Kernelization

Every block of stone has a statue inside it and it is the task of the sculptor to discover it.

— Michelangelo Buonarroti

Lecture #6

Roohani Sharma

November 16, 2021

Kernelization

Every block of stone has a statue inside it and it is the task of the sculptor to discover it.

— Michelangelo Buonarroti

Lecture #6

Roohani Sharma

November 16, 2021

Kernelization

- Efficient pre-processing with guarantees for parameterized problems

Runs in polynomial time

I, k Π

$\equiv$

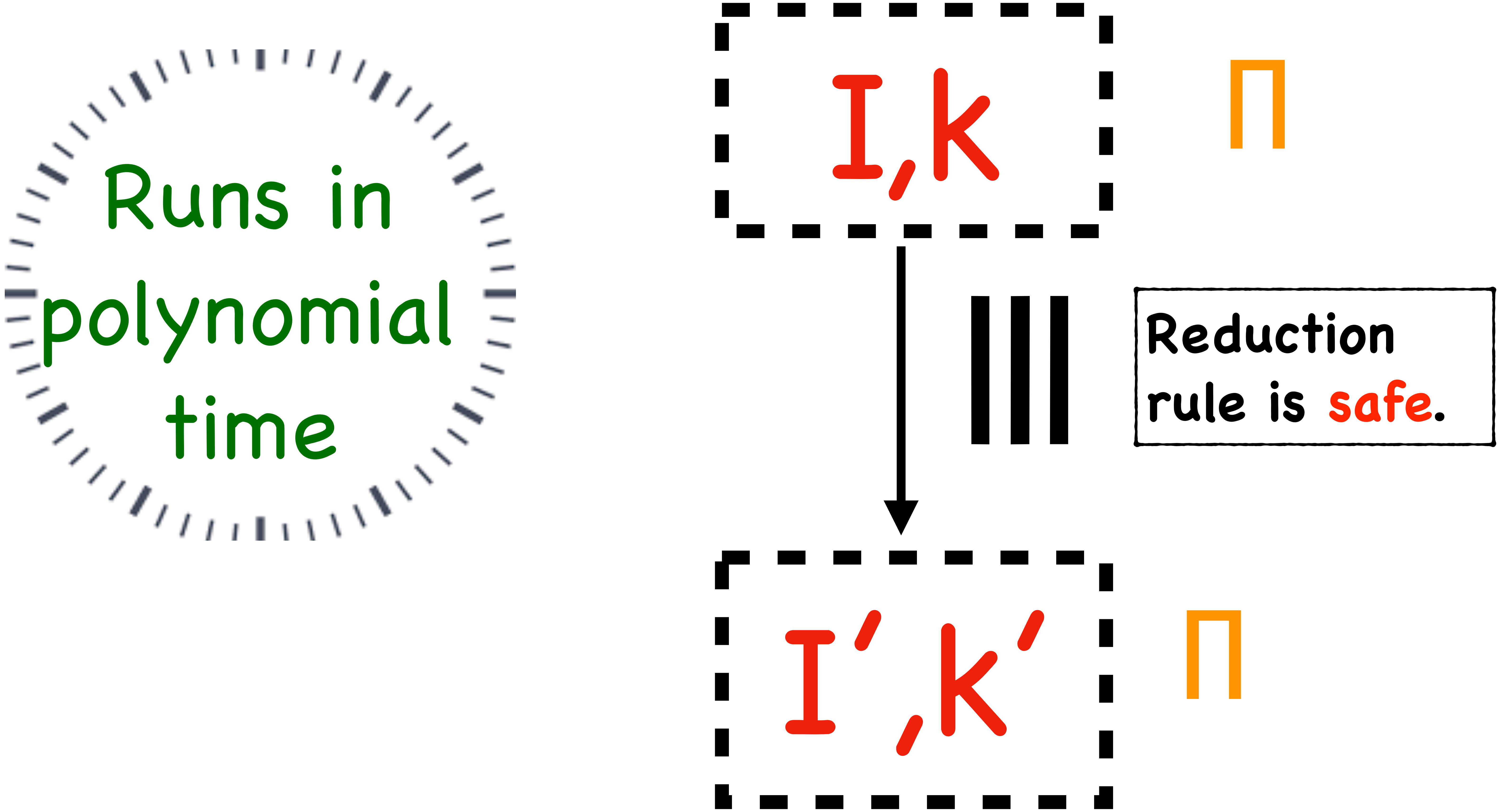
I', k' Π

$|I'|, k' \leq g(k)$

Π admits a kernel of size $g(k)$.

If $g(k)$ is a polynomial/exponential function, then Π admits a polynomial/exponential kernel.

Reduction rules (RR)



- A kernelization algorithm comprises of possibly several (polynomially many) safe reduction rules.
- Each reduction rules should be applicable only a polynomial number of times.
- This is followed by an analysis showing that if none of the designed reduction rules are applicable, then the size of the instance is bounded by some g(k).

Summary from Lecture #4

Vertex Cover	k^2+k vertices and k^2 edges	
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	
d-Hitting Set	$O(k^d)$ sets and elements	
d-Set Packing	$O(k^d)$ sets and elements	

Sunflower Lemma

Summary from Lecture #4

Vertex Cover	k^2+k vertices and k^2 edges	
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	
d-Hitting Set	$O(k^d)$ sets and elements	
d-Set Packing	$O(k^d)$ sets and elements	

Sunflower Lemma

Vertex Cover	k^2+k vertices and k^2 edges	$O(k^{2-\varepsilon})$ size
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	Polynomial kernel
d-Hitting Set	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size
d-Set Packing	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size

Sunflower Lemma

Vertex Cover	k^2+k vertices and k^2 edges	$O(k^{2-\varepsilon})$ size
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	Polynomial kernel
d-Hitting Set	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size
d-Set Packing	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size

Sunflower Lemma

Vertex Cover	k^2+k vertices and k^2 edges	$O(k^{2-\varepsilon})$ size
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	Polynomial kernel
d-Hitting Set	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size
d-Set Packing	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size

Sunflower Lemma

Under reasonable complexity assumptions

Vertex Cover	k^2+k vertices and k^2 edges	~~$O(k^{2-\varepsilon})$ size~~
Feedback Arc Set in Tournaments	$O(k^2)$ vertices	
Edge Clique Cover	2^k-1 vertices	~~Polynomial kernel~~
d-Hitting Set	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~
d-Set Packing	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~

Sunflower Lemma

Under reasonable complexity assumptions

Sunflower Lemma

Vertex Cover	k^2+k vertices and k^2 edges	$O(k^{2-\varepsilon})$ size	$O(k)$ vertices?
Feedback Arc Set in	$O(k^2)$ vertices		
Edge Clique Cover	2^k-1 vertices	Polynomial kernel	
d-Hitting Set	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size	$O(k^{d-1})$ elements?
d-Set Packing	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size	$O(k^{d-1})$ elements?

Under reasonable complexity assumptions

Vertex Cover	k^2+k vertices and k^2 edges	$O(k^{2-\varepsilon})$ size	$O(k)$ vertices?
Feedback Arc Set in	$O(k^2)$ vertices		
Edge Clique Cover	2^k-1 vertices	Polynomial kernel	
d-Hitting Set	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size	$O(k^{d-1})$ elements?
d-Set Packing	$O(k^d)$ sets and elements	$O(k^{d-\varepsilon})$ size	$O(k^{d-1})$ elements?

Sunflower Lemma

Under reasonable complexity assumptions

Vertex Cover	k^2+k vertices and k^2 edges	~~$O(k^{2-\varepsilon})$ size~~	$O(k)$ vertices?
Feedback Arc Set in	$O(k^2)$ vertices		
Edge Clique Cover	2^k-1 vertices	~~Polynomial kernel~~	
d-Hitting Set	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~	$O(k^{d-1})$ elements?
d-Set Packing	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~	$O(k^{d-1})$ elements?

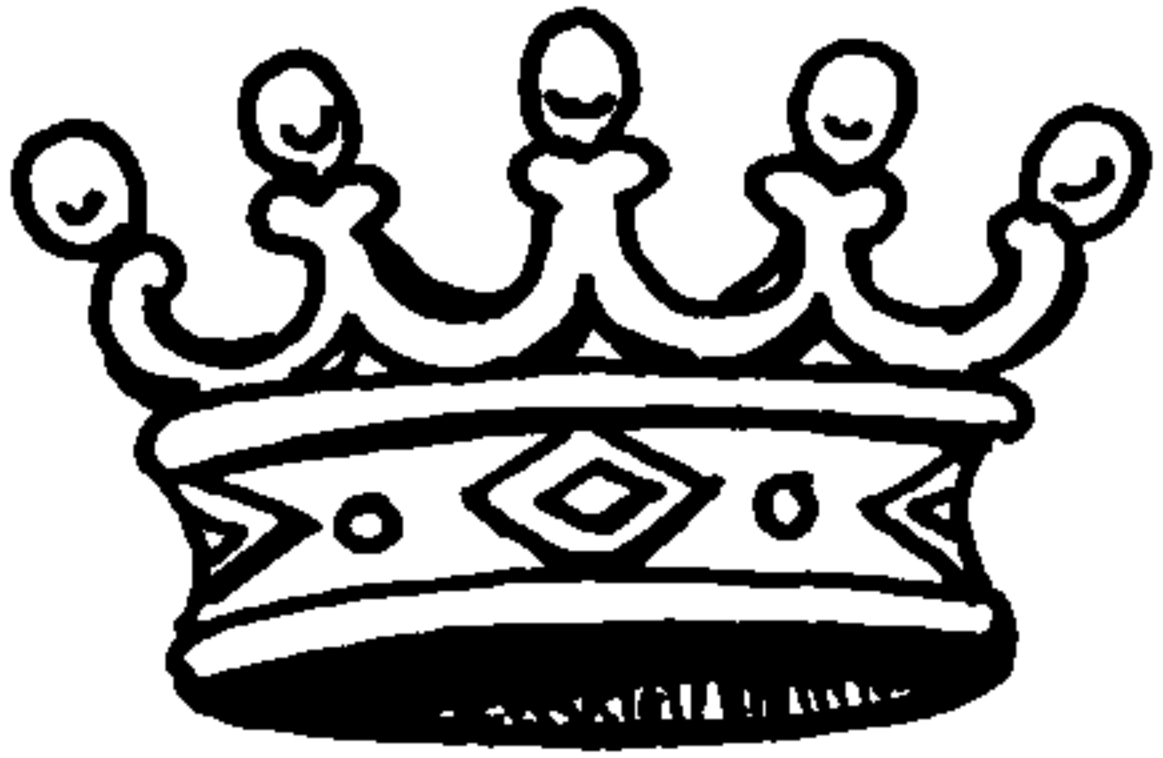
Sunflower Lemma

Under reasonable complexity assumptions

Vertex Cover	k^2+k vertices and k^2 edges	~~$O(k^{2-\varepsilon})$ size~~	$O(k)$ vertices?
Feedback Arc Set in	$O(k^2)$ vertices		
Edge Clique Cover	2^k-1 vertices	~~Polynomial kernel~~	
d-Hitting Set	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~	$O(k^{d-1})$ elements?
d-Set Packing	$O(k^d)$ sets and elements	~~$O(k^{d-\varepsilon})$ size~~	$O(k^{d-1})$ elements?

Sunflower Lemma

Under reasonable complexity assumptions



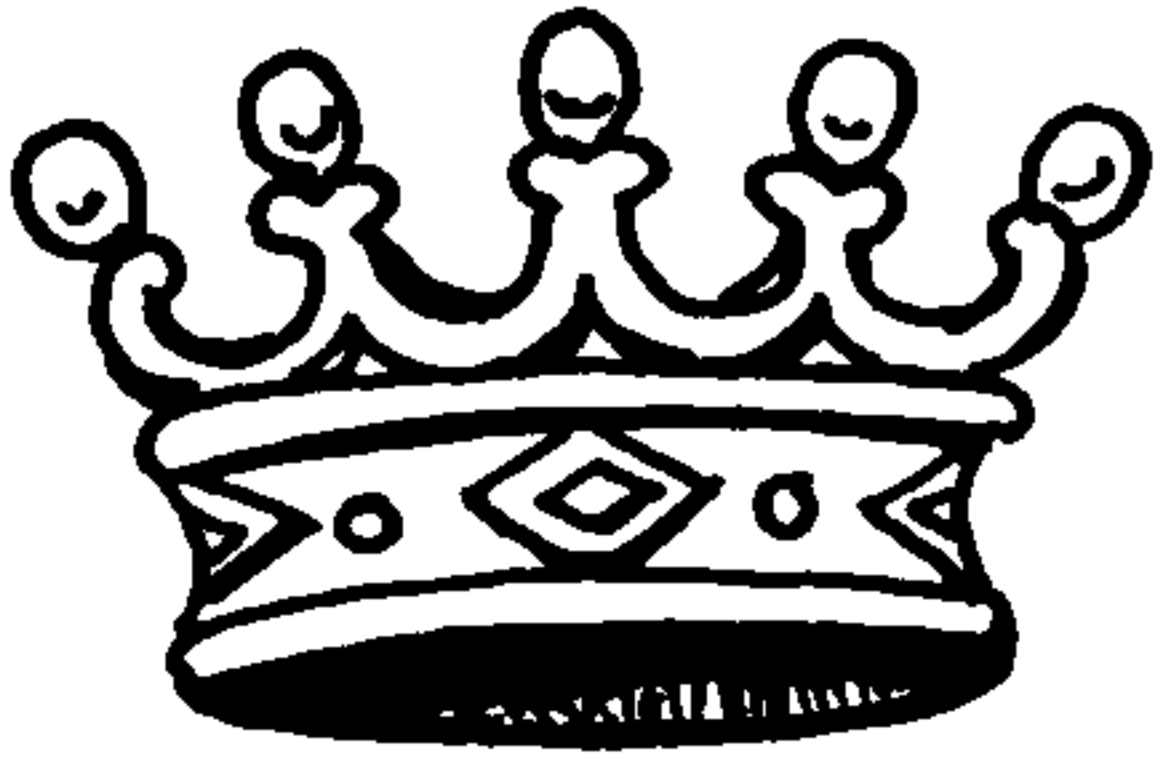
Crown Decomposition

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1$$

$$0 \leq x_i \leq 1$$

Linear Programming based kernels



Crown Decomposition

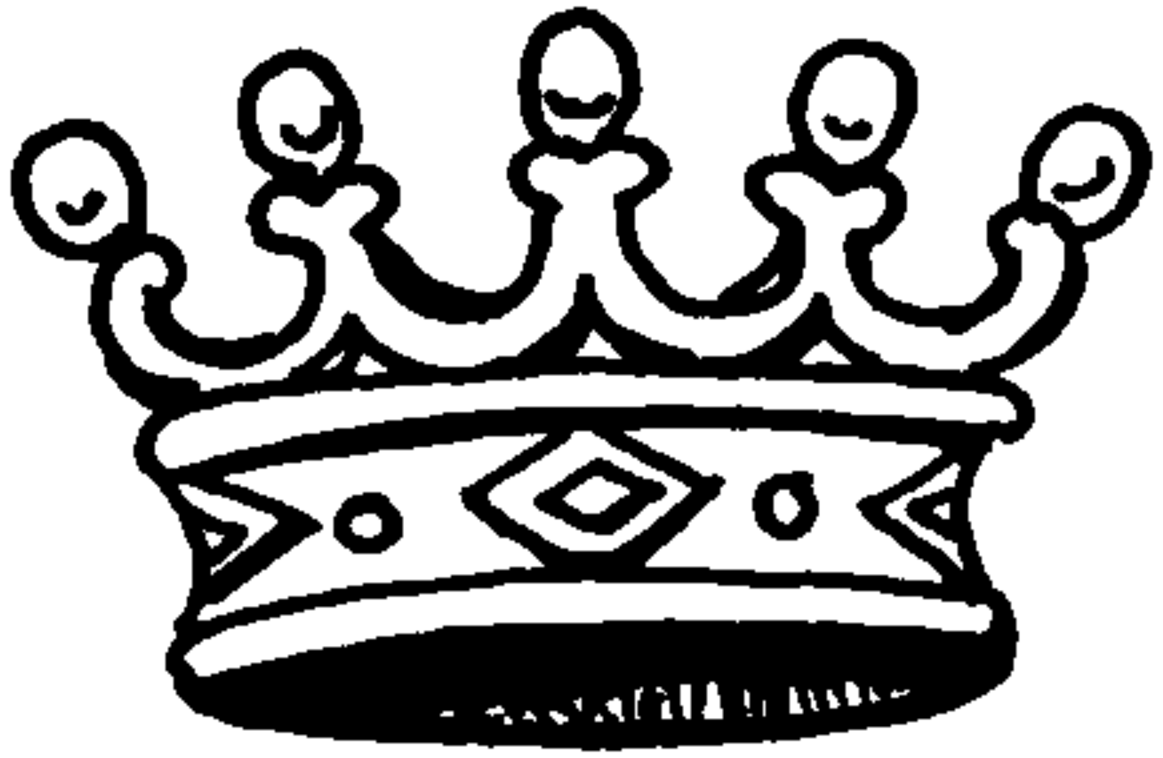
VERTEX COVER admits a kernel with 3k vertices.

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1$$

$$0 \leq x_i \leq 1$$

Linear Programming based kernels



$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1$$

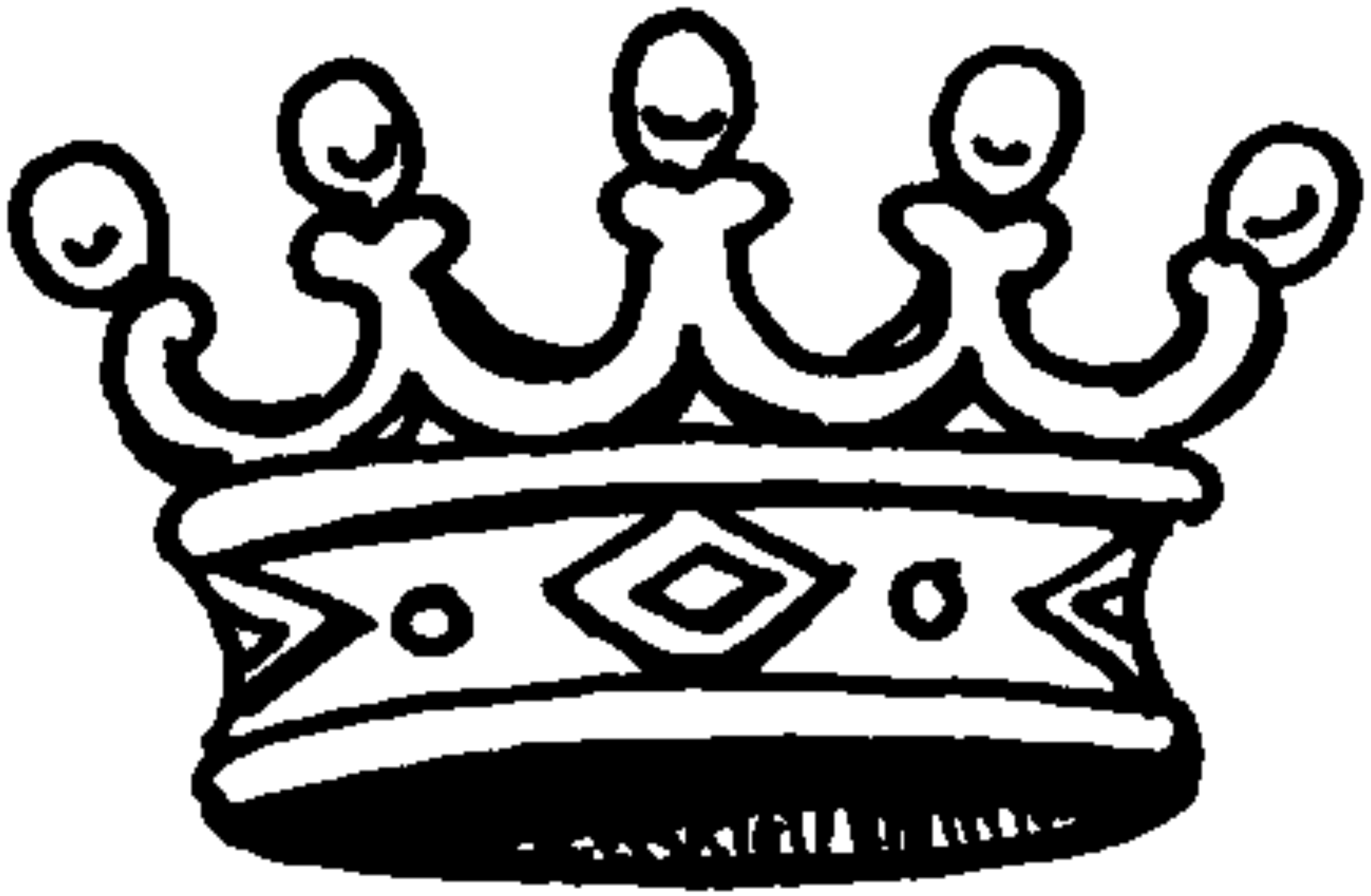
$$0 \leq x_i \leq 1$$

Crown Decomposition

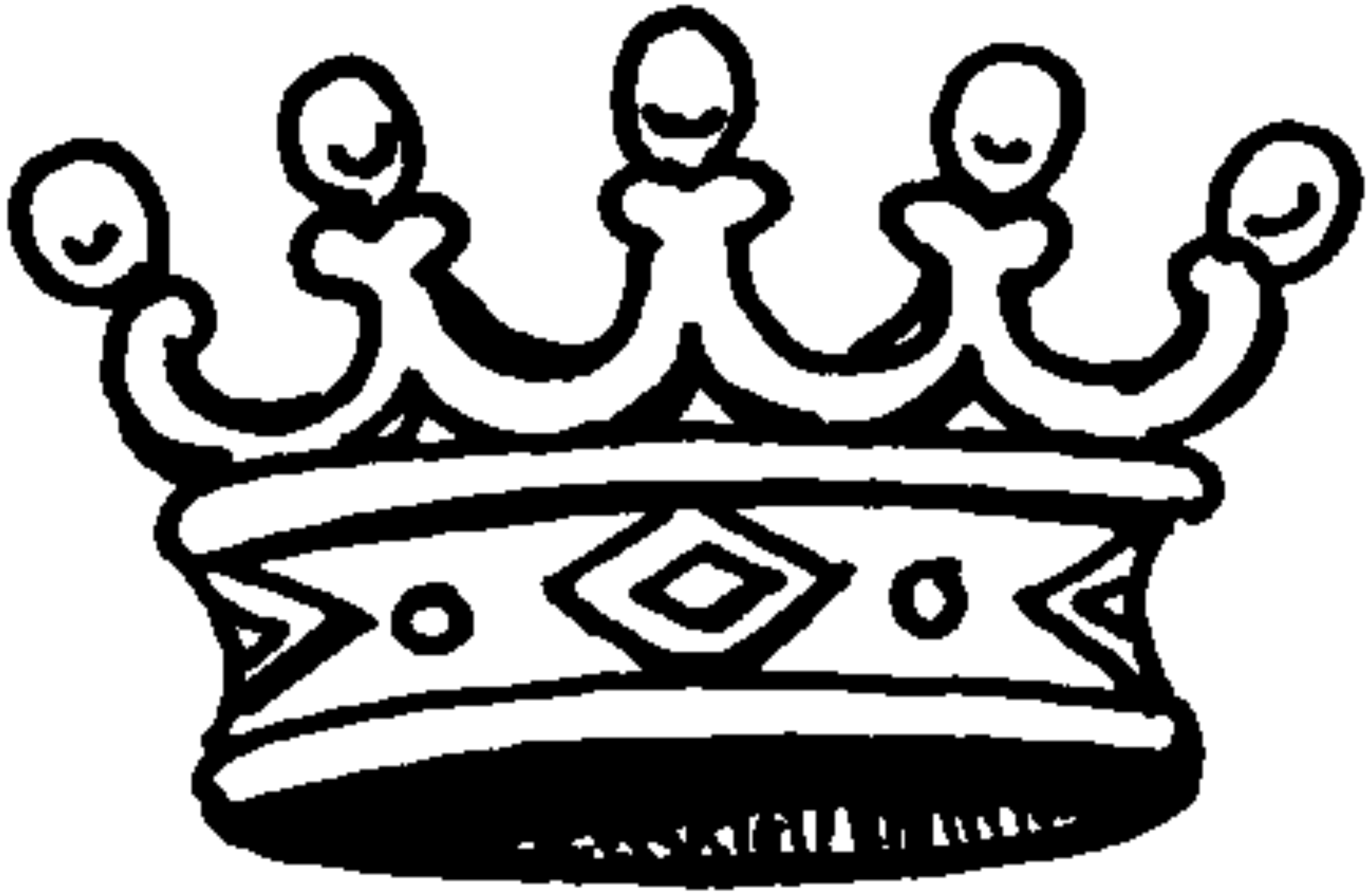
VERTEX COVER admits a kernel with 3k vertices.

Linear Programming based kernels

VERTEX COVER admits a kernel with 2k vertices.



