Julius-Maximilians-UNIVERSITÄT WÜRZBURG

Lehrstuhl für INFORMATIK I Algorithmen & Komplexität

Institut für Informatik

Exact Algorithms

Sommer Term 2020

Lecture 12 Iterative Compression

Based on: [Parameterized Algorithms: §4, 4.1, 4.2]

(slides by J. Spoerhase, Th. van Dijk, S. Chaplick, and A. Wolff)

Alexander Wolff

Lehrstuhl für Informatik I

Iterative Compression

Idea: Is it useful for an exact algorithm to start from a near-optimal solution?

Iterative Compression

Idea: Is it useful for an exact algorithm to start from a near-optimal solution?

Idea: Greedy algorithms do not solve NP-hard problems *exactly*, but might give us an incremental approach.

Iterative Compression

Idea: Is it useful for an exact algorithm to start from a near-optimal solution?

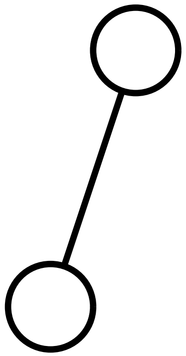
Idea: Greedy algorithms do not solve NP-hard problems *exactly*, but might give us an incremental approach.

k-VERTEX COVER

Given:	Graph $G = (V, E)$
Parameter:	Integer k
Question:	Does G have a vertex cover of size $\leq k$?

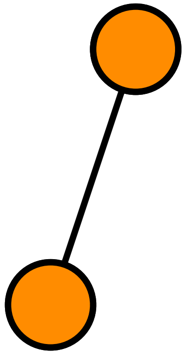
Vertex Cover

Vertex Cover with 2 vertices?



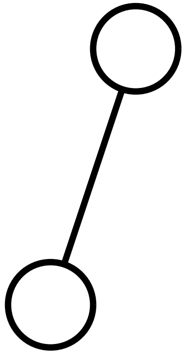
Vertex Cover

Vertex Cover with 2 vertices?



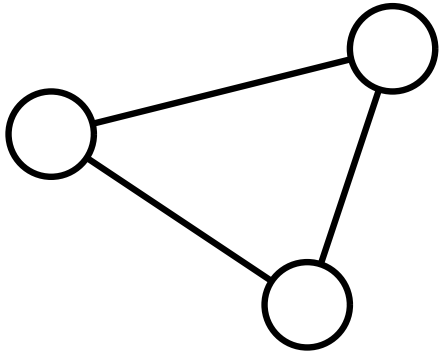
Vertex Cover

Vertex Cover with 2 vertices?



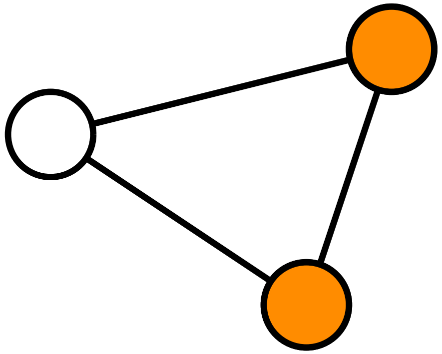
Vertex Cover

Vertex Cover with 2 vertices?



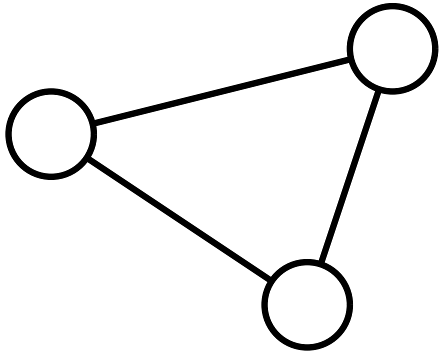
Vertex Cover

Vertex Cover with 2 vertices?



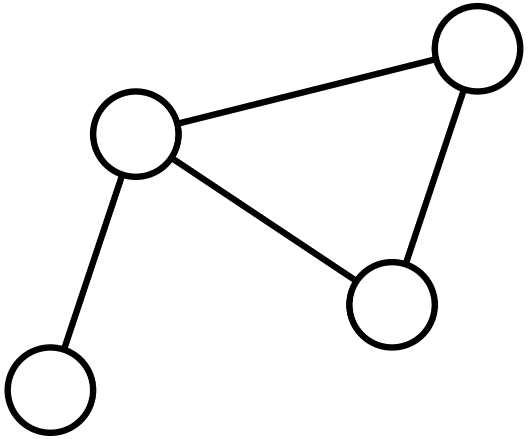
Vertex Cover

Vertex Cover with 2 vertices?



Vertex Cover

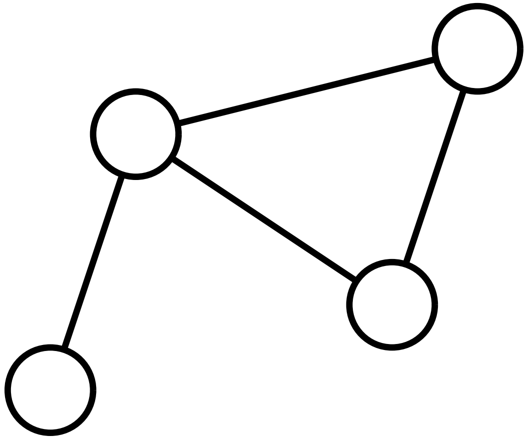
Vertex Cover with 2 vertices?



Vertex Cover

Vertex Cover with 2 vertices?

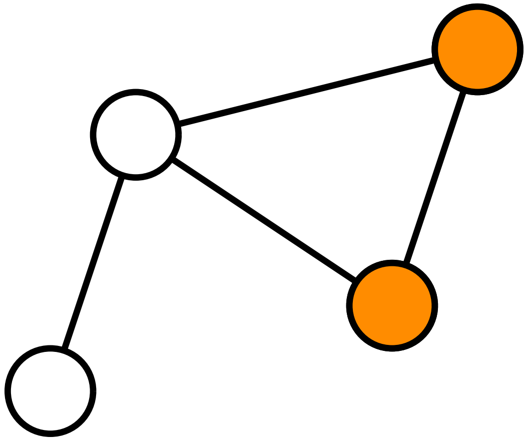
3 is easy...



Vertex Cover

Vertex Cover with 2 vertices?

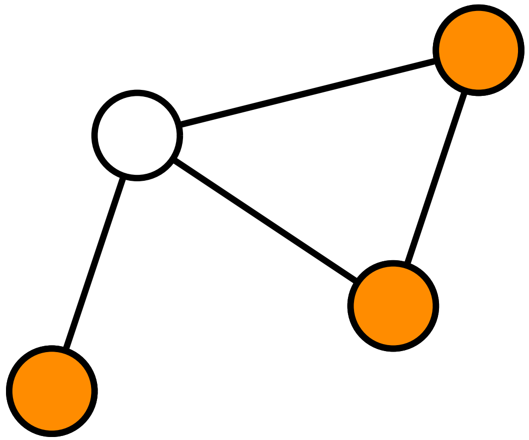
3 is easy...



Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

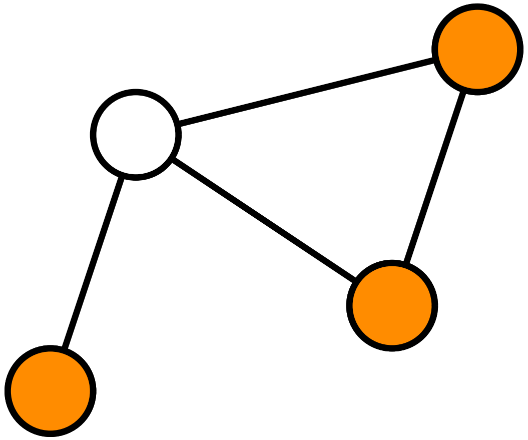


Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?



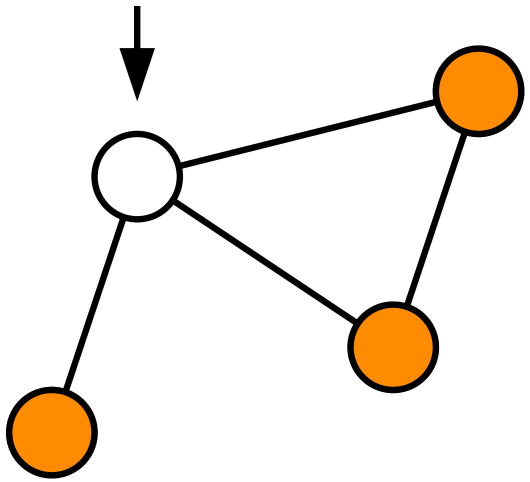
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



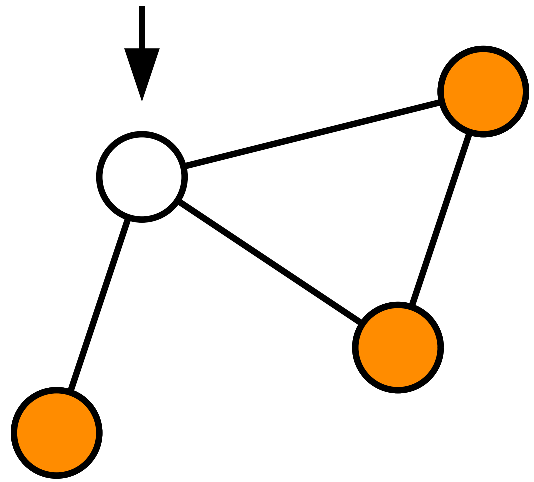
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



Vertex Cover Compression

Given: Graph $G = (V, E)$,
vertex cover $C \subseteq V$ with $|C| = 3$,

Find: Vertex cover $X \subseteq V$ with $|X| = 2$,
or answer: No.

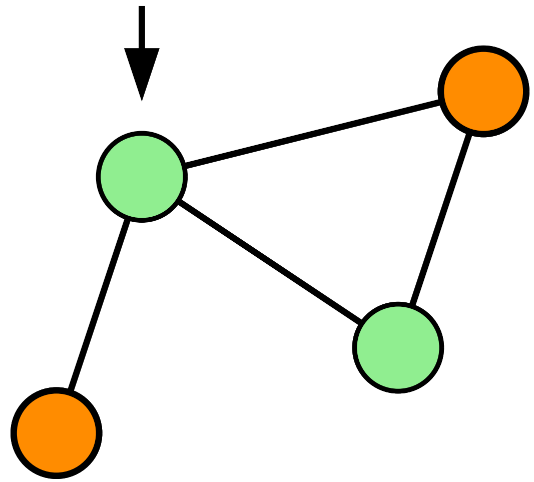
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



Vertex Cover Compression

Given: Graph $G = (V, E)$,
vertex cover $C \subseteq V$ with $|C| = 3$,

Find: Vertex cover $X \subseteq V$ with $|X| = 2$,
or answer: No.

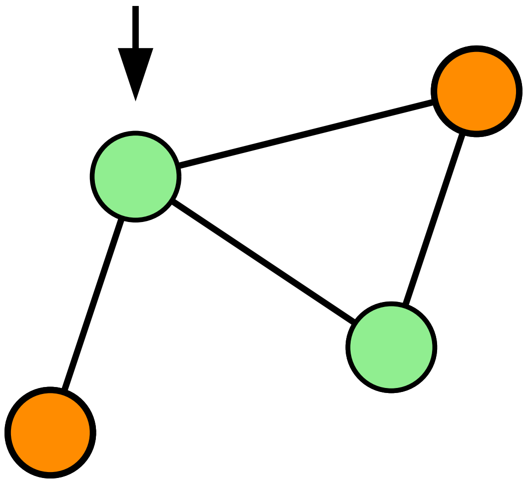
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



Vertex Cover Compression

Given: Graph $G = (V, E)$, number k, vertex cover $C \subseteq V$ with $|C| = k+1$,

Find: Vertex cover $X \subseteq V$ with $|X| = k$, or answer: No.

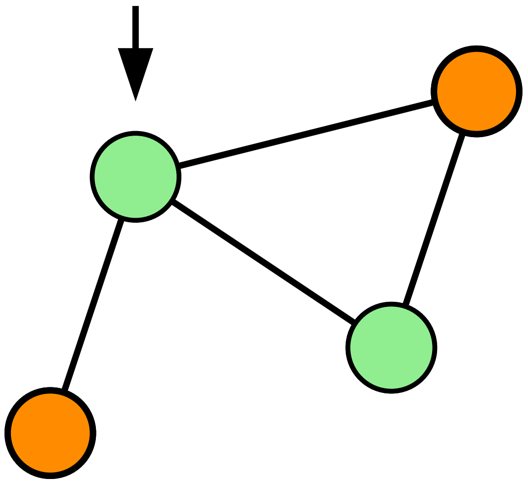
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



VERTEX COVER COMPRESSION

Given: Graph $G = (V, E)$, number k, vertex cover $C \subseteq V$ with $|C| = k+1$,

Find: Vertex cover $X \subseteq V$ with $|X| = k$, or answer: NO.

Complexity of VERTEX COVER COMPRESSION?

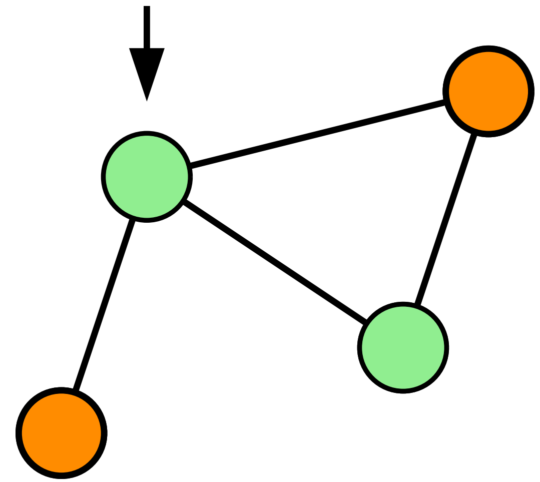
Vertex Cover

Vertex Cover with 2 vertices?

3 is easy...

But does 2 work?

Need to use other vertices!



VERTEX COVER COMPRESSION

Given: Graph $G = (V, E)$, number k,
vertex cover $C \subseteq V$ with $|C| = k+1$,

Find: Vertex cover $X \subseteq V$ with $|X| = k$,
or answer: NO.

Complexity of VERTEX COVER COMPRESSION?

Not in P, otherwise VERTEX COVER in P!

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

2

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

2

3 Profit

Vertex Cover by Compression

1. Start with a k-vertex subgraph.

   In such a graph, k-vertex cover is trivial :-)

2. Add an unvisited vertex to our graph and our vertex cover.

3. Profit

Vertex Cover by Compression

1. Start with a k-vertex subgraph.

   In such a graph, k-vertex cover is trivial :-)

2. Add an unvisited vertex to our graph and our vertex cover.

   This yields a $(k + 1)$-vertex cover.

3. Profit

Vertex Cover by Compression

1. Start with a k-vertex subgraph.

   In such a graph, k-vertex cover is trivial :-)

2. Add an unvisited vertex to our graph and our vertex cover.

   This yields a $(k+1)$-vertex cover.

   Compress to a k-vertex cover

3. Profit

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

2 Add an unvisited vertex to our graph and our vertex cover.

This yields a $(k+1)$-vertex cover.

Compress to a k-vertex cover

3 Profit

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

2 Add an unvisited vertex to our graph and our vertex cover.

This yields a $(k+1)$-vertex cover.

Compress to a k-vertex cover, or answer: NO.

3 Profit

Vertex Cover by Compression

1 Start with a k-vertex subgraph.

In such a graph, k-vertex cover is trivial :-)

2 Add an unvisited vertex to our graph and our vertex cover.

This yields a $(k+1)$-vertex cover.

Compress to a k-vertex cover, or answer: NO.

3 Profit

Runtime: $O(n)$ compression steps.

Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$

Find: Vertex cover X, $|X| = k$

Vertex Cover Compression

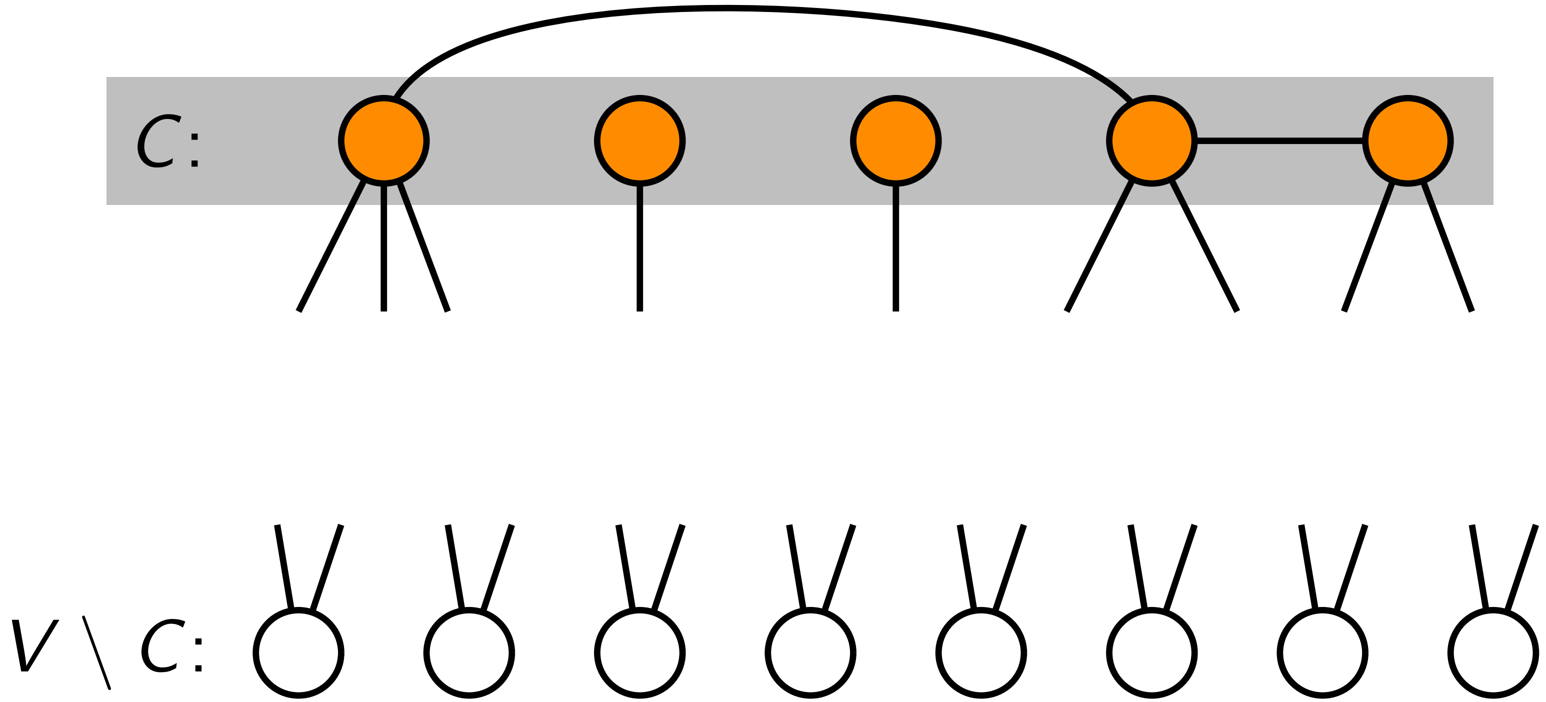
Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.

Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

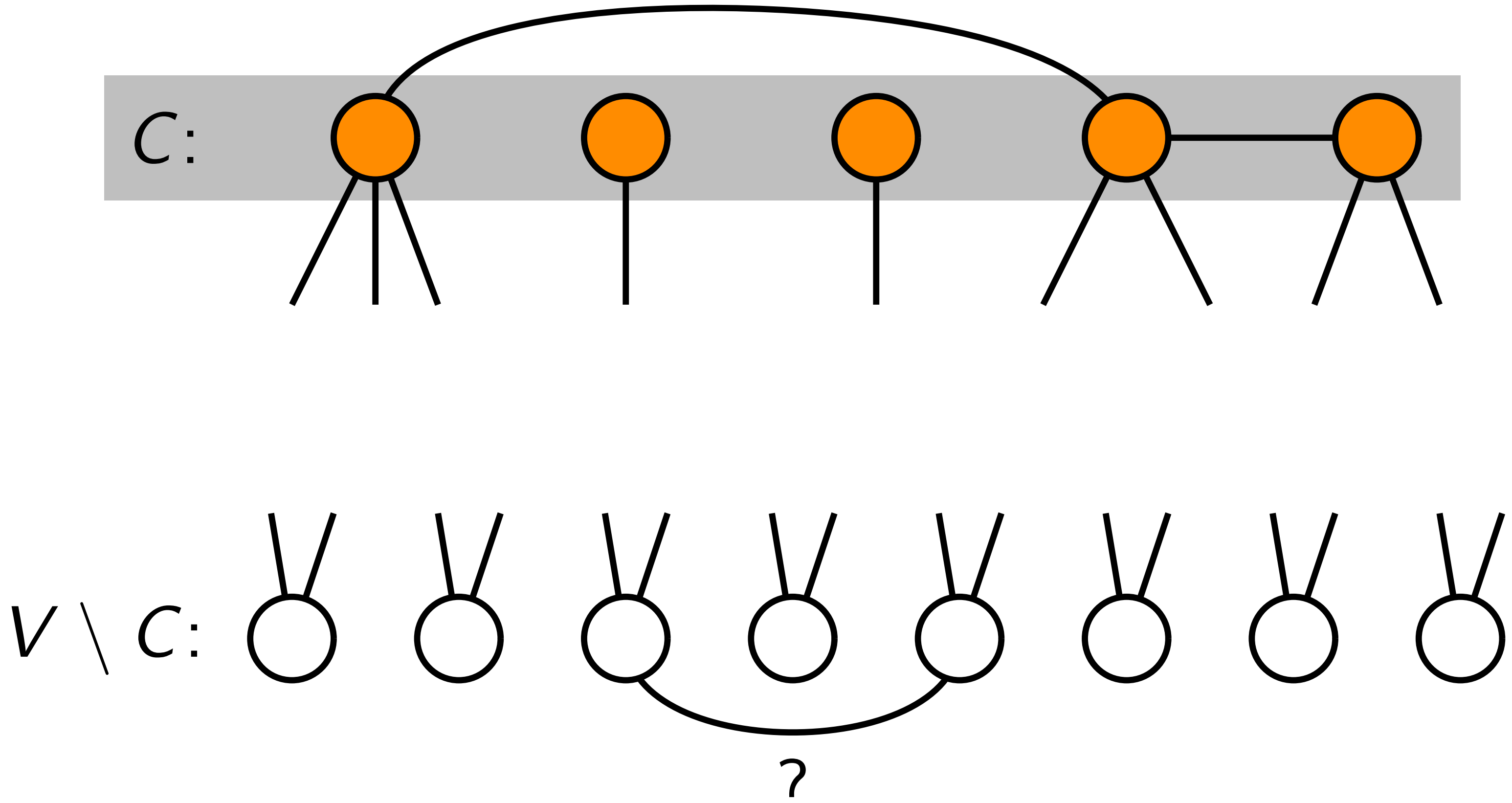
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

