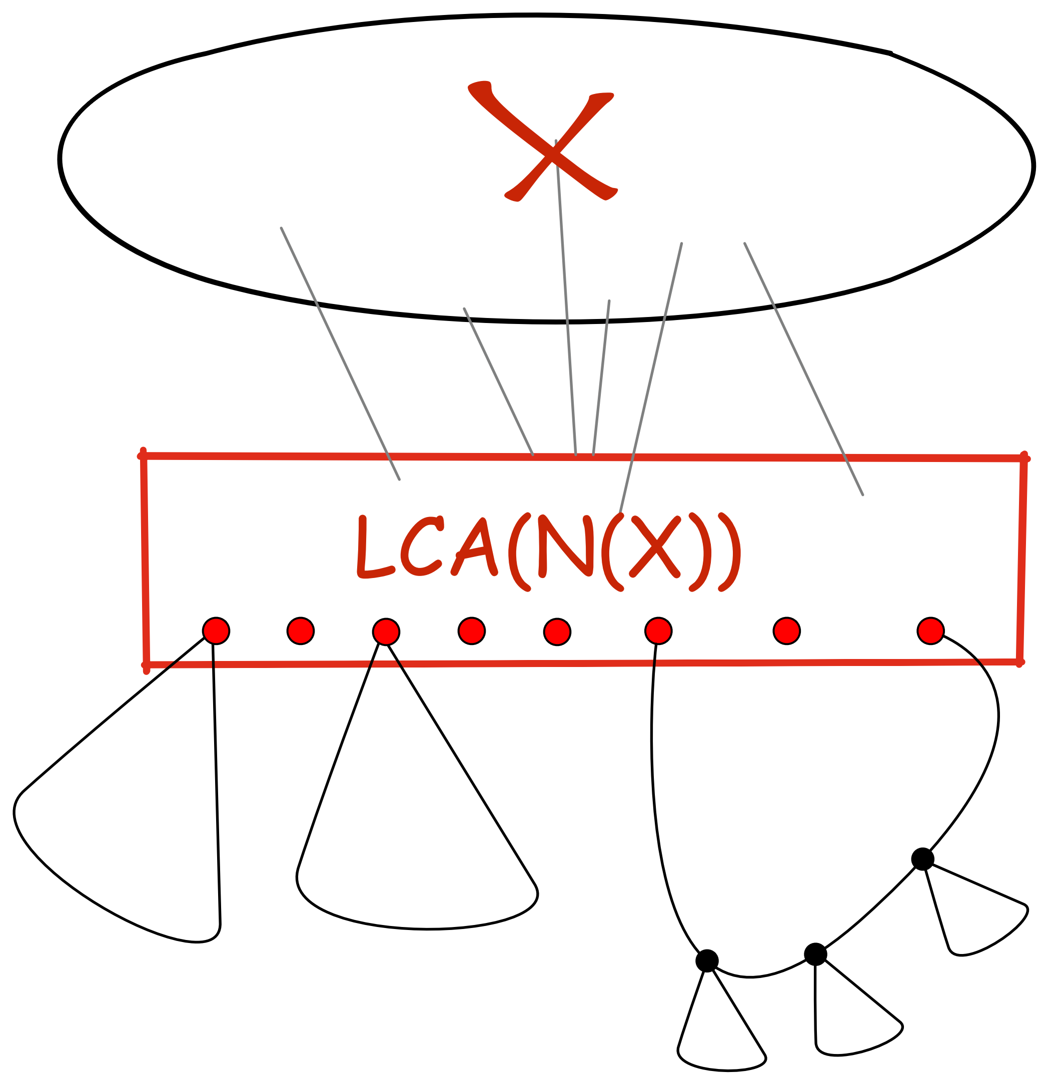
Size of this set —>
$O(d\ |X|)=O(d\ k)$

1. Easy to describe.
2. Easy to `reduce' without changing solution.

Feedback Vertex Set: A general perspective

Size of this set —>
$|X| = k$

Size of this set —>
$O(d\,|X|)=O(d\,k)$



CONSTANT Treewidth subgraph connected to the rest of the graph through a constant number of vertices.