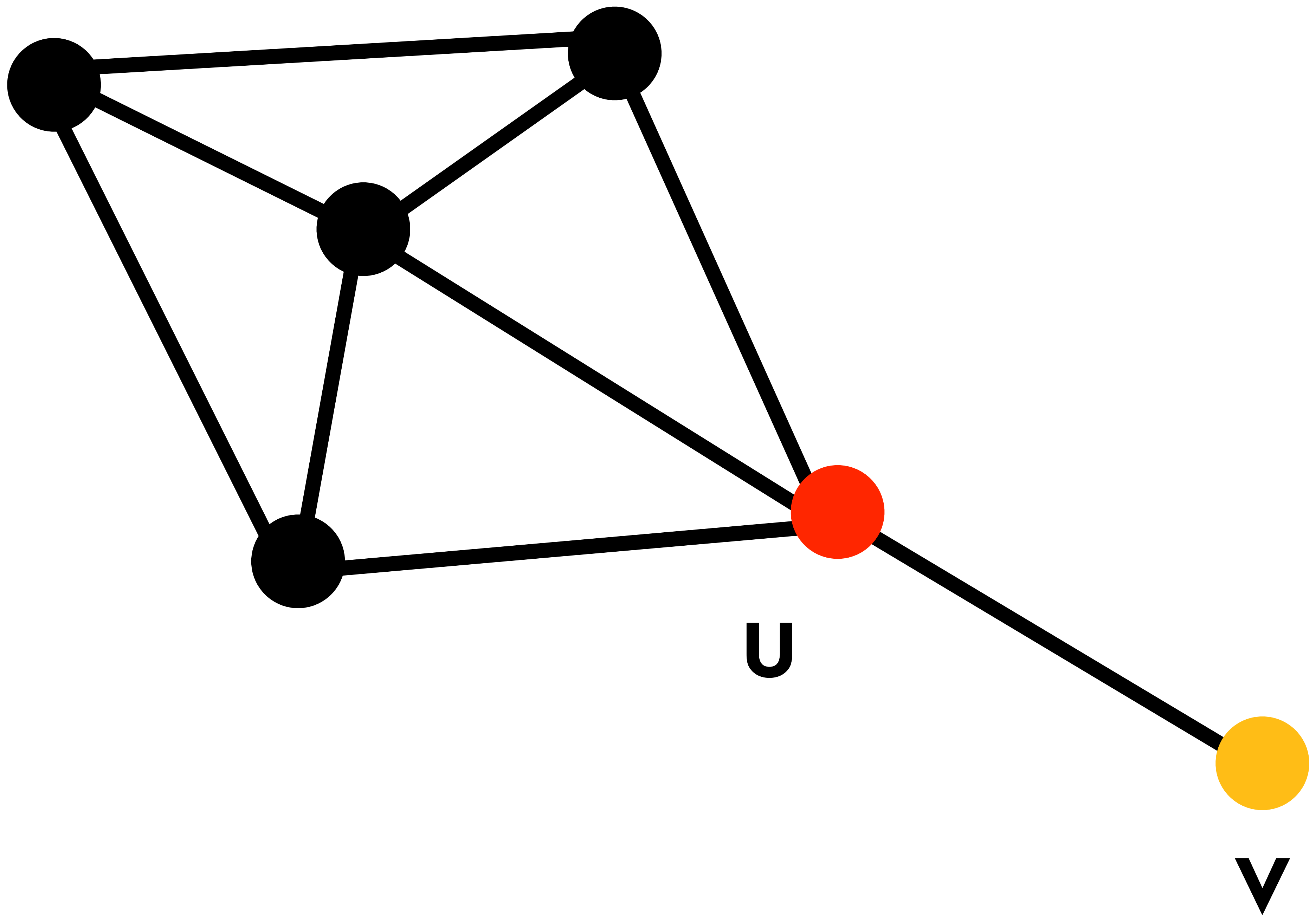
Crown Decomposition



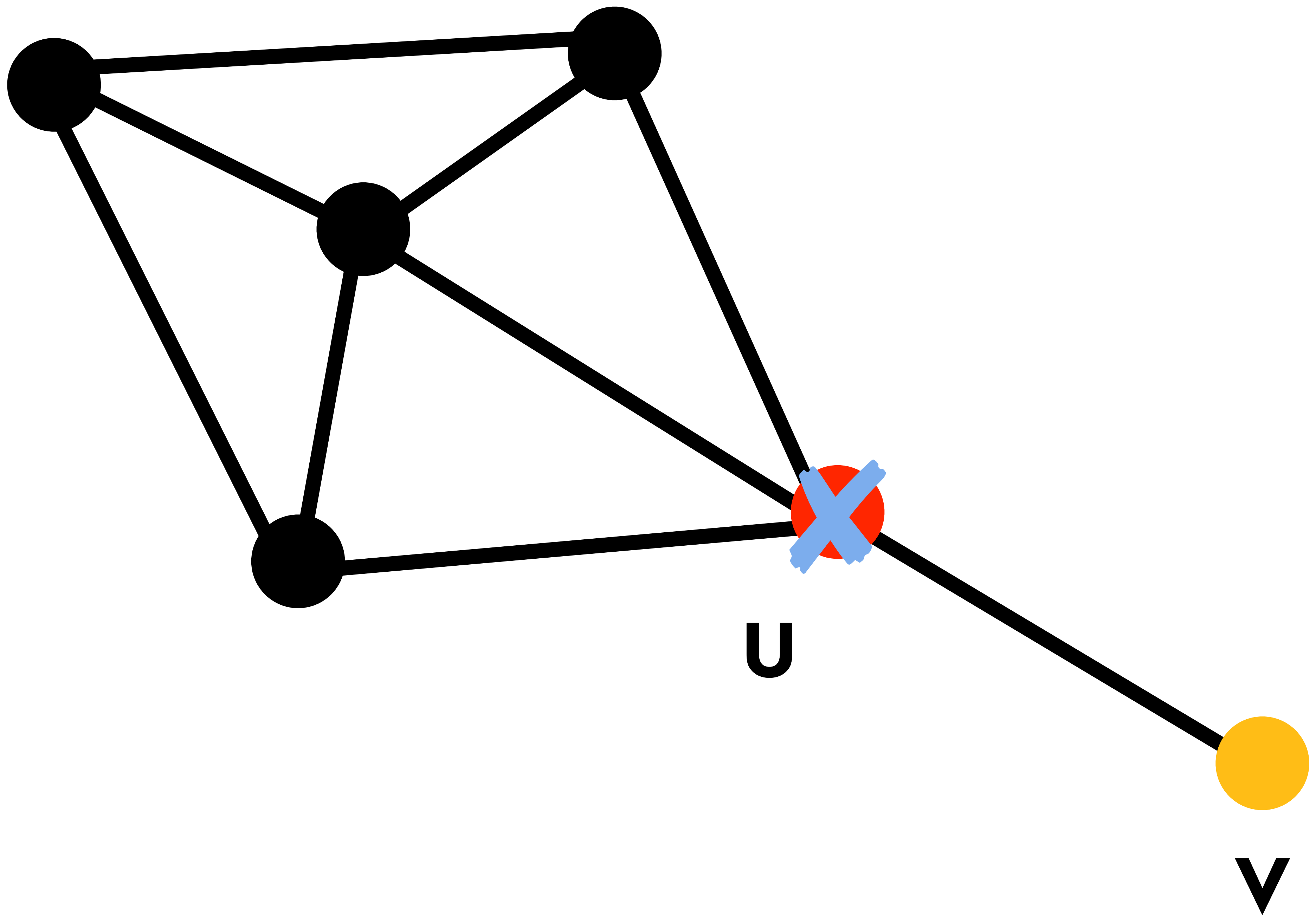
Crown Decomposition

VERTEX COVER admits a kernel with $3k$ vertices.

VERTEX COVER: Deleting degree 1 vertices



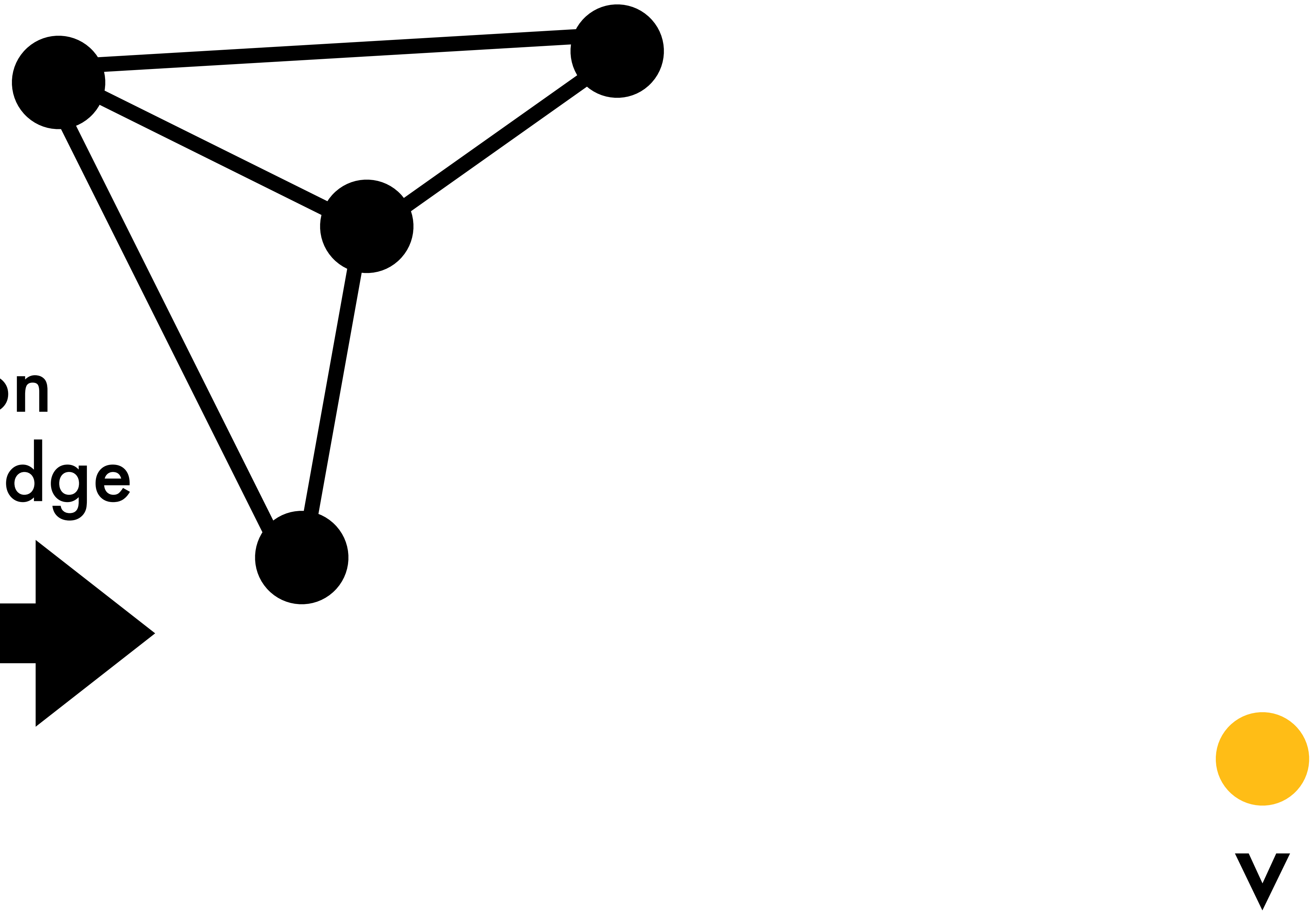
VERTEX COVER: Deleting degree 1 vertices



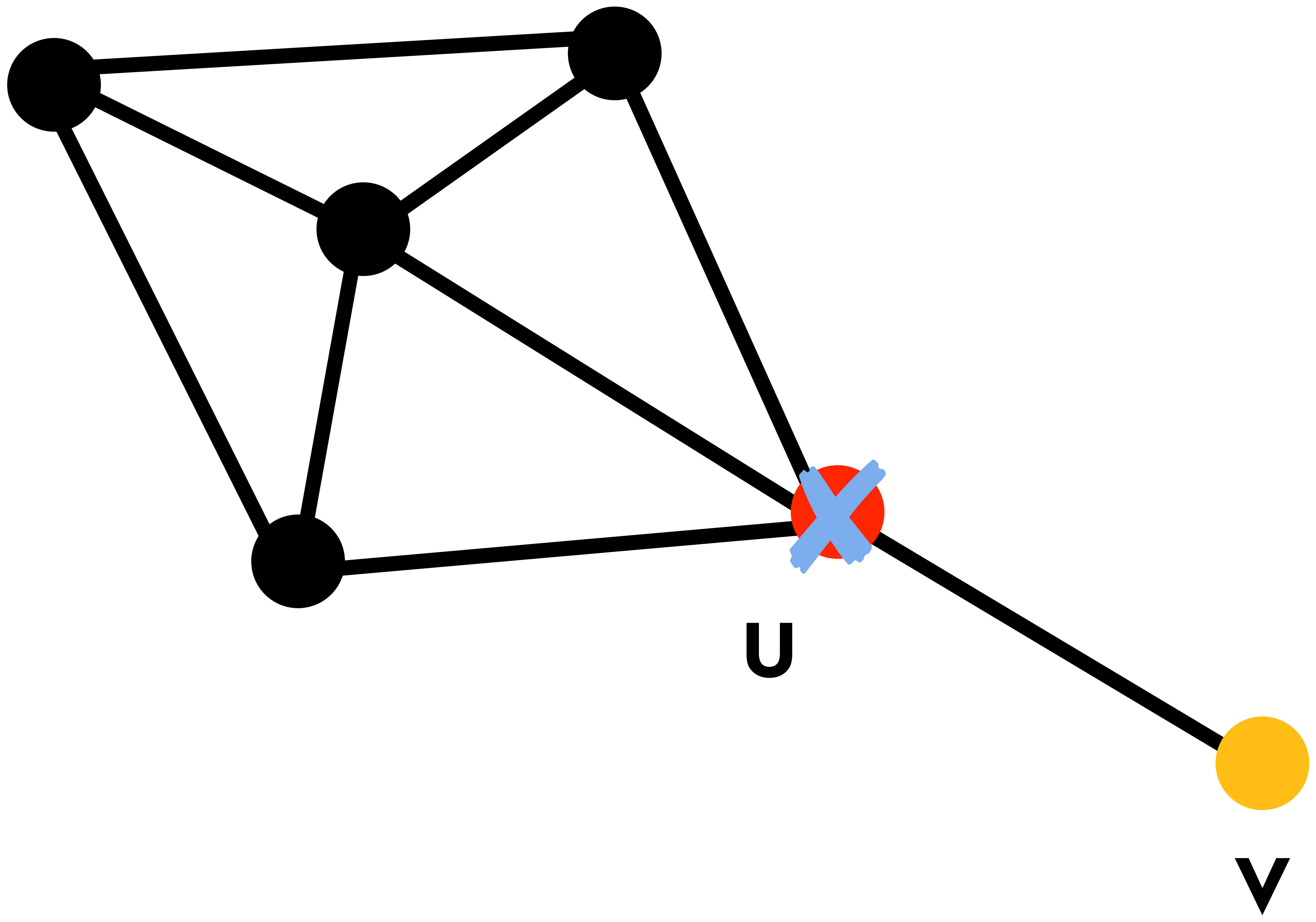
Reduction rule:

Be greedy and
pick u in the solution
to cover the (u,v) edge

Drop budget by 1



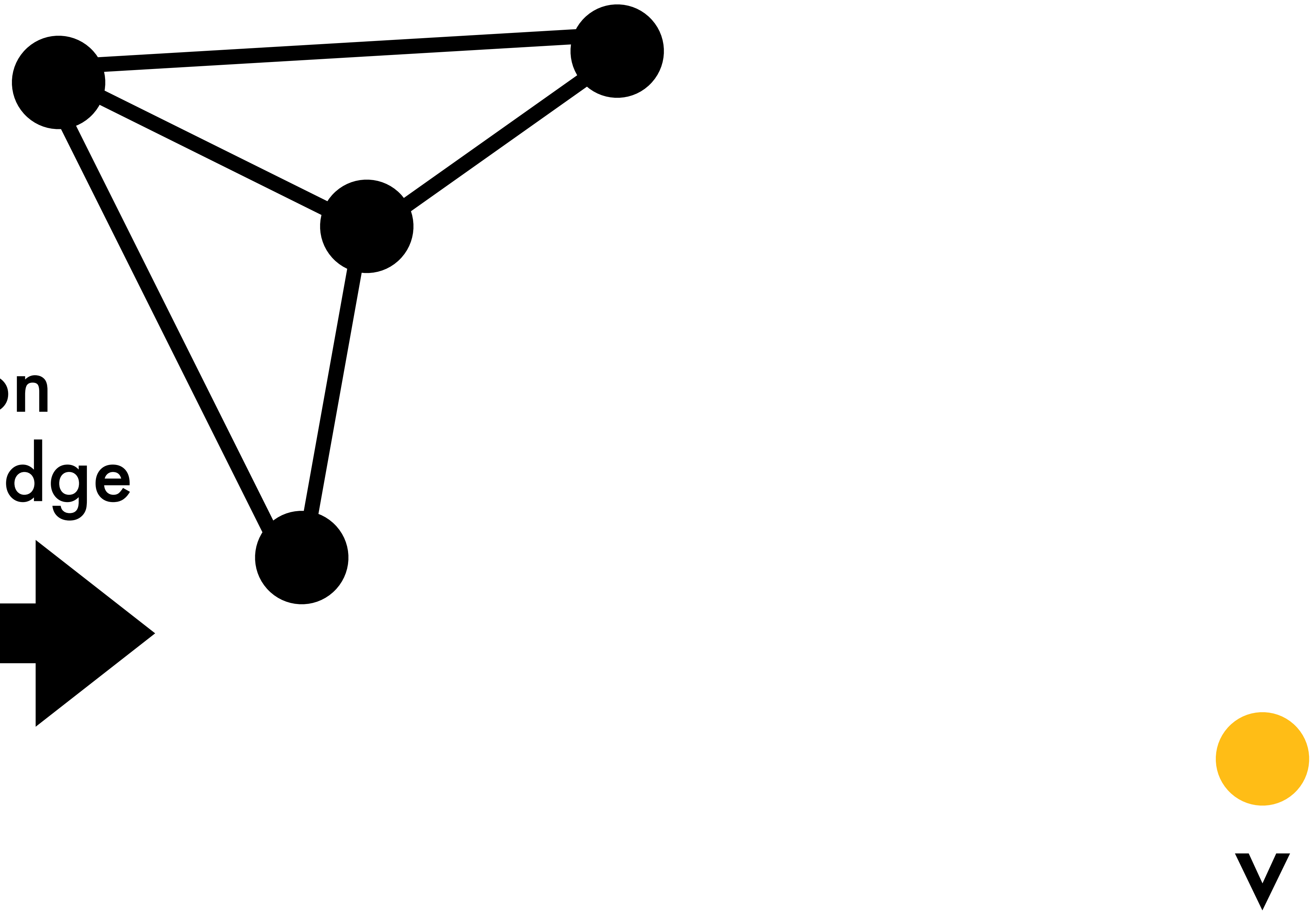
VERTEX COVER: Deleting degree 1 vertices



Reduction rule:

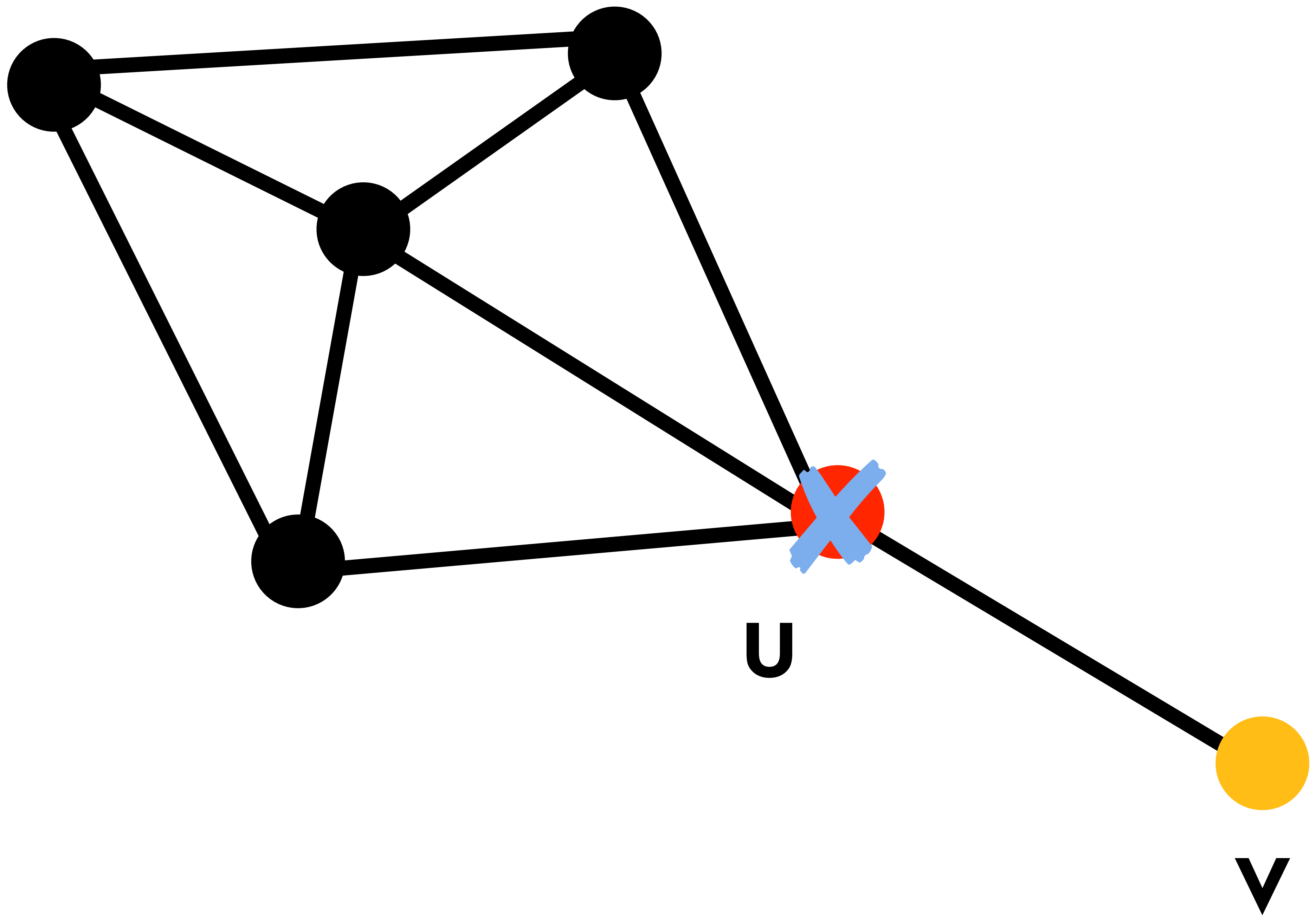
Be greedy and
pick u in the solution
to cover the (u,v) edge

Drop budget by 1



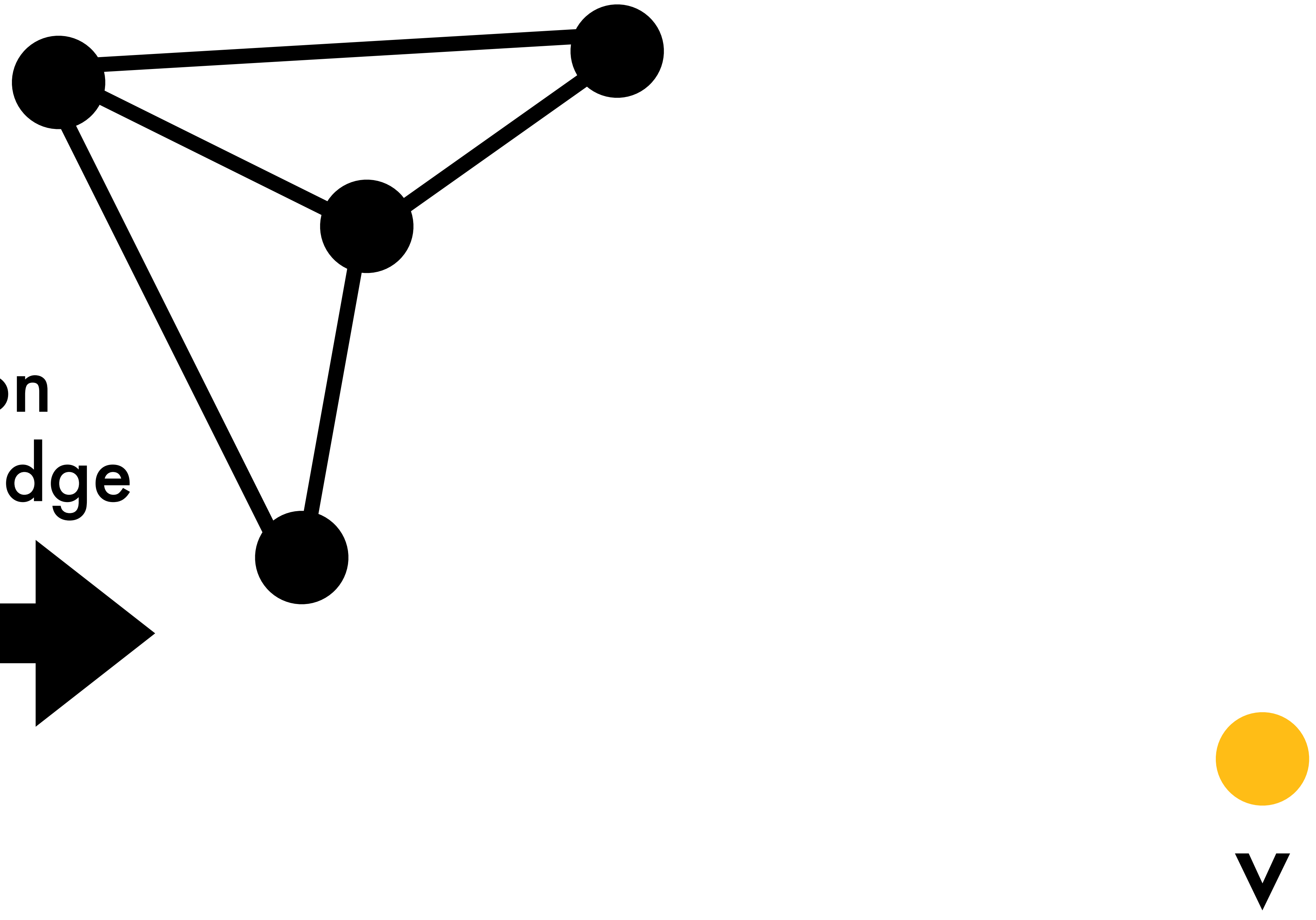
v becomes isolated,
So delete v too.