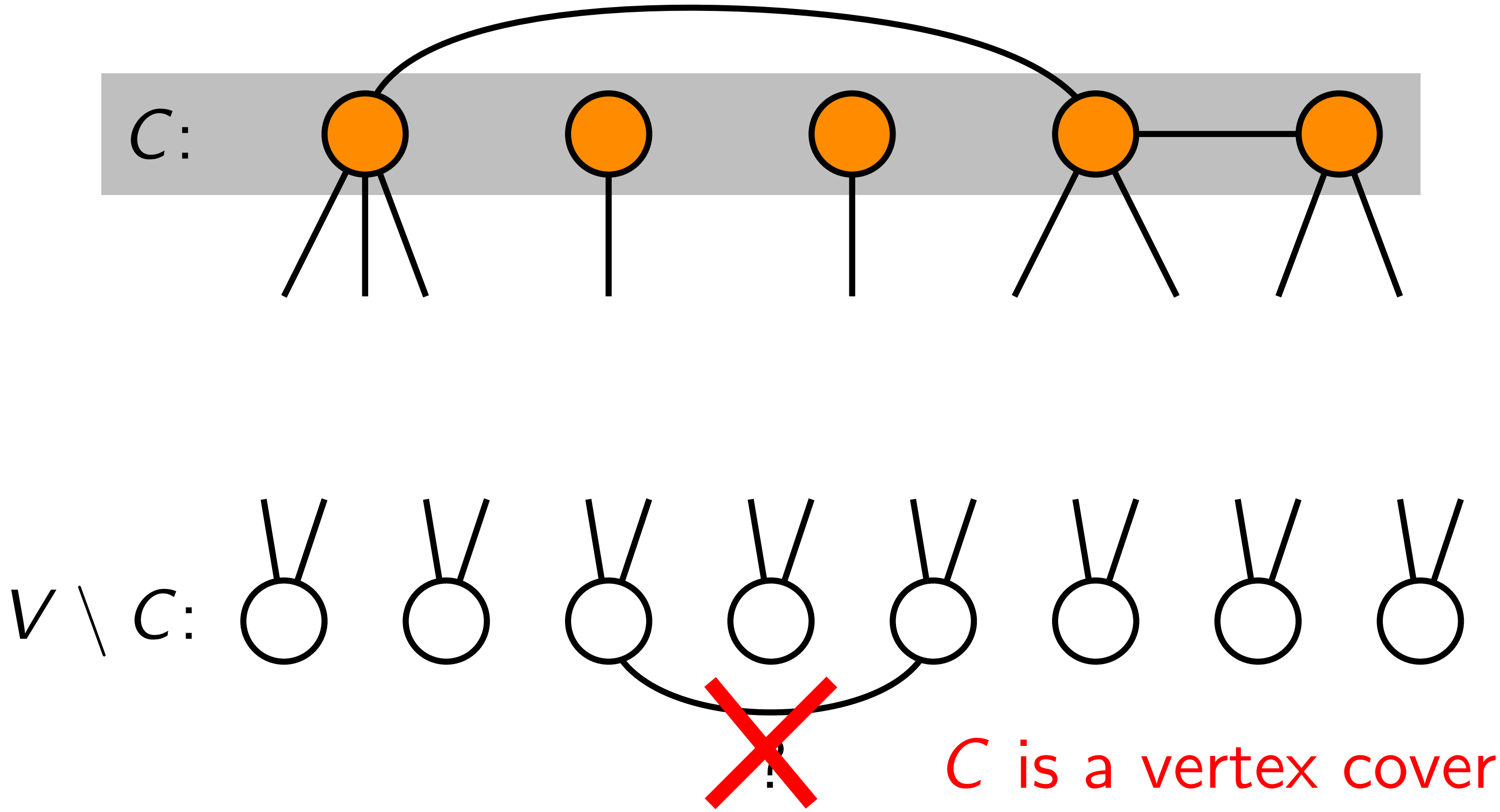
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

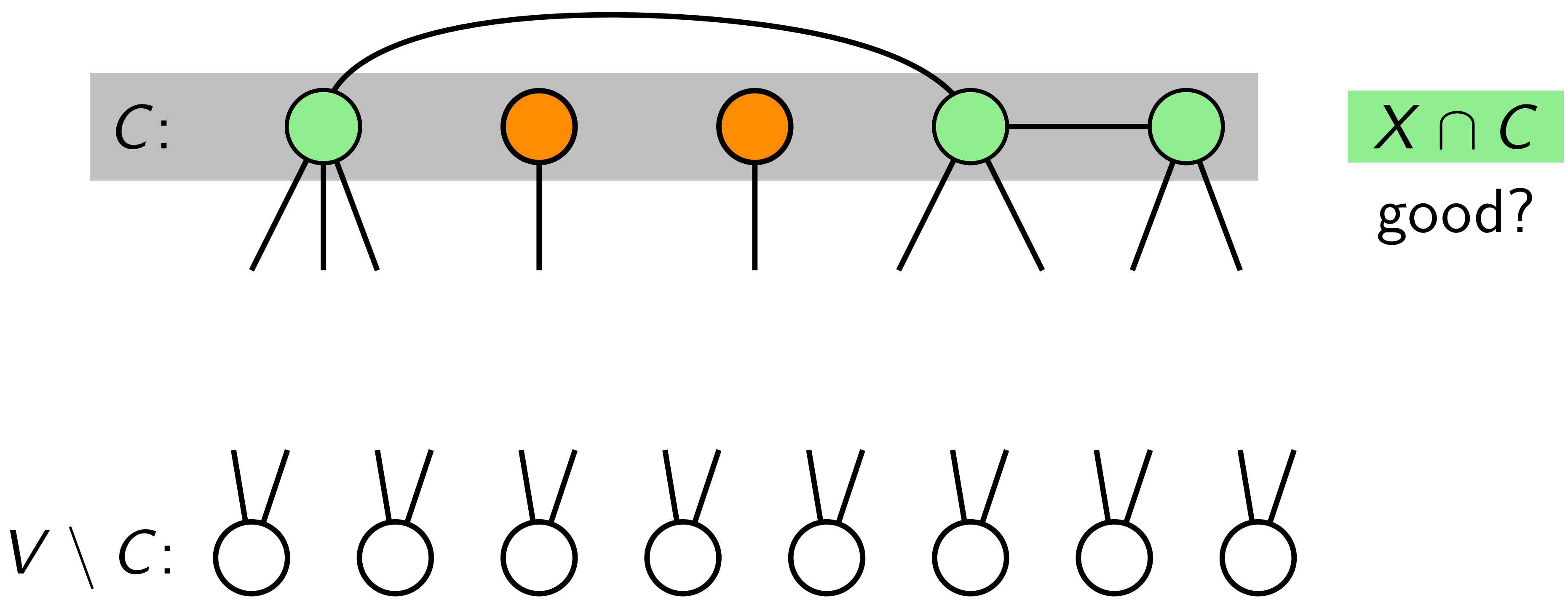
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

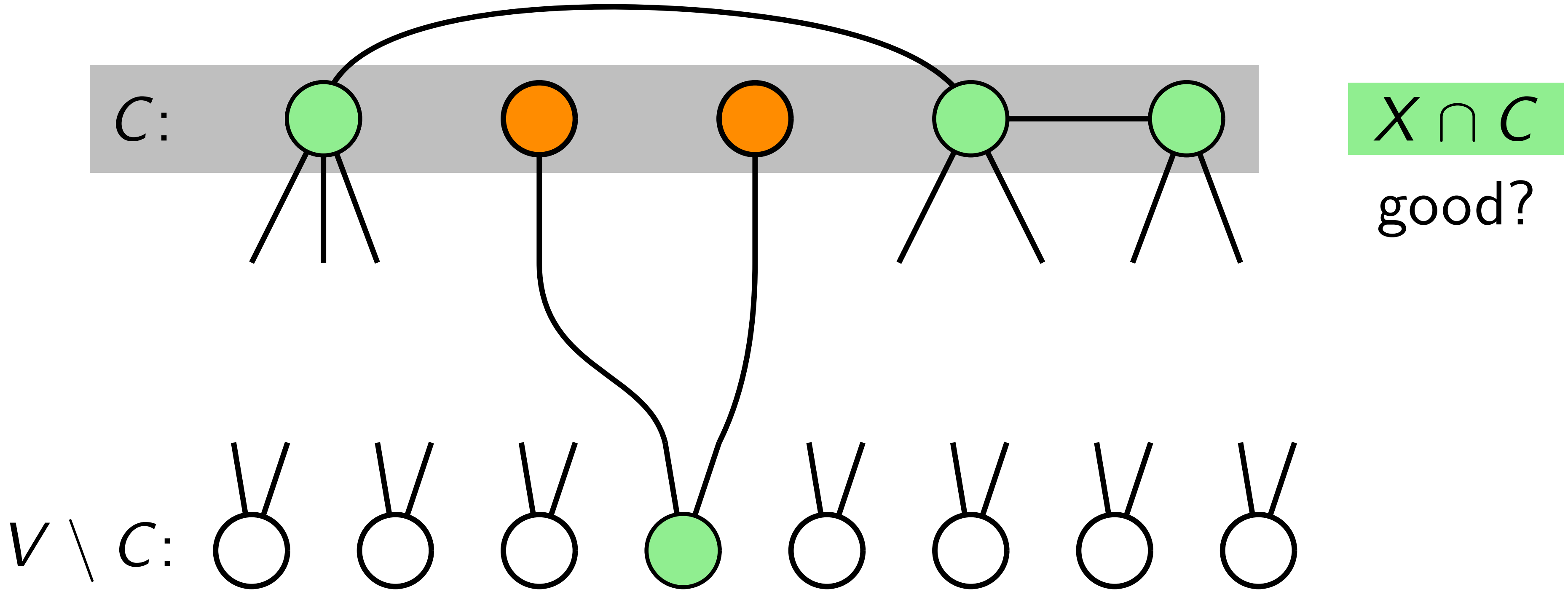
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

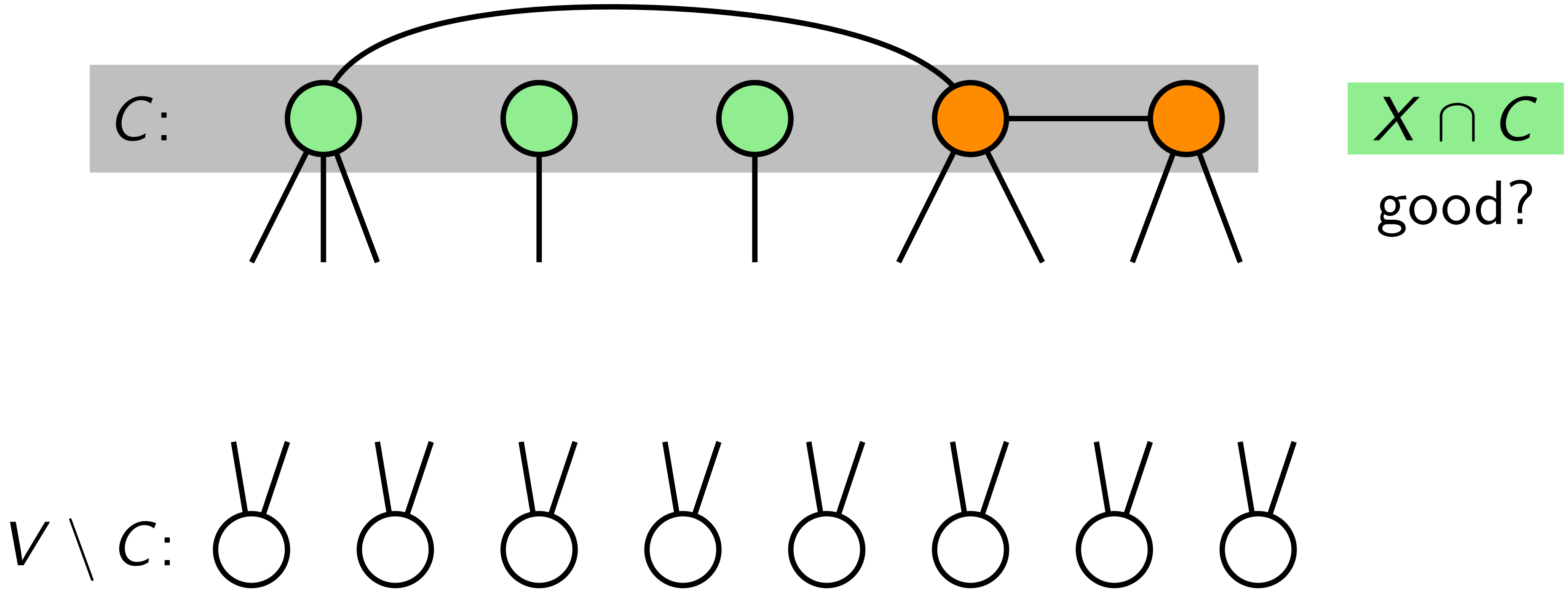
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

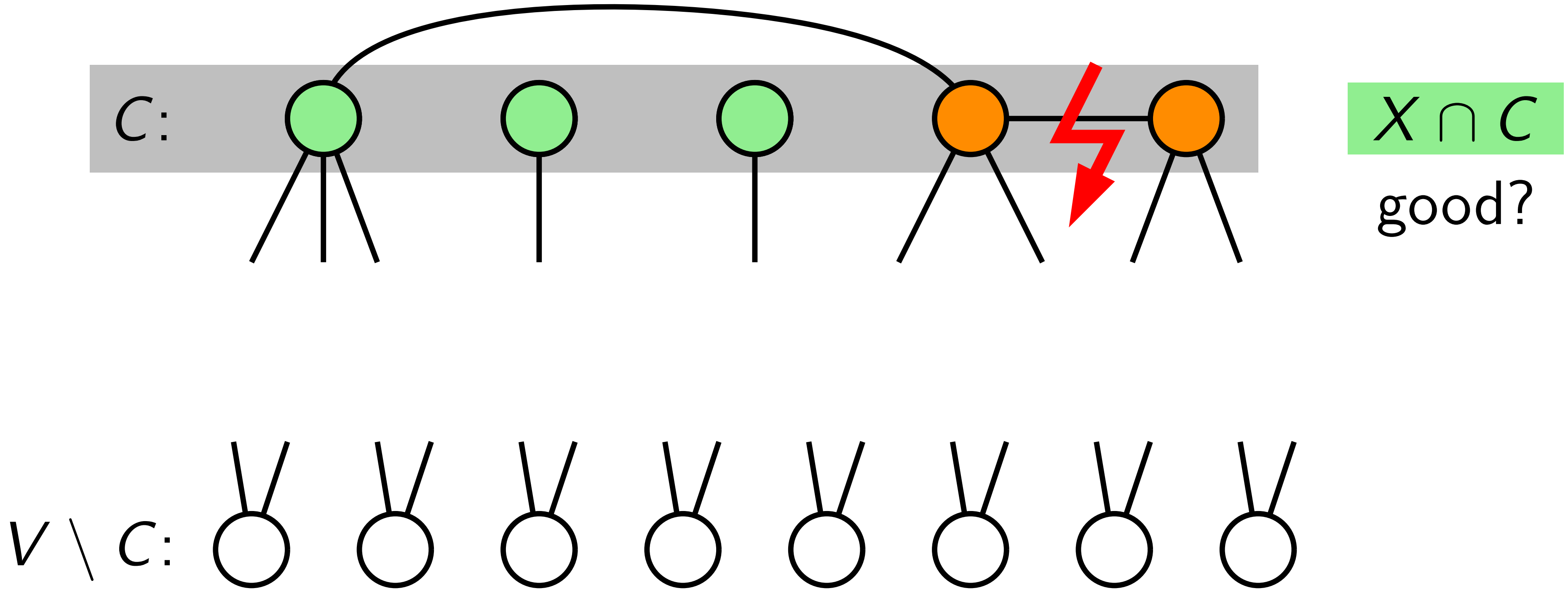
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

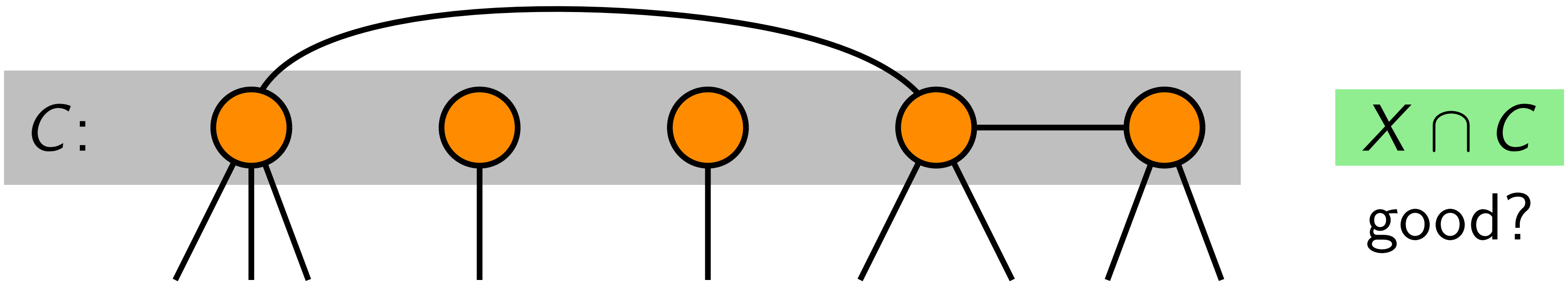
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k+1$
Find: Vertex cover X, $|X| = k$

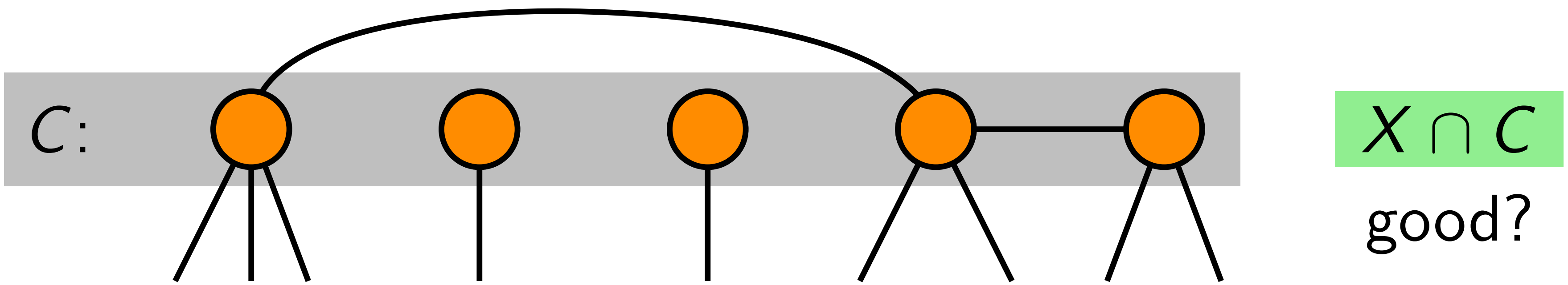
Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



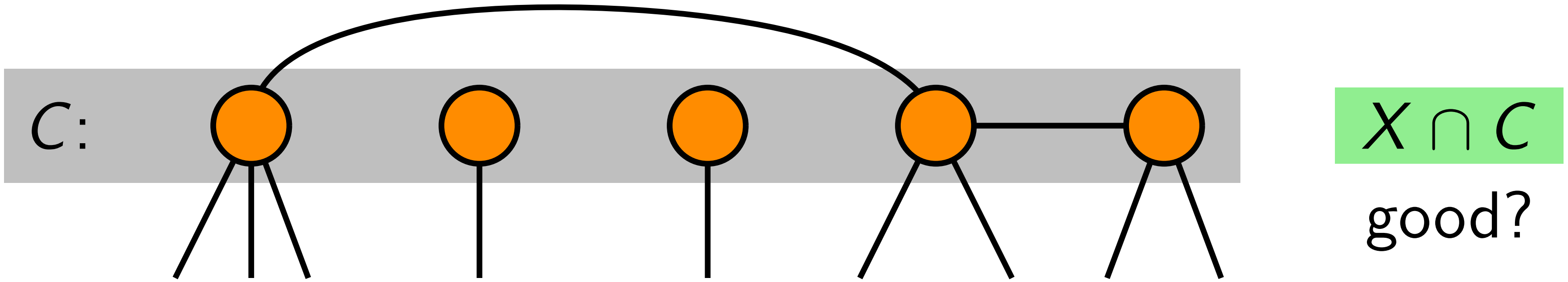
Note: $X \cap C$ is valid if

- $C \setminus X$ is independent

Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



Note: $X \cap C$ is valid if

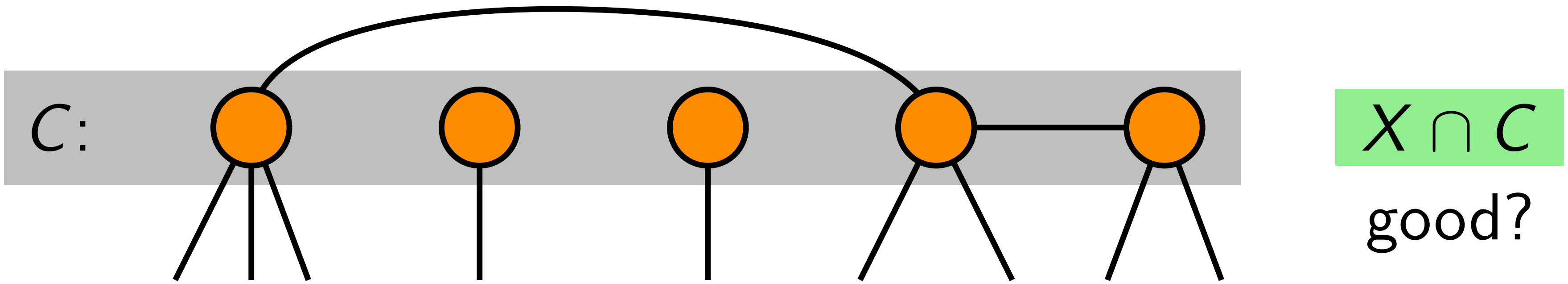
- $C \setminus X$ is independent
- $|(X \cap C) \cup N(C \setminus X)| \leq k$

Runtime:

Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



Note: $X \cap C$ is valid if

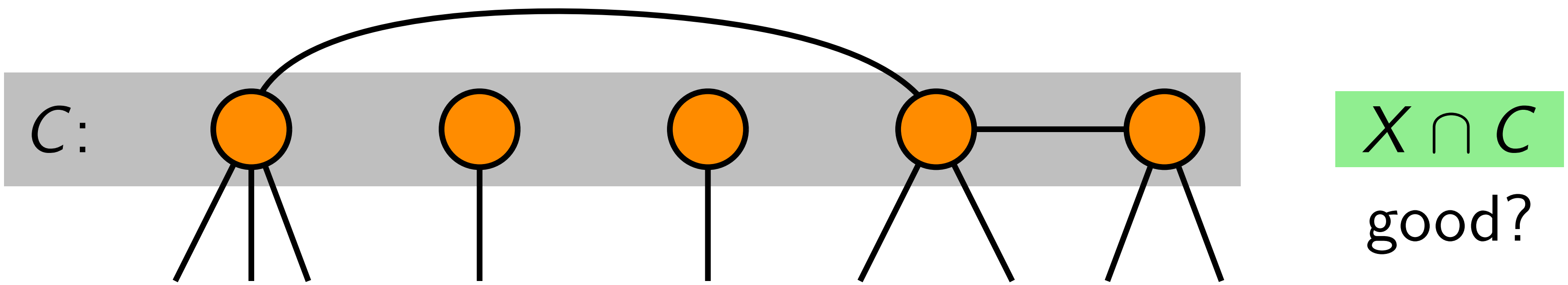
- $C \setminus X$ is independent
- $|(X \cap C) \cup N(C \setminus X)| \leq k$

Runtime: $< 2^k$ options for $X \cap C$; polytime for each one

Vertex Cover Compression

Given: Vertex cover C, $|C| = k+1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



Note: $X \cap C$ is valid if

- $C \setminus X$ is independent
- $|(X \cap C) \cup N(C \setminus X)| \leq k$

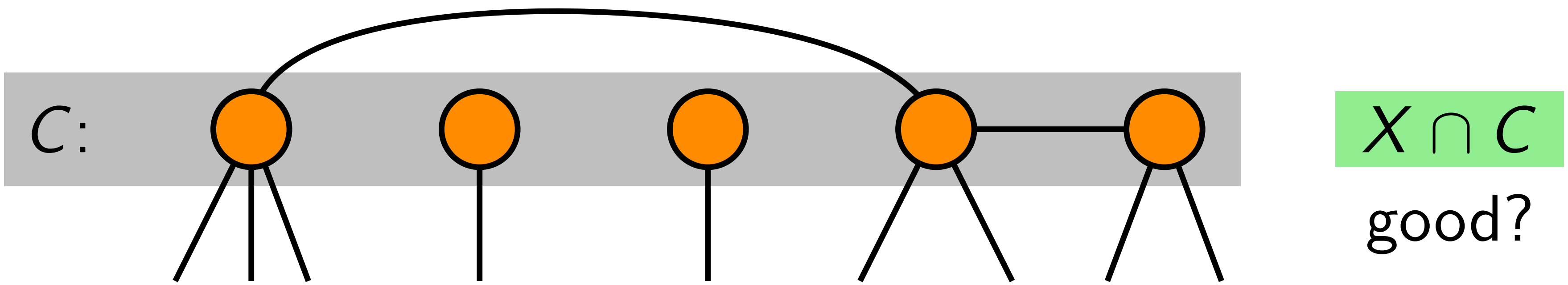
Runtime: $< 2^k$ options for $X \cap C$; polytime for each one

Runtime for Vertex Cover:

Vertex Cover Compression

Given: Vertex cover C, $|C| = k + 1$
Find: Vertex cover X, $|X| = k$

Strategy: Fix $X \cap C$, i.e., determine what to keep from C.
For each potential $X \cap C$, X is unique.