1. Easy to describe.
2. Easy to `reduce' without changing solution.

Pumpkin Hitting Set

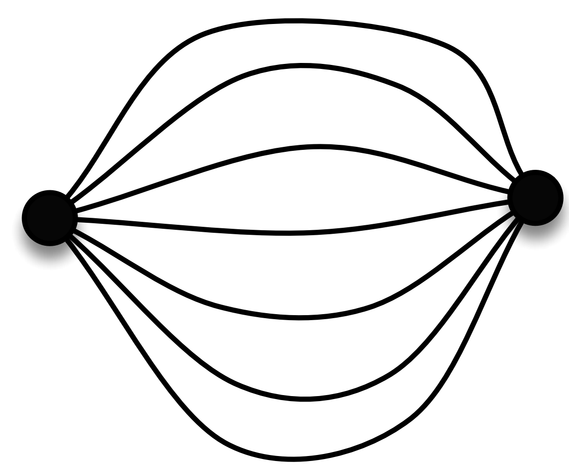
Pumpkin Hitting Set

Input: Graph G, integer k

Question: Is there a t-pumpkin hitting set of size at most k in G?

A subset of vertices S such that G-S has no t-pumpkin minors

7-



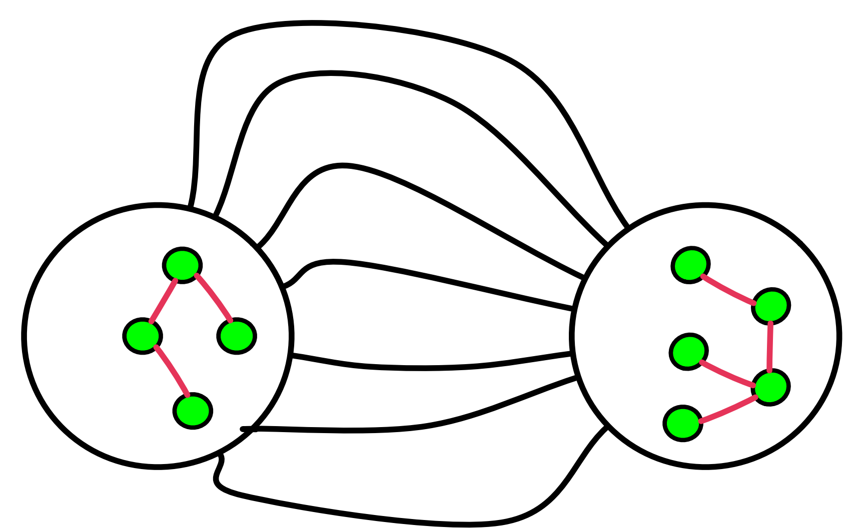
Pumpkin Hitting Set

Input: Graph G, integer k

Question: Is there a t-pumpkin hitting set of size at most k in G?

A subset of vertices S such that G-S has no t-pumpkin minors

7-



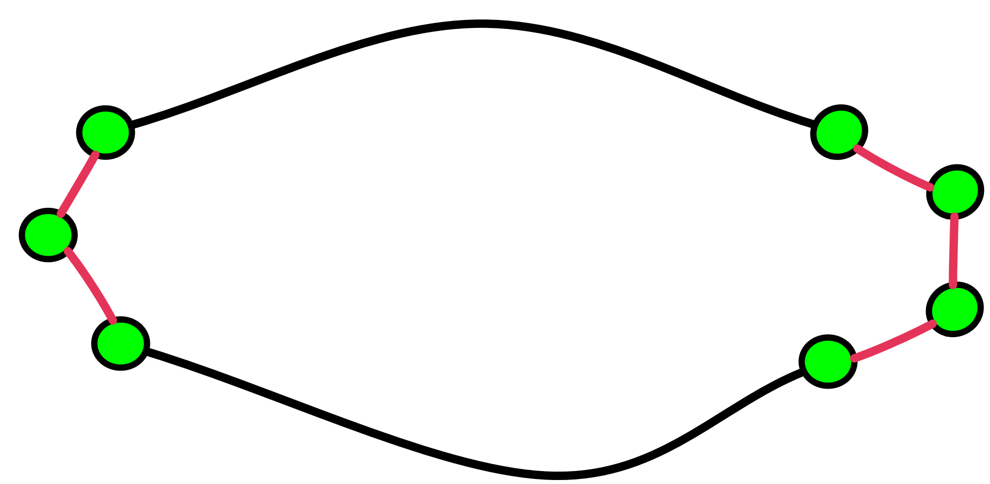
Pumpkin Hitting Set

Input: Graph G, integer k

Question: Is there a t-pumpkin hitting set of size at most k in G?

A subset of vertices S such that G-S has no t-pumpkin minors

2- 



Pumpkin Hitting Set

Input: Graph G, integer k

Question: Is there a t-pumpkin hitting set of size at most k in G?