VERTEX COVER: Deleting degree 1 vertices



Reduction rule:

Be greedy and
pick u in the solution
to cover the (u,v) edge



Drop budget by 1

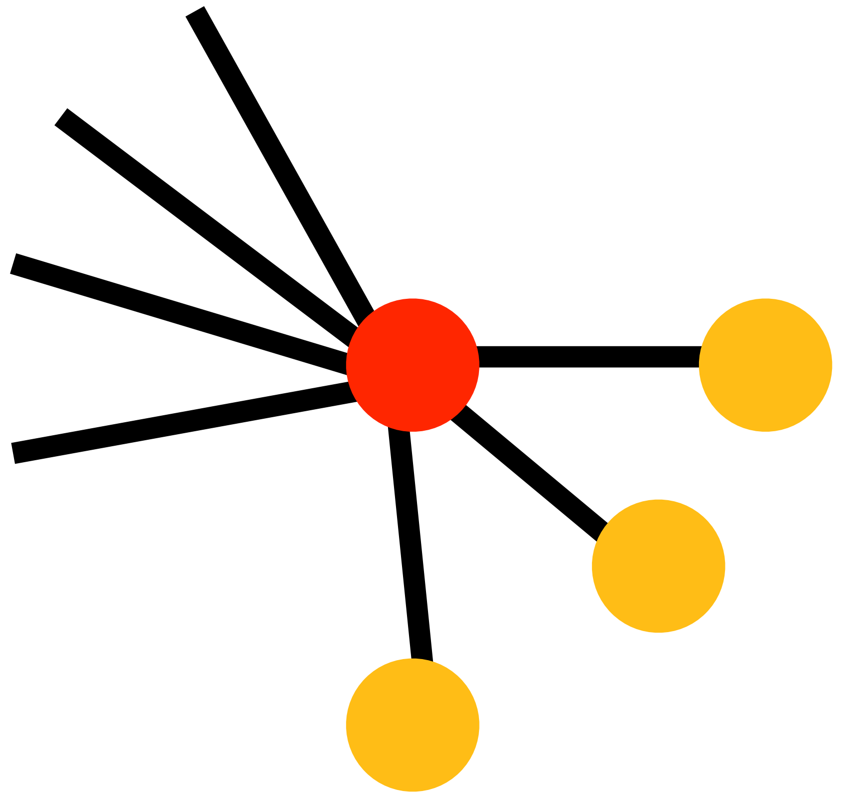
v becomes isolated,
So delete v too.

Essence of the safeness (Delegation principle):

(yellow to red): You can do what I can do, and at least one of us needs to do the job!

VERTEX COVER:

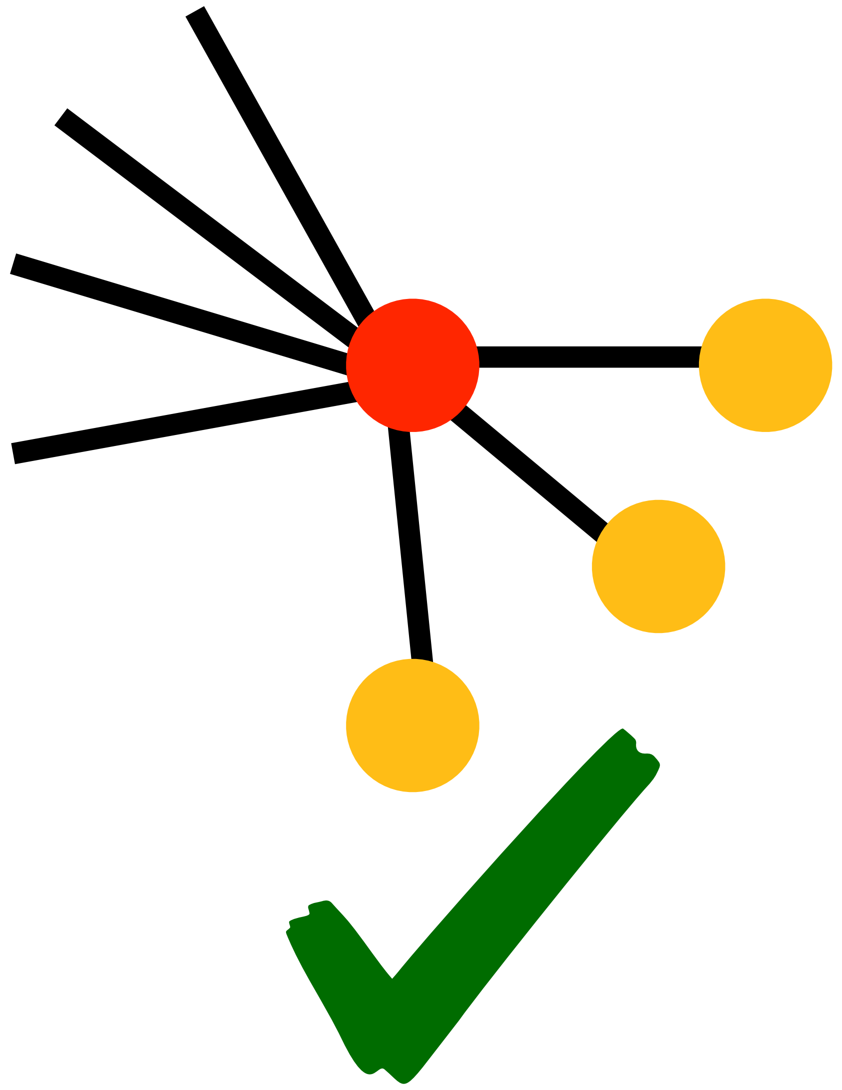
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

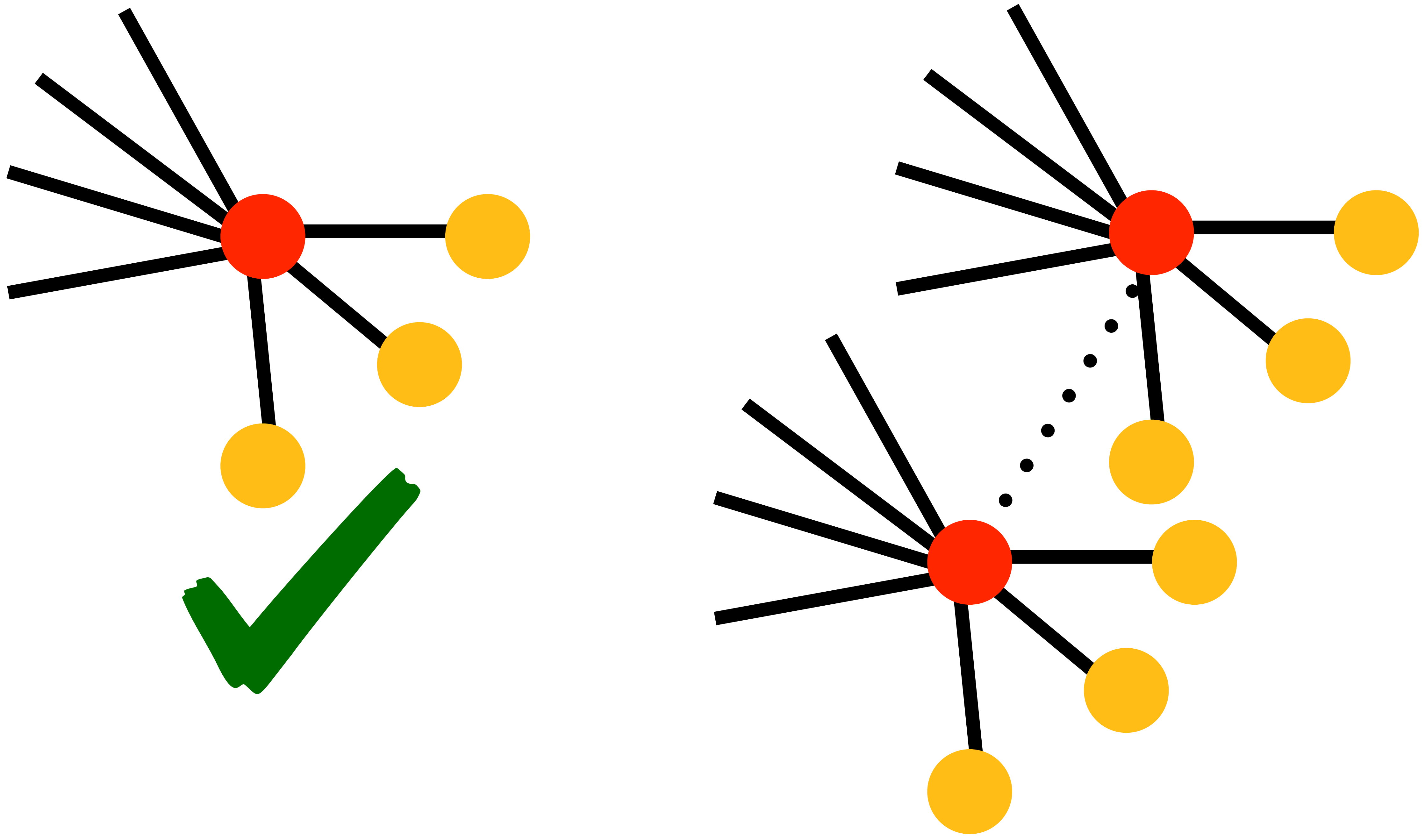
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

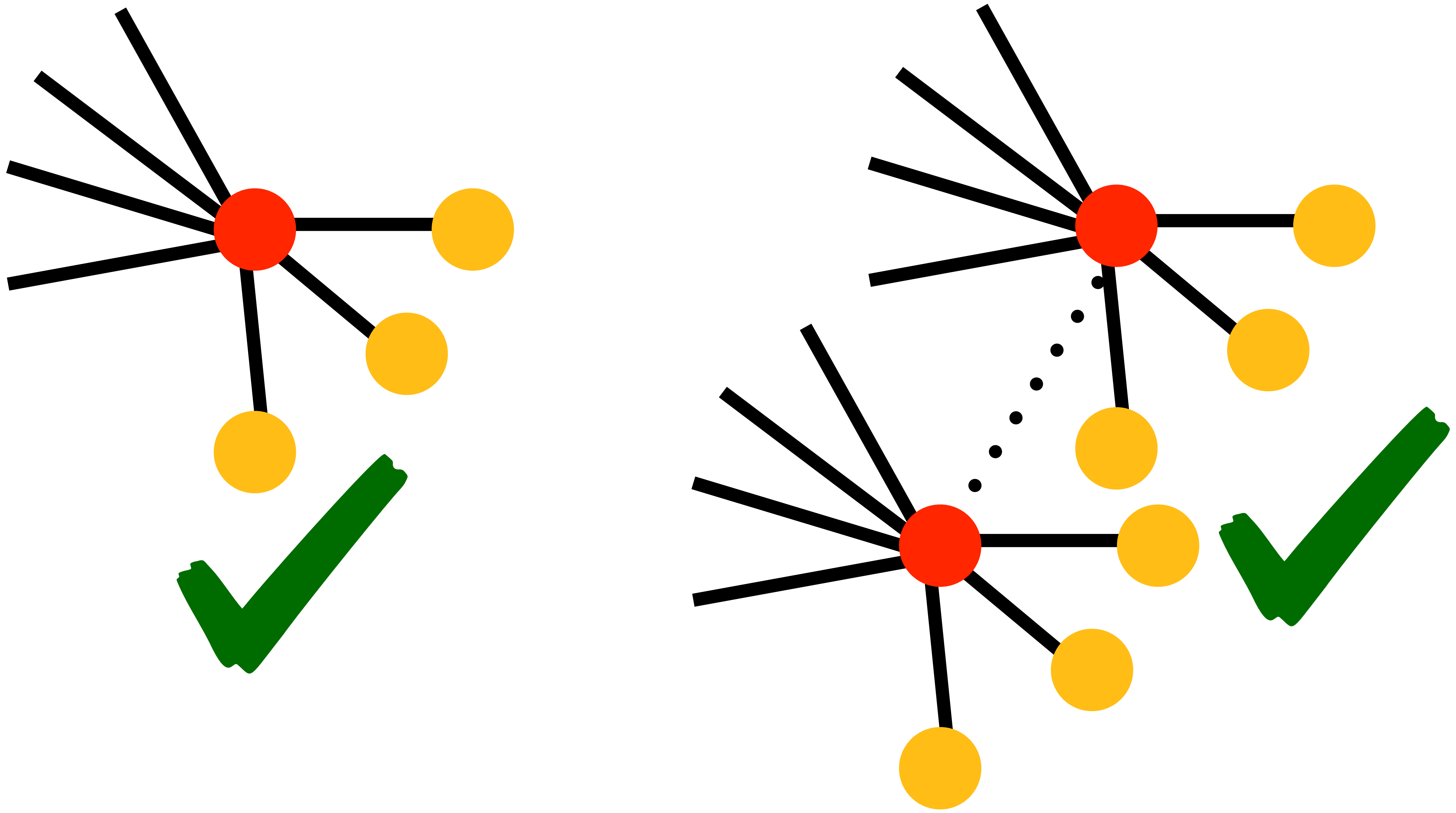
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

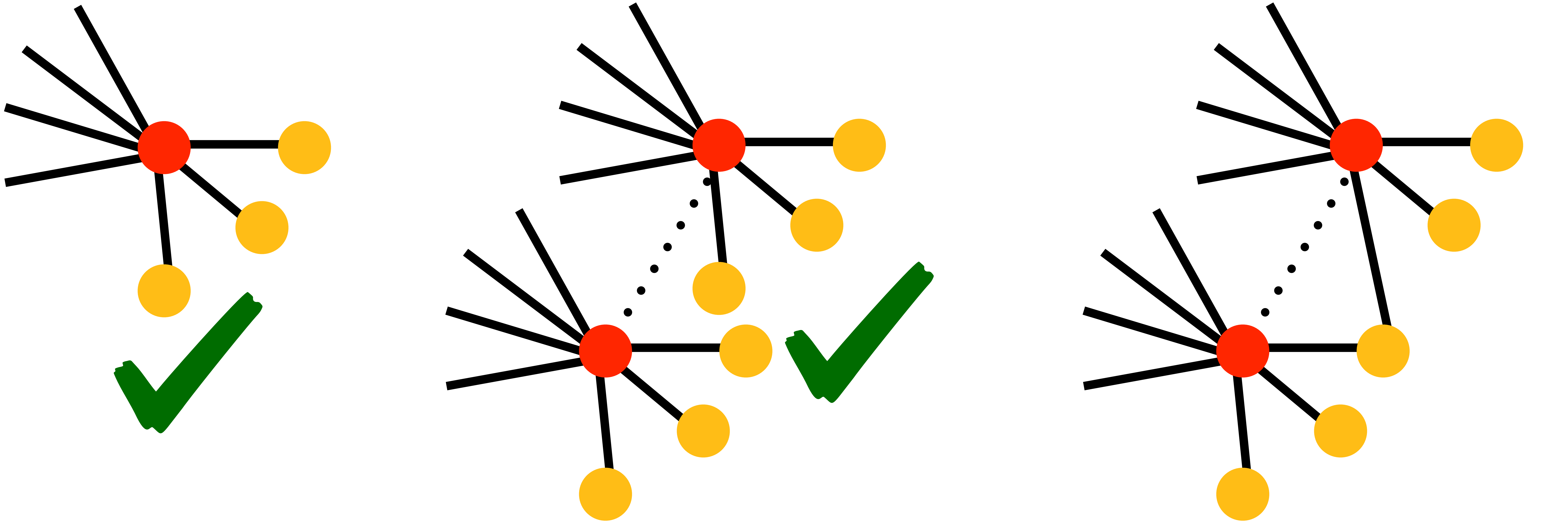
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

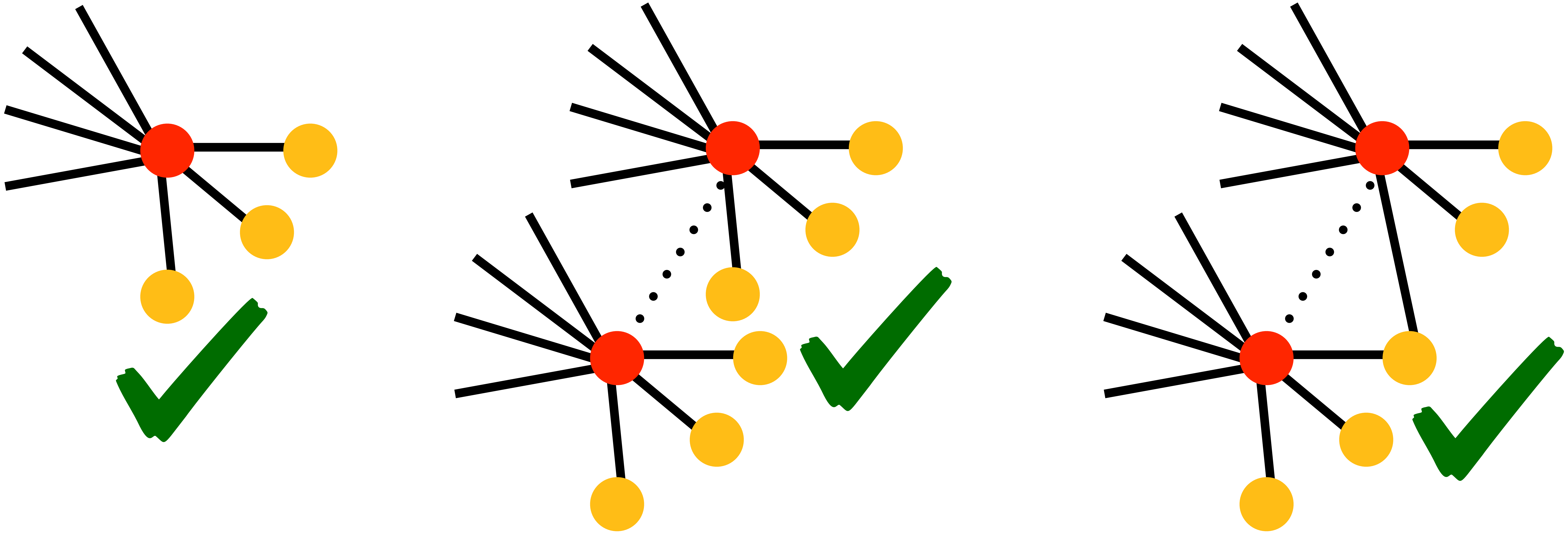
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

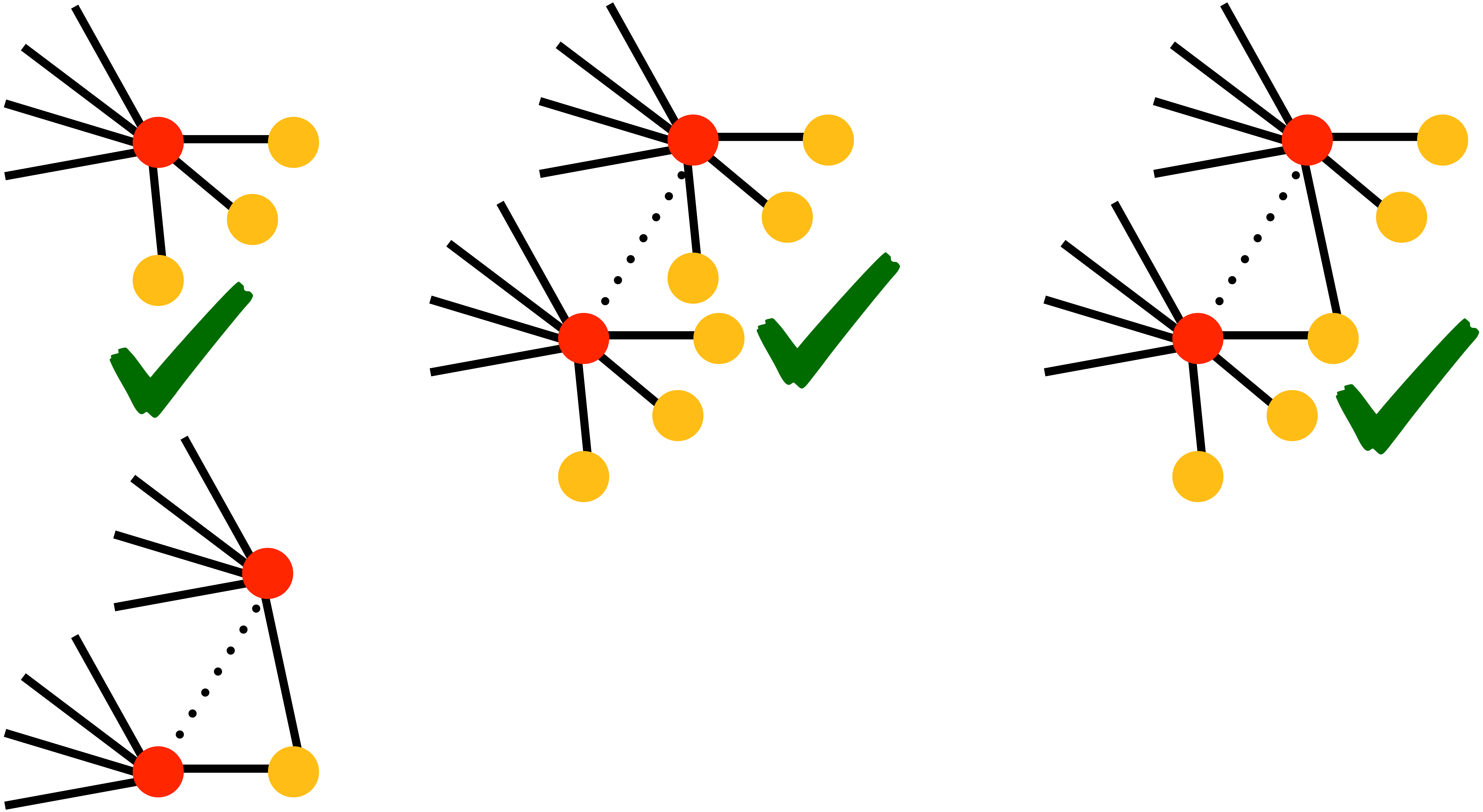
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

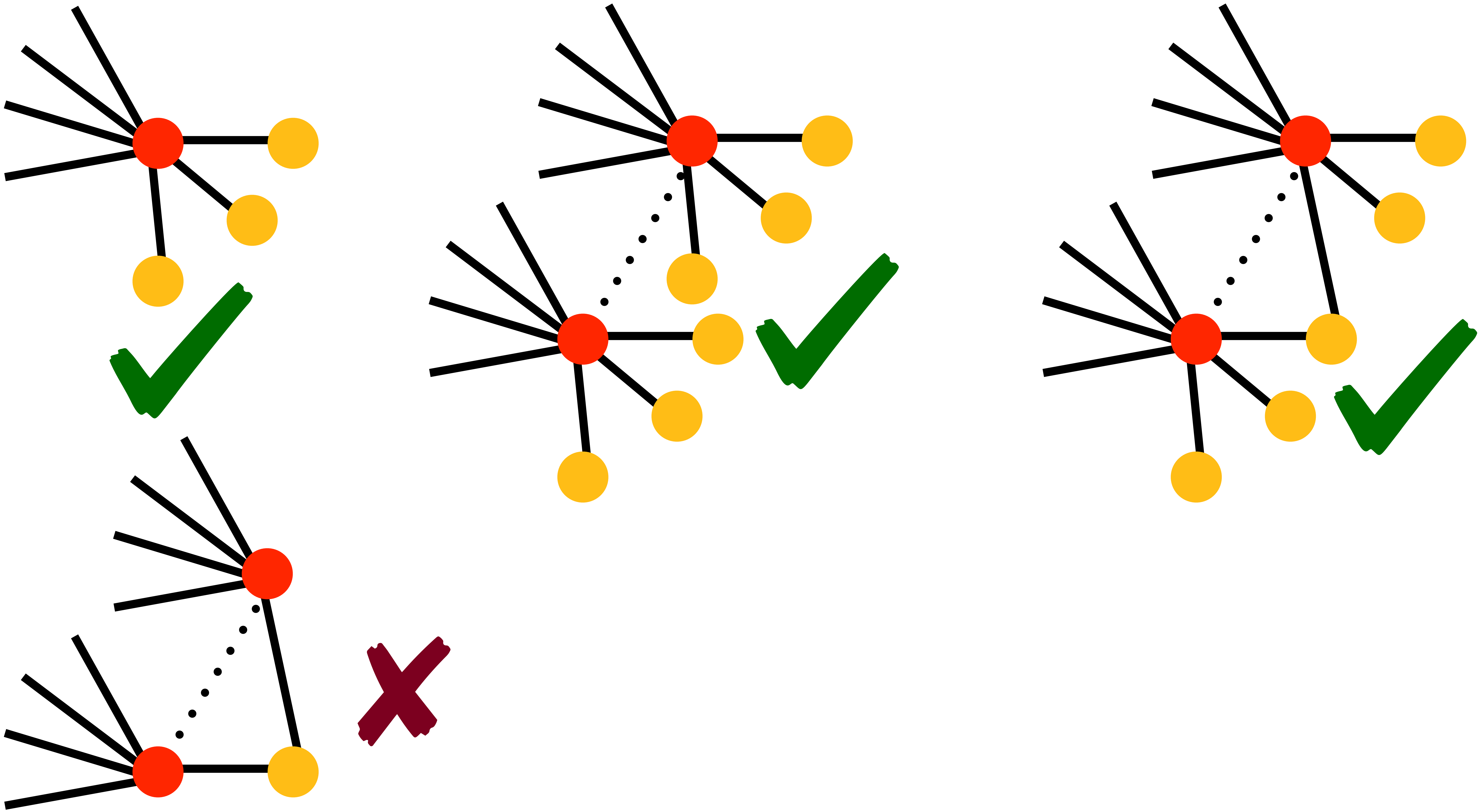
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