Note: $X \cap C$ is valid if

- $C \setminus X$ is independent
- $|(X \cap C) \cup N(C \setminus X)| \leq k$

Runtime: $< 2^k$ options for $X \cap C$; polytime for each one

Runtime for Vertex Cover: $O^*(2^k)$

Dominating Set

Dominating Set (Compression)

Given: Graph $G = (V, E)$, number k,
dominating set $D \subseteq V$, $|D| = k + 1$

Find: Dominating set $X' \subseteq V$, $|X'| = k$,
or answer: No

Dominating Set

DOMINATING SET (COMPRESSION)

Given: Graph $G = (V, E)$, number k,
dominating set $D \subseteq V$, $|D| = k + 1$

Find: Dominating set $X' \subseteq V$, $|X'| = k$,
or answer: NO

Complexity of DOMINATING SET (COMPRESSION)?

Dominating Set

DOMINATING SET (COMPRESSION)

Given: Graph $G = (V, E)$, number k,
dominating set $D \subseteq V$, $|D| = k + 1$

Find: Dominating set $X' \subseteq V$, $|X'| = k$,
or answer: NO

Complexity of DOMINATING SET (COMPRESSION)?

Not FPT in k since DOMINATING SET is not FPT in k!

FVS in Tournaments

FEEDBACK VERTEX SET (TOURNAMENTS)

Given: Tournament $T = (V, E)$, number k

Question: $\exists\, X \subseteq V$ such that $|X| \leq k$ and $T[V \setminus X]$ is acyclic?

FVS in Tournaments

FEEDBACK VERTEX SET (TOURNAMENTS)

Given: Tournament $T = (V, E)$, number k

Question: $\exists\, X \subseteq V$ such that $|X| \leq k$ and $T[V \setminus X]$ is acyclic?

oriented clique

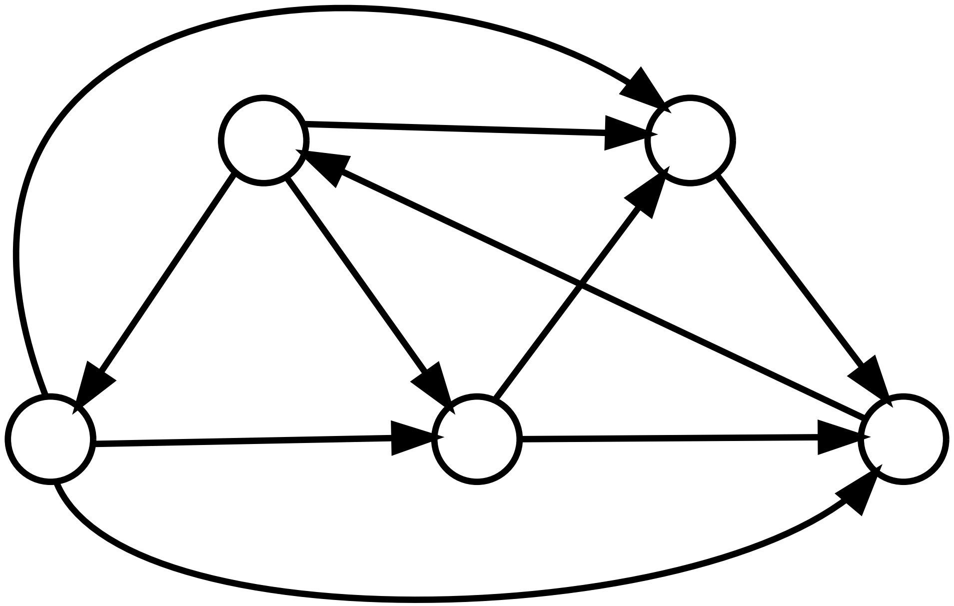
FVS in Tournaments

FEEDBACK VERTEX SET (TOURNAMENTS)

oriented clique

Given: Tournament $T = (V, E)$, number k

Question: $\exists\, X \subseteq V$ such that $|X| \leq k$ and $T[V \setminus X]$ is acyclic?



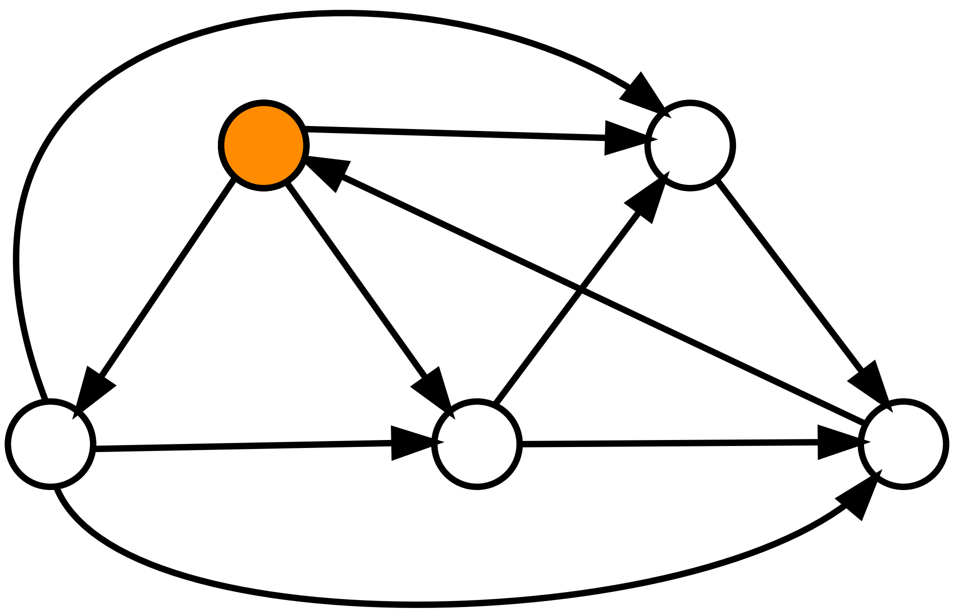
FVS in Tournaments

FEEDBACK VERTEX SET (TOURNAMENTS)

Given: Tournament $T = (V, E)$, number k

Question: $\exists\ X \subseteq V$ such that $|X| \leq k$ and $T[V \setminus X]$ is acyclic?

oriented clique



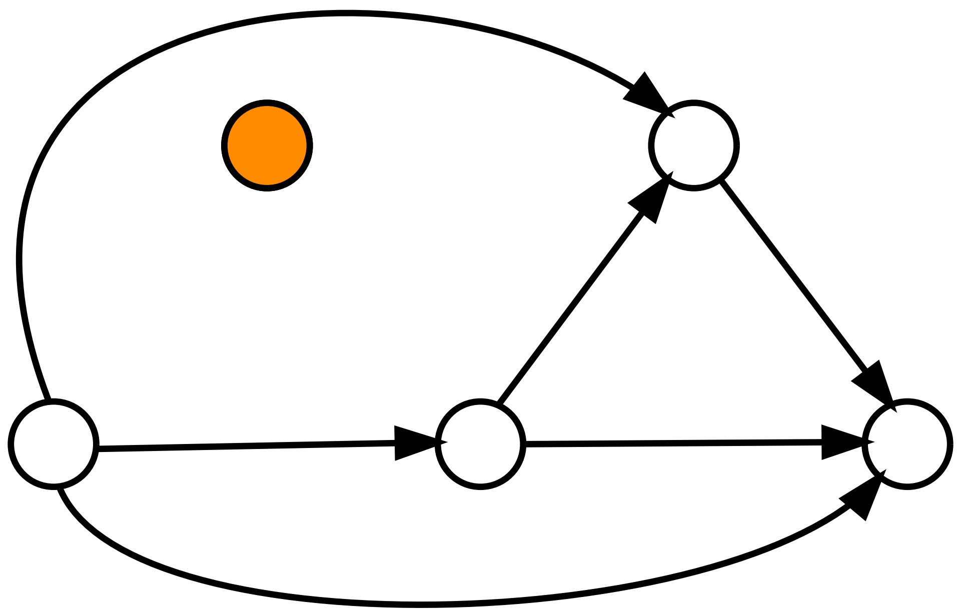
FVS in Tournaments

FEEDBACK VERTEX SET (TOURNAMENTS)

oriented clique

Given: Tournament $T = (V, E)$, number k

Question: $\exists\ X \subseteq V$ such that $|X| \leq k$ and $T[V \setminus X]$ is acyclic?

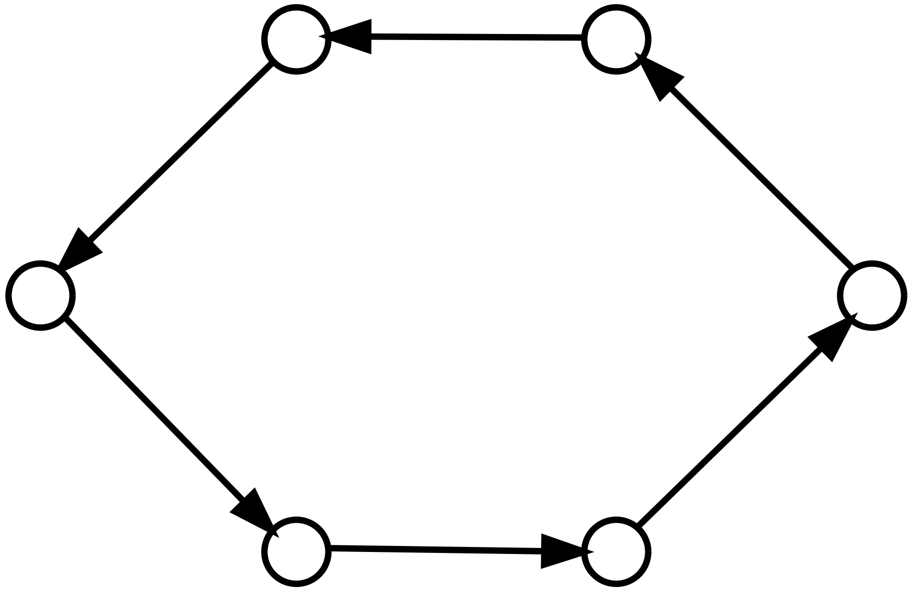


Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.

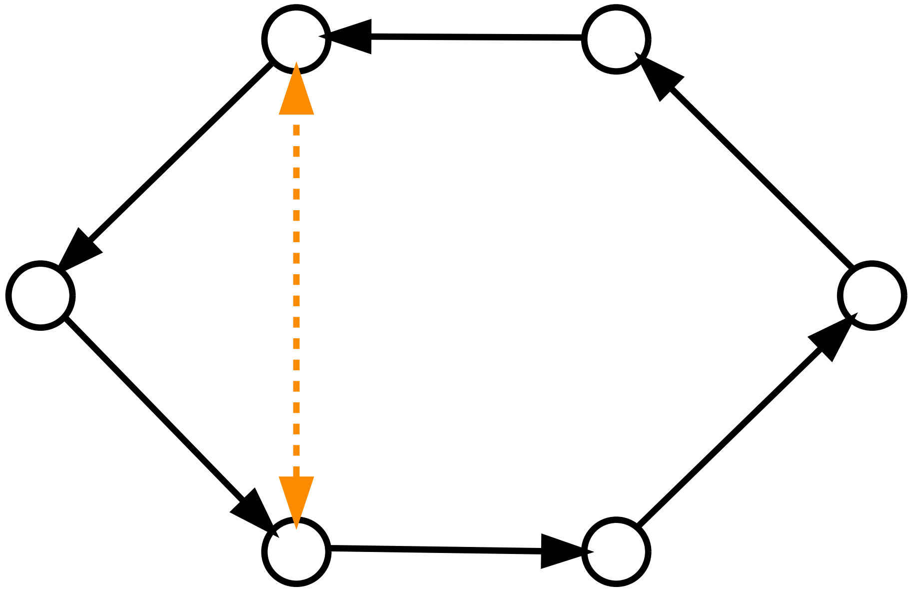
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



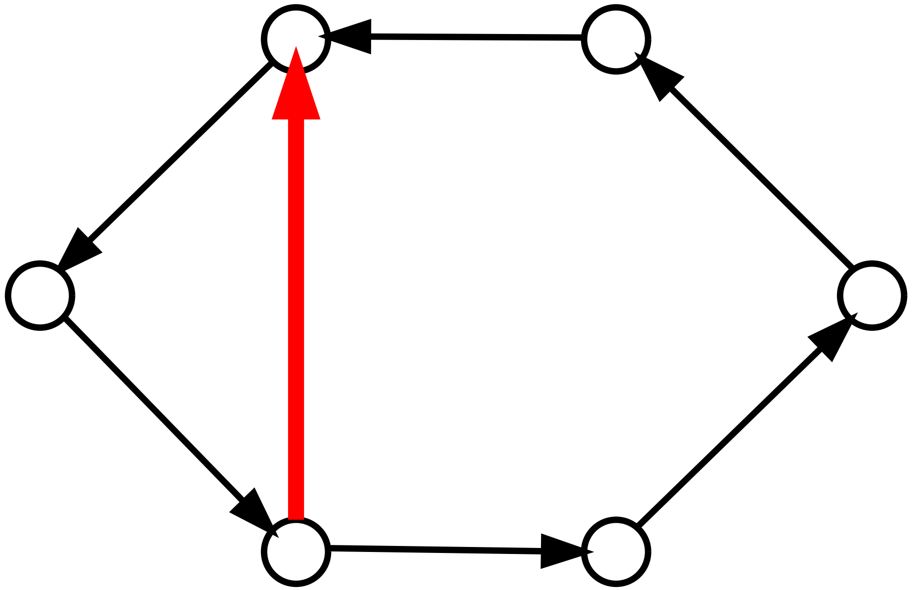
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



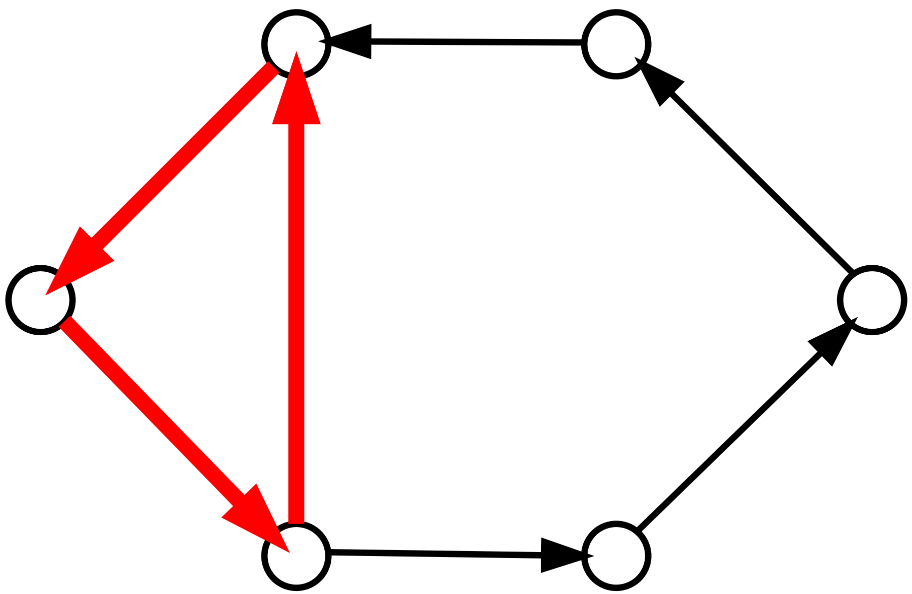
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



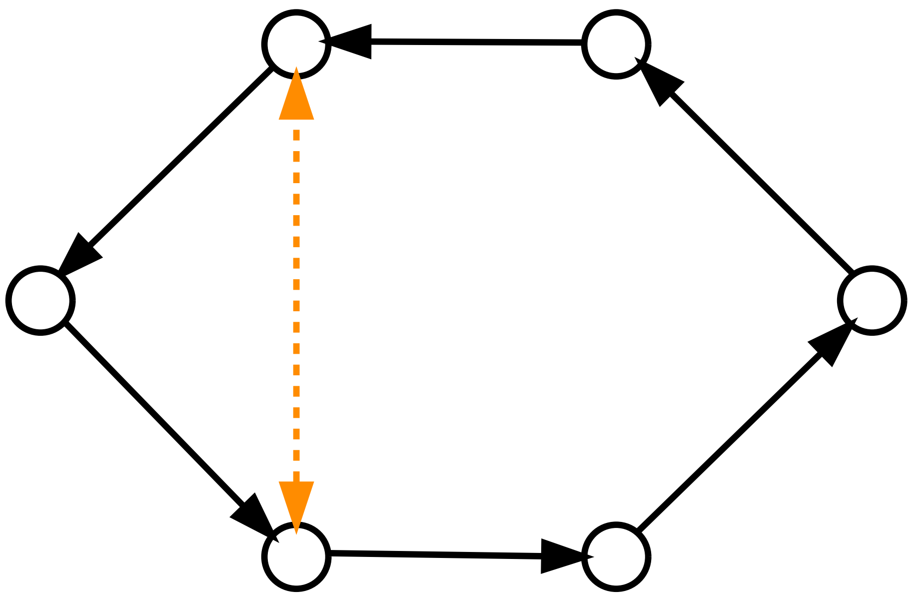
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



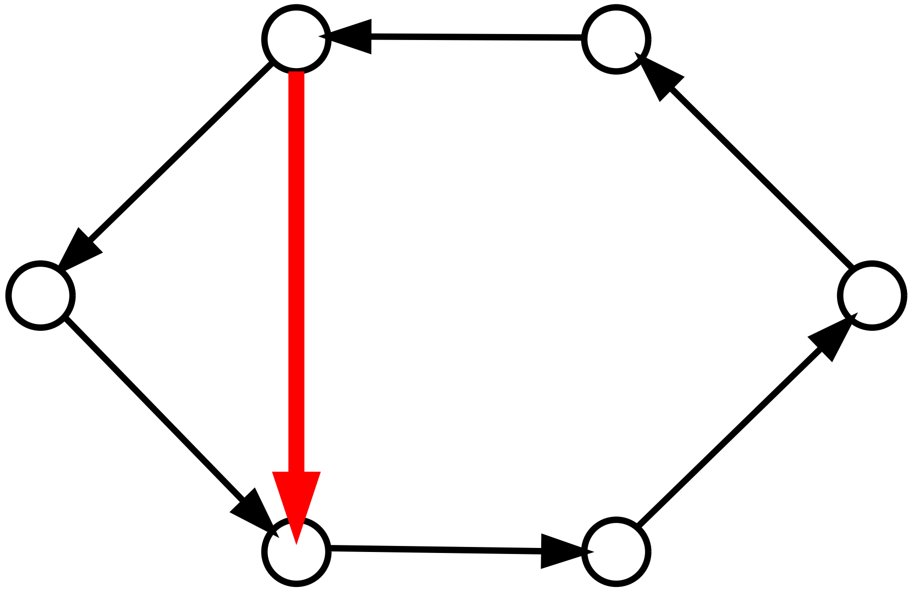
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



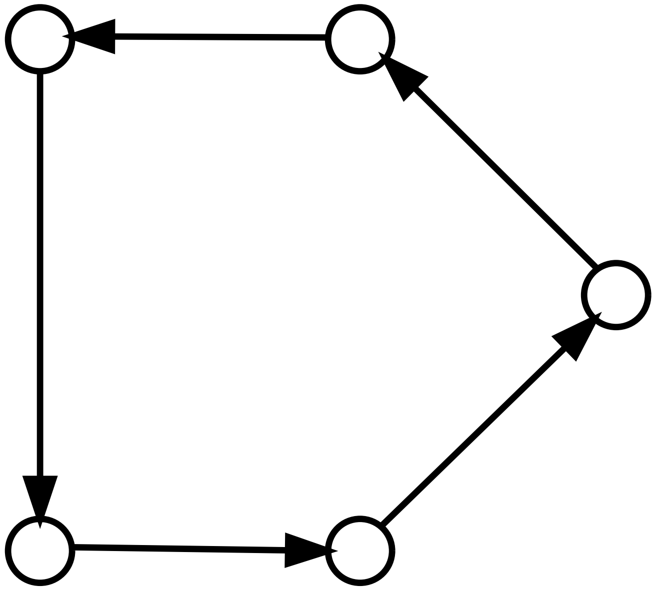
Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



Cycles in Tournaments

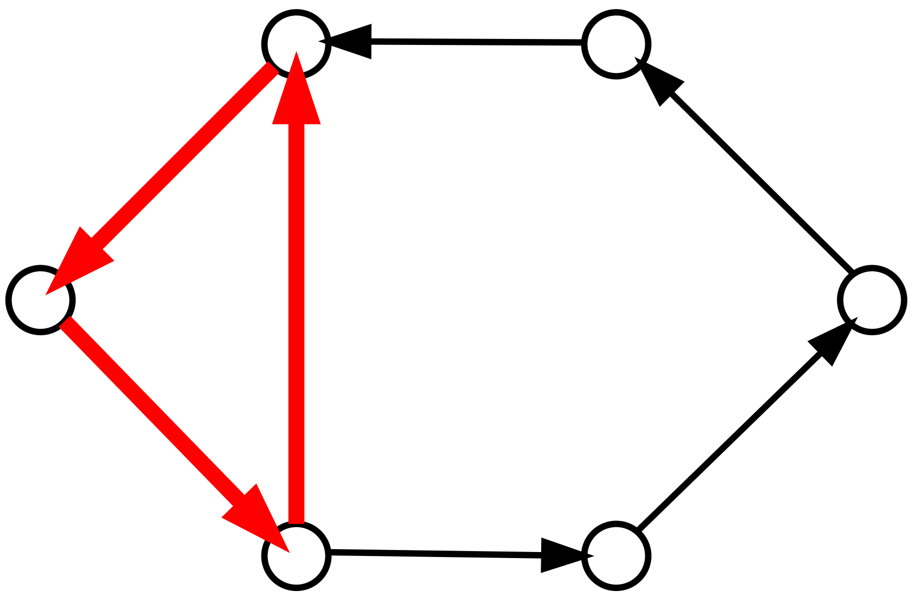
Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.



Cycles in Tournaments

Lemma. A tournament contains a cycle ⇔ it contains a length-3 cycle.

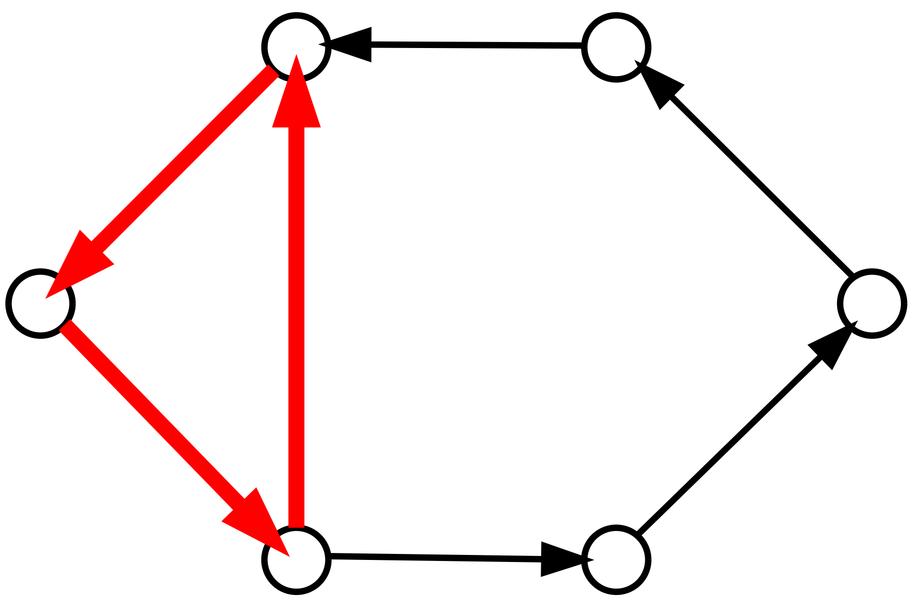
Can we use this lemma algorithmically?



Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.

Can we use this lemma algorithmically?
FVS must contain ≥ 1 vertex of each 3-cycle.

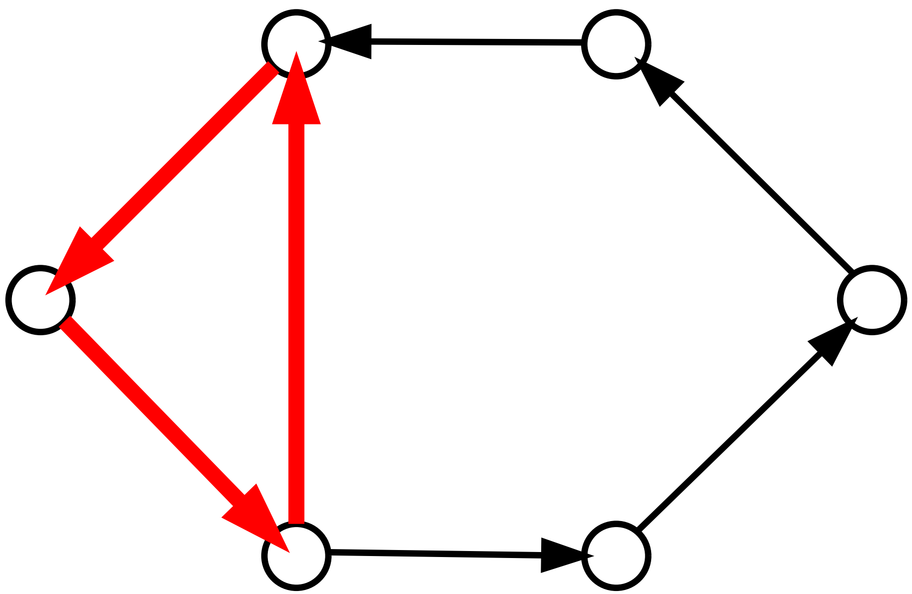


Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.

Can we use this lemma algorithmically?
FVS must contain ≥ 1 vertex of each 3-cycle.

Algorithm: Branch $(1, 1, 1)$.

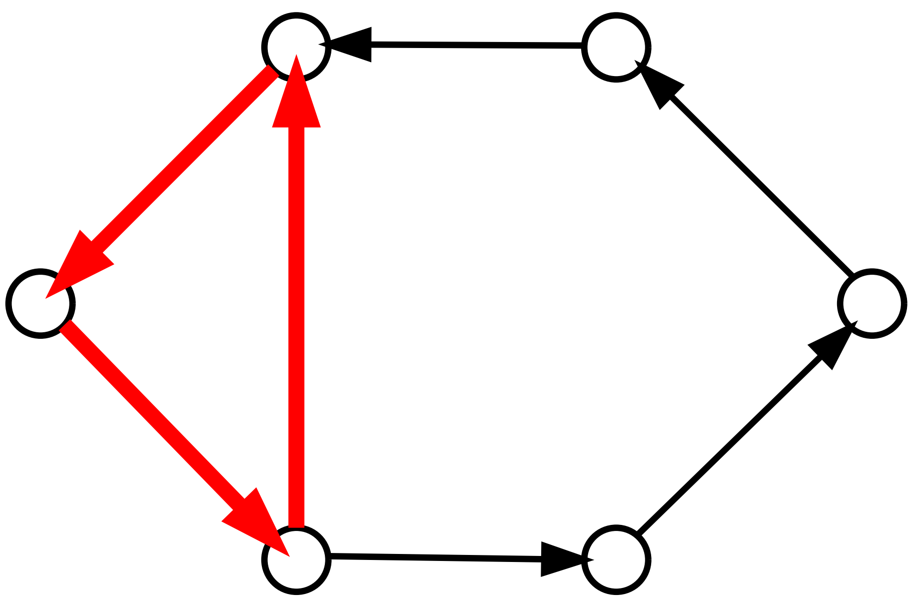


Cycles in Tournaments

Lemma. A tournament contains a cycle $\Leftrightarrow$ it contains a length-3 cycle.

Can we use this lemma algorithmically?
FVS must contain ≥ 1 vertex of each 3-cycle.

Algorithm: Branch $(1, 1, 1)$.



Theorem. FVS in Tournaments can be solved in $O^*(3^k)$ time.

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k.

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k, feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\, X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k, feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\, X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.

FVS in Tournaments

FVS (TOURNAMENTS) COMPRESSION

Given: Tournament $T = (V, E)$, number k,
feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.
Then:

FVS in Tournaments

FVS (TOURNAMENTS) COMPRESSION

Given: Tournament $T = (V, E)$, number k, feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\, X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.

Then: delete R and

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k,
feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.
Then: delete R and *forbid* F.

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k,
feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.
Then: delete R and *forbid* F.

Disjoint FVS (Tournaments) Compression

Given: Tournament $T' = (V', E')$, number k',
feedback vertex set $S' \subseteq V$ with $|S'| \leq k' + 1$

Question: $\exists X \subseteq V' \setminus S'$ so that $|X| \leq k'$ and $T' - X$ is acyclic?

FVS in Tournaments

FVS (Tournaments) Compression

Given: Tournament $T = (V, E)$, number k,
feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.
Then: delete R and *forbid* F.

Disjoint FVS (Tournaments) Compression

Given: Tournament $T' = (V', E')$, number k',
feedback vertex set $S' \subseteq V$ with $|S'| \leq k' + 1$

Question: $\exists X \subseteq V' \setminus S'$ so that $|X| \leq k'$ and $T' - X$ is acyclic?

Reduction: FVS-T Comp. by 2^k calls to Disj.-FVS-T Comp.

FVS in Tournaments

FVS (TOURNAMENTS) COMPRESSION

Given: Tournament $T = (V, E)$, number k, feedback vertex set $S \subseteq V$ with $|S| \leq k + 1$

Question: $\exists\ X \subseteq V$ so that $|X| \leq k$ and $T \setminus X$ is acyclic?

Strategy: Fix $X \cap S$. Let $R = X \cap S$ and $F = S \setminus X$.
Then: delete R and *forbid* F.

DISJOINT FVS (TOURNAMENTS) COMPRESSION

Given: Tournament $T' = (V', E')$, number k', feedback vertex set $S' \subseteq V$ with $|S'| \leq k' + 1$

Question: $\exists X \subseteq V' \setminus S'$ so that $|X| \leq k'$ and $T' - X$ is acyclic?

Reduction: FVS-T COMP. by 2^k calls to DISJ.-FVS-T COMP.
For each call, set $T' := T - R$, $S' := F$, $k' := |F| - 1$.

Disjoint FVS in Tournaments

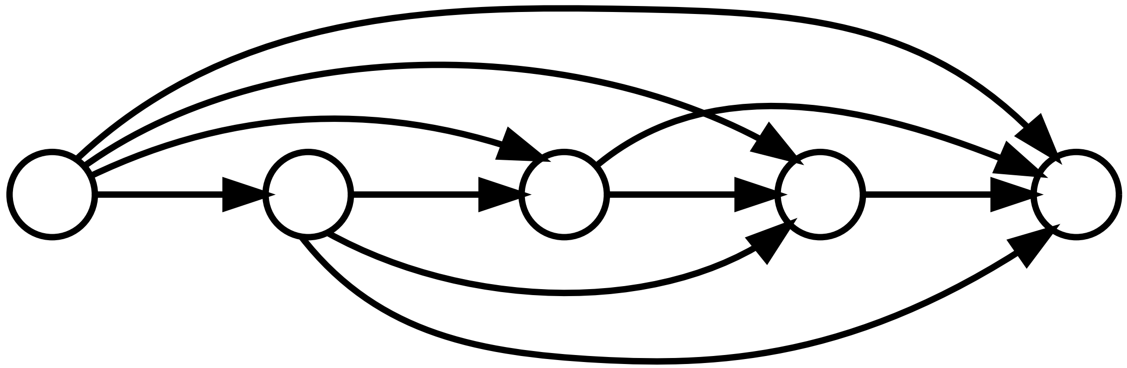
Reduction Rule: If $T'[F]$ is not acyclic, answer: No.

Disjoint FVS in Tournaments

Reduction Rule: If $T'[F]$ is not acyclic, answer: No.

Obs. Acyclic tournaments have a total order on vertices.

F

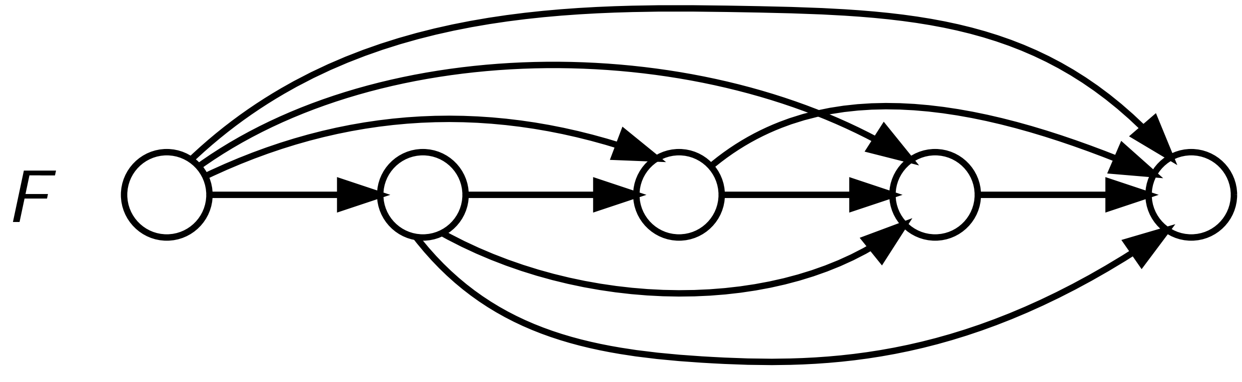


Disjoint FVS in Tournaments

Reduction Rule: If $T'[F]$ is not acyclic, answer: $\textsc{No}$.

Obs. Acyclic tournaments have a total order on vertices.

Let $A := V' \setminus F$ be the set of *chooseable* vertices.



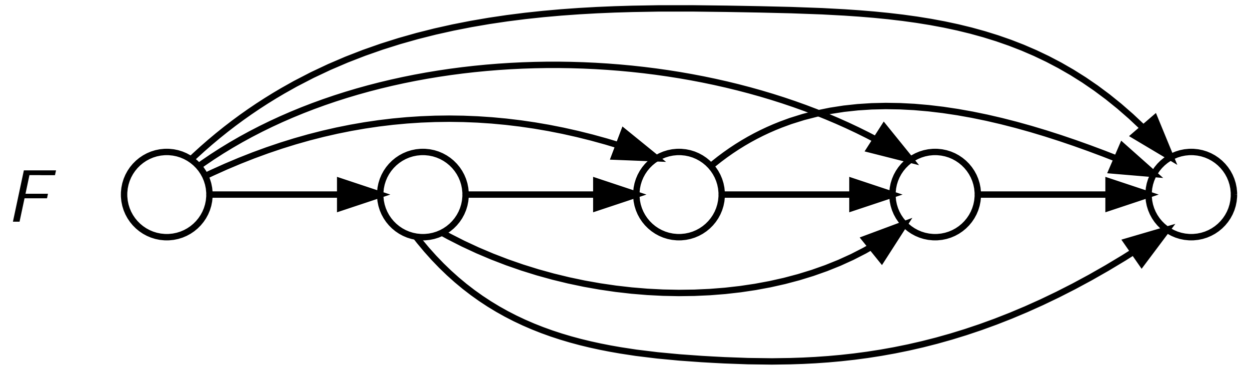
Disjoint FVS in Tournaments

Reduction Rule: If $T'[F]$ is not acyclic, answer: NO.

Obs. Acyclic tournaments have a total order on vertices.

Let $A := V' \setminus F$ be the set of *chooseable* vertices.

Reduction Rule: Let $v \in A$. If $T'[F \cup \{v\}]$ is not acyclic, then delete v and set $k' \leftarrow k' - 1$.



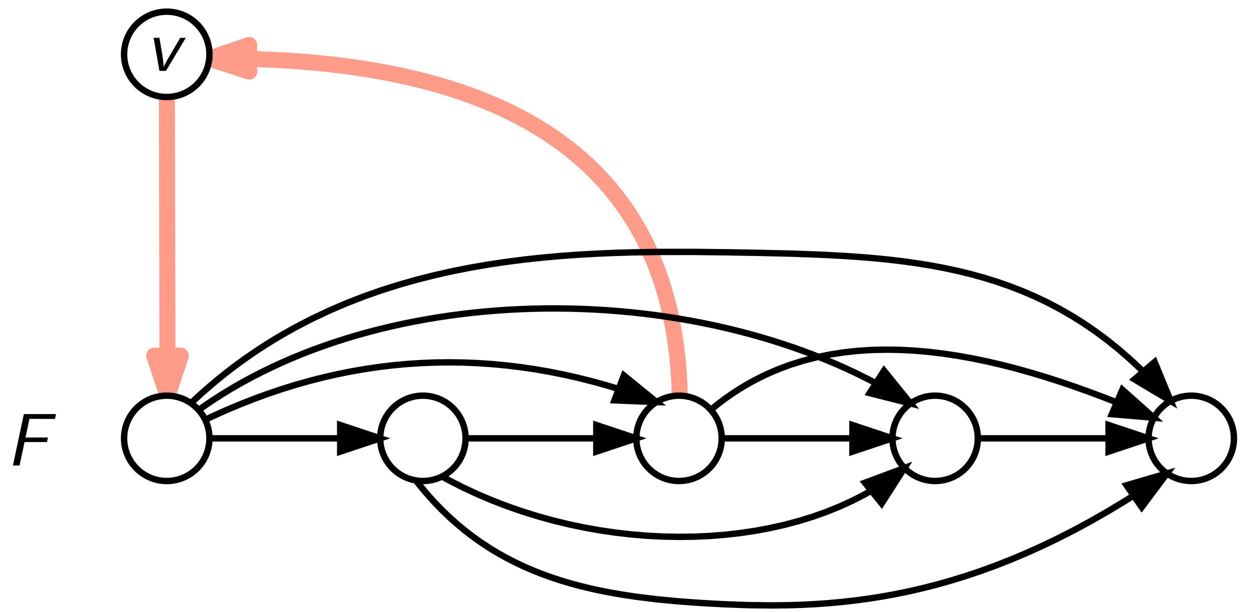
Disjoint FVS in Tournaments

Reduction Rule: If $T'[F]$ is not acyclic, answer: No.

Obs. Acyclic tournaments have a total order on vertices.

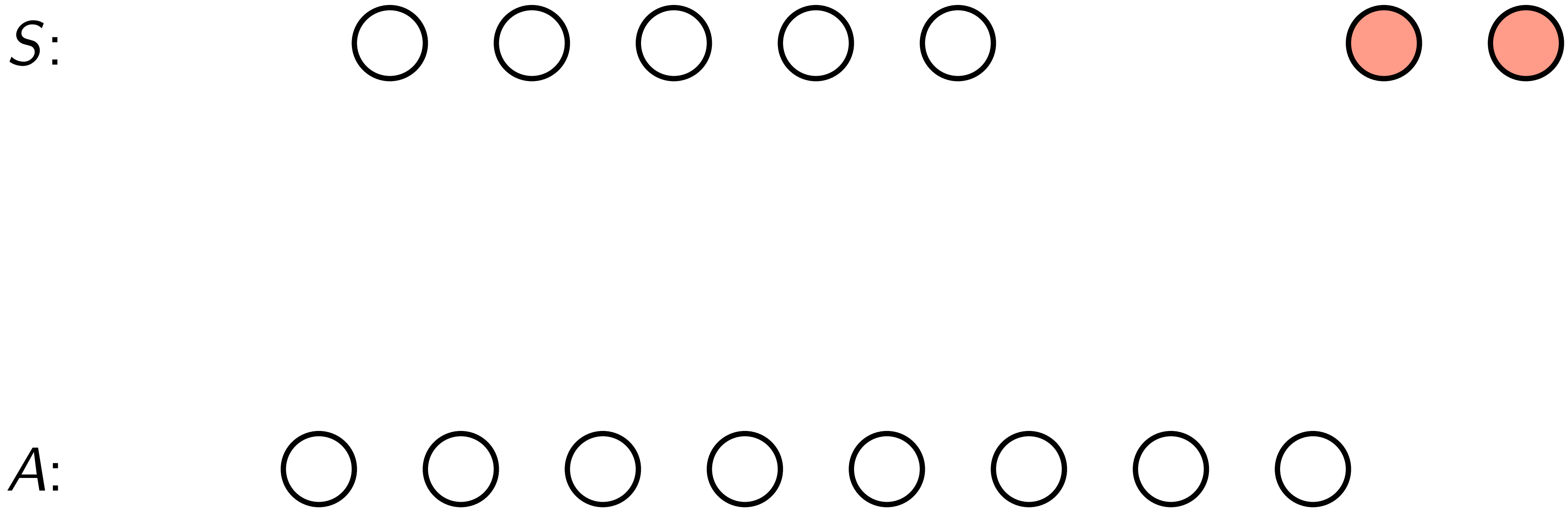
Let $A := V' \setminus F$ be the set of *chooseable* vertices.

Reduction Rule: Let $v \in A$. If $T'[F \cup \{v\}]$ is not acyclic, then delete v and set $k' \leftarrow k' - 1$.



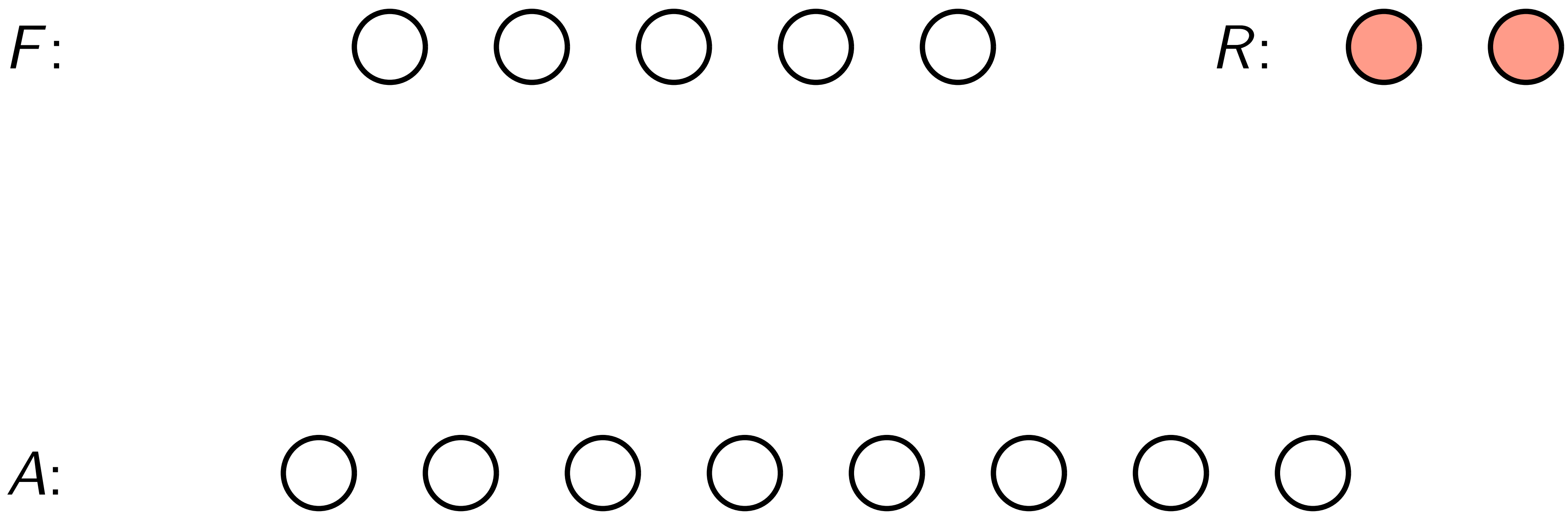
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \uplus R$ is FVS, $A = V \setminus S$



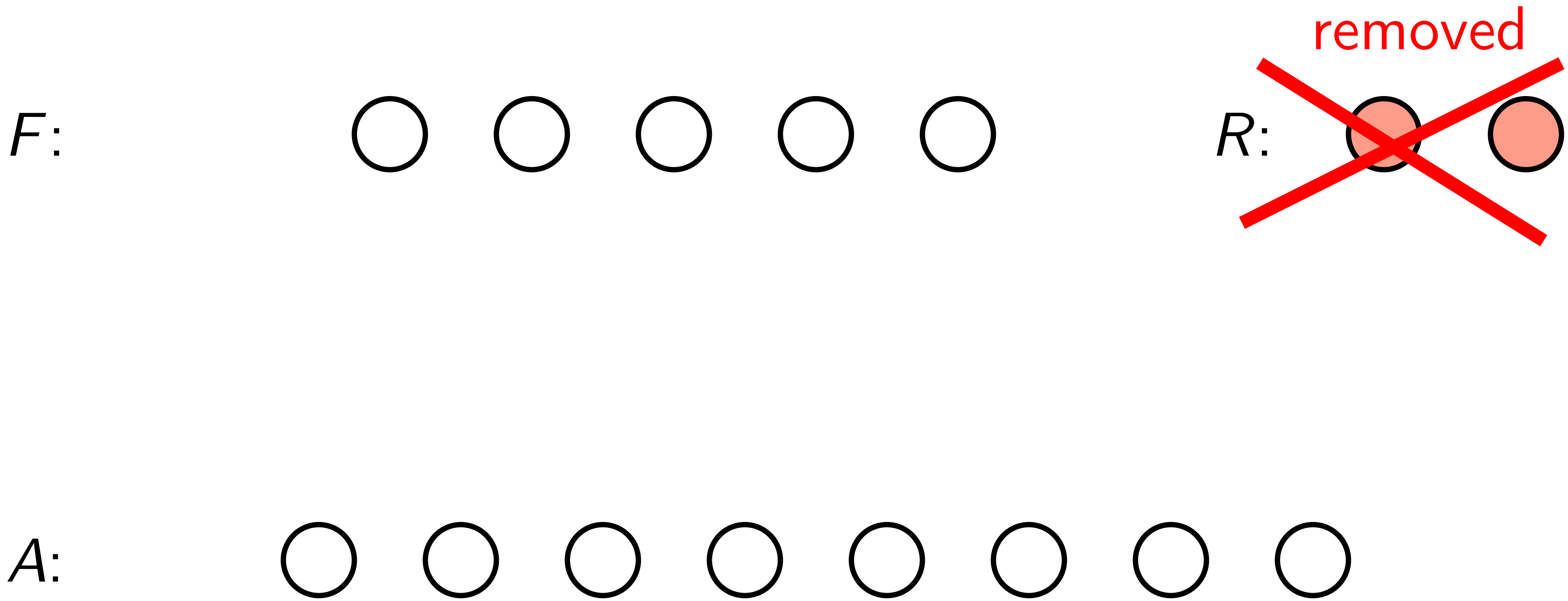
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $\quad S = F \uplus R$ is FVS, $\quad A = V \setminus S$