Template

1. Design `protrusion reduction' rules.

2. If no rules apply, then bounding the degree is sufficient.

3. Design a reduction rule to bound degree.

Pumpkin Hitting Set

Input: Graph G, integer k

Question: Is there a t-pumpkin hitting set of size at most k in G?

EXERCISE: What do the protrusions for hitting 3-pumpkins look like? Find an easy description.

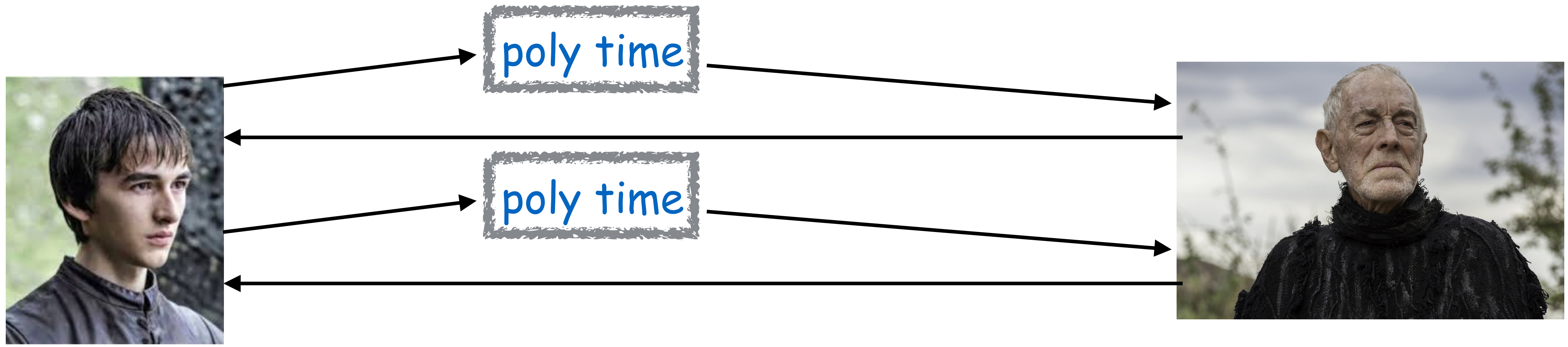
Summary so far

- q-expansion lemma : a more powerful crown type reduction.

- Finding and reducing protrusions helps.

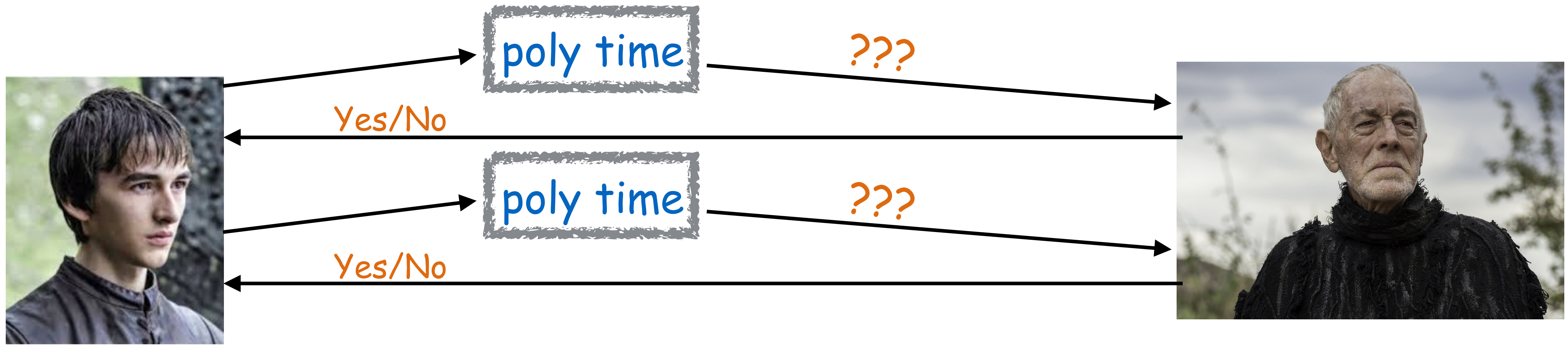
Turing Kernelization

Turing Kernelization



- You get the input (G,k) for problem P and you can ask polynomially many queries to the oracle where