VERTEX COVER:

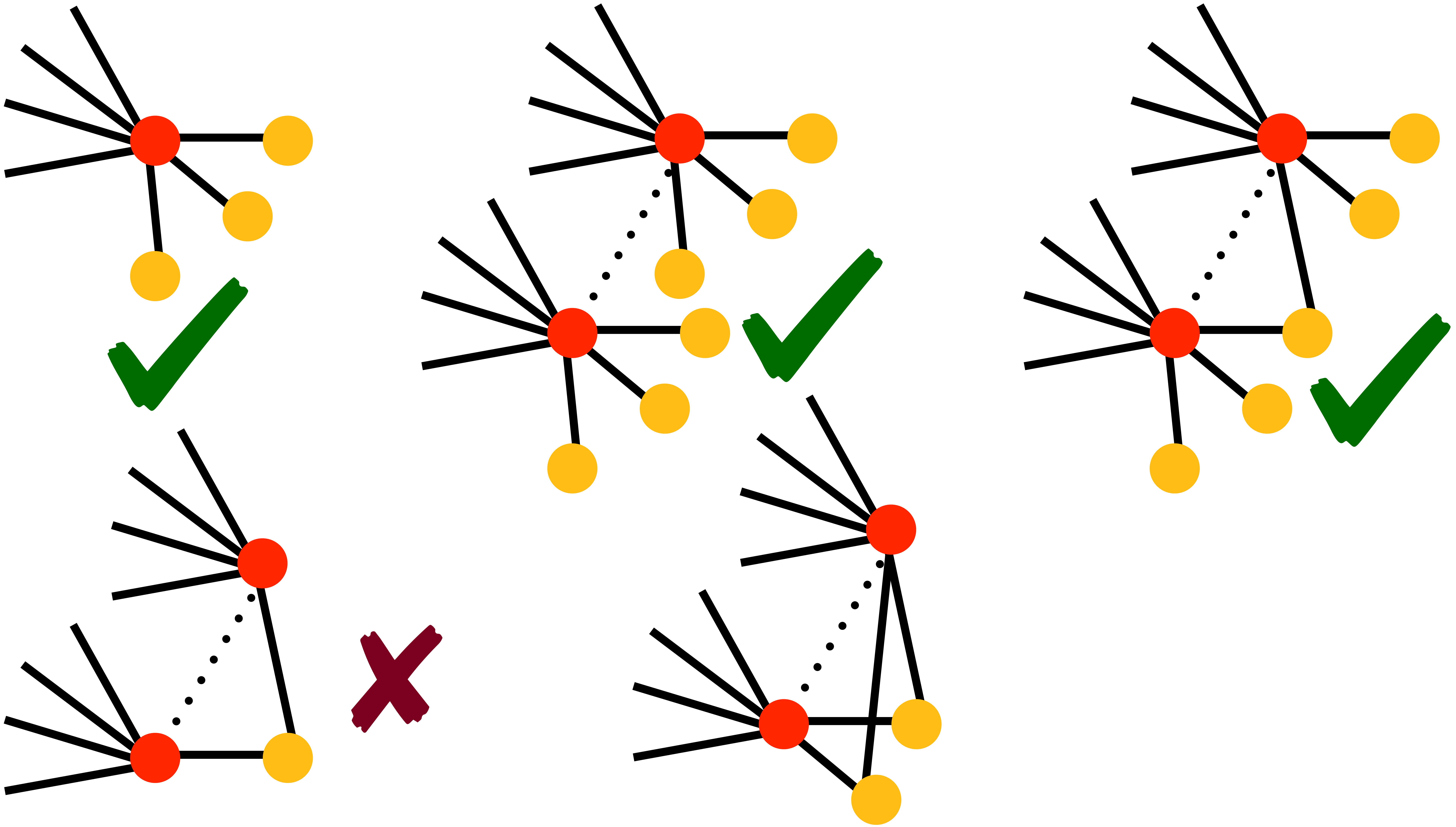
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

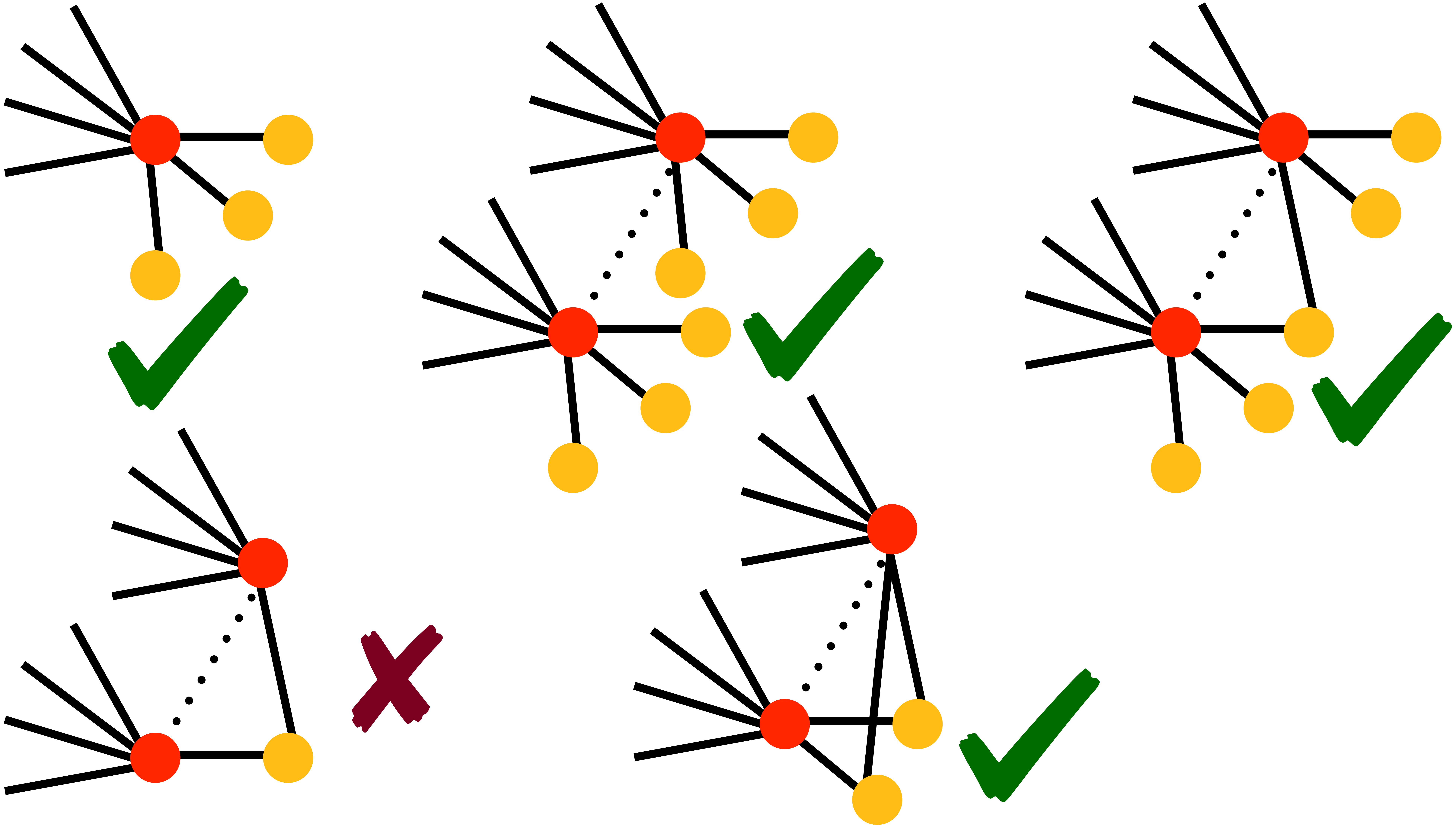
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

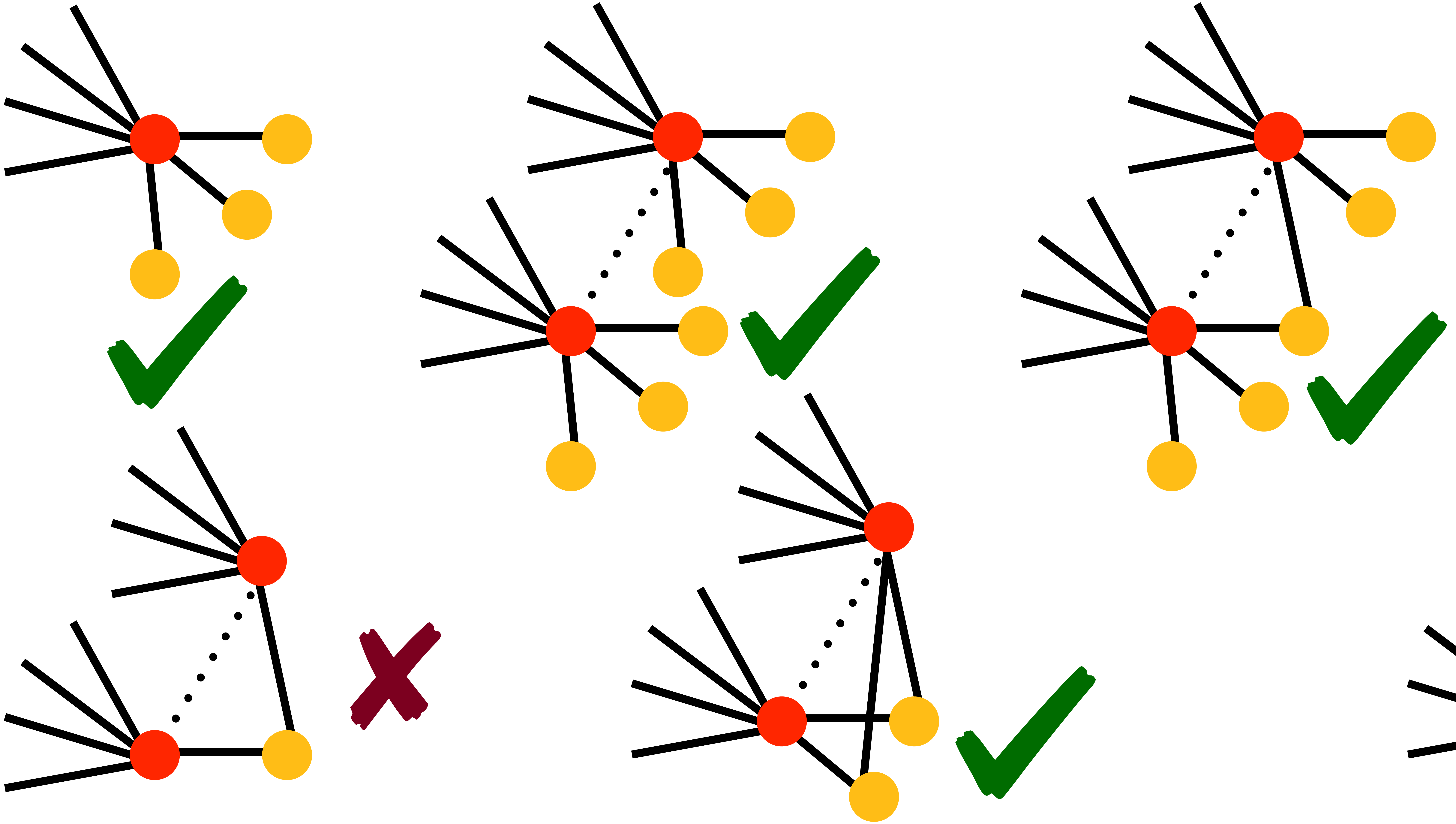
You can do what I can do, and at least one of us needs to do the job!



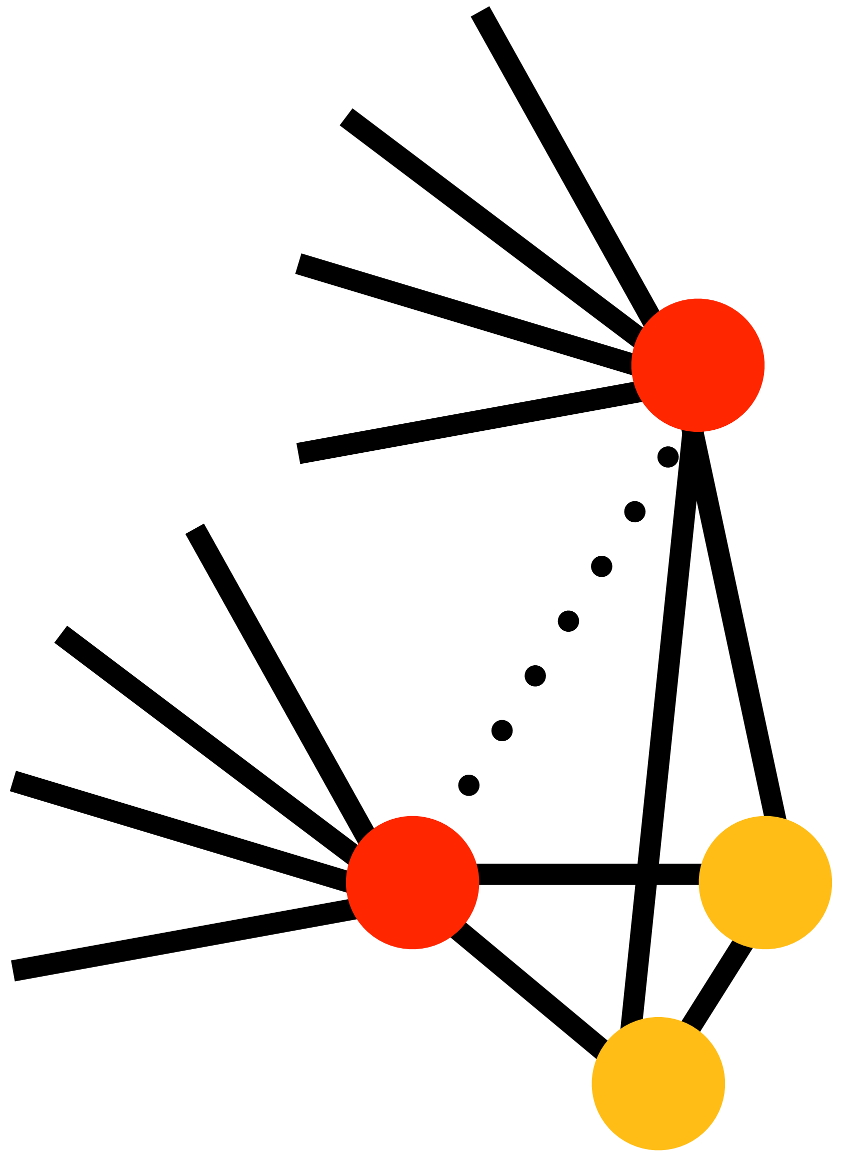
Be greedy and pick ALL the red vertices in the solution ?

Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!

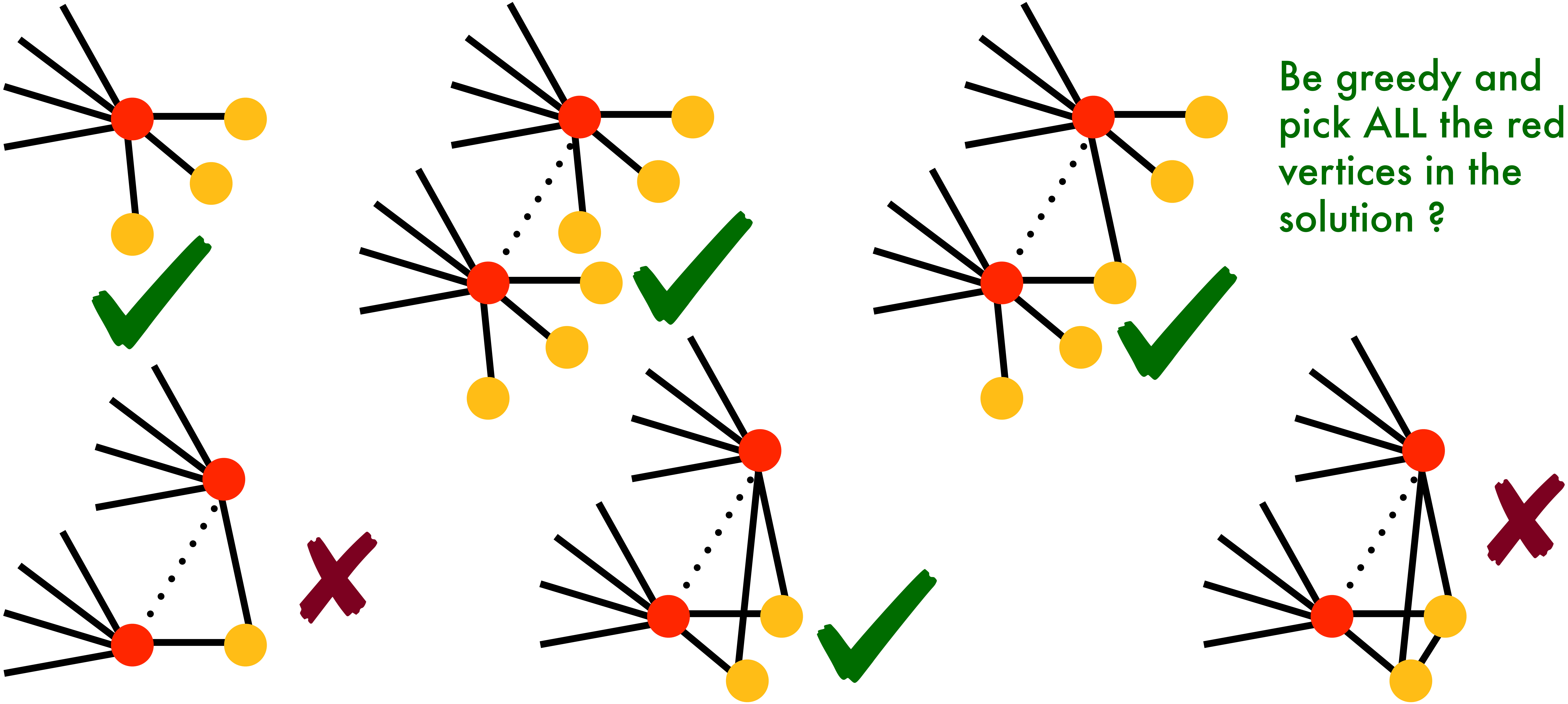


Be greedy and pick ALL the red vertices in the solution ?



Vertex Cover:

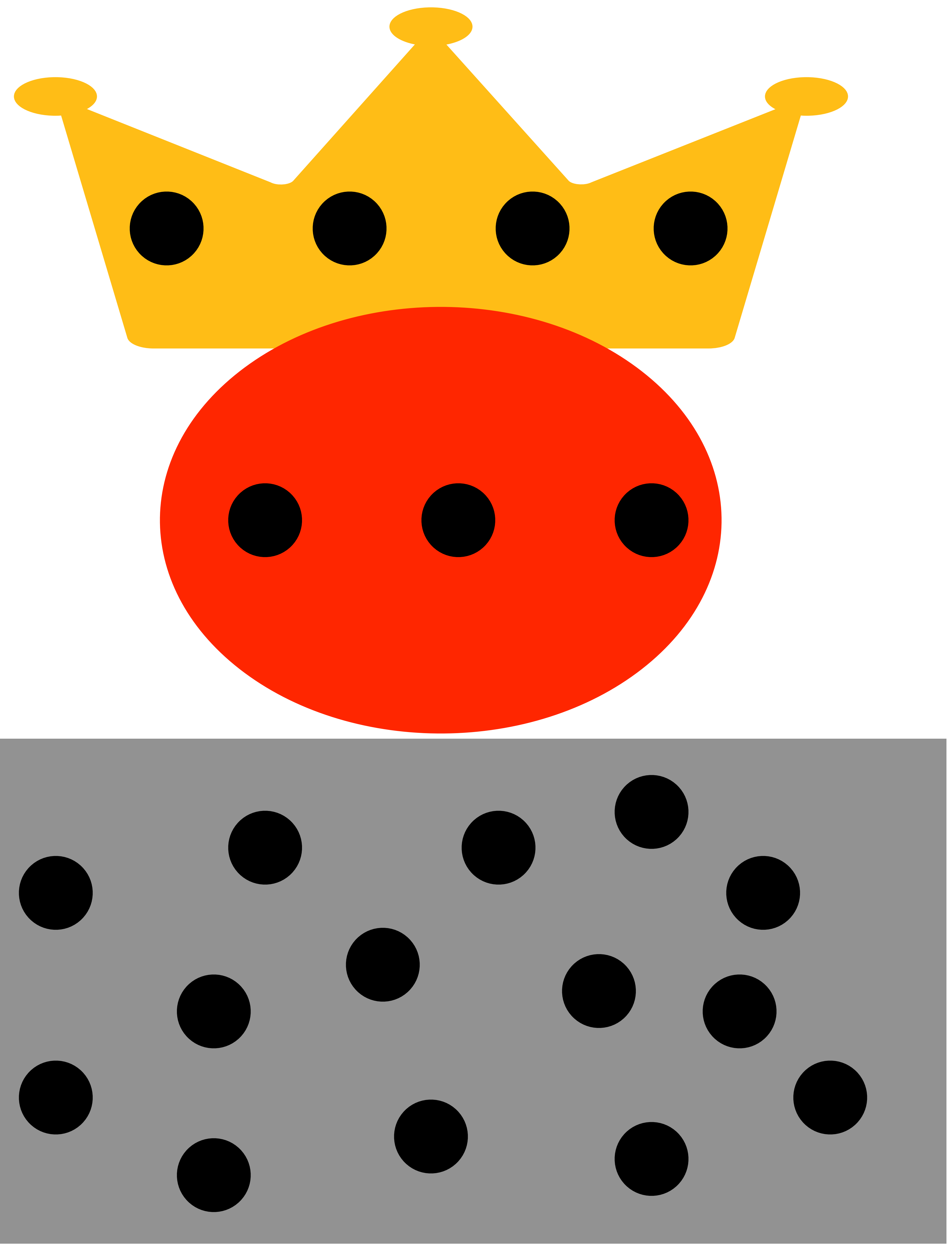
You can do what I can do, and at least one of us needs to do the job!



Be greedy and pick ALL the red vertices in the solution ?