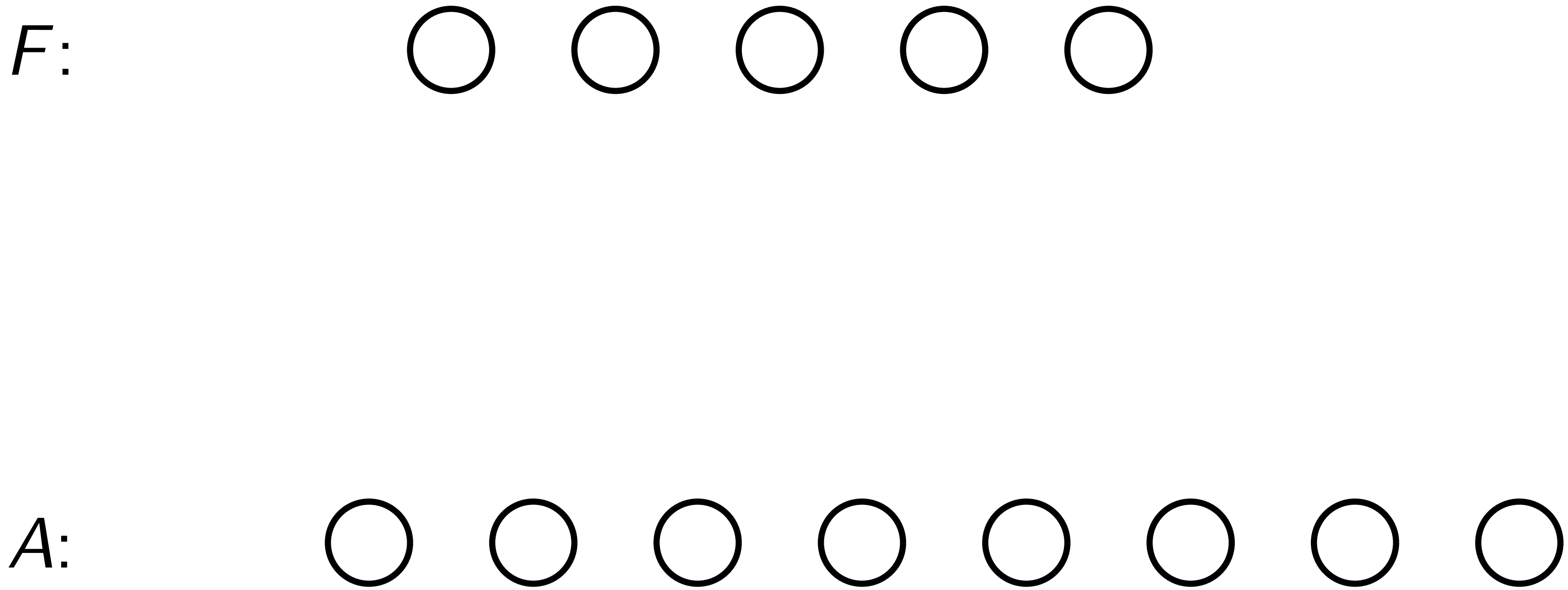
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \uplus R$ is FVS, $A = V \setminus S$



Disjoint FVS in Tournaments (cont'd)

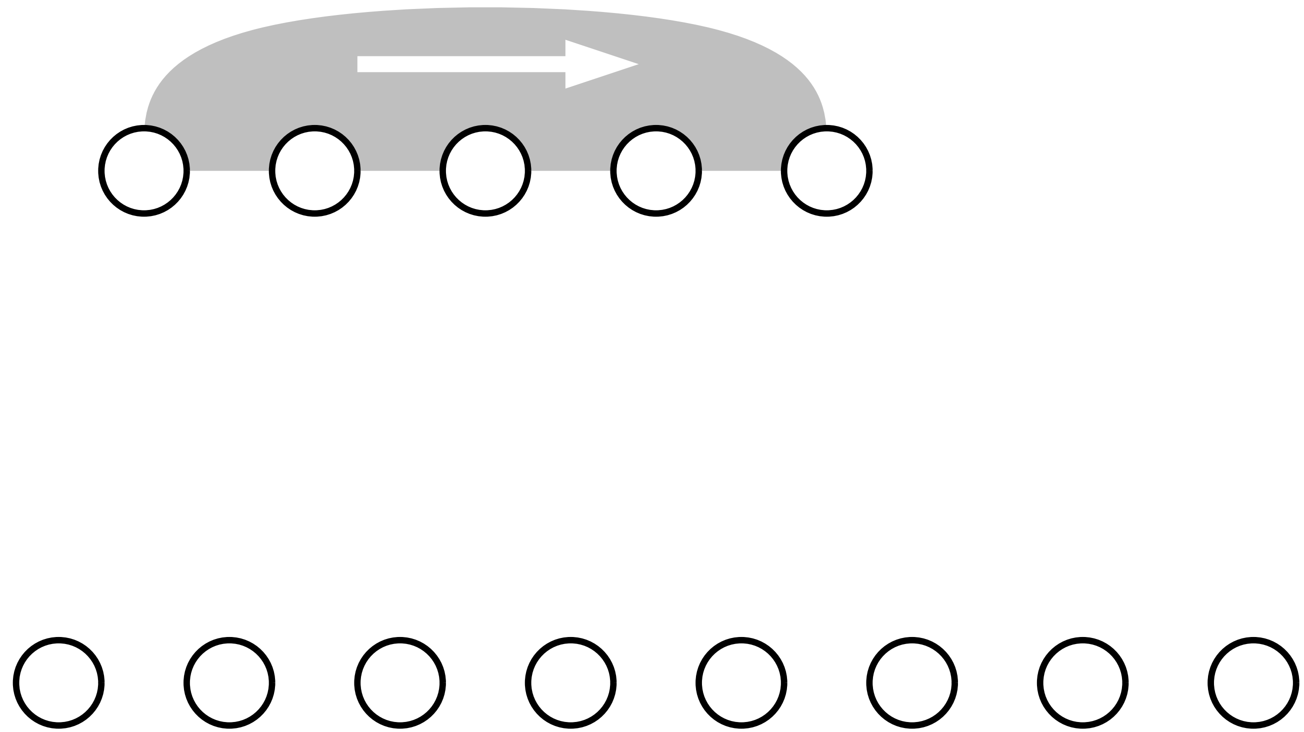
$T = (V, E)$, $\quad S = F \,\dot\cup\, R$ is FVS, $\quad A = V \setminus S$



Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$

F:

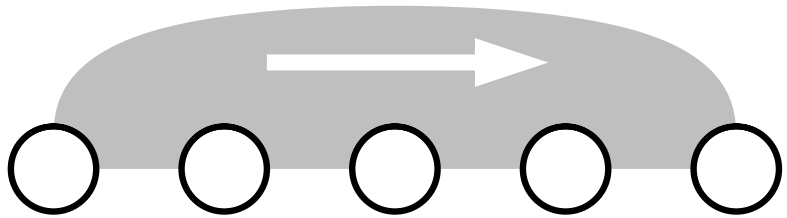


A:

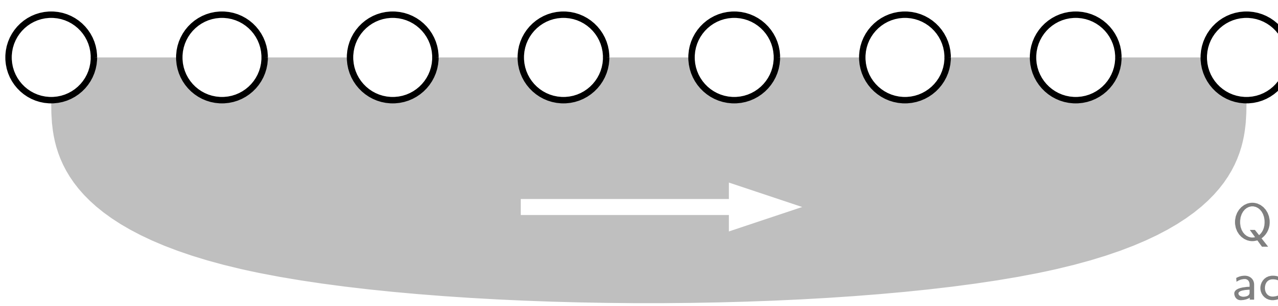
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$

F:



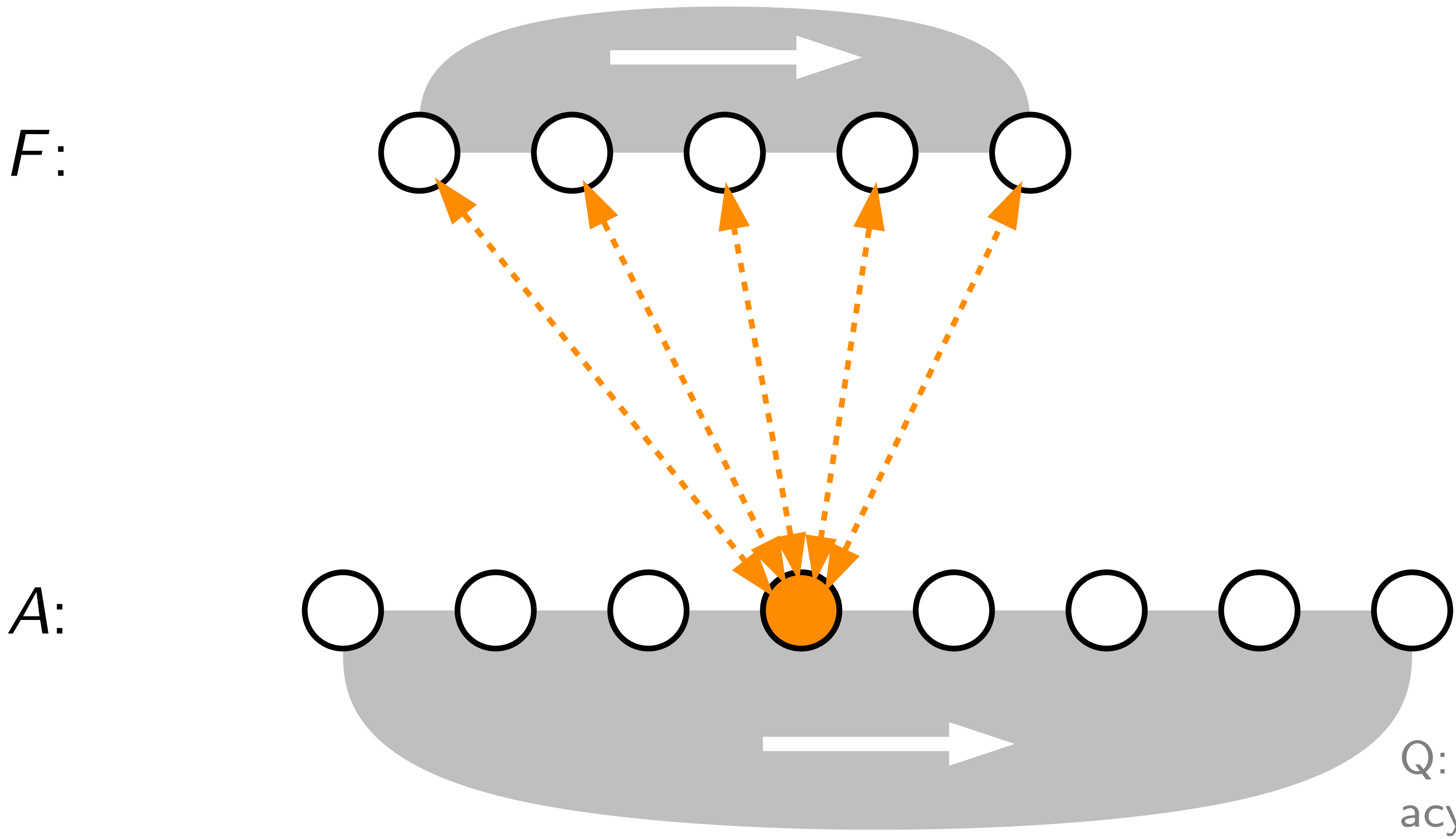
A:



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

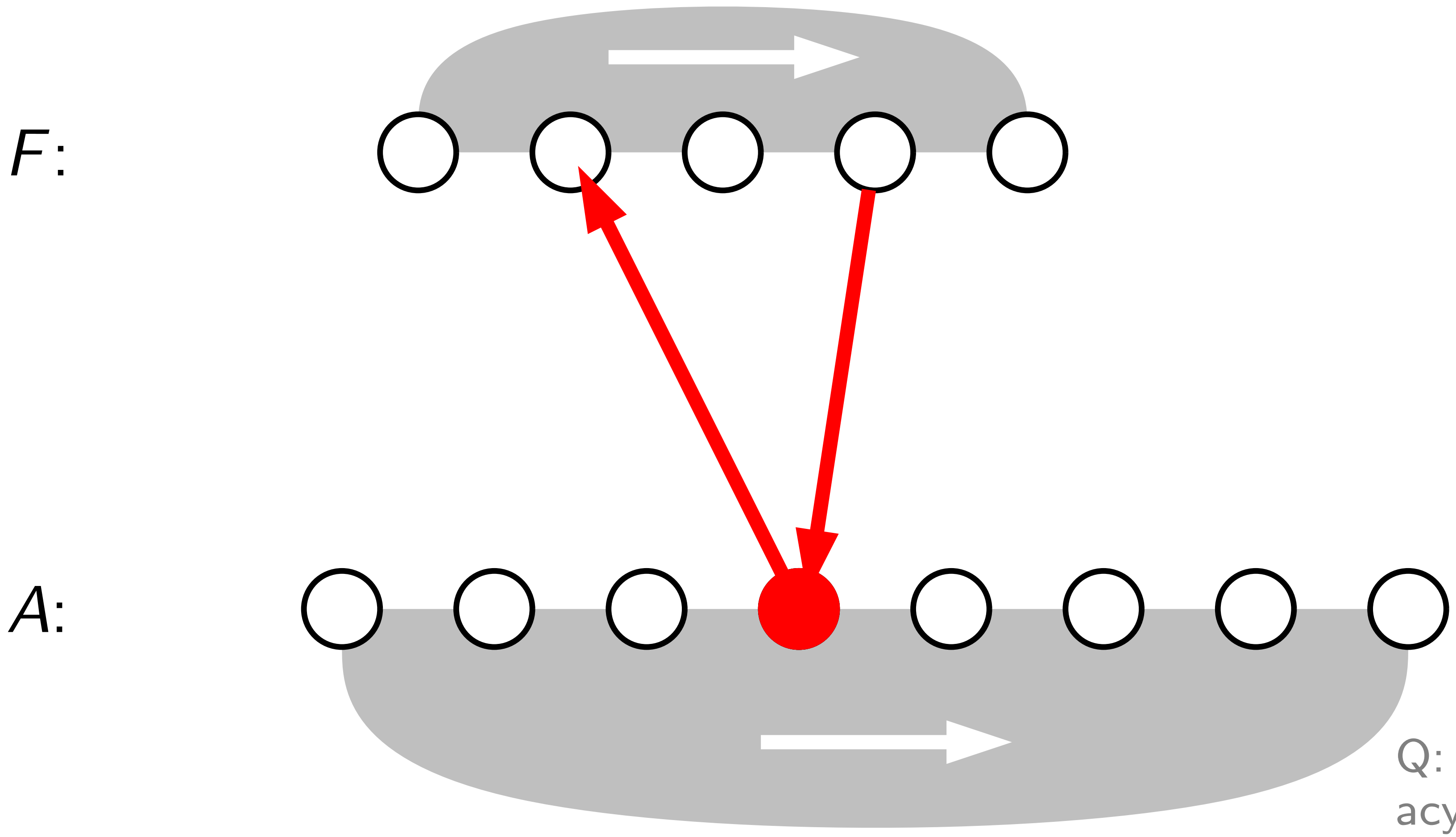
$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

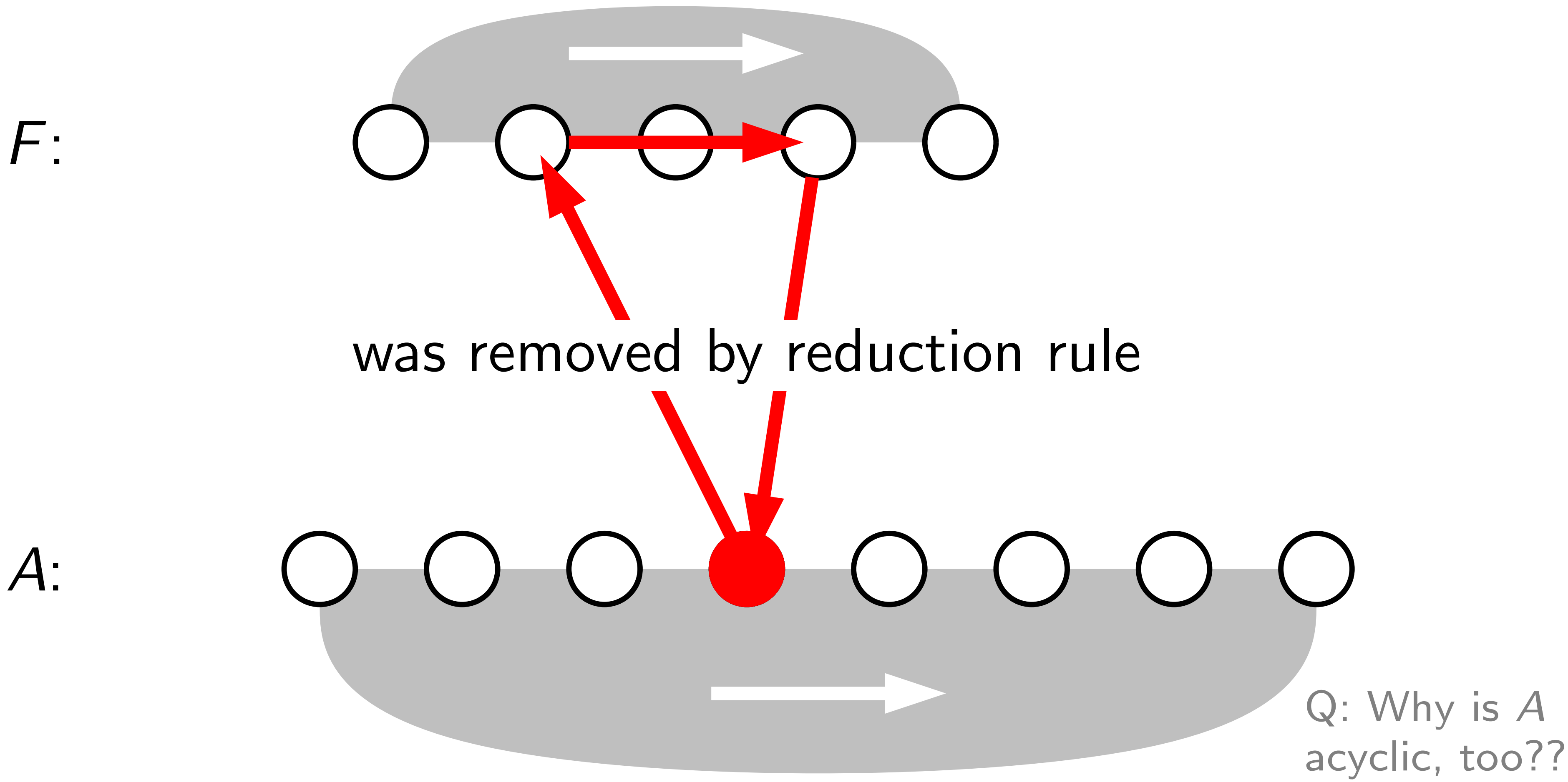
$T = (V, E)$, $S = F \uplus R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

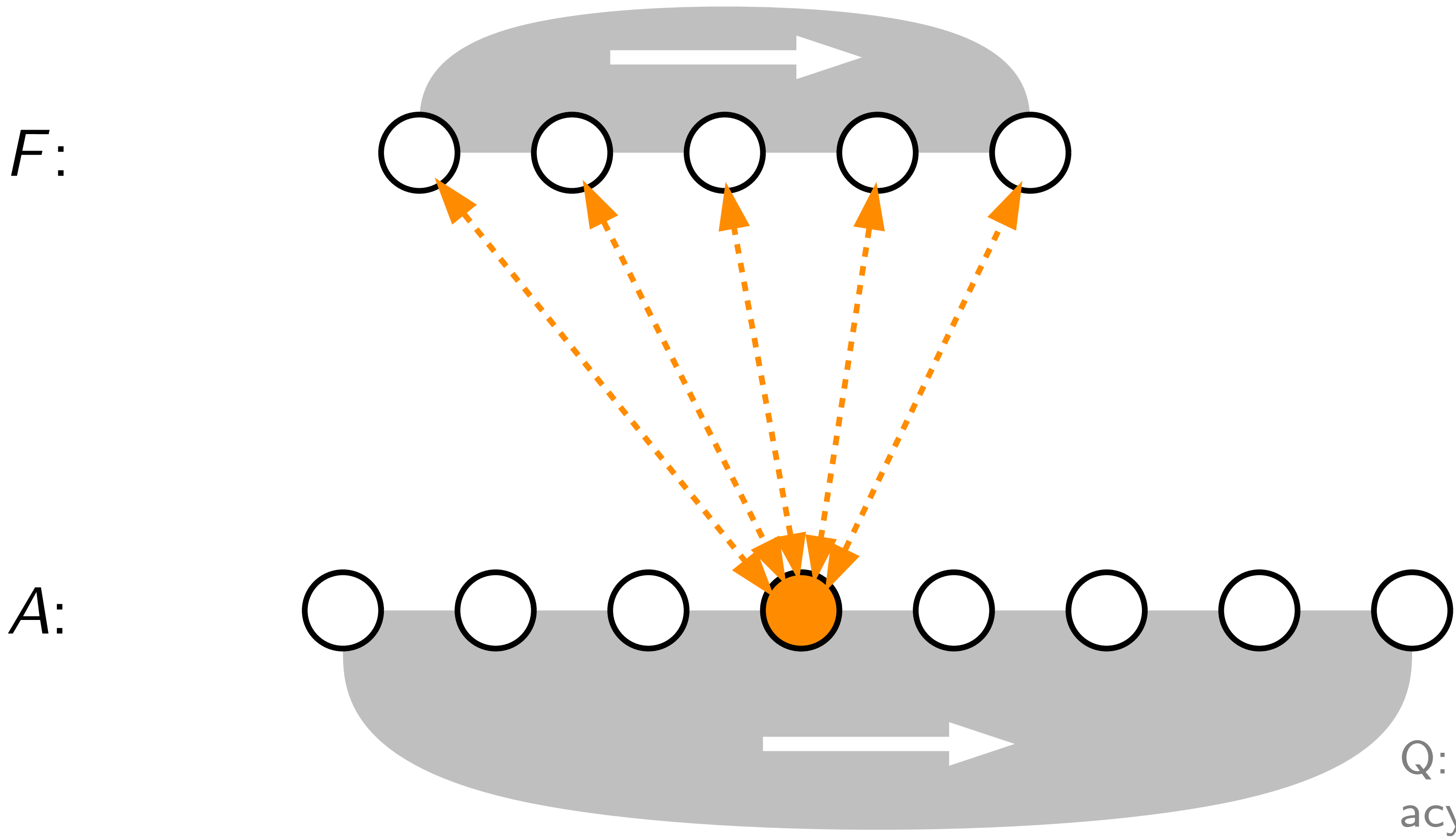
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



Disjoint FVS in Tournaments (cont'd)