Turing Kernelization



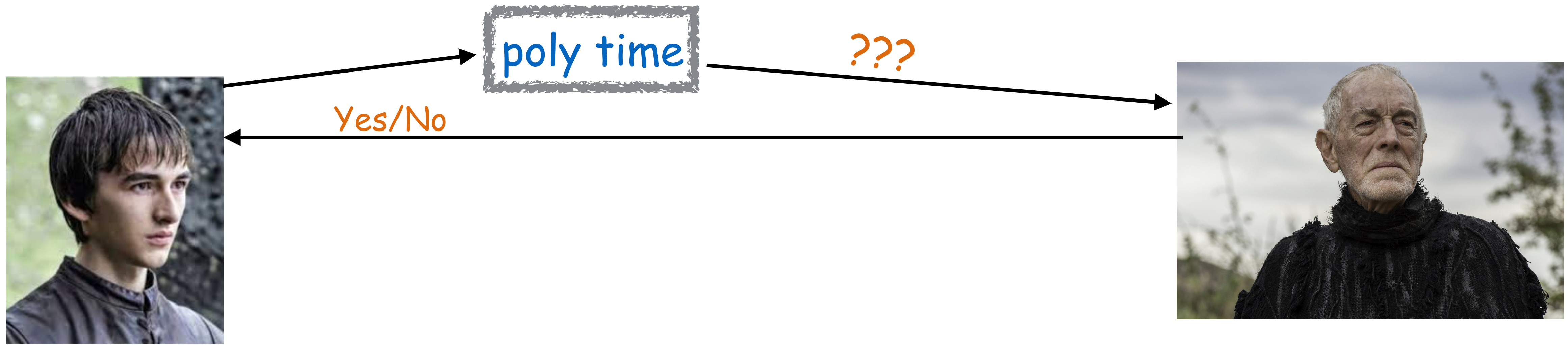
- You get the input (G,k) for problem P and you can ask polynomially many queries to the oracle where

EACH query must have size at most f(k) and looks like this:

Oh Great Oracle! Please tell me whether (Q_i,k_i) is a yes-instance of the problem P.

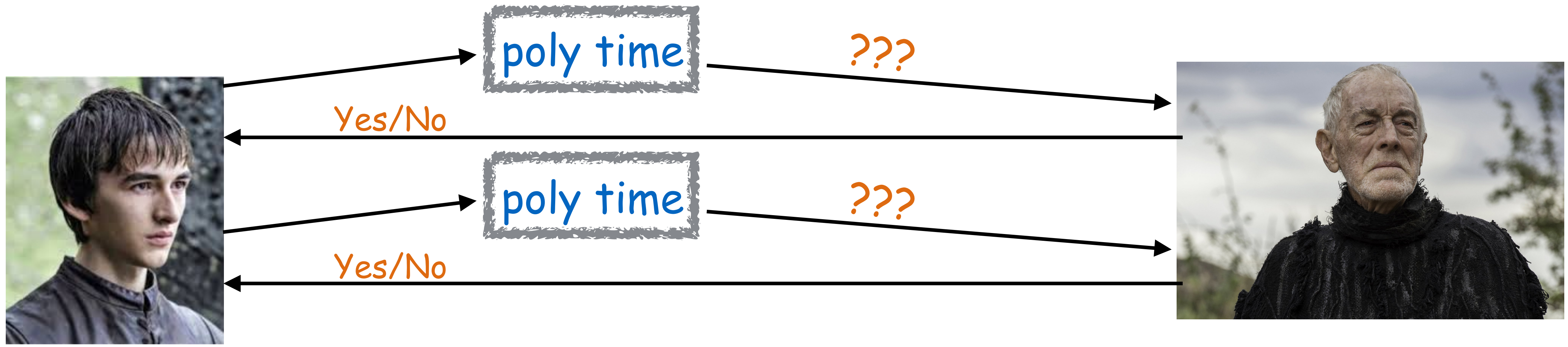
In the end, you must solve it.

Turing Kernelization



If problem P has a polynomial kernel, then P also has a polynomial Turing kernel.

Turing Kernelization



There are many interesting problems for which we don't expect polynomial kernels to even exist, but for which we already know polynomial TURING kernels.

General Take home message

- Kernelization is a subfield of Algorithms in its own right.

- Rich theory of upper and lower bounds.

- Lots and lots of interesting research in this domain.

- Many many open problems.

- Upcoming textbook on Kernelization [Fedor Fomin, Daniel Lokshtanov, Saket Saurabh, Meirav Zehavi]

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!