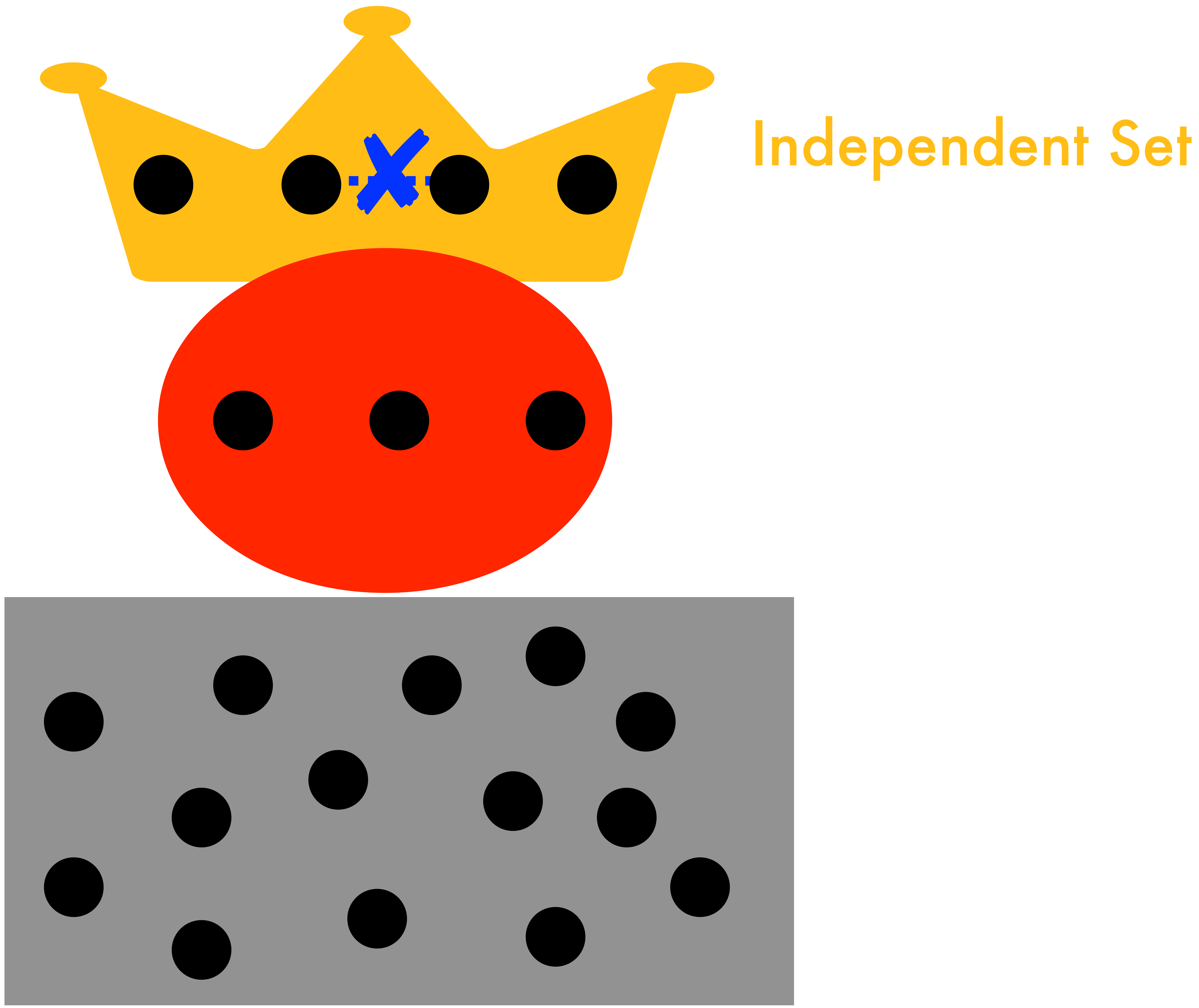
Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!



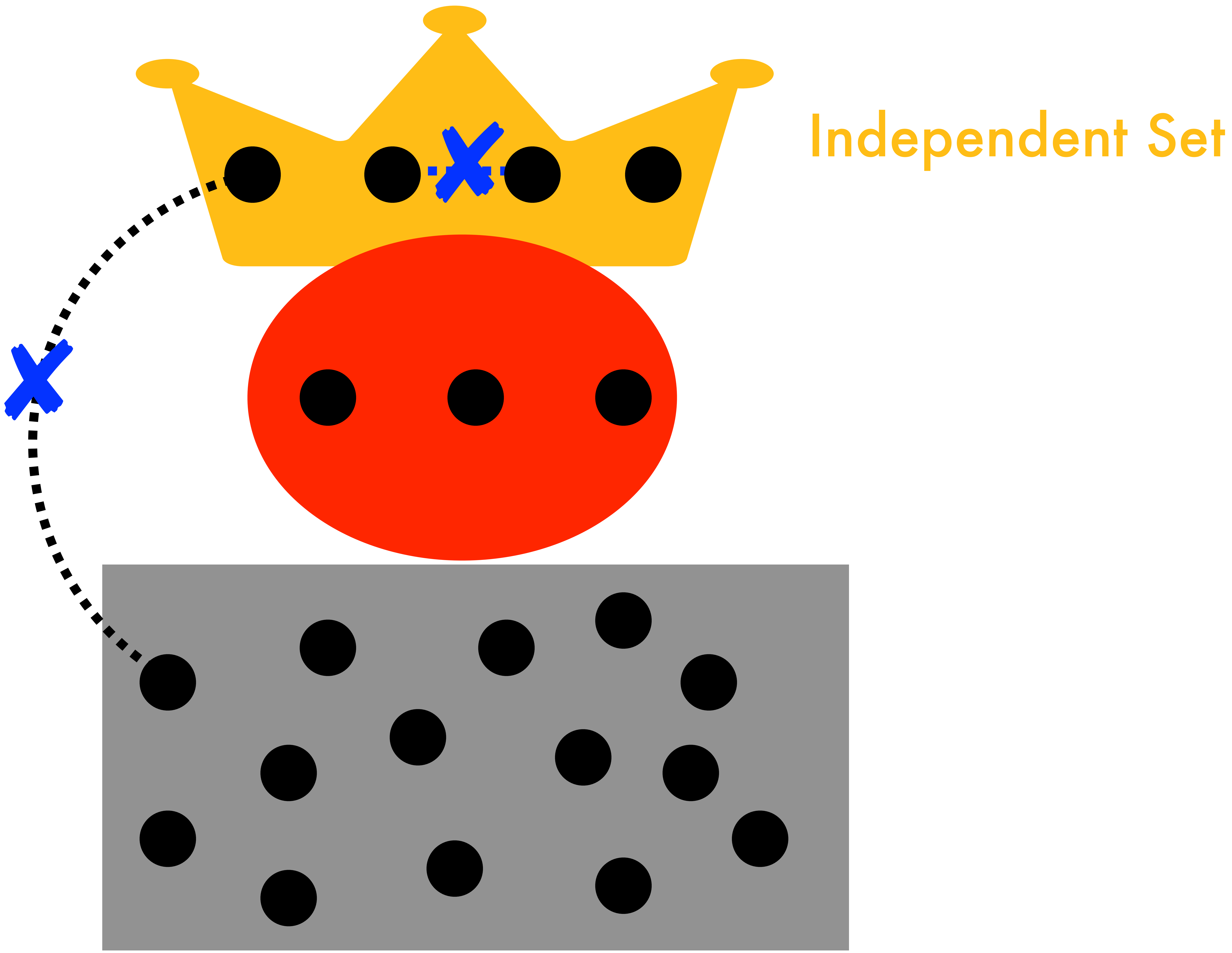
Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!



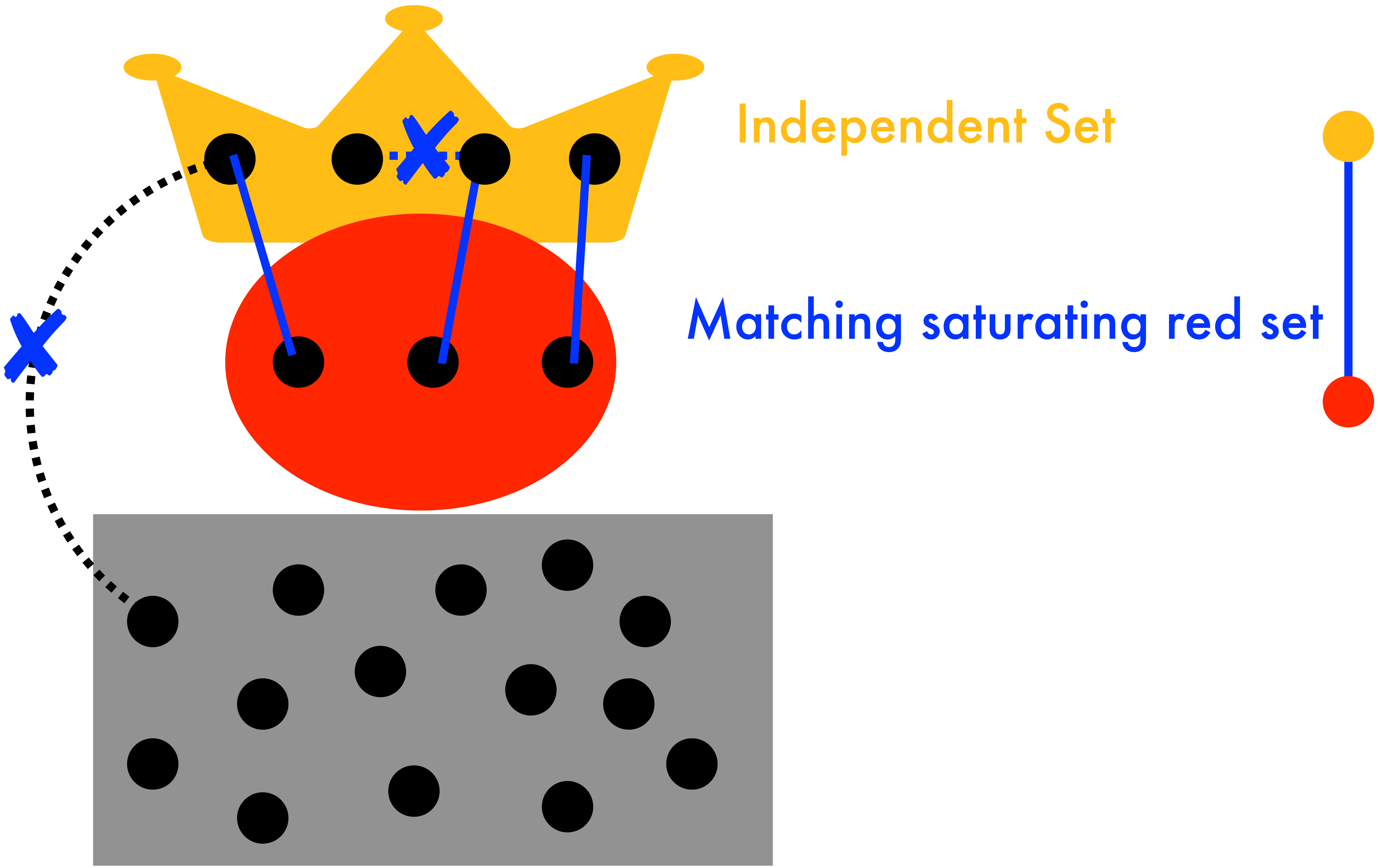
Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!



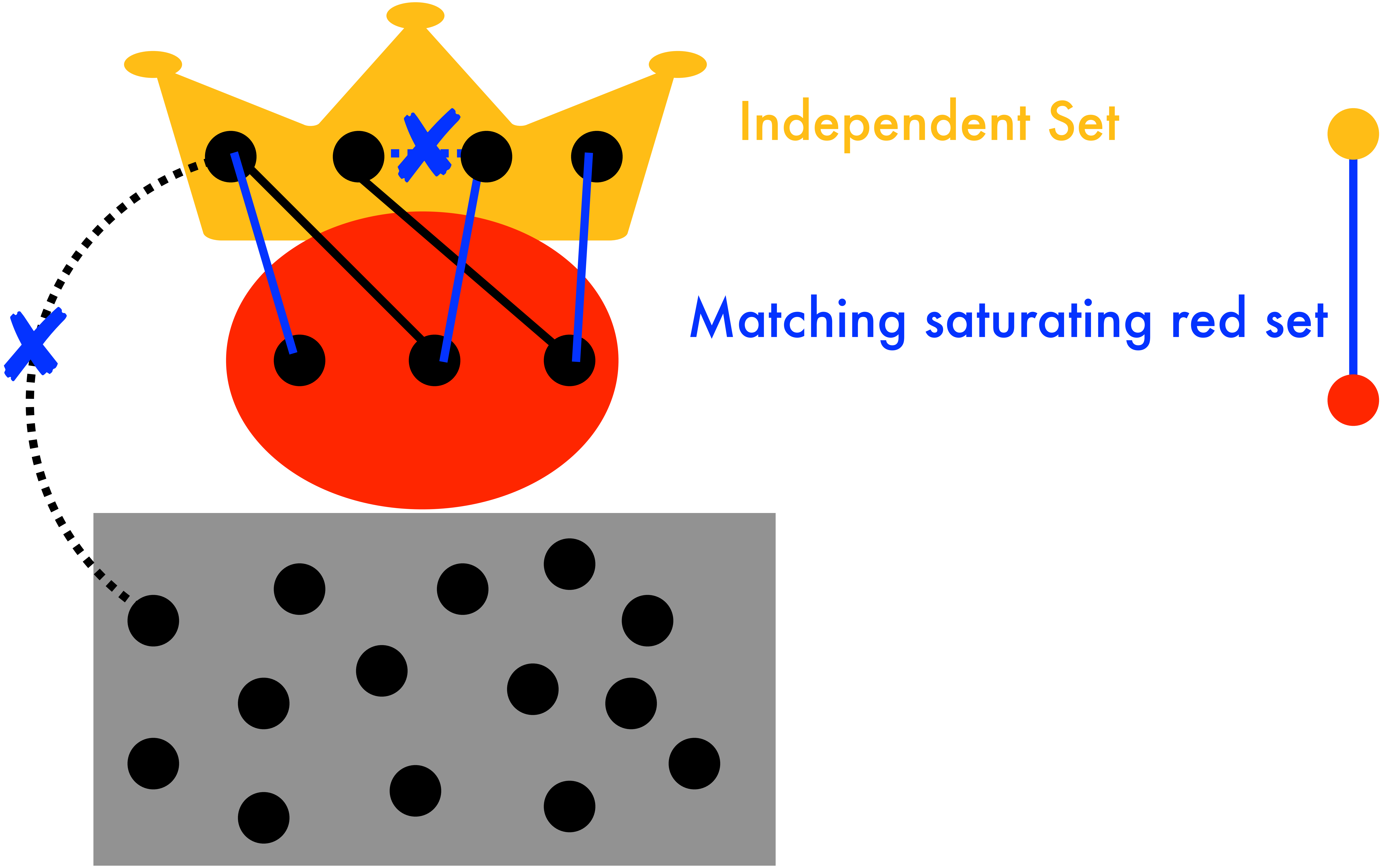
VERTEX COVER:

You can do what I can do, and at least one of us needs to do the job!



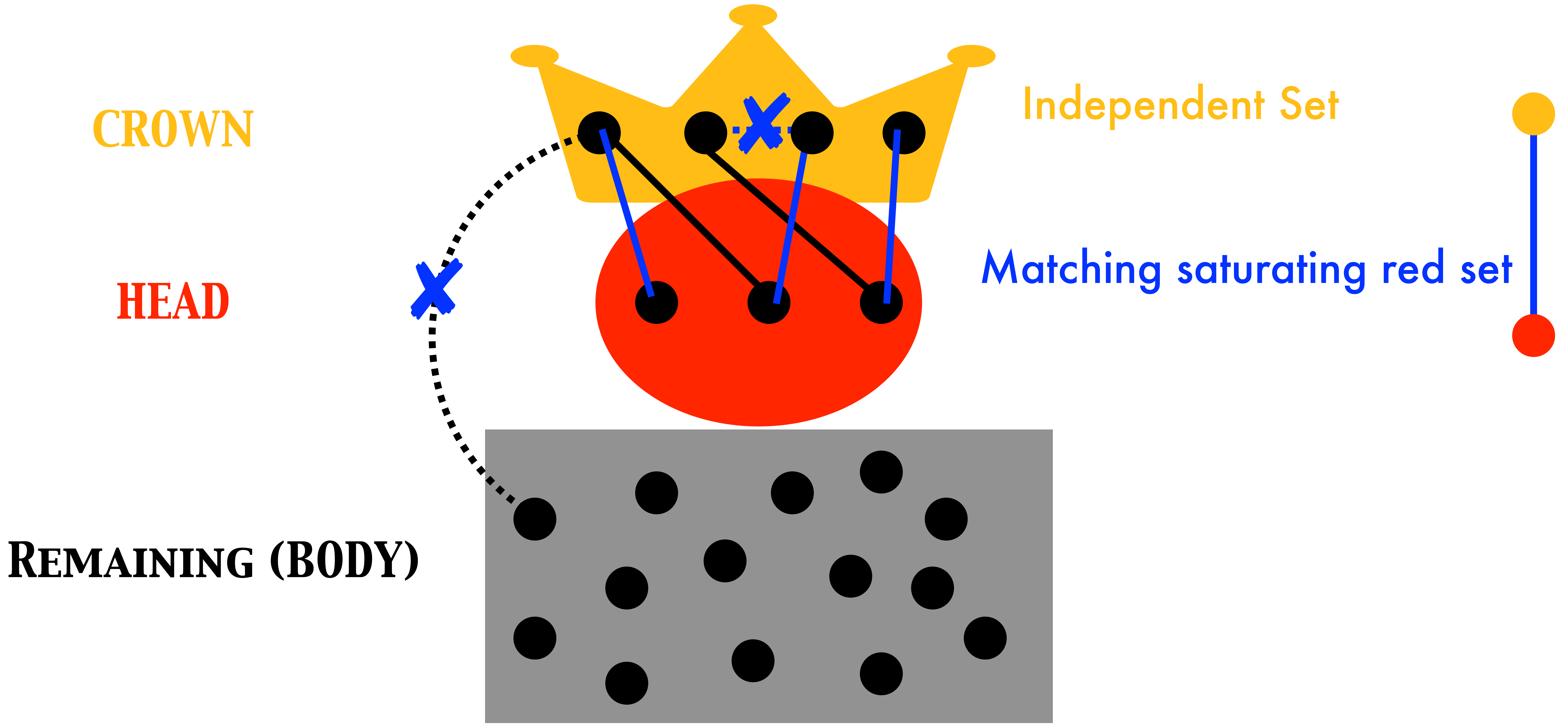
Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!



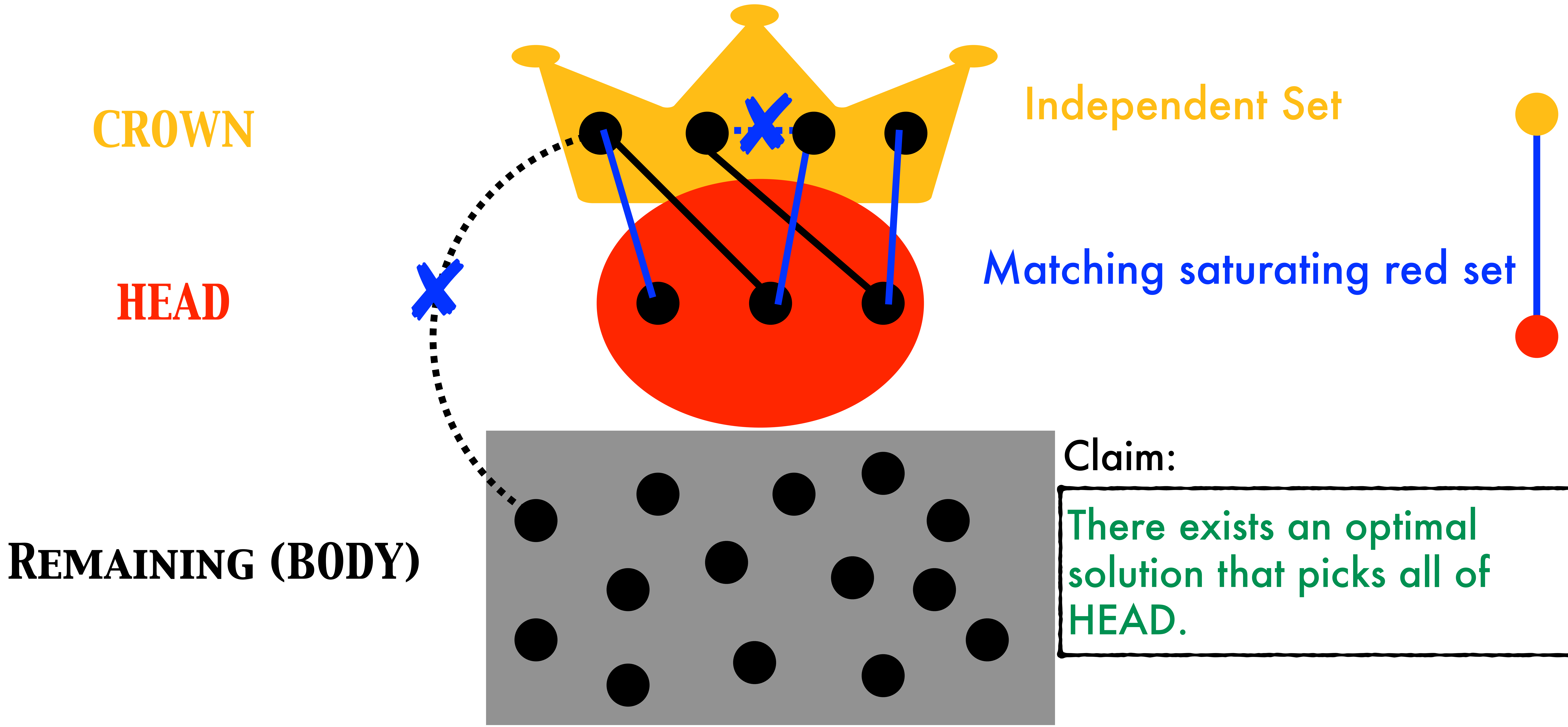
VERTEX COVER:

You can do what I can do, and at least one of us needs to do the job!



Vertex Cover:

You can do what I can do, and at least one of us needs to do the job!



Claim:

There exists an optimal solution that picks all of HEAD.

Vertex Cover:

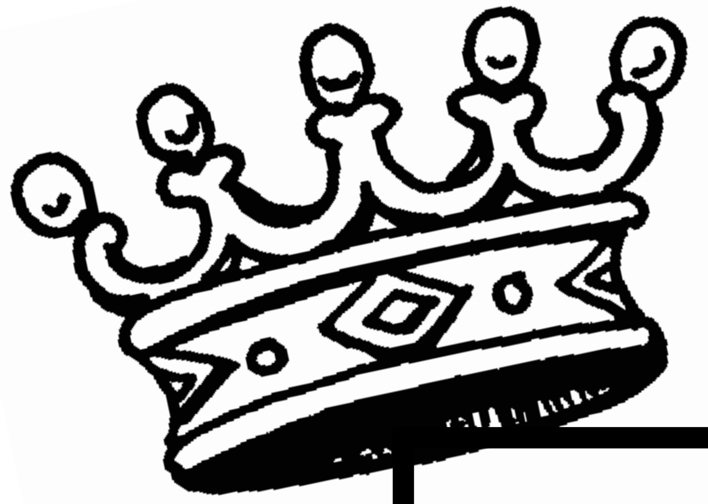
You can do what I can do, and at least one of us needs to do the job!

Reduction rule:

Pick the HEAD in the solution.
Decrease the budget by |HEAD|.
CROWN becomes isolated, so deleted it too.

How to find crown decompositions?

Why should they exist in graphs?

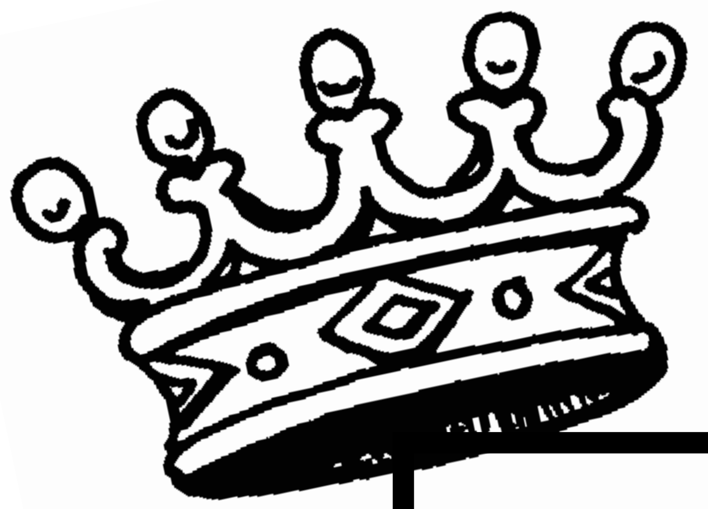


Crown Decomposition Lemma

For every graph G with no isolated vertices and integer k, either

- G has a crown decomposition (with non-empty head and crown), or
- G has a matching of size at least $k+1$, or
- $|V(G)| \leq 3k$

Such a decomposition can be found in polynomial time.



Crown Decomposition Lemma

For every graph G with no isolated vertices and integer k, either

- G has a crown decomposition (with non-empty head and crown), or
- G has a matching of size at least $k+1$, or
- $|V(G)| \leq 3k$

Such a decomposition can be found in polynomial time.

Vertex Cover admits a kernel with 3k vertices.

Remembering the Classics

Remembering the Classics

Kőnig's theorem

In a bipartite graph,
|minimum vertex cover|=
|size of maximum matching|.

Remembering the Classics

Kőnig's theorem

In a bipartite graph,
|minimum vertex cover|=
|size of maximum matching|.

Hall's theorem

In a bipartite graph $G=(A,B)$,
there exists a matching saturating A (into B)
if and only if
$|N(X)| \geq |X|$, $X \subseteq A$.

REMEMBERING THE CLASSICS

Kőnig's theorem

In a bipartite graph,
|minimum vertex cover|=
|size of maximum matching|.

Hall's theorem

In a bipartite graph G=(A,B),
there exists a matching saturating A (into B)
if and only if
$|N(X)| \geq |X|$, $X \subseteq A$.

Hopcroft-Karp algorithm

Given a bipartite graph G=(A,B) on n vertices and m edges,
one can find in $O(m\, n^{1/2})$ time a maximum matching and a minimum vertex cover in G.
Moreover, one can also find a matching saturating A or an inclusion-wise minimal set $X \subseteq A$
such that $|N(X)| < |X|$.