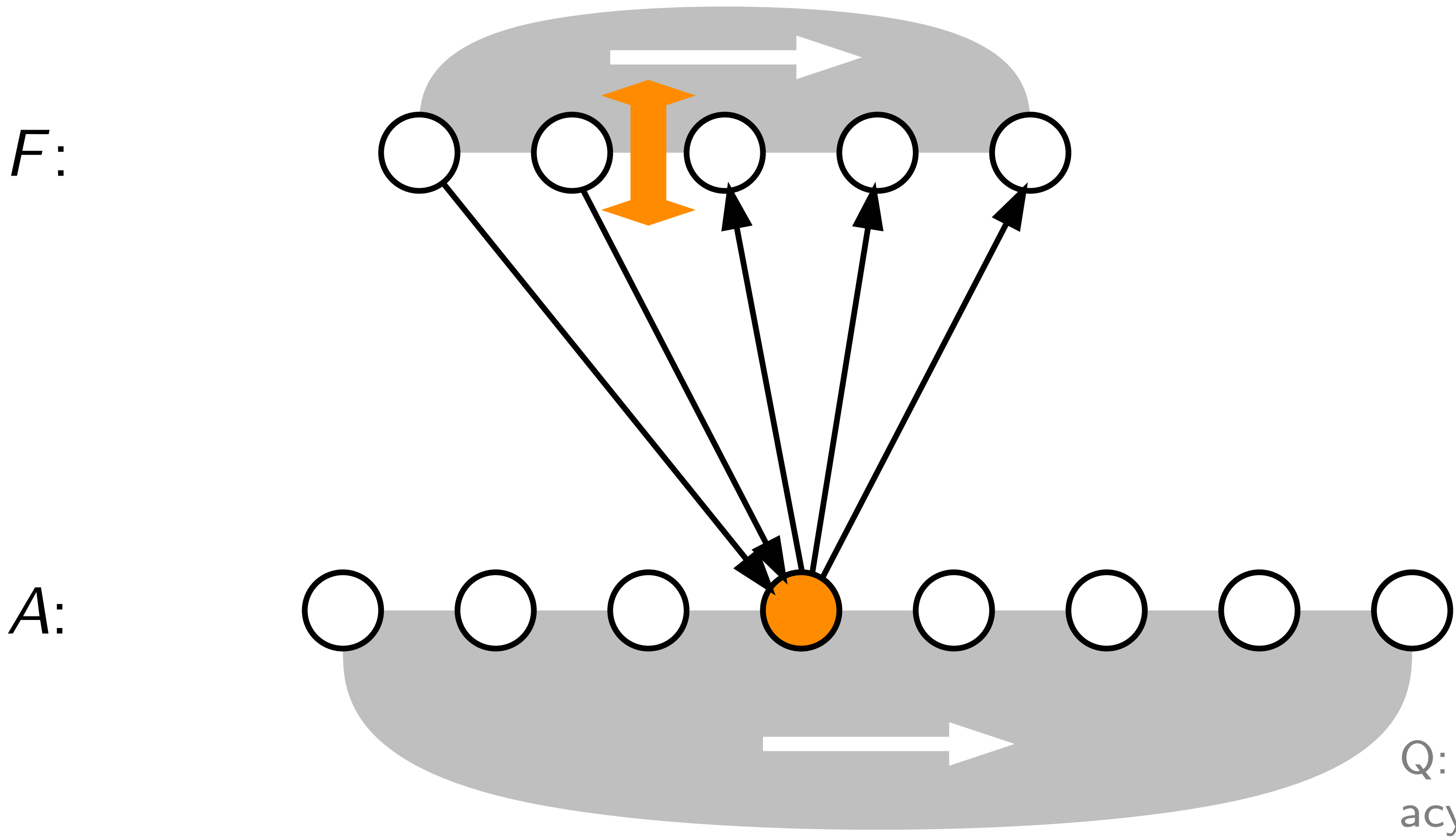
$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

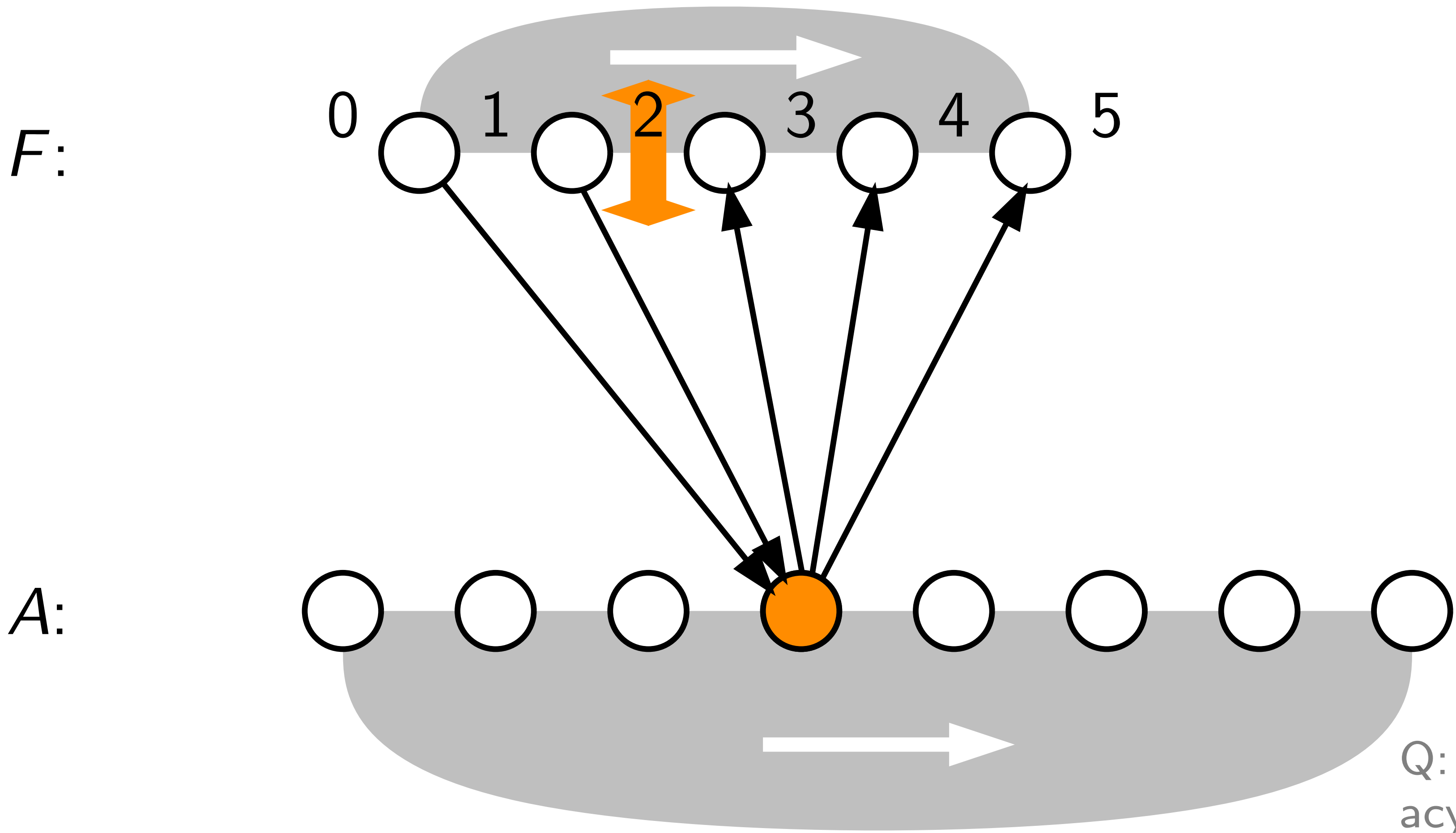
$T = (V, E)$, $S = F \cup R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

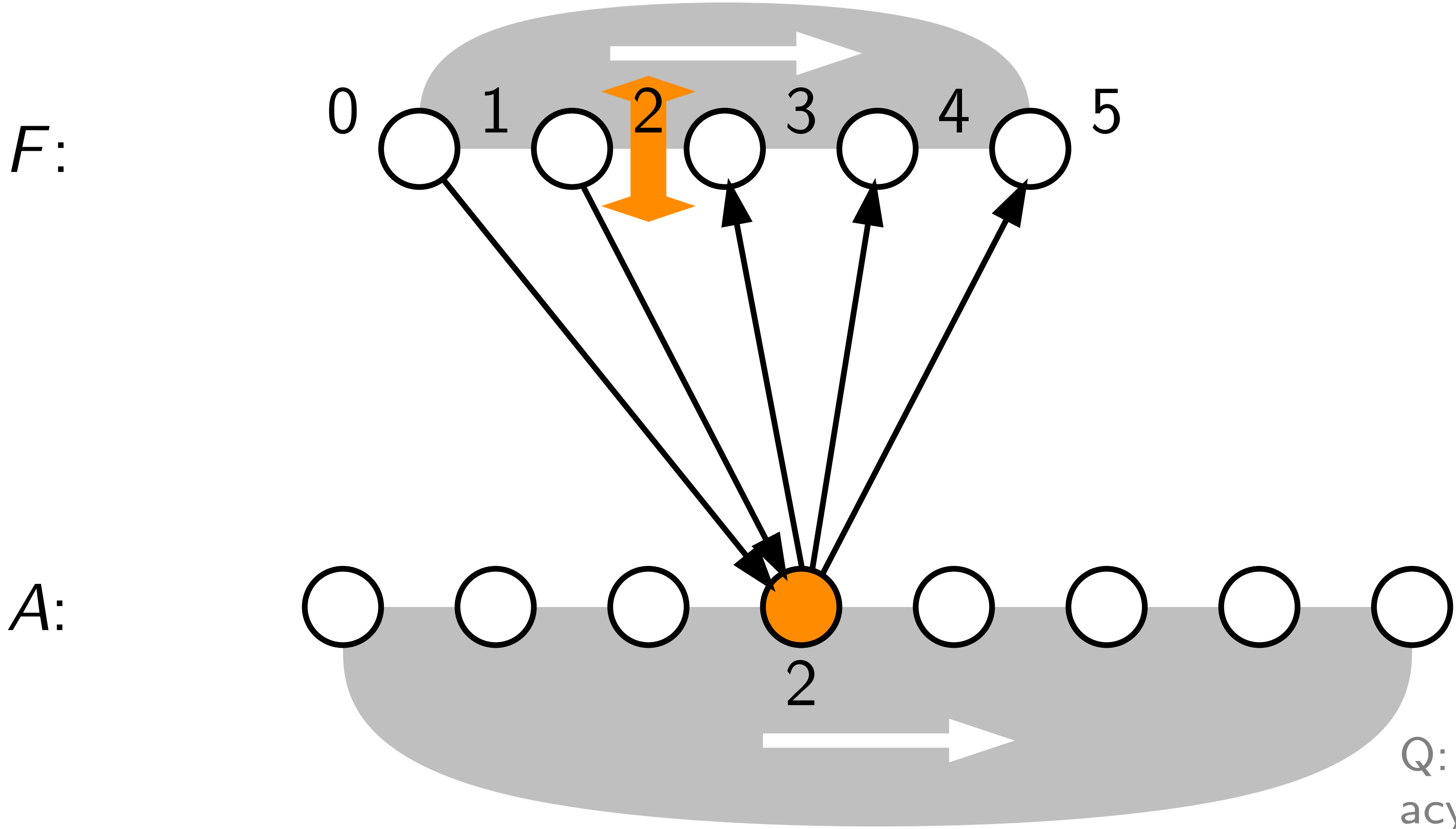
$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

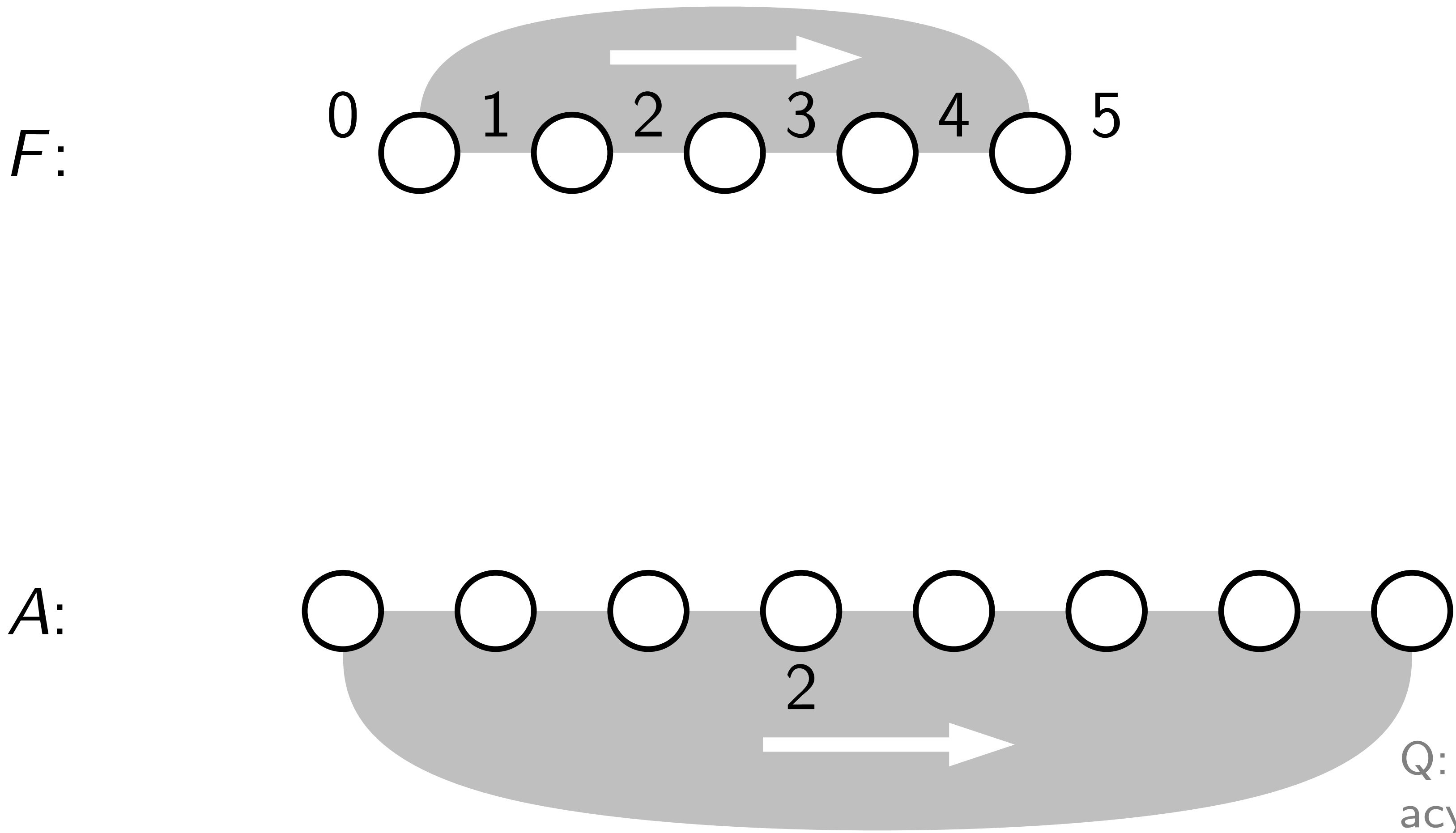
$T = (V, E)$, $S = F \cupdot R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

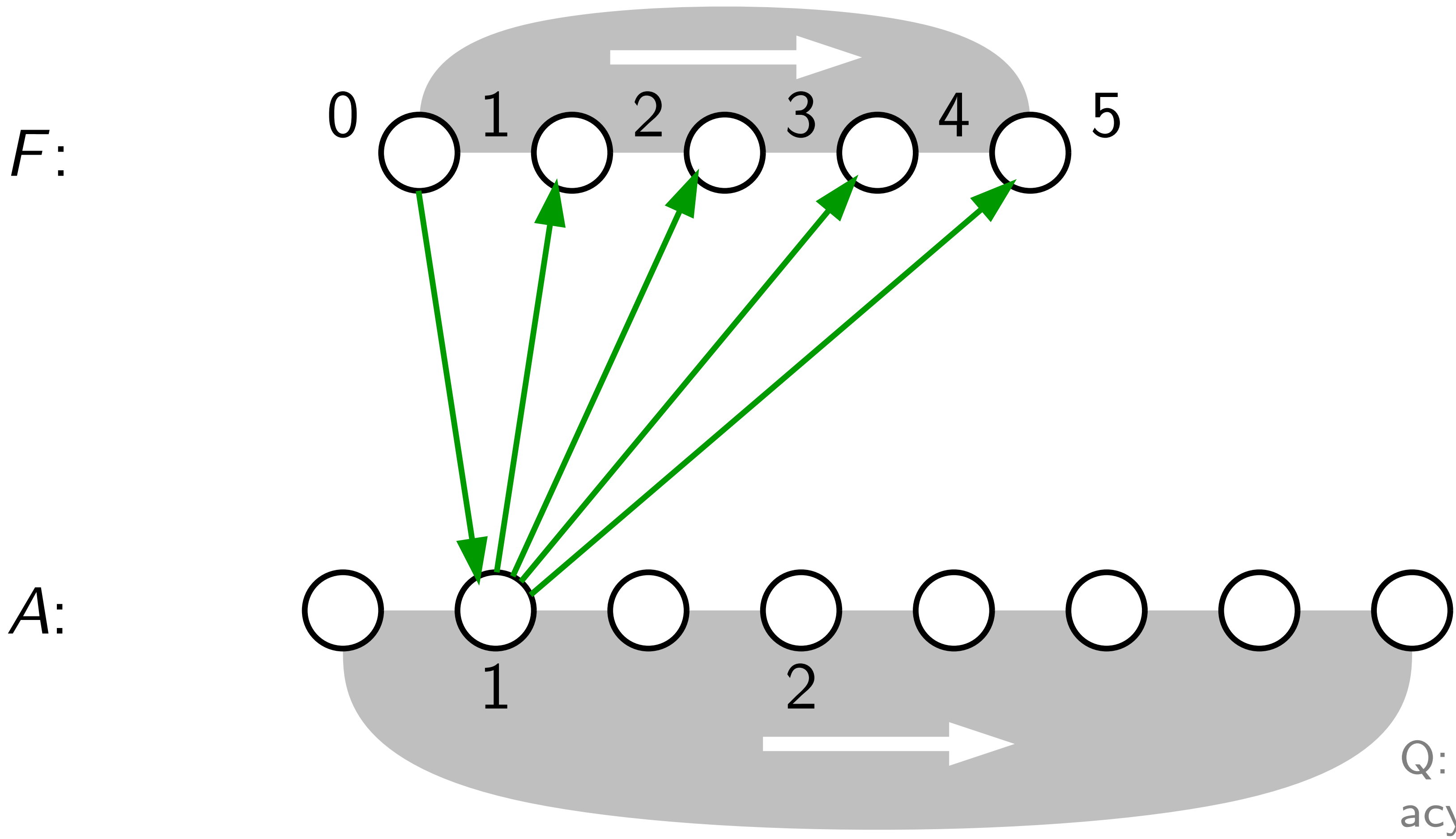
$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

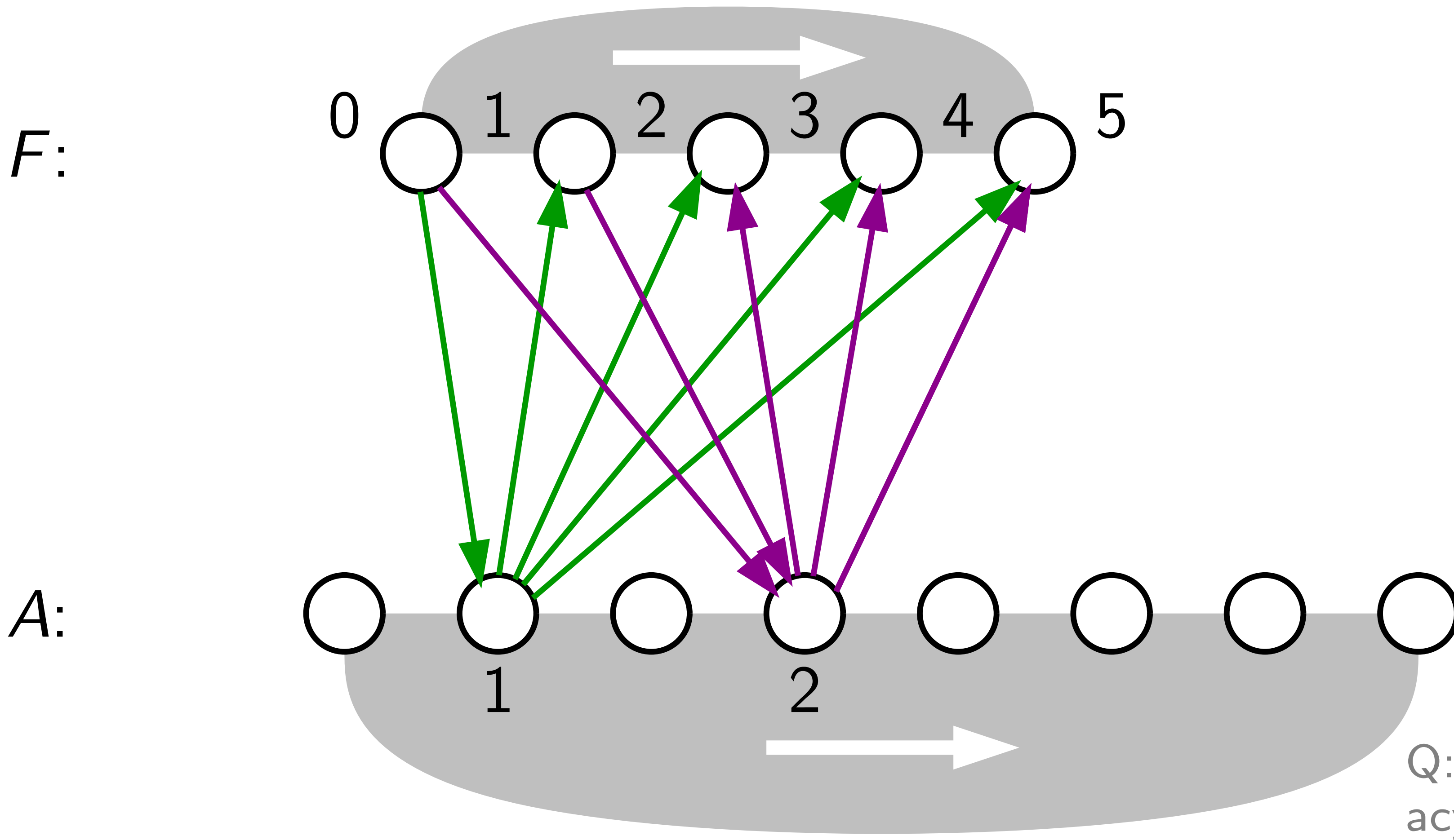
$T = (V, E)$, $\quad S = F \,\dot{\cup}\, R$ is FVS, $\quad A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

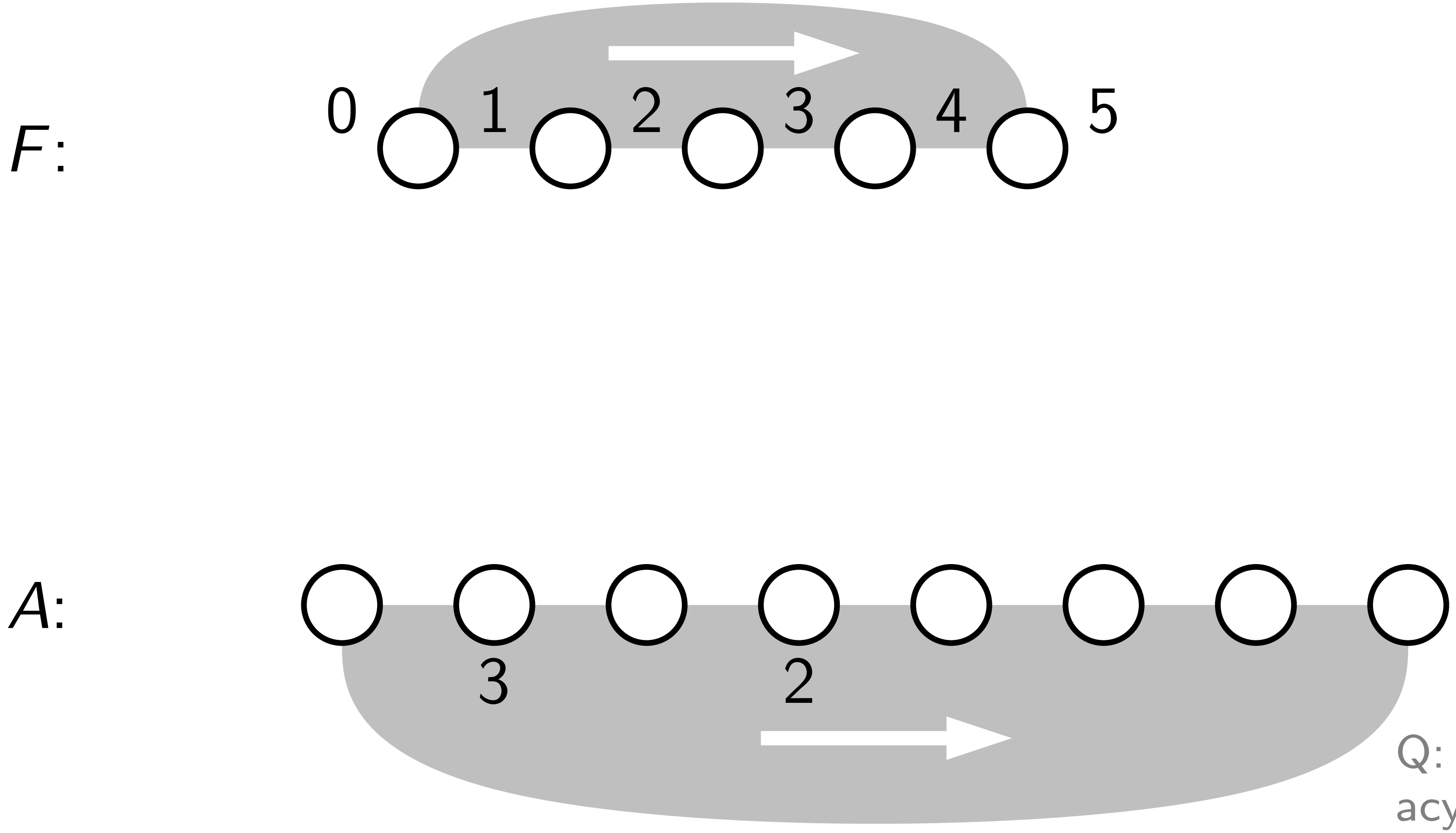
$T = (V, E)$, $S = F \uplus R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

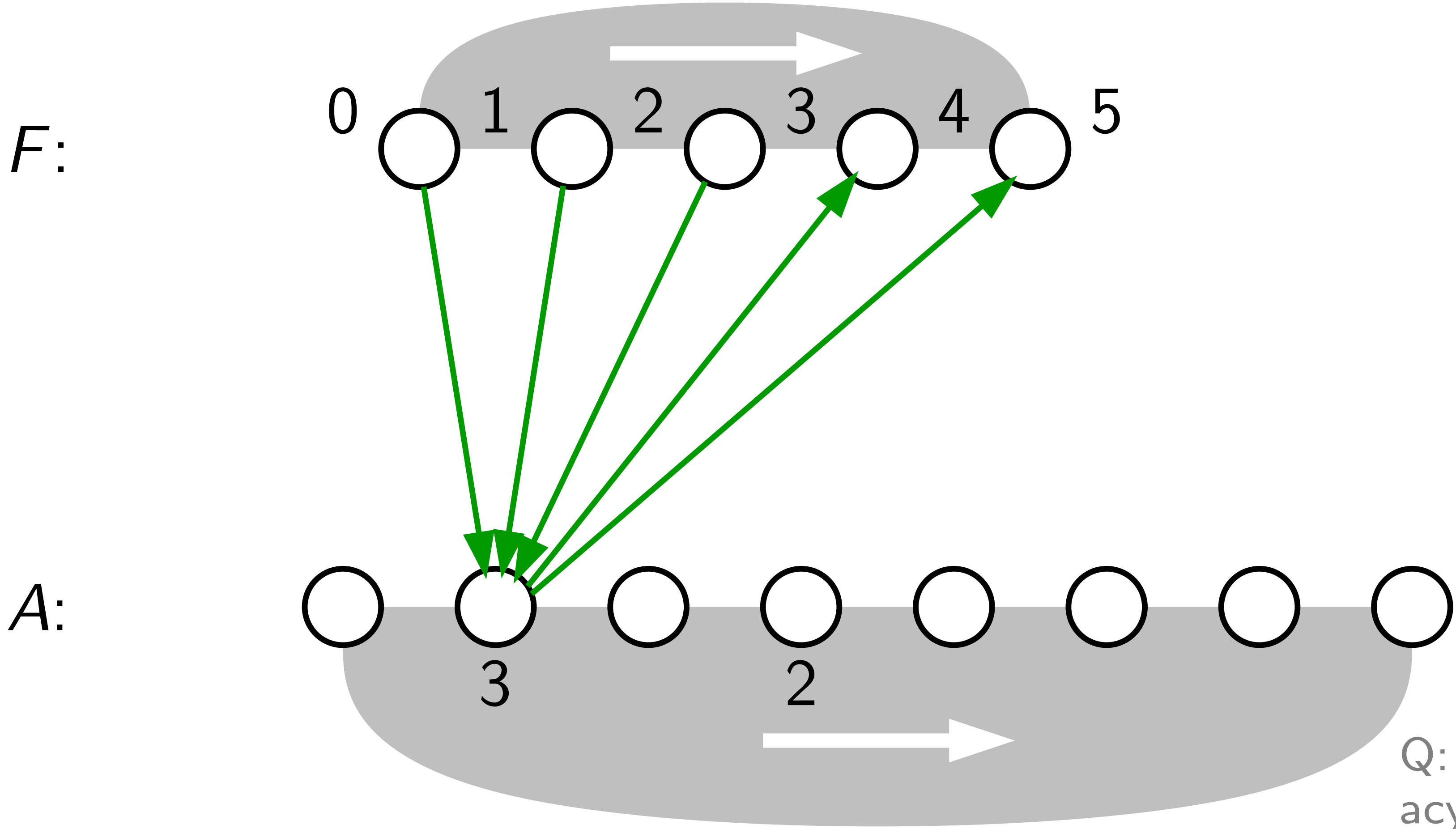
$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

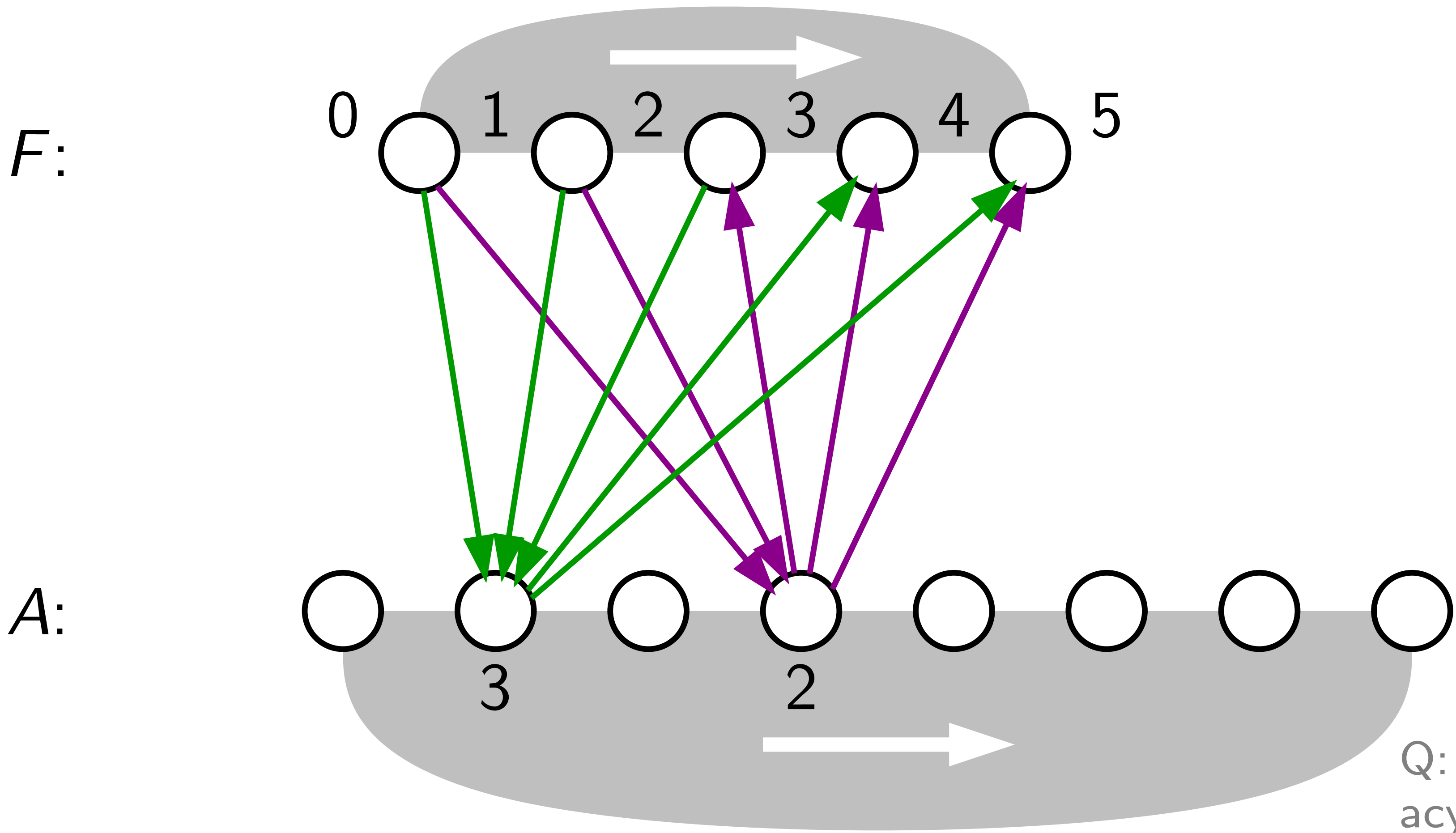
$T = (V, E)$, $\quad S = F \uplus R$ is FVS, $\quad A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

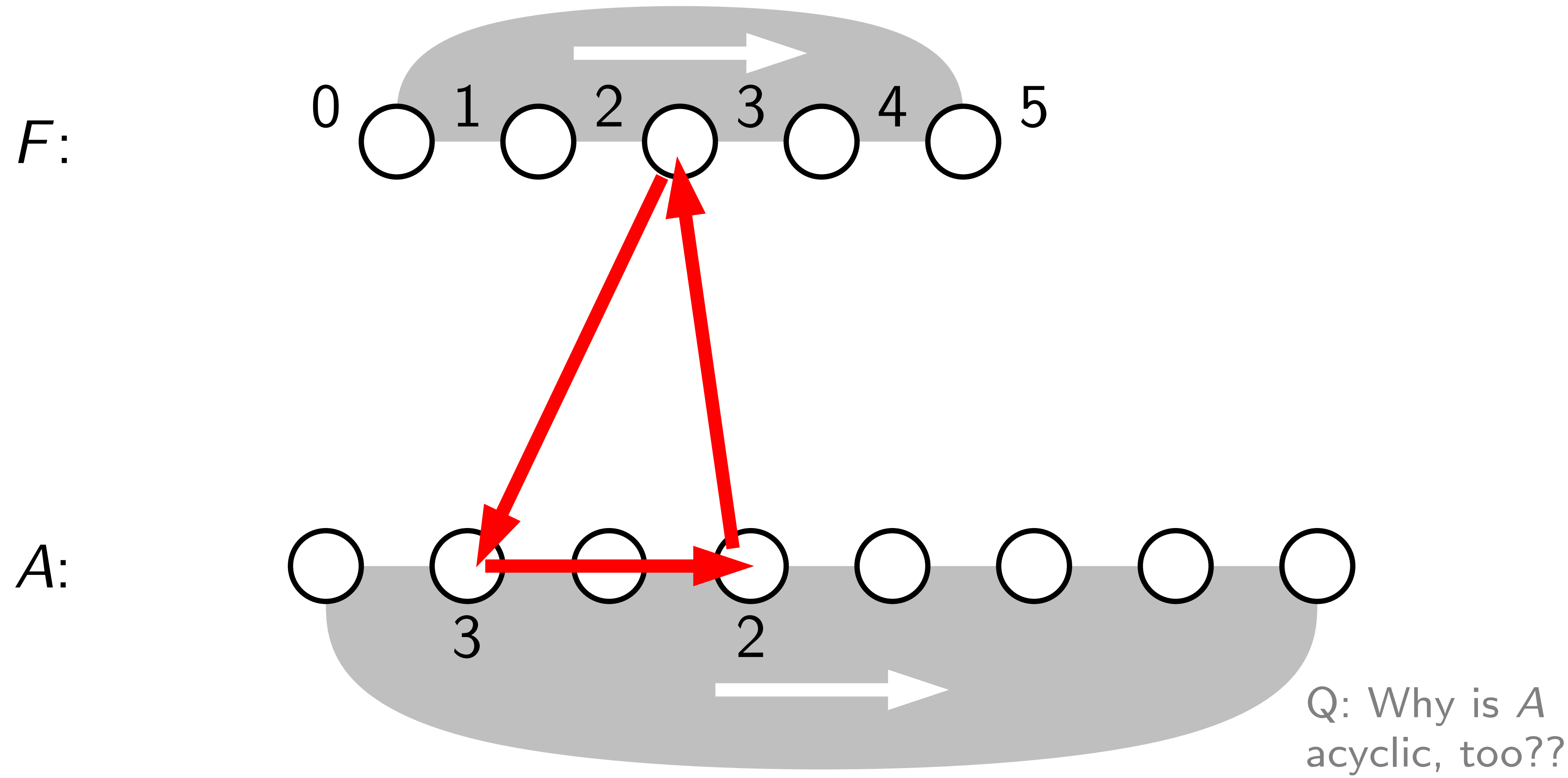
$T = (V, E)$, $S = F \,\dot\cup\, R$ is FVS, $A = V \setminus S$



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

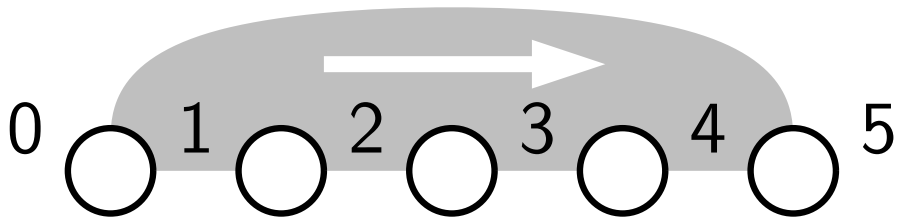
$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



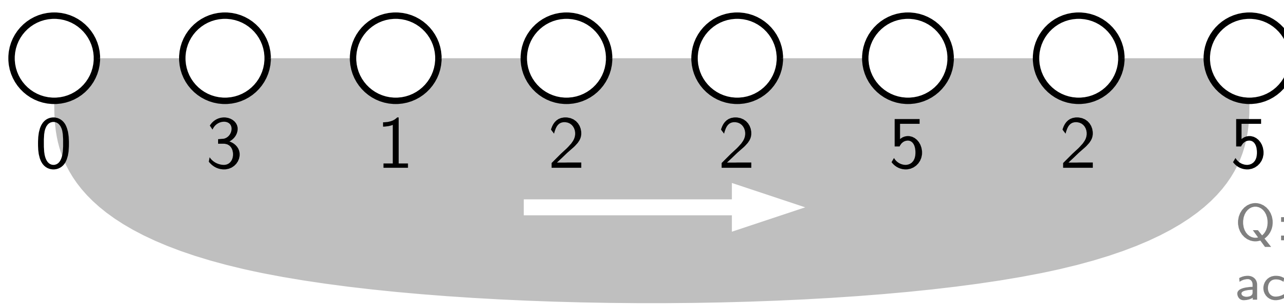
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \dot{\cup} R$ is FVS, $A = V \setminus S$

F:



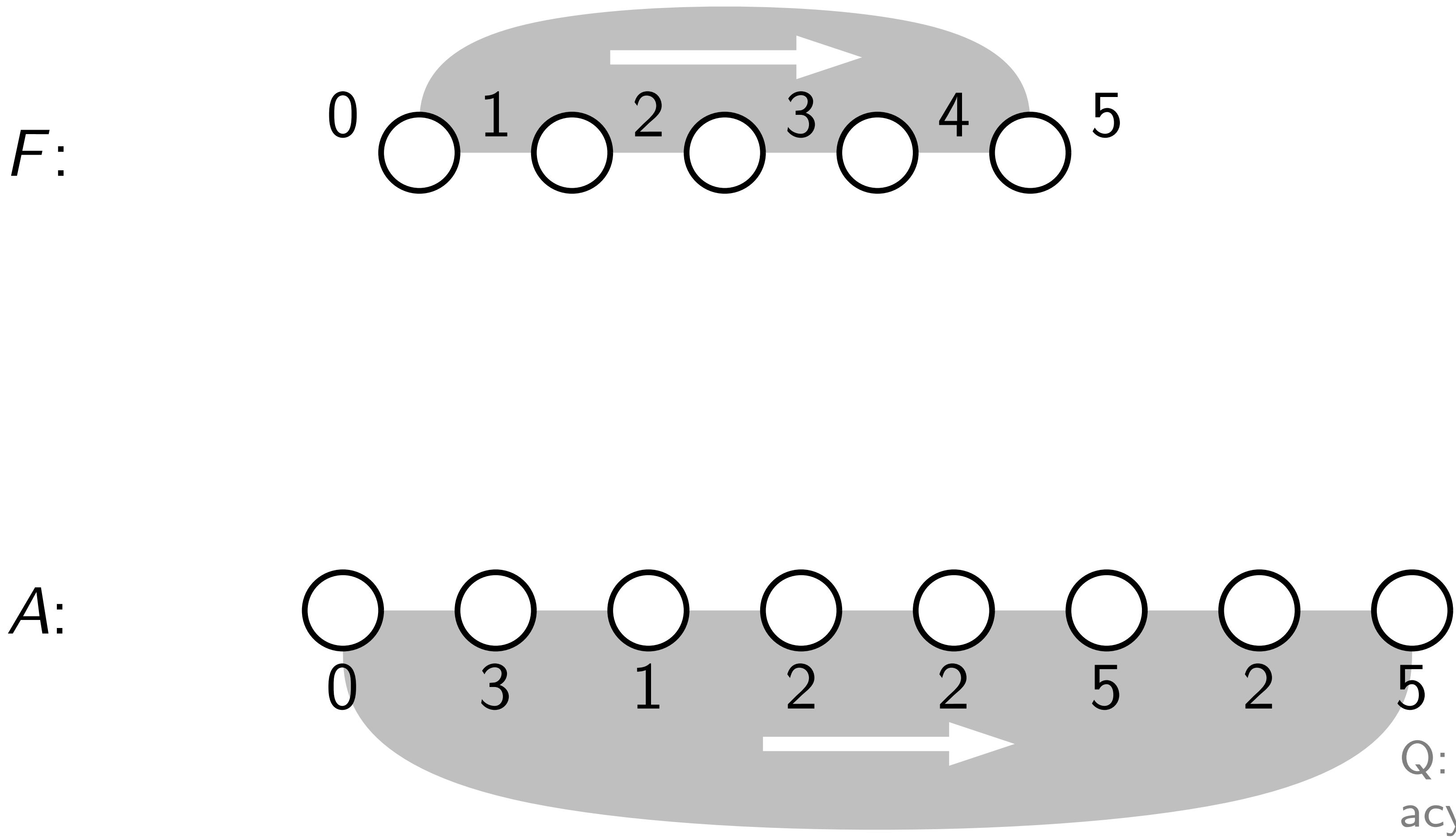
A:



Q: Why is A acyclic, too??

Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$



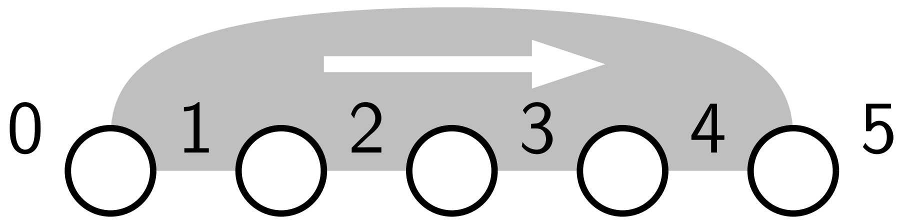
Q: Why is A acyclic, too??

Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

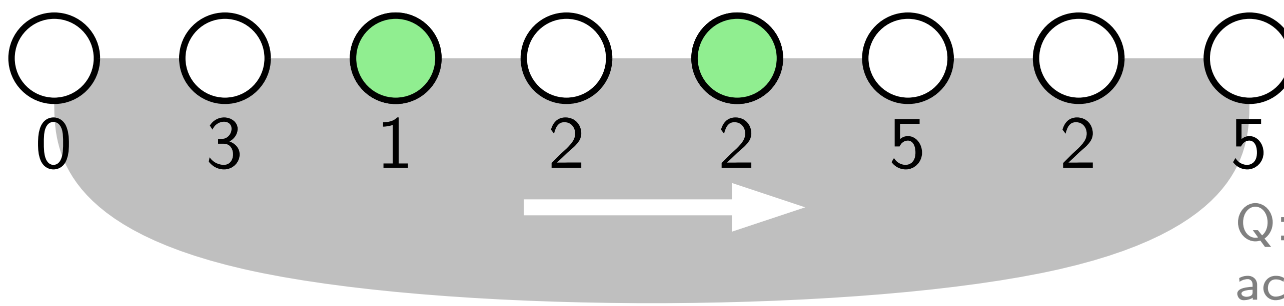
$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$

F:



might be possible to keep both

A:



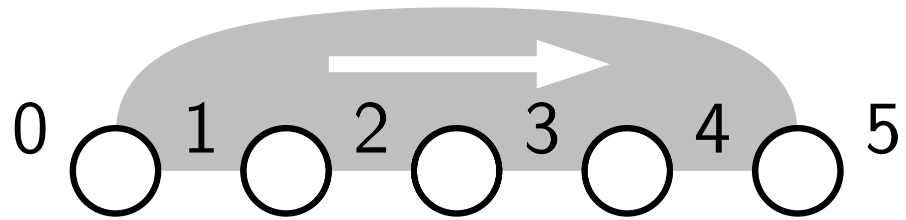
Q: Why is A acyclic, too??

Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

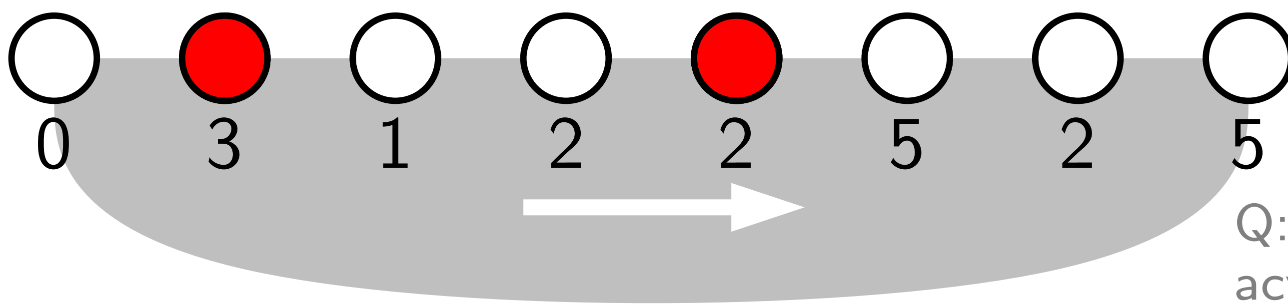
$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$

F:



cannot keep both, otherwise cycle

A:

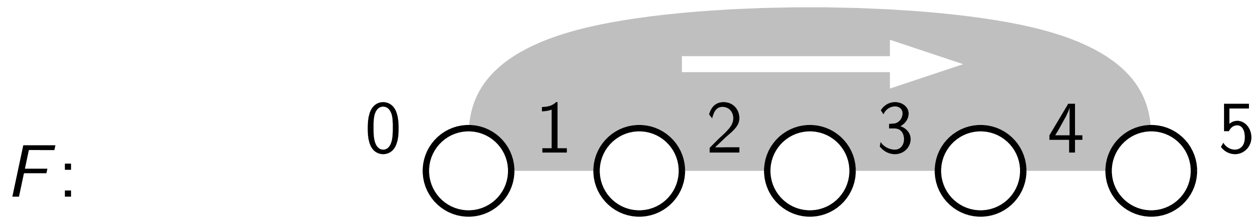


Q: Why is A acyclic, too??