Remembering the Classics

Kőnig's theorem

In a bipartite graph,
|minimum vertex cover|=
|size of maximum matching|.

Hall's theorem

In a bipartite graph $G=(A,B)$,
there exists a matching saturating A (into B)
if and only if
$|N(X)| \geq |X|$, $X \subseteq A$.

Hopcroft-Karp algorithm

Given a bipartite graph $G=(A,B)$ on n vertices and m edges,
one can find in $O(m\, n^{1/2})$ time a maximum matching and a minimum vertex cover in G.
Moreover, one can also find a matching saturating A or an inclusion-wise minimal set $X \subseteq A$
such that $|N(X)| < |X|$.

Hall's set

Proof of the crown Decomposition Lemma

Proof of the crown Decomposition Lemma

Approximate Vertex Cover: Find a maximal matching M.
If its size is ≥ k+1, then return the matching M.
Otherwise, get vertex cover (V(M)) of size at most 2k.

PROOF OF THE CROWN DECOMPOSITION LEMMA

Independent Set

$|V(M)| \leq 2k$

Approximate Vertex Cover: Find a maximal matching M.
If its size is ≥ k+1, then return the matching M.
Otherwise, get vertex cover (V(M)) of size at most 2k.

Proof of the crown Decomposition Lemma

Independent Set

$|I| > k$

$|V(M)| \leq 2k$

Proof of the crown Decomposition Lemma

Independent Set

$|I| > k$

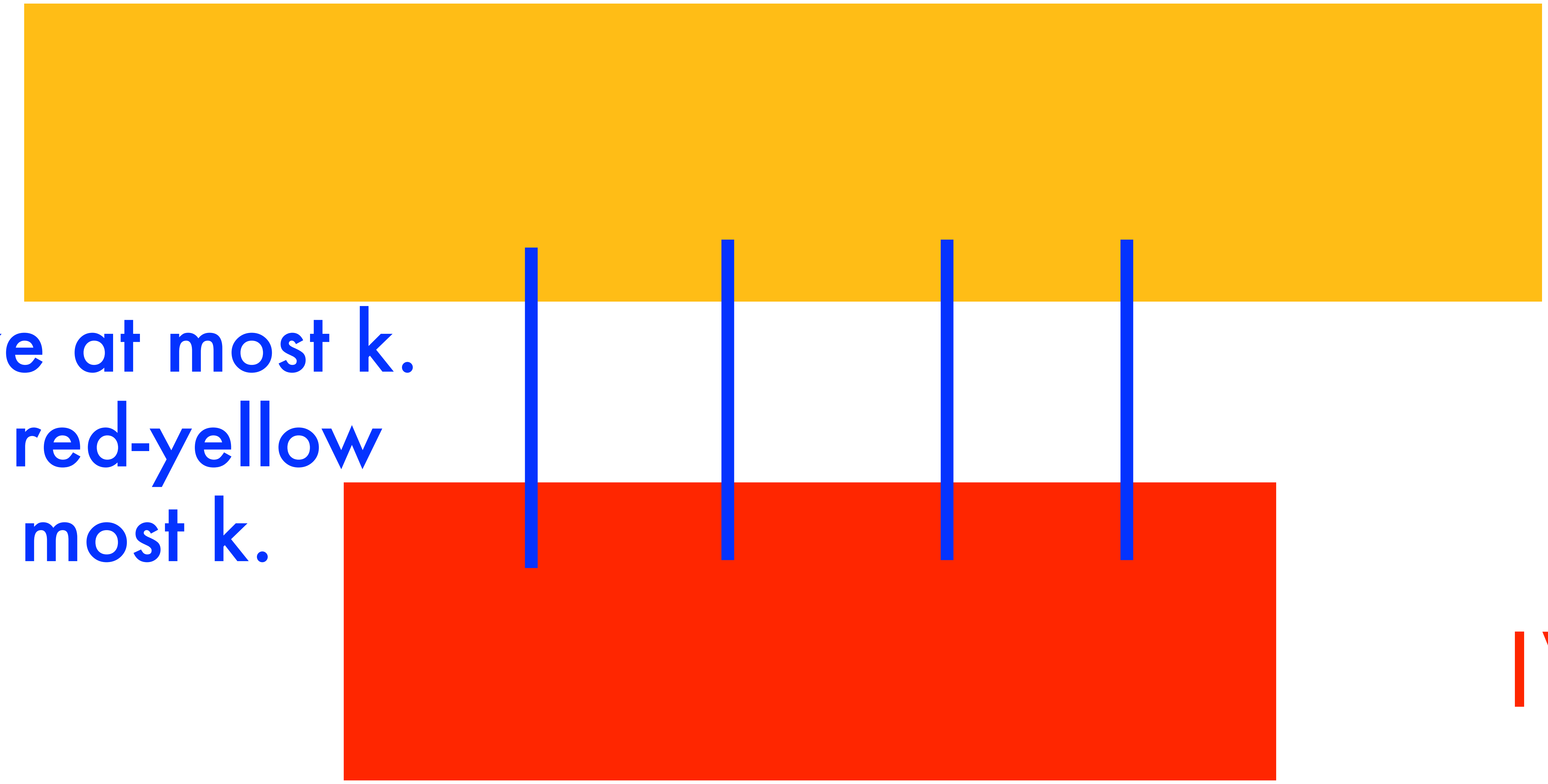
$|V(M)| \leq 2k$

Find a maximum matching between V(M) and I.
If size is at least k+1, return the matching.

PROOF OF THE CROWN DECOMPOSITION LEMMA

Independent Set

$|I| > k$



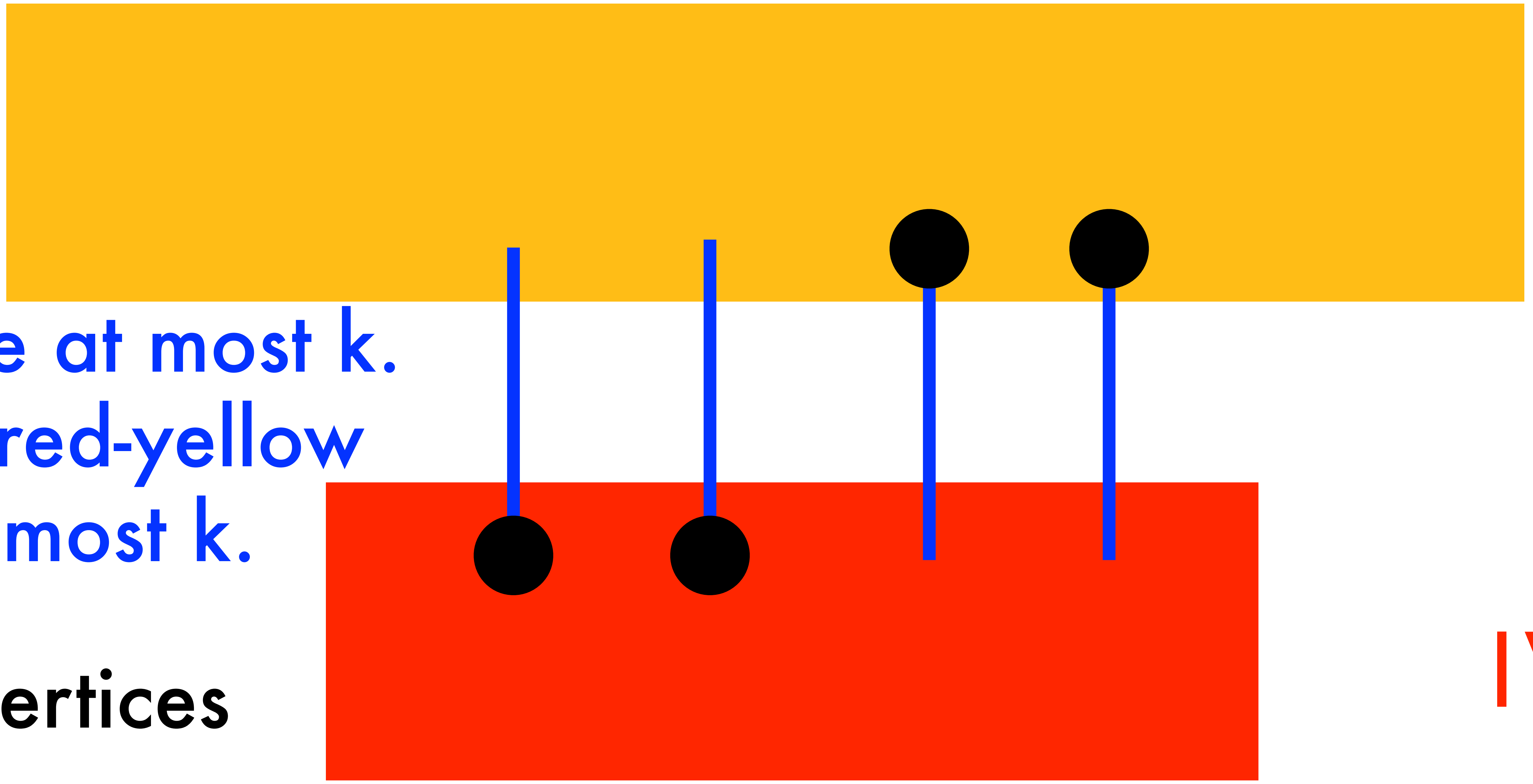
Blue matching of size at most k.
Vertex Cover of the red-yellow bipartite graph is at most k.

$|V(M)| \leq 2k$

Find a maximum matching between V(M) and I.
If size is at least k+1, return the matching.

PROOF OF THE CROWN DECOMPOSITION LEMMA

Independent Set



$|I| > k$

Blue matching of size at most k.
Vertex Cover of the red-yellow bipartite graph is at most k.

● Vertex Cover vertices

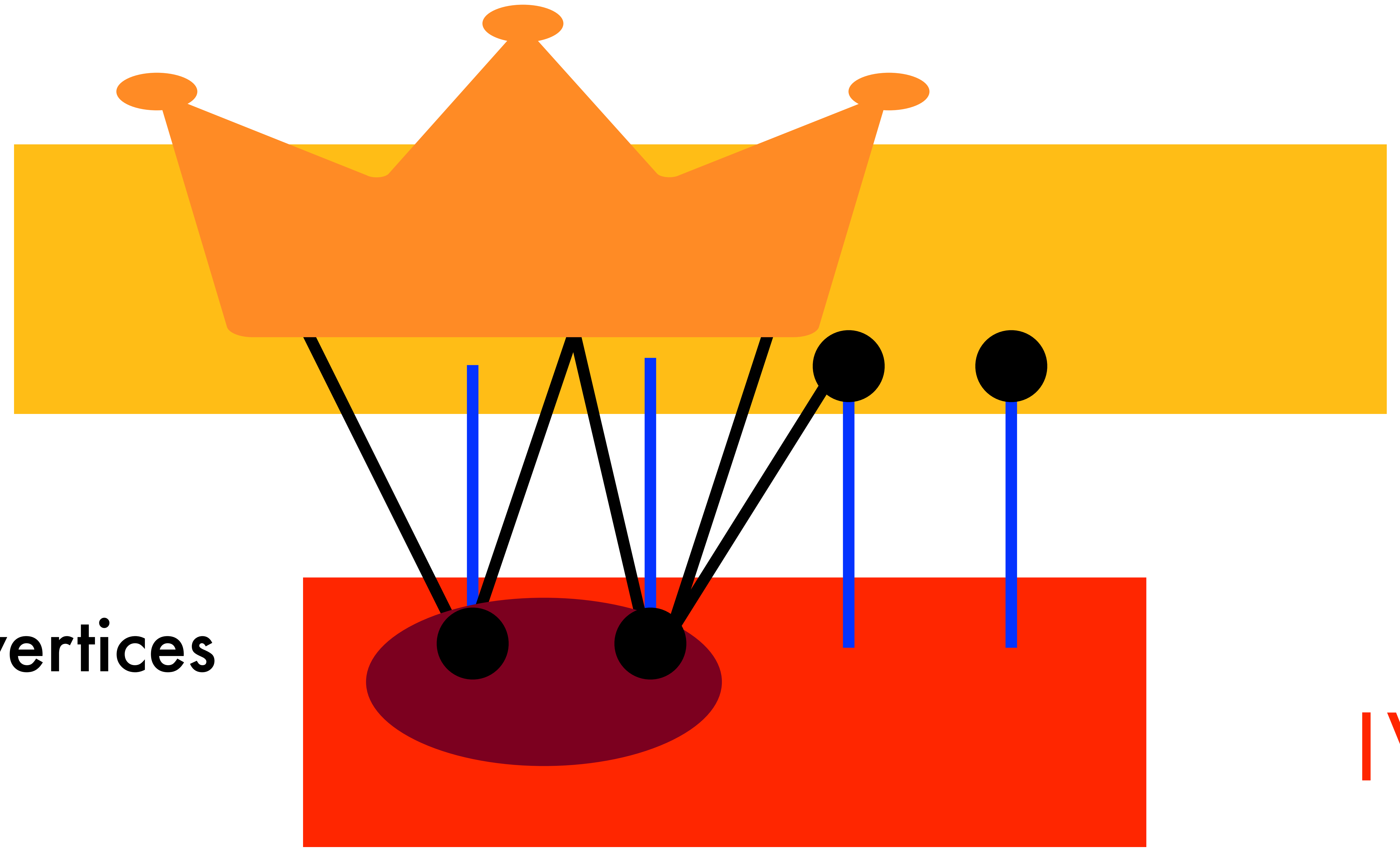
$|V(M)| \leq 2k$

Find a maximum matching between V(M) and I.
If size is at least k+1, return the matching.

Proof of the crown Decomposition Lemma

Independent Set

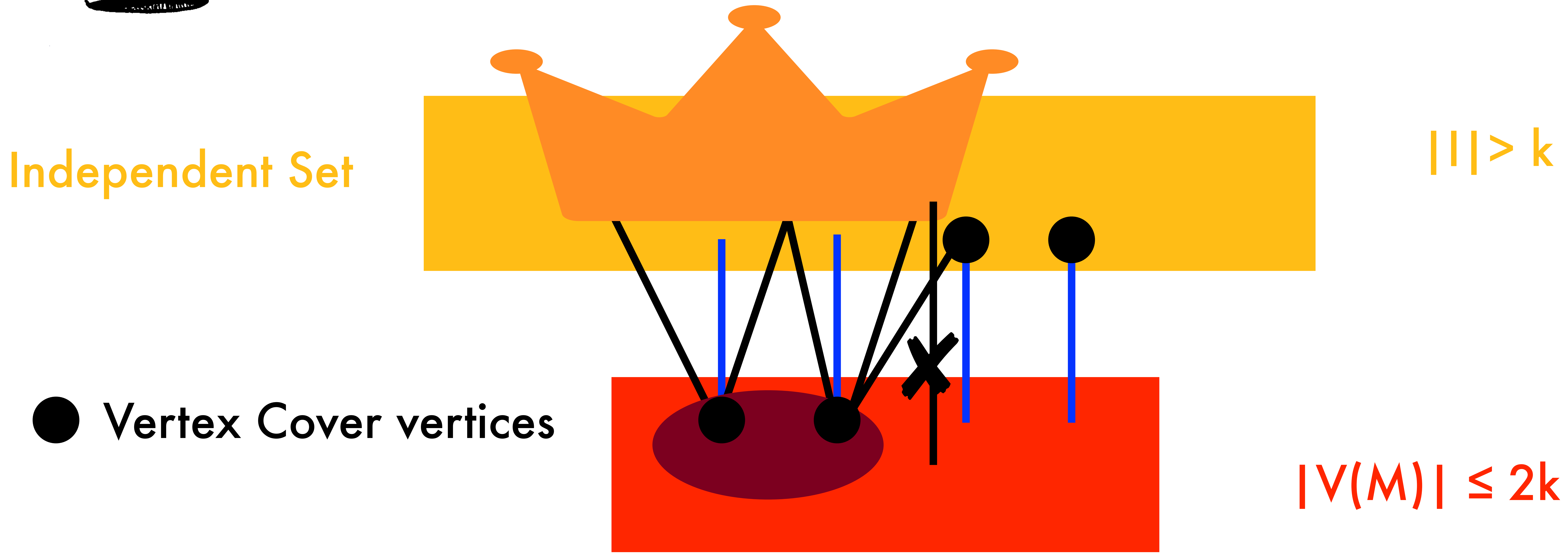
$|I| > k$



● Vertex Cover vertices

$|V(M)| \leq 2k$

Proof of the crown Decomposition Lemma



Linear Programming (LP) based kernel for VERTEX COVER

VERTEX COVER admits a kernel with $2k$ vertices.

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

Linear Program (LP) for Vertex Cover

P time

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0,1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

OPT_{LP}

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

Linear Program (LP) for Vertex Cover

P time

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

OPT_{ILP}

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

OPT_{LP}

$$\boxed{\text{minimize} \sum_{i \in \{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

Linear Program (LP) for Vertex Cover

P time

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$OPT_{ILP} \geq OPT_{LP}$

OPT_{ILP}

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

OPT_{LP}

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

Linear Program (LP) for Vertex Cover

P time

LP for Vertex Cover

Graph G, $V(G)=\{v_1, \ldots, v_n\}$

$OPT_{ILP} \geq OPT_{LP}$

OPT_{ILP}

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$x_i \in \{0, 1\} \qquad \forall v_i \in V(G)$$

Integer Linear Program (ILP) for Vertex Cover

NP-Complete

OPT_{LP}

$$\boxed{\text{minimize} \sum_{i\in\{1,\ldots,n\}} x_i}$$

$$x_i + x_j \geq 1 \qquad \forall (v_i, v_j) \in E(G)$$

$$0 \leq x_i \leq 1 \qquad \forall v_i \in V(G)$$

Linear Program (LP) for Vertex Cover

P time

$$V(G) = V_{<1/2} \uplus V_{=1/2} \uplus V_{>1/2}$$

$$V_{<1/2} = \{v_i \in V(G) : x_i < 1/2\}$$

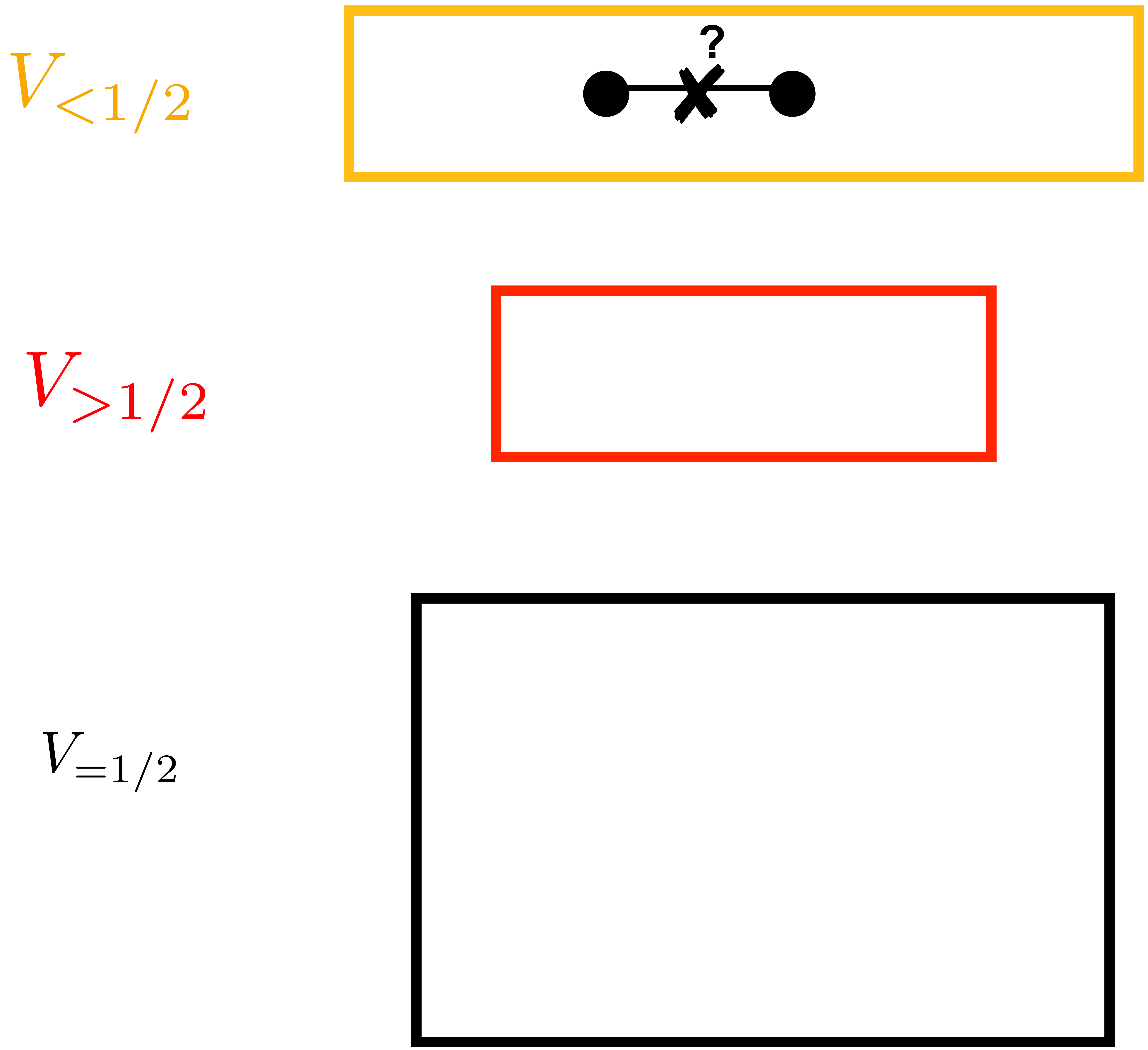
$$V_{=1/2} = \{v_i \in V(G) : x_i = 1/2\}$$

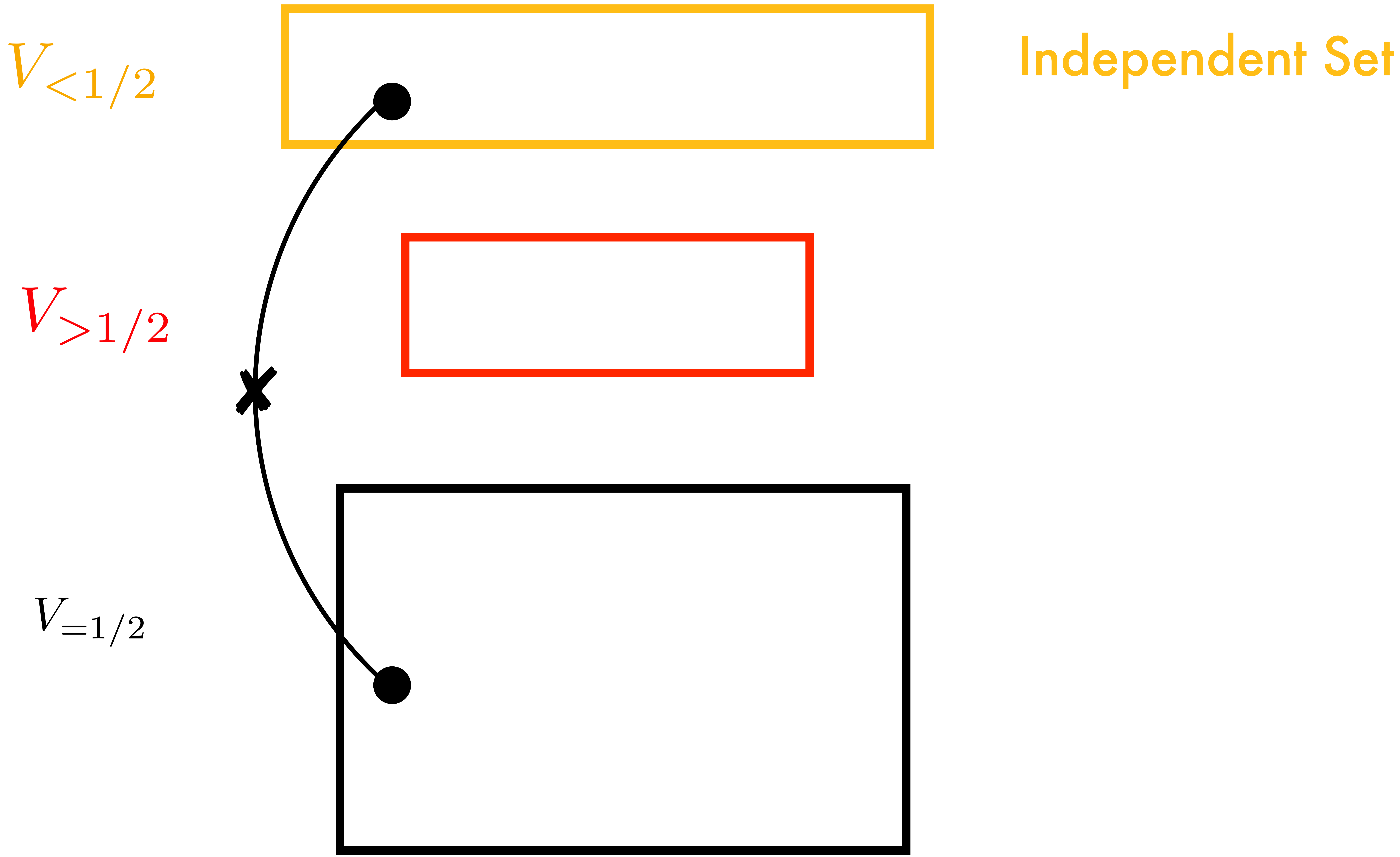
$$V_{>1/2} = \{v_i \in V(G) : x_i > 1/2\}$$