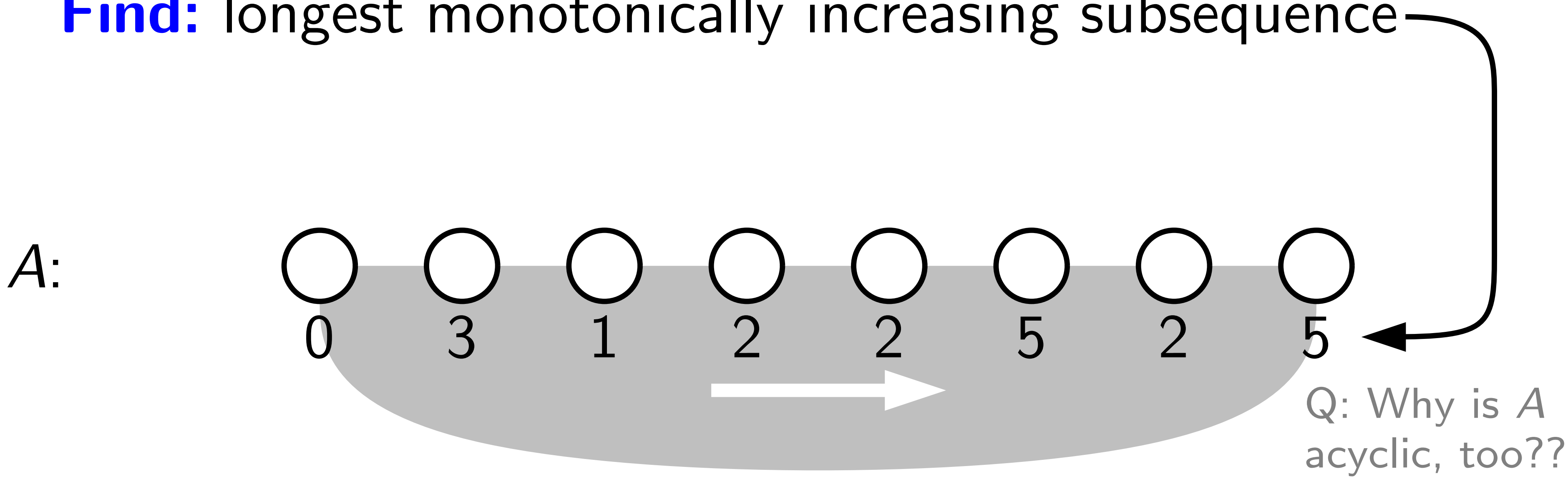
Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$



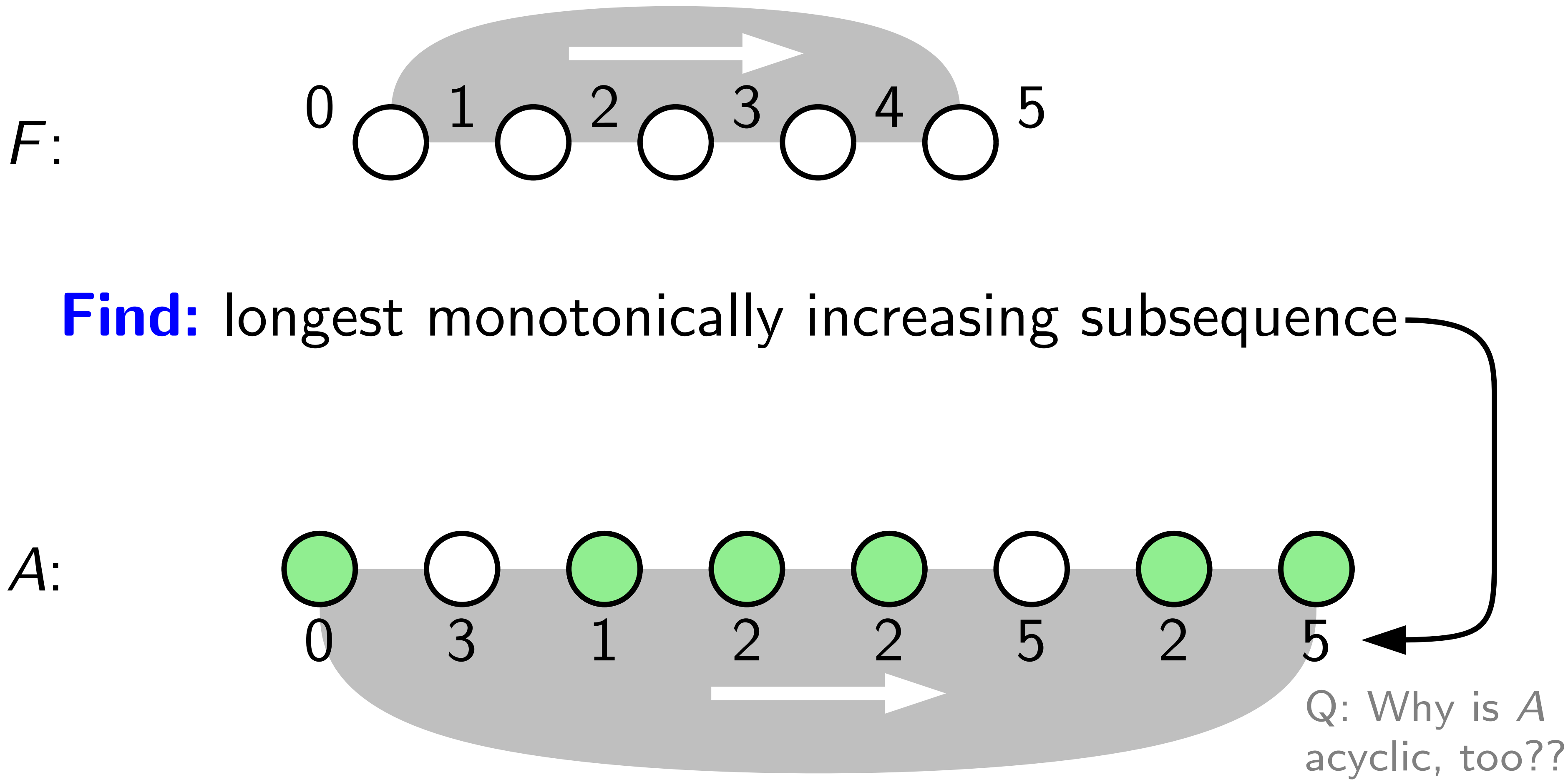
Find: longest monotonically increasing subsequence



Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \dot\cup R$ is FVS, $A = V \setminus S$

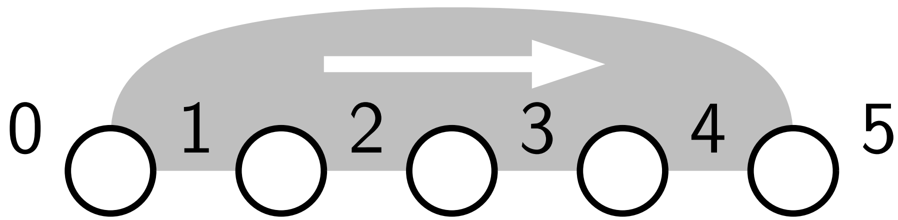


Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

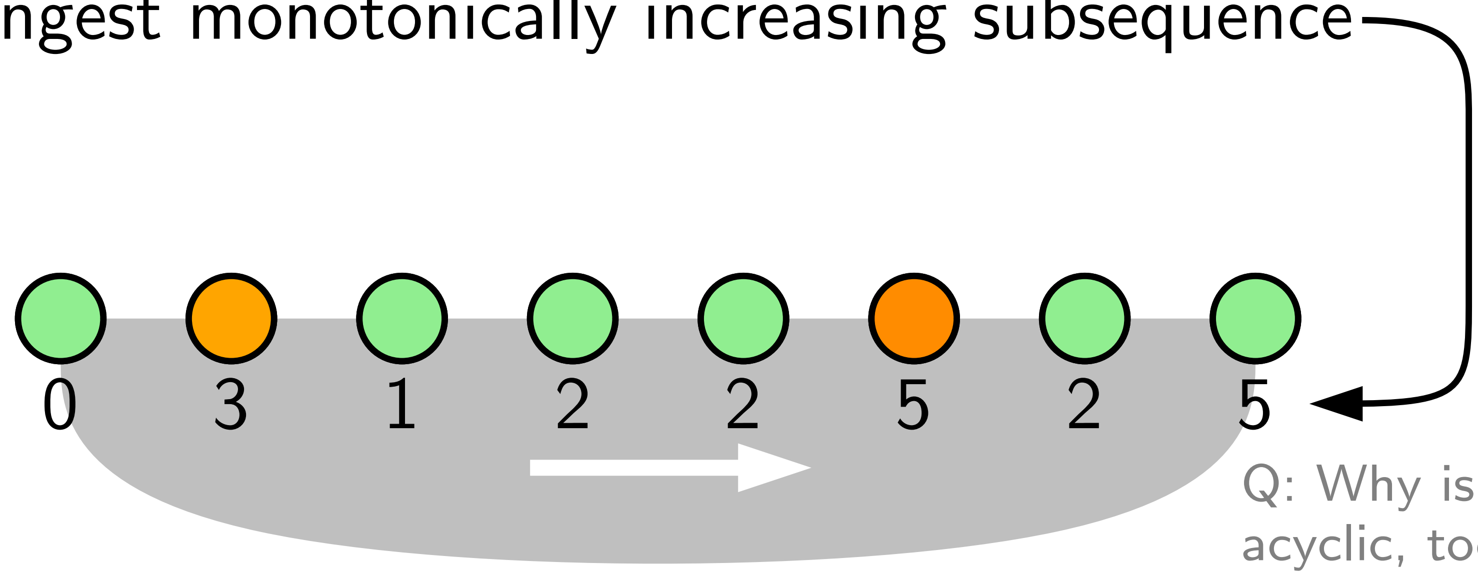
$T = (V, E)$, $\quad S = F \mathbin{\dot\cup} R$ is FVS, $\quad A = V \setminus S$

F:



Find: longest monotonically increasing subsequence

A: X:

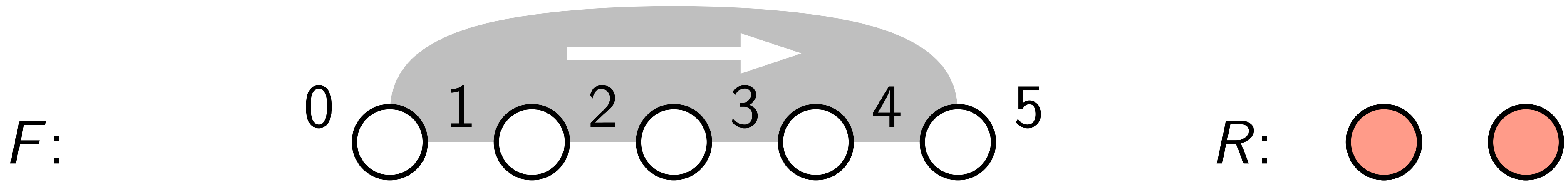


Q: Why is A acyclic, too??

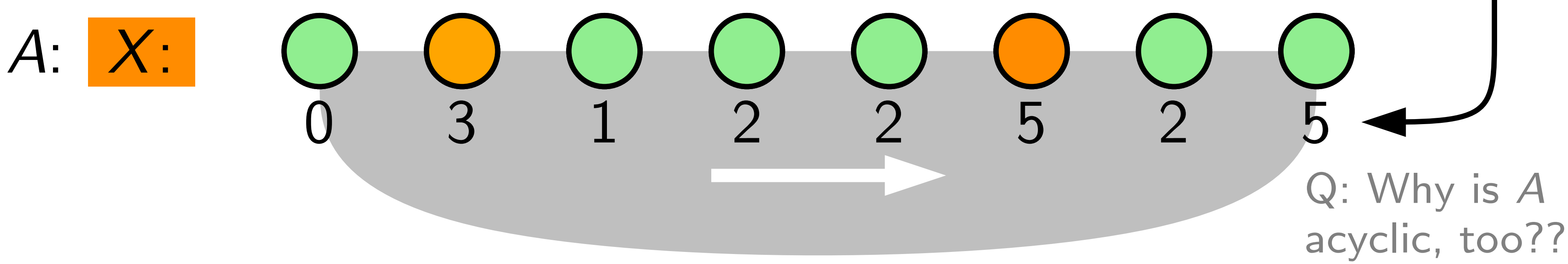
Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $\quad S = F \mathbin{\dot\cup} R$ is FVS, $\quad A = V \setminus S$



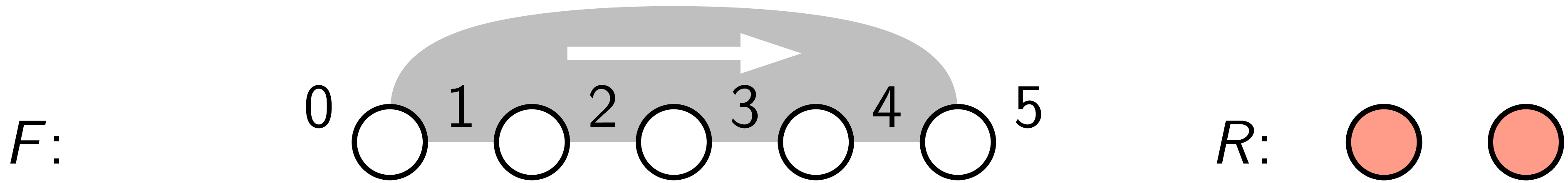
Find: longest monotonically increasing subsequence



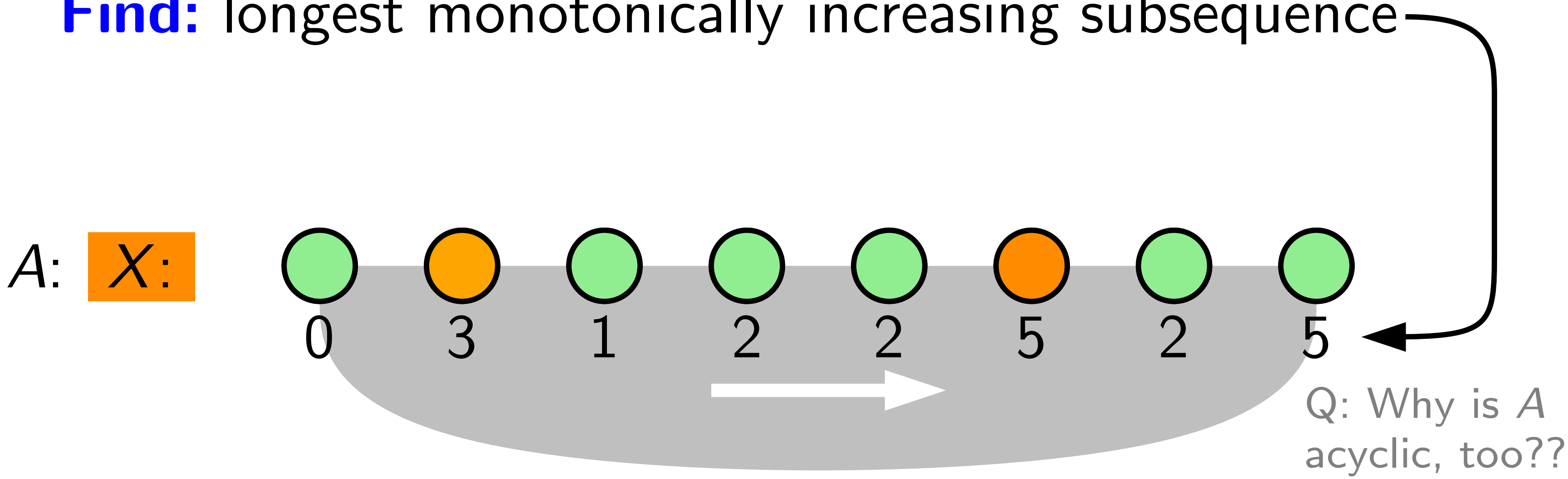
Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \cupdot R$ is FVS, $A = V \setminus S$



Find: longest monotonically increasing subsequence

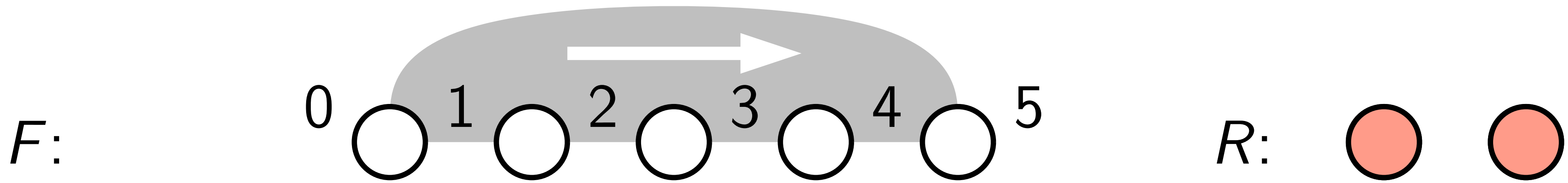


Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

$\Rightarrow X \cup R$ is a FVS of T!

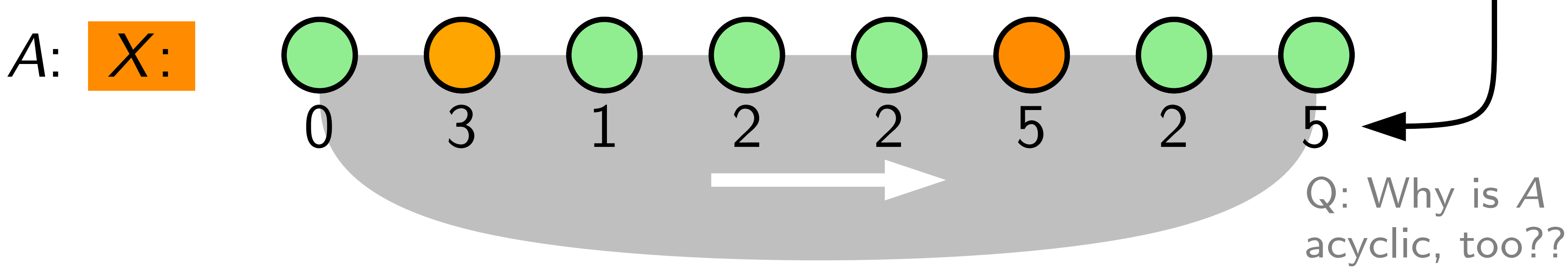
Disjoint FVS in Tournaments (cont'd)

$T = (V, E)$, $S = F \mathbin{\dot\cup} R$ is FVS, $A = V \setminus S$



Find: longest monotonically increasing subsequence

easy DP exercise for polytime



Find: minimum FVS $X \subseteq A$ of $T[F \cup A]$

$\Rightarrow X \cup R$ is a FVS of T!

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k+1$.

// Compress using Disj. FVS-T Compression:

For each $F \subseteq S$:

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k+1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k+1$.

// Compress using Disj. FVS-T Compression:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

Compute longest increasing subsequence
via labels defined before.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer]

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k+1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

Compute longest increasing subsequence

via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$.

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

Runtime?

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION:

For each $F \subseteq S$:

Remove $R_F = S \setminus F$. poly(n)

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

Runtime?

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices:

Partial graph gains a vertex, and so does S; now $|S| \leq k+1$.

// Compress using DISJ. FVS-T COMPRESSION: $O(2^k)$

For each $F \subseteq S$:

Remove $R_F = S \setminus F$. poly(n)

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

Runtime?

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices: $n - k$

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION: $O(2^k)$

For each $F \subseteq S$:

Remove $R_F = S \setminus F$. poly(n)

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

Runtime?

FVS in Tournaments by Compression

Start with any k-vertex subgraph G_k of G, and set $S = V(G_k)$.

Iteratively add vertices: $n - k$

Partial graph gains a vertex, and so does S; now $|S| \leq k + 1$.

// Compress using DISJ. FVS-T COMPRESSION: $O(2^k)$

For each $F \subseteq S$:

Remove $R_F = S \setminus F$. poly(n)

Apply reduction rules.

Compute longest increasing subsequence via labels defined before.

$\rightsquigarrow$ min. FVS X_F of $T[F \cup A]$

If some set X_F fulfills $|X_F| + |R_F| \leq k$, then $S \leftarrow X_F \cup R_F$.

Otherwise, answer NO.

Theorem. FVS in Tournaments can be solved in $O^*(2^k)$ time.

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.

see Parameterized Algorithms §9.1

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.

see Parameterized Algorithms §9.1

Kernel brute-force $\rightsquigarrow$ runtime ...

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.
see Parameterized Algorithms §9.1

Kernel brute-force $\rightsquigarrow$ runtime ... $2^{O(k^2)}\text{poly}(n)$

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.

see Parameterized Algorithms §9.1

Kernel brute-force $\rightsquigarrow$ runtime ... $2^{O(k^2)}\text{poly}(n)$

Iterative Compression $\rightsquigarrow$ $O^*(5^k)$ time

see Parameterized Algorithms §4.3

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.
see Parameterized Algorithms §9.1

Kernel brute-force $\rightsquigarrow$ runtime ... $2^{O(k^2)}\text{poly}(n)$

Iterative Compression $\rightsquigarrow$ $O^*(5^k)$ time
see Parameterized Algorithms §4.3

Current "best" algorithm: $O^*(3^k)$ randomised (Monte Carlo)
see Parameterized Algorithms §11.2.1

FVS in general graphs

Theorem. FVS has a kernel with $O(k^2)$ vertices and edges.
see Parameterized Algorithms §9.1

Kernel brute-force $\rightsquigarrow$ runtime ... $2^{O(k^2)}\text{poly}(n)$

Iterative Compression $\rightsquigarrow$ $O^*(5^k)$ time
see Parameterized Algorithms §4.3

Current "best" algorithm: $O^*(3^k)$ randomised (Monte Carlo)
see Parameterized Algorithms §11.2.1

Lemma. If FVS $\leq k$, then treewidth $\leq k + 1$.

Summary: Iterative Compression

- Show: (PROBLEM) solvable
  if (PROBLEM)-COMPRESSION solvable

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable

- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable

- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable

- Solve: Disjoint-(Problem)

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable

- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable

- Solve: Disjoint-(Problem) (This is the hard part!)

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable
- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable
- Solve: Disjoint-(Problem) (This is the hard part!)

What makes the Disjoint-(Problem) easier?

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable
- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable
- Solve: Disjoint-(Problem) (This is the hard part!)

What makes the Disjoint-(Problem) easier?

- Size-$(k+1)$ solution Y is given.

Summary: Iterative Compression

- Show: (Problem) solvable
  if (Problem)-Compression solvable

- Show: (Problem)-Compression solvable
  if Disjoint-(Problem) solvable

- Solve: Disjoint-(Problem) (This is the hard part!)

What makes the Disjoint-(Problem) easier?

- Size-$(k+1)$ solution Y is given.
- Complement of Y has special properties (here: acyclic).

Summary: Iterative Compression

- Show: $(\textsc{Problem})$ solvable
  if $(\textsc{Problem})$-$\textsc{Compression}$ solvable
- Show: $(\textsc{Problem})$-$\textsc{Compression}$ solvable
  if $\textsc{Disjoint}$-$(\textsc{Problem})$ solvable
- Solve: $\textsc{Disjoint}$-$(\textsc{Problem})$ (This is the hard part!)

What makes the Disjoint-(Problem) easier?

- Size-$(k+1)$ solution Y is given.
- Complement of Y has special properties (here: acyclic).
- Splitting $Y = F \uplus R$ implies special properties of F (here: acyclic).