$V_{<1/2}$

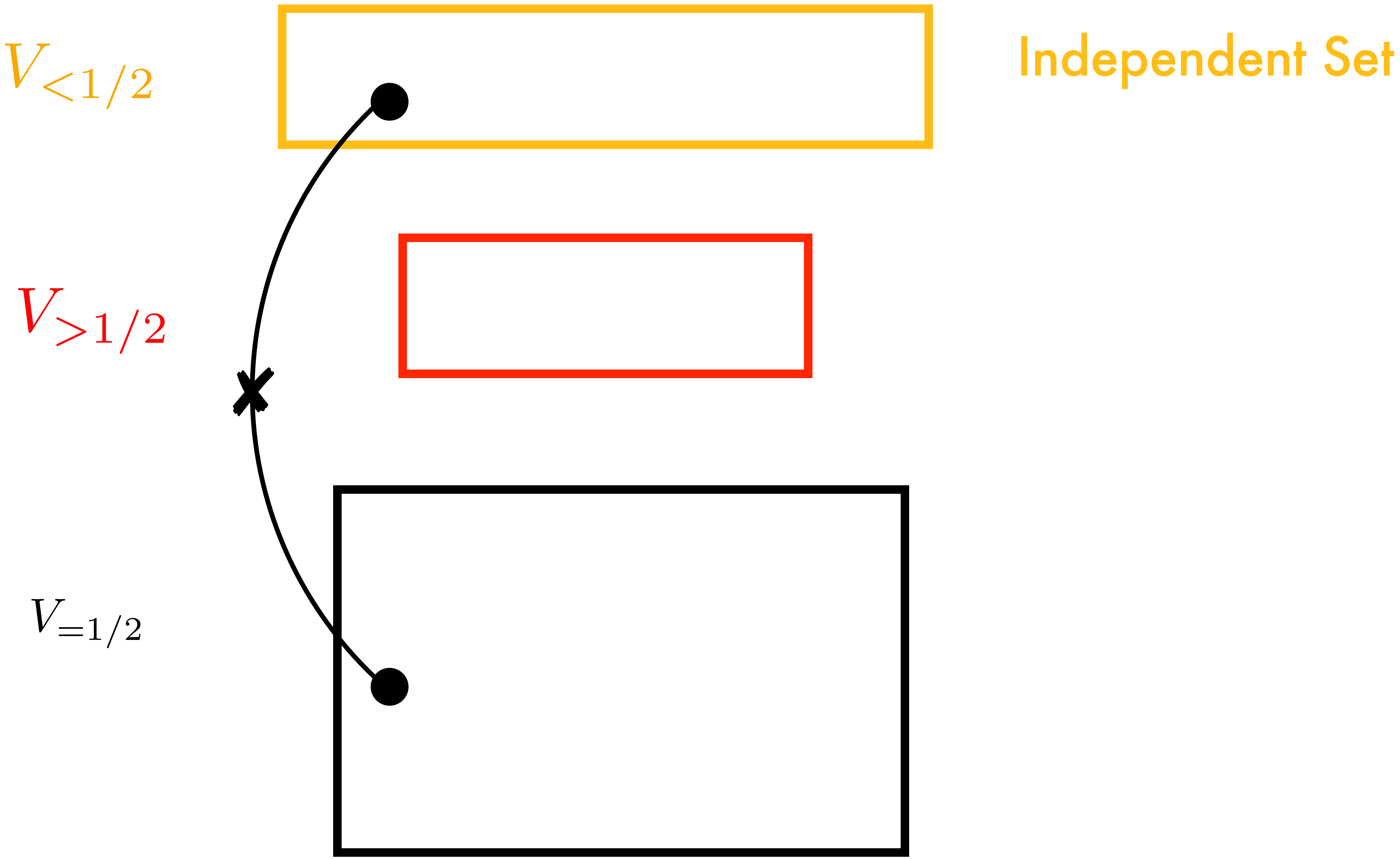
$V_{>1/2}$

$V_{=1/2}$

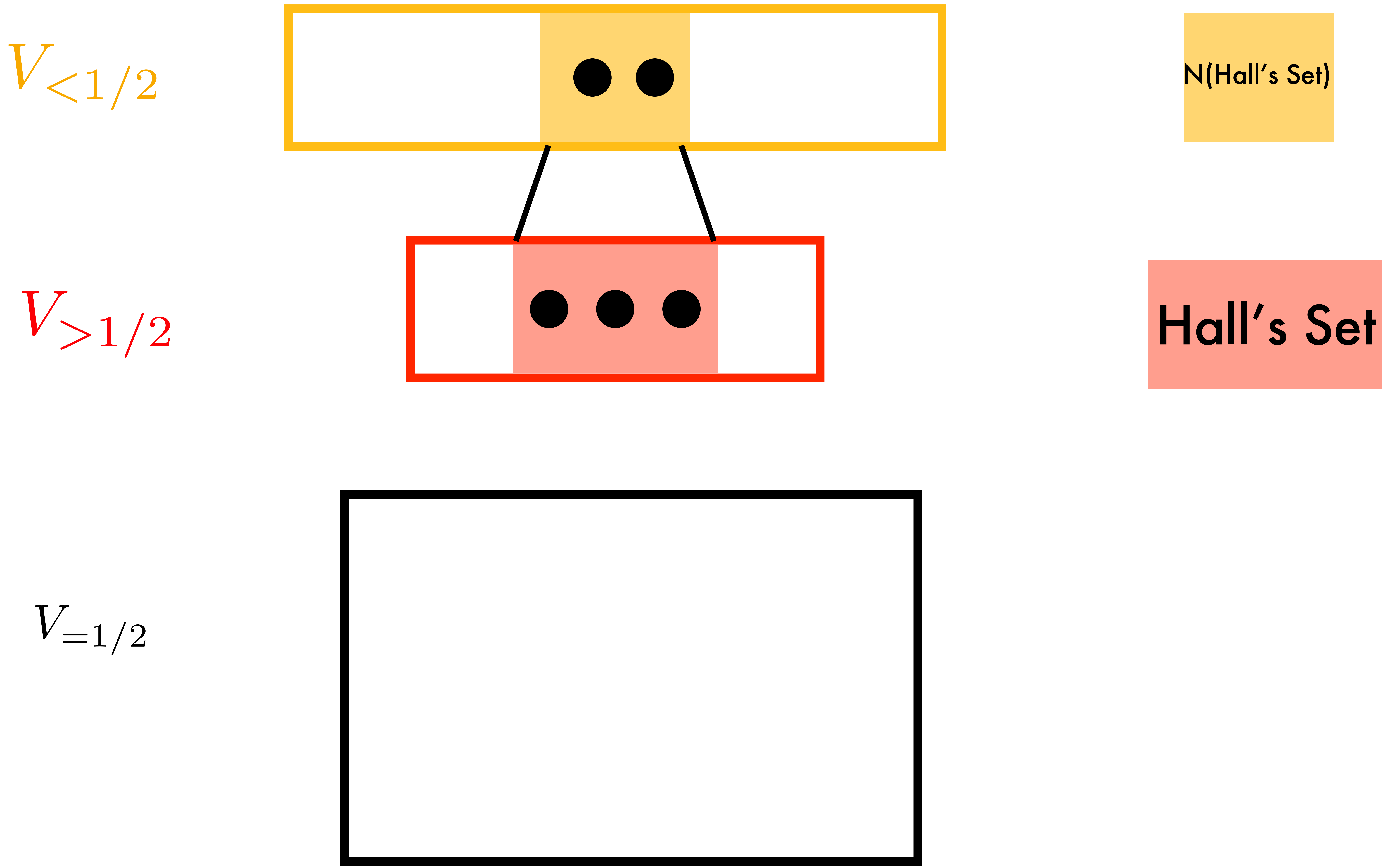

$V_{<1/2}$
?
$V_{>1/2}$
$V_{=1/2}$


$V_{<1/2}$
Independent Set
$V_{>1/2}$
$V_{=1/2}$

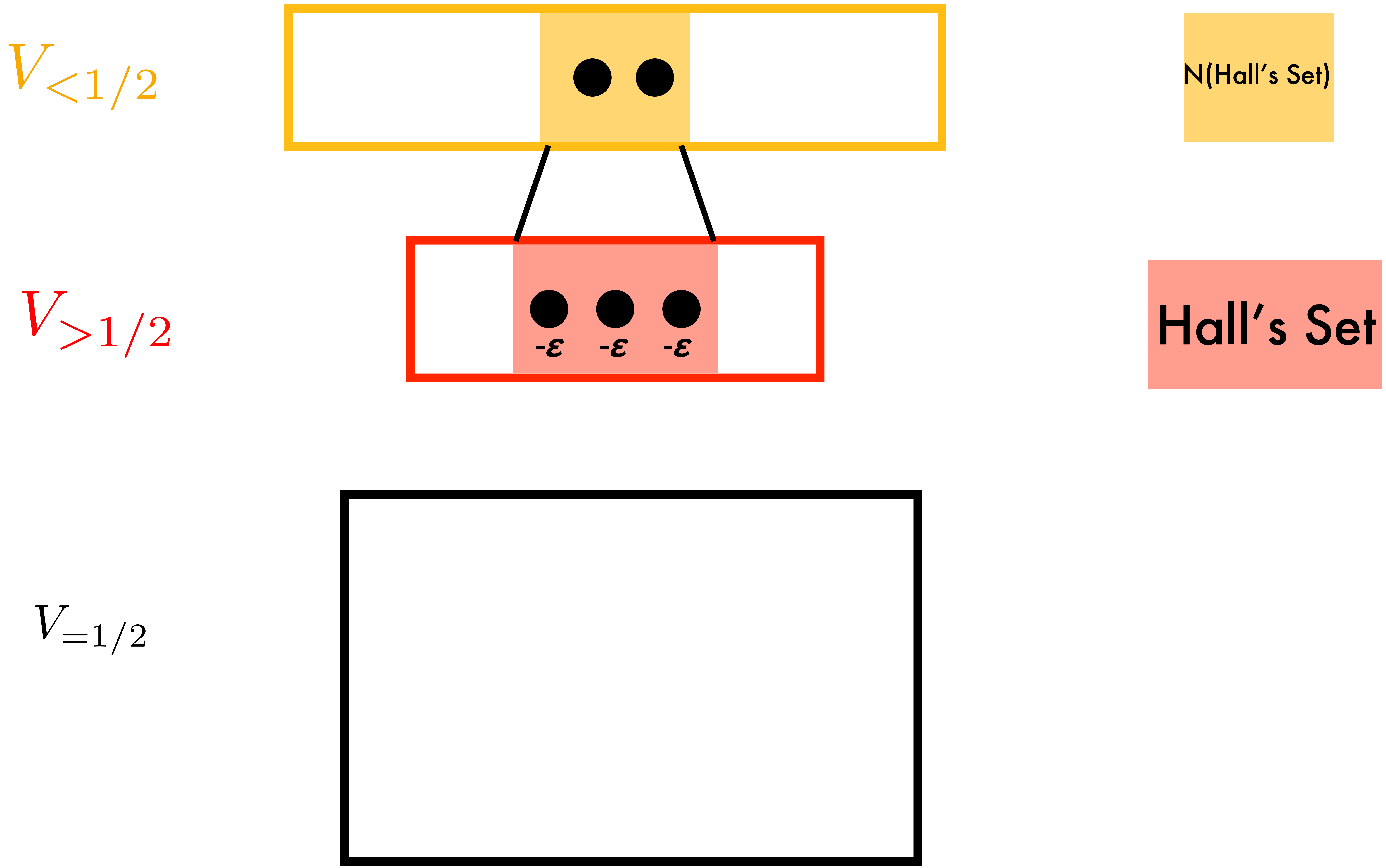
Show that there is a matching saturating $V_{>1/2}$ into $V_{<1/2}$.



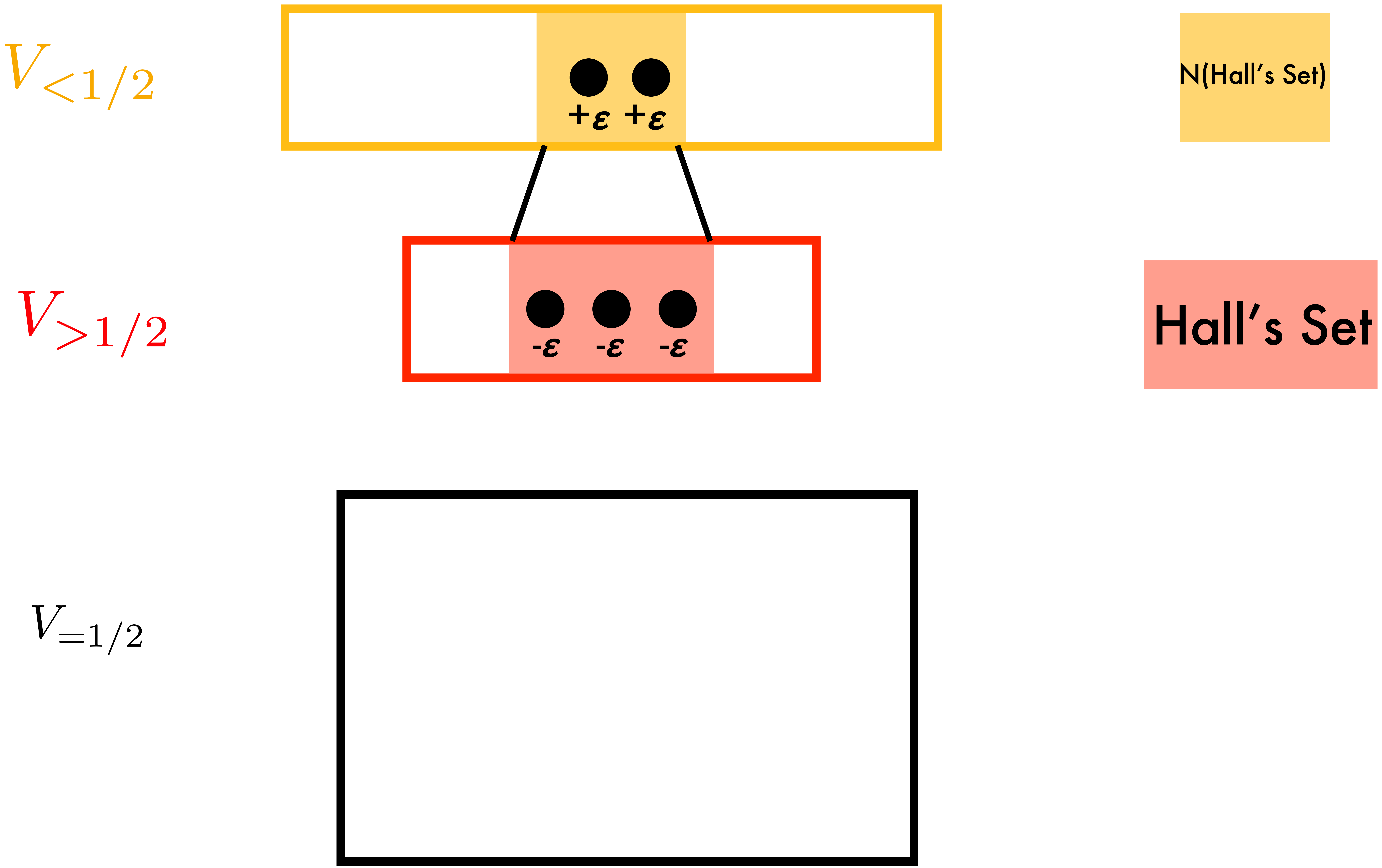
Show that there is a matching saturating $V_{>1/2}$ into $V_{<1/2}$.



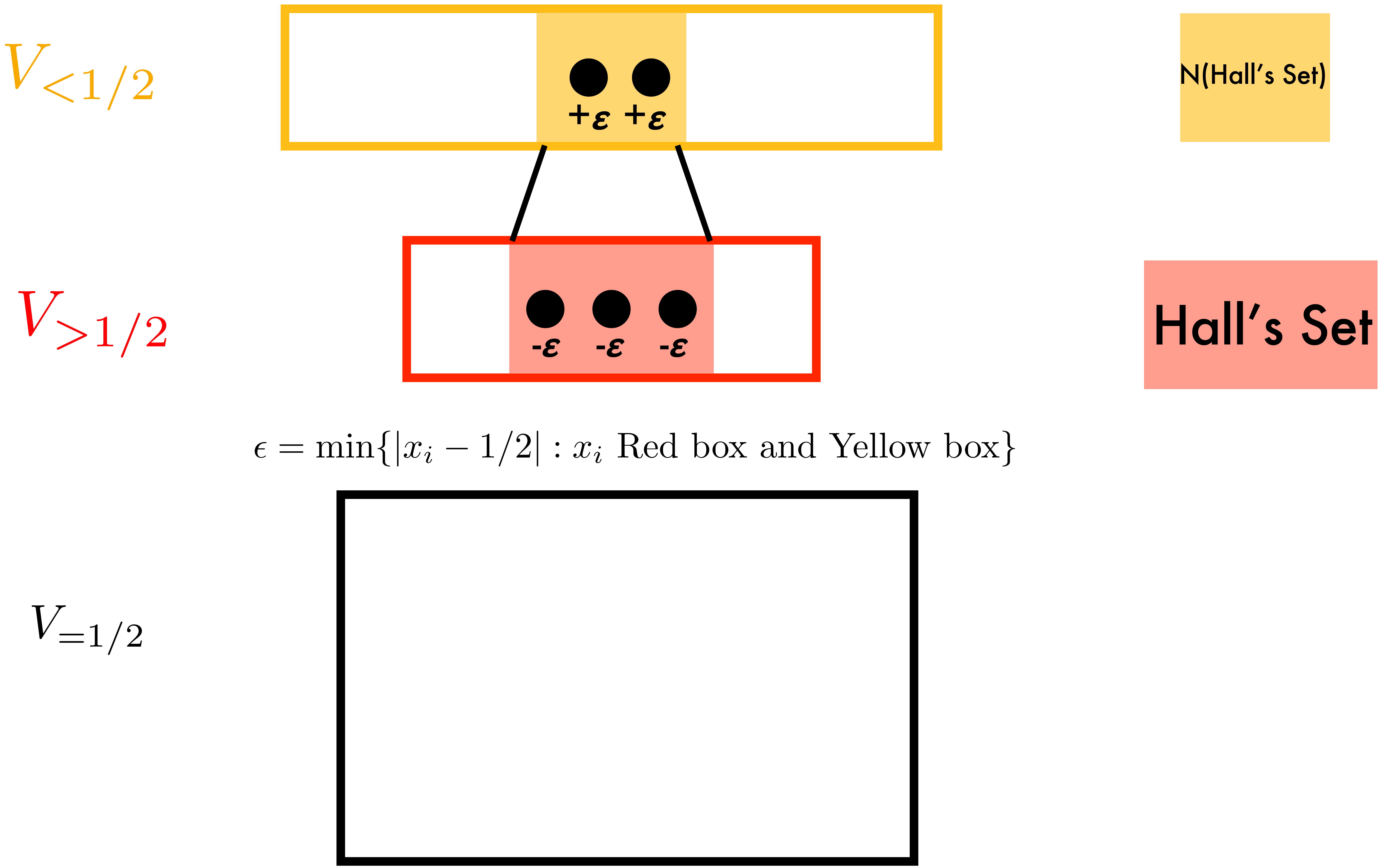
Show that there is a matching saturating $V_{>1/2}$ into $V_{<1/2}$.



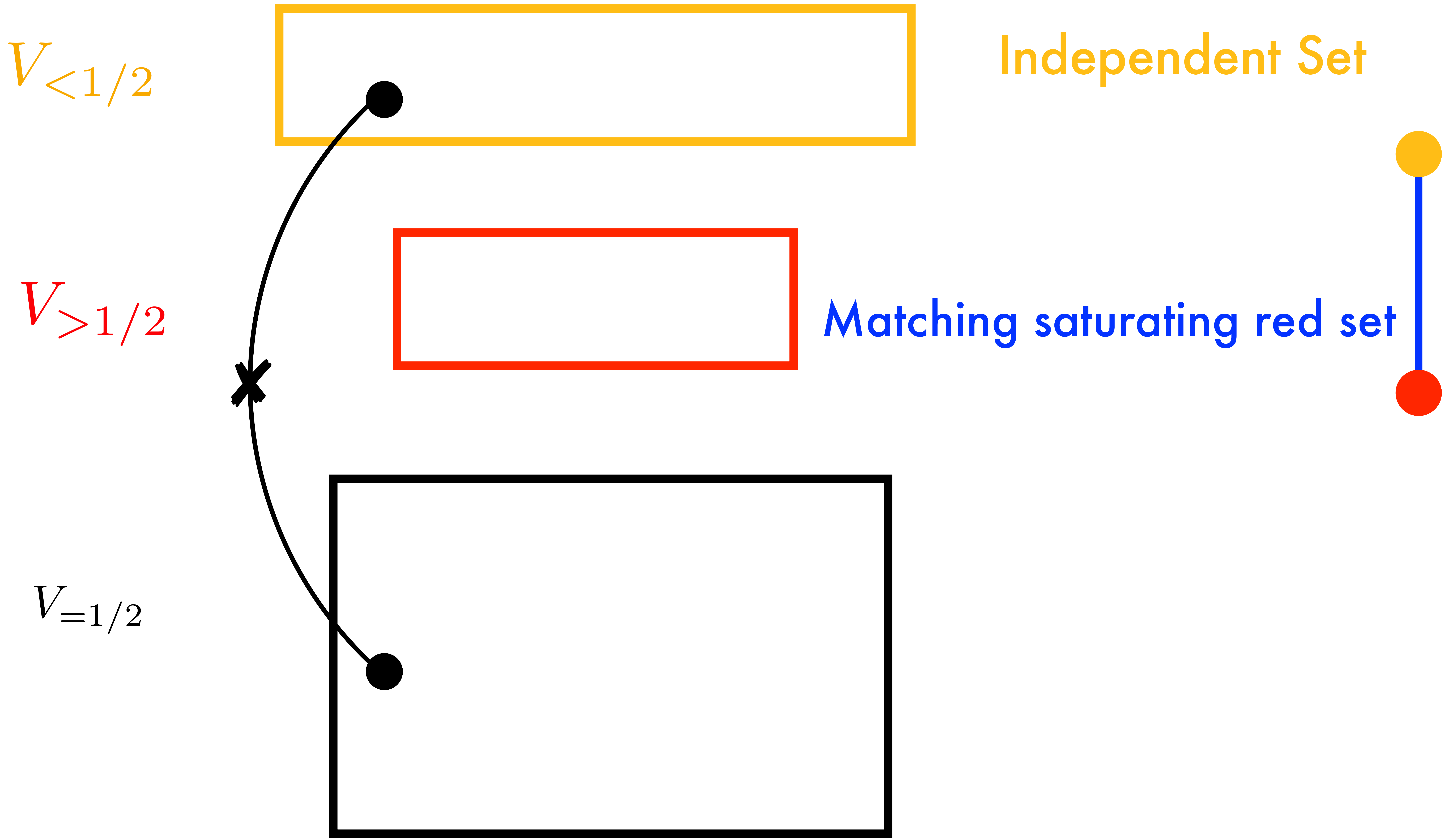
Show that there is a matching saturating $V_{>1/2}$ into $V_{<1/2}$.



Show that there is a matching saturating $V_{>1/2}$ into $V_{<1/2}$.



Crown Decomposition



Reduction Rule using LP values

Nemhausser-Trotter's Theorem:

There is a minimum vertex cover, say S, of G such

$$V_{>1/2} \subseteq S \subseteq V_{>1/2} \cup V_{=1/2}$$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1,\ldots,n\} x_i = 1/2$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1, \ldots, n\} x_i = 1/2$

$$OPT_{LP} = \sum_{i \in \{1,\ldots,n\}} x_i = n/2$$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1, \dots, n\} x_i = 1/2$

$$OPT_{ILP} \geq \; OPT_{LP} = \sum_{i \in \{1, \dots, n\}} x_i = n/2$$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1, \ldots, n\} x_i = 1/2$

$$k \geq OPT_{ILP} \geq OPT_{LP} = \sum_{i \in \{1, \ldots, n\}} x_i = n/2$$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1,\ldots,n\} x_i = 1/2$

$$k \geq OPT_{ILP} \geq \; OPT_{LP} = \sum_{i \in \{1,\ldots,n\}} x_i = n/2$$

$$n \leq 2k$$

When N-T rule is not applicable?

All vertices have LP values =1/2. $\forall i \in \{1, \ldots, n\} x_i = 1/2$

$$k \geq OPT_{ILP} \geq OPT_{LP} = \sum_{i \in \{1,\ldots,n\}} x_i = n/2$$

$$n \leq 2k$$

VC admits a kernel with 2k vertices.

Expansion Lemma

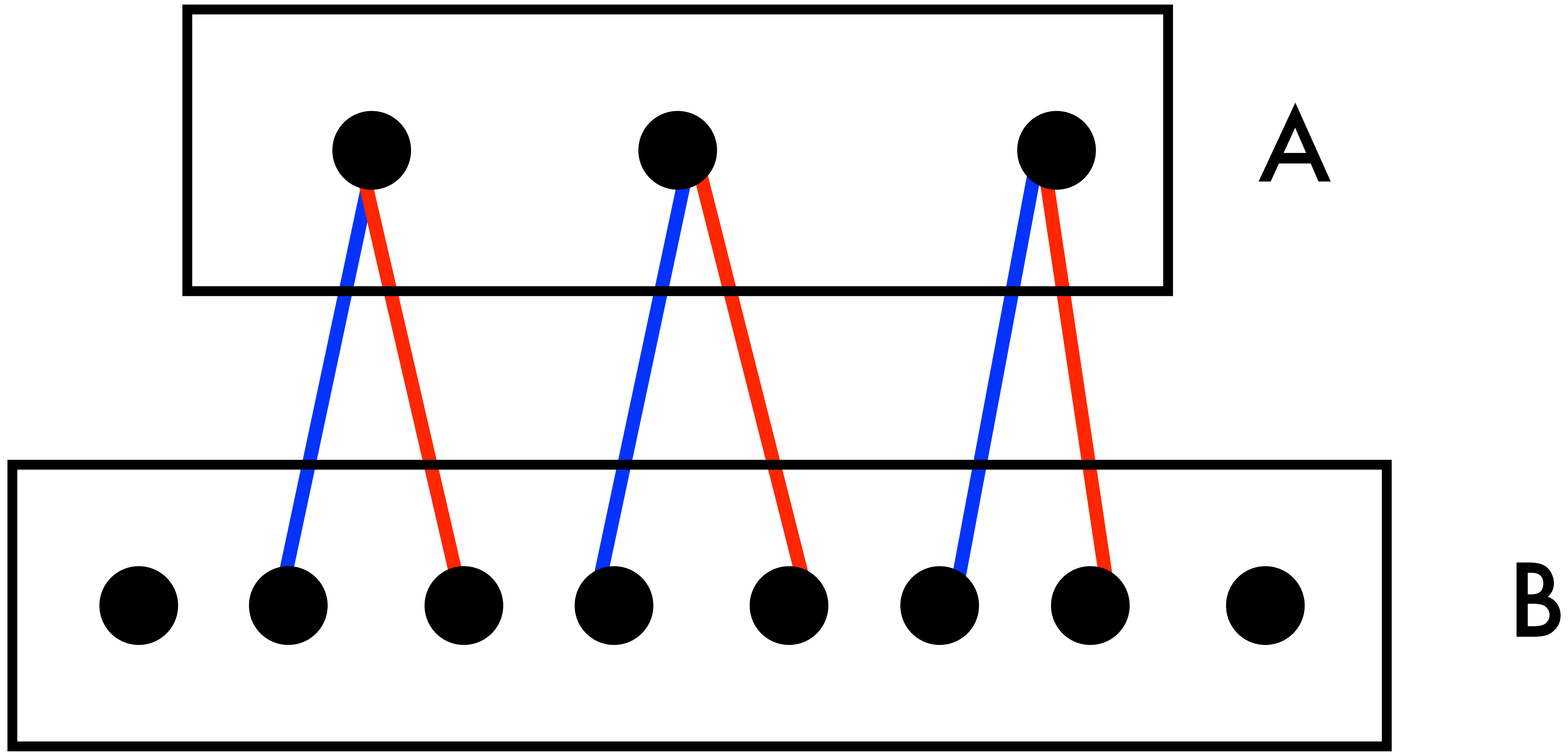
q-CROWN DECOMPOSITION

q-Expansion

Bipartite graph G=(A,B),
a set of edges M is a q-expansion of A into B if
M is a disjoint union of |A| q-stars each with its centre vertex in A
and the leaves in B.

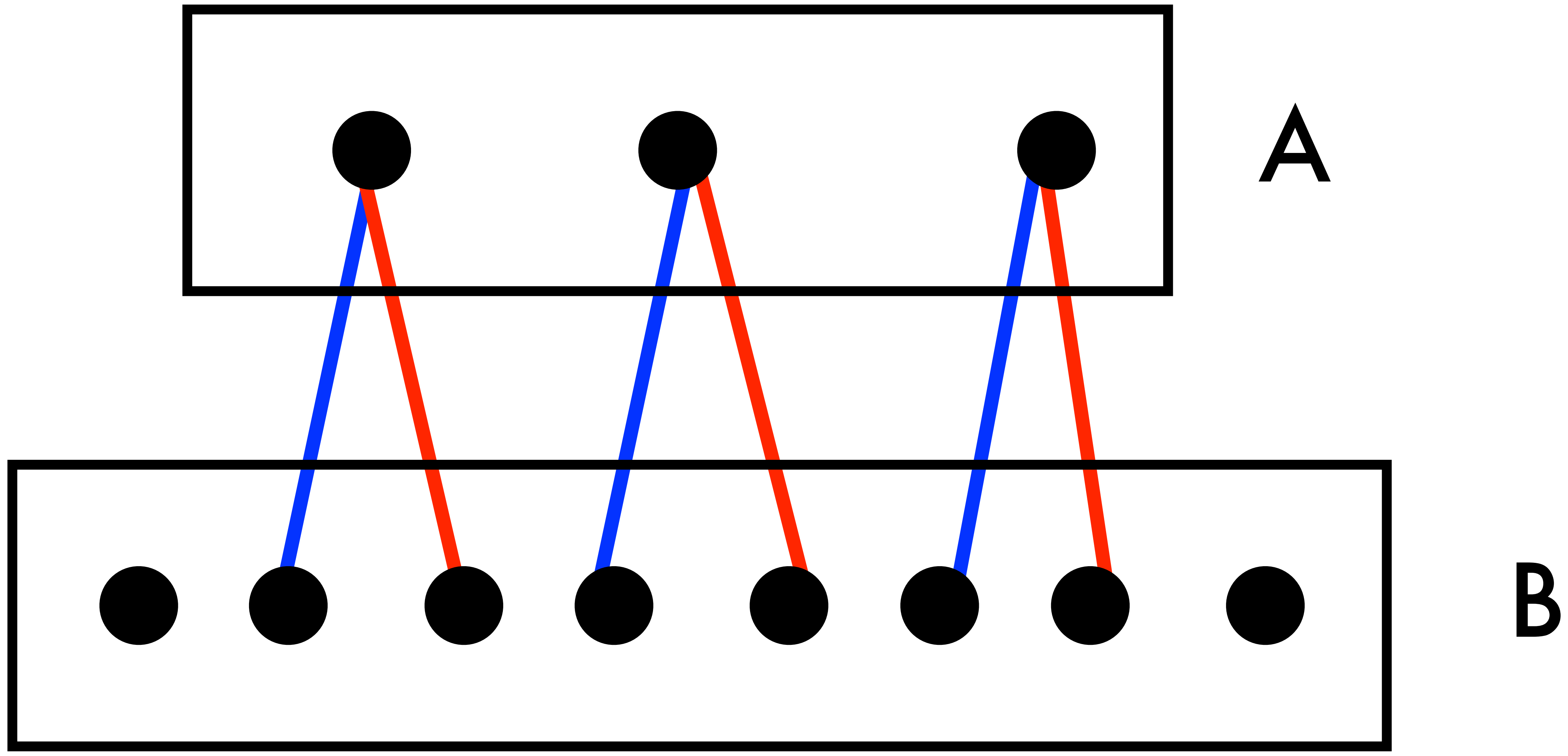
q-Expansion

Bipartite graph G=(A,B),
a set of edges M is a q-expansion of A into B if
M is a disjoint union of |A| q-stars each with its centre vertex in A
and the leaves in B.



q-Expansion

Bipartite graph G=(A,B),
a set of edges M is a q-expansion of A into B if
M is a disjoint union of |A| q-stars each with its centre vertex in A
and the leaves in B.



1-Expansion is a matching

Expansion Lemma

Expansion Lemma

Bipartite graph $G=(A,B)$, integer q such that $|B| \geq q\,|A|$, no isolated vertices in B:
then $\exists\, H \subseteq A$ and $C \subseteq B$ such that there is a q-expansion from A into B and $N(C) \subseteq H$.

Expansion Lemma

Bipartite graph G=(A,B), integer q such that
|B| ≥ q |A|, no isolated vertices in B:
then ∃ H ⊆ A and C ⊆ B such that there is a q-expansion from A into B and N(C) ⊆ H.

Can be proved using arguments similar to the proof of Hall's theorem.

Component Order Connectivity (COC)

Input: A graph G, an integers k,p

Parameter: k,p

Question: Does there exist a set of at most k vertices, say S, such that each connected component of G-S has at most p vertices?

Component Order Connectivity (COC)

Input: A graph G, an integers k,p
Parameter: k,p
Question: Does there exist a set of at most k vertices, say S, such that each connected component of $G-S$ has at most p vertices?

$p=1$ is the Vertex Cover problem.

Component Order Connectivity

Every solution hits a connected set of size at least p+1 in the graph.

Start by finding an approximate solution.

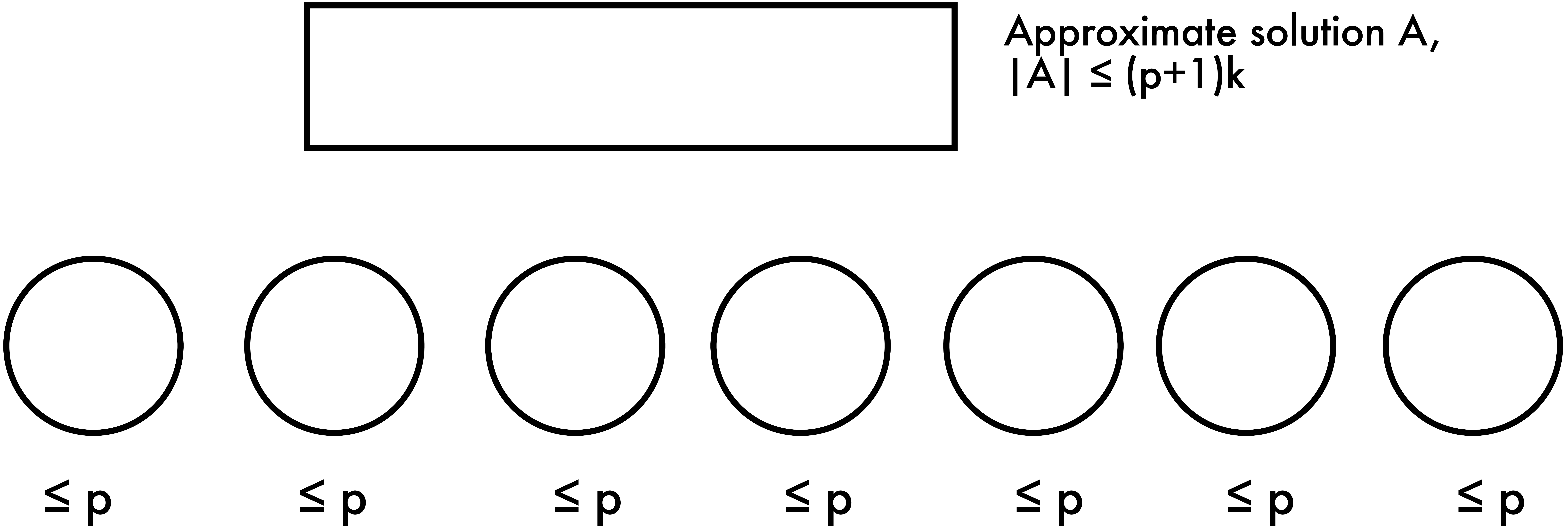
Component Order Connectivity

Every solution sets a connected set of size at least p+1 in the graph.

Approximate solution (A):

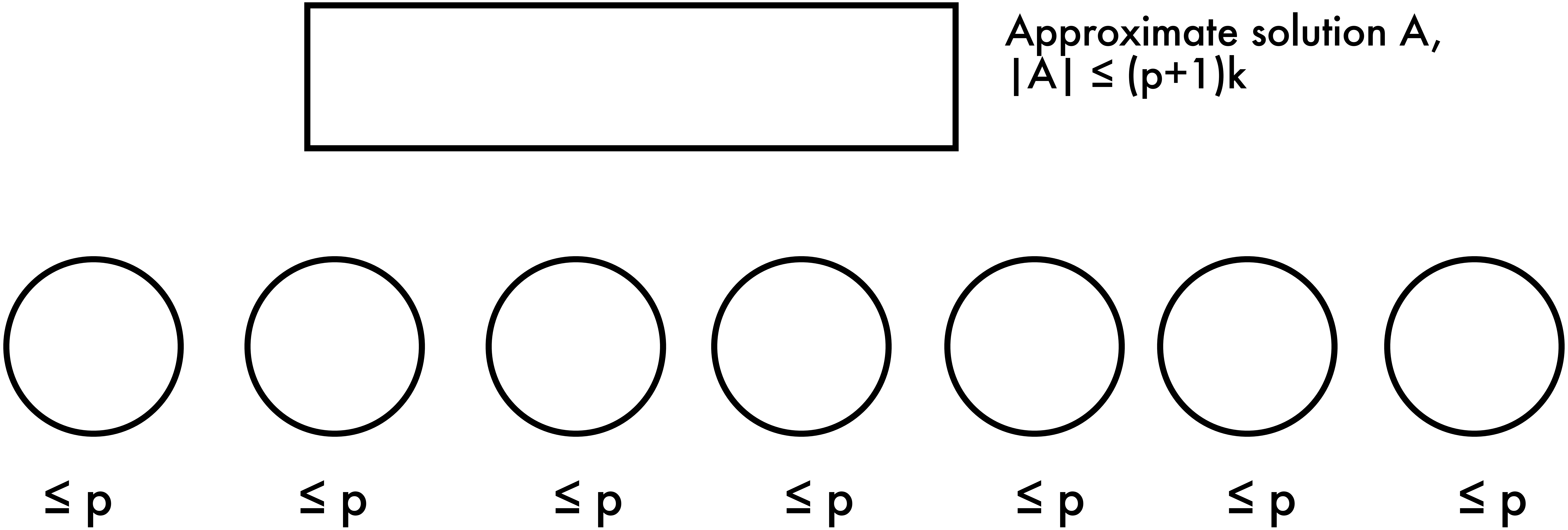
Greedily find a maximal collection $\mathcal{C}$ of connected sets of size p+1.
If $\mathcal{C} \geq k+1$, then return NO.
Otherwise, A be the union of the vertices in the sets of $\mathcal{C}$.
Then $|A| \leq (p+1)k$.

Component Order Connectivity



If the number of connected components of G-A is at most p|A|-1, then we get a kernel with O(p^3 k) vertices.

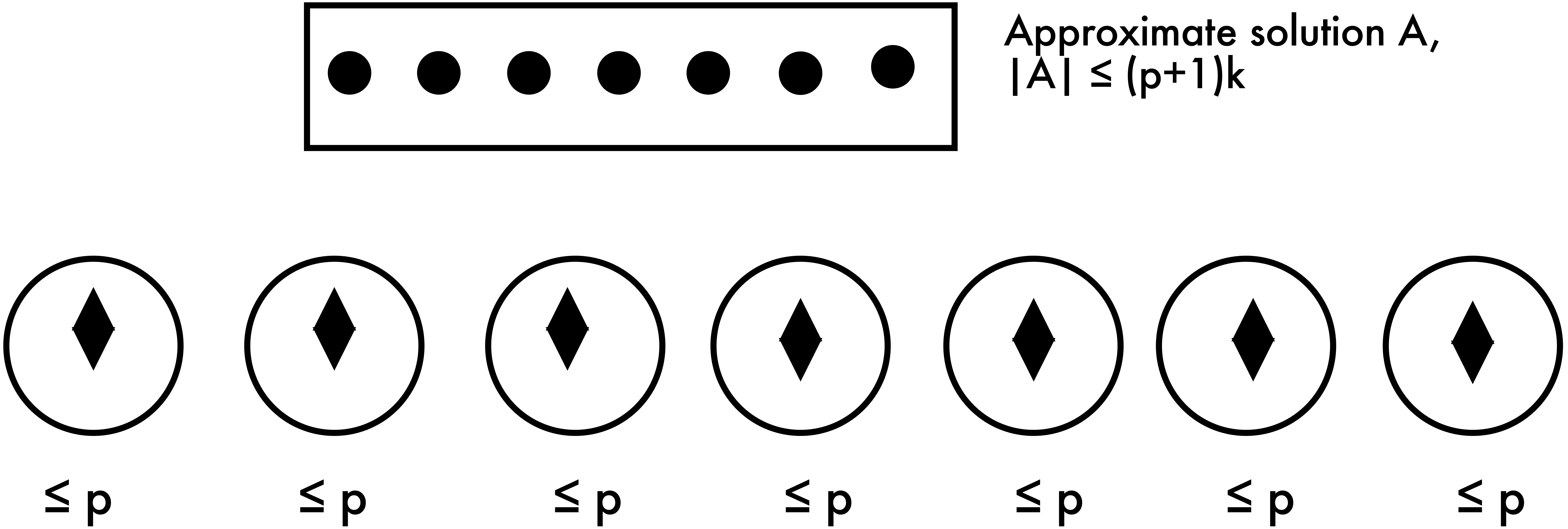
Component Order Connectivity



The number of connected components of G-A is at least p|A|.

Use p-Expansion Lemma.

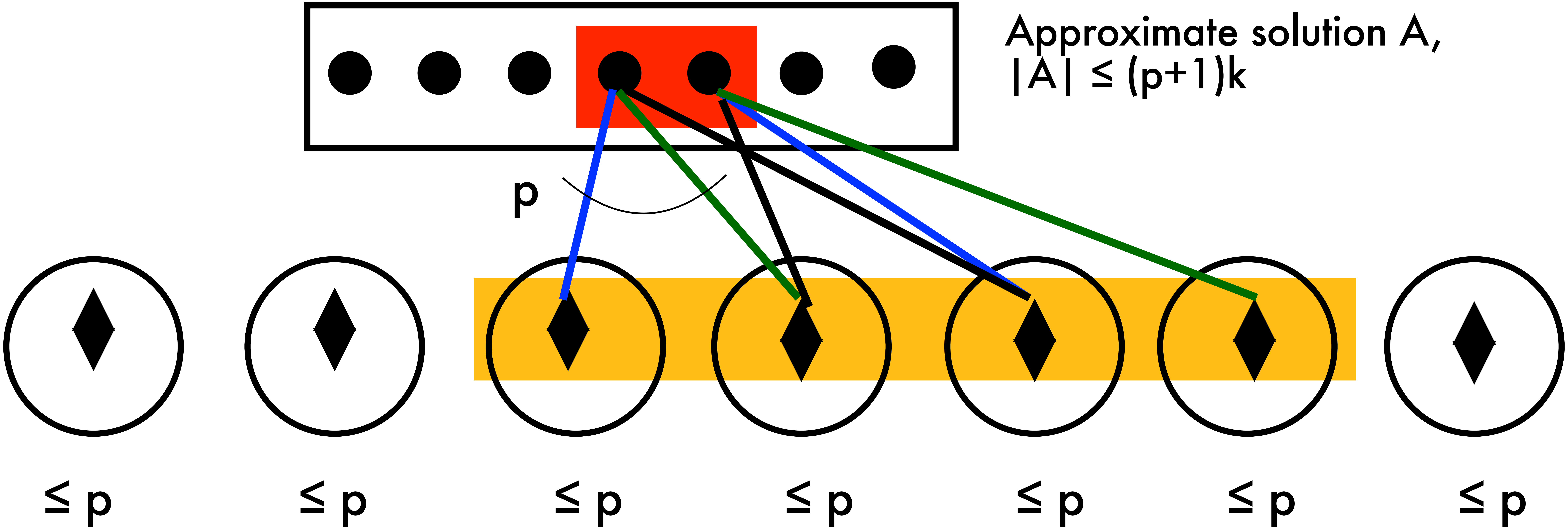
Component Order Connectivity



The number of connected components of G-A is at least p|A|.

Use p-Expansion Lemma.

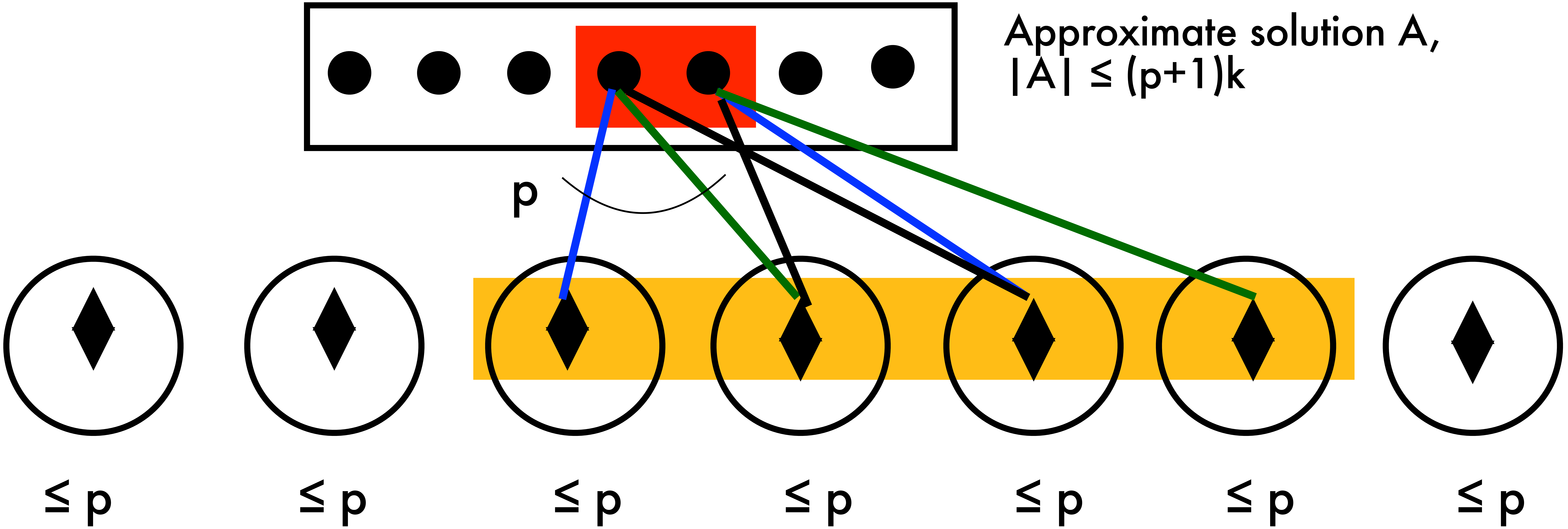
Component Order Connectivity



The number of connected components of G-A is at least p|A|.

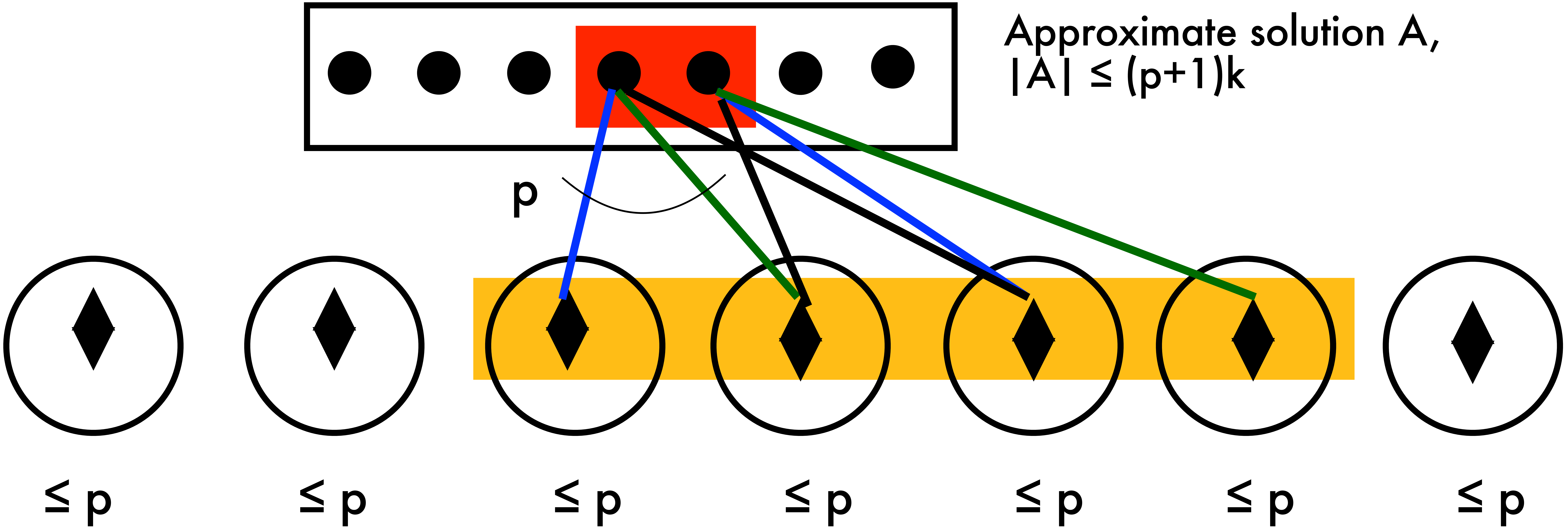
Use p-Expansion Lemma.

Component Order Connectivity



You can do what I can do, and at least one of us needs to do the job!

Component Order Connectivity



When p-Expansion lemma is not applicable, the number of connected components is at most $p|A|+1$.

Kernel with $O(p^3 k)$ vertices.