Parameterized Algorithms

- Iterative Compression (2004)
- Dynamic Programming over Subsets

Roohani Sharma
Lecture #3
November 02, 2021

Iterative Compression

(Illustrate with examples)

FEEDBACK VERTEX SET (FVS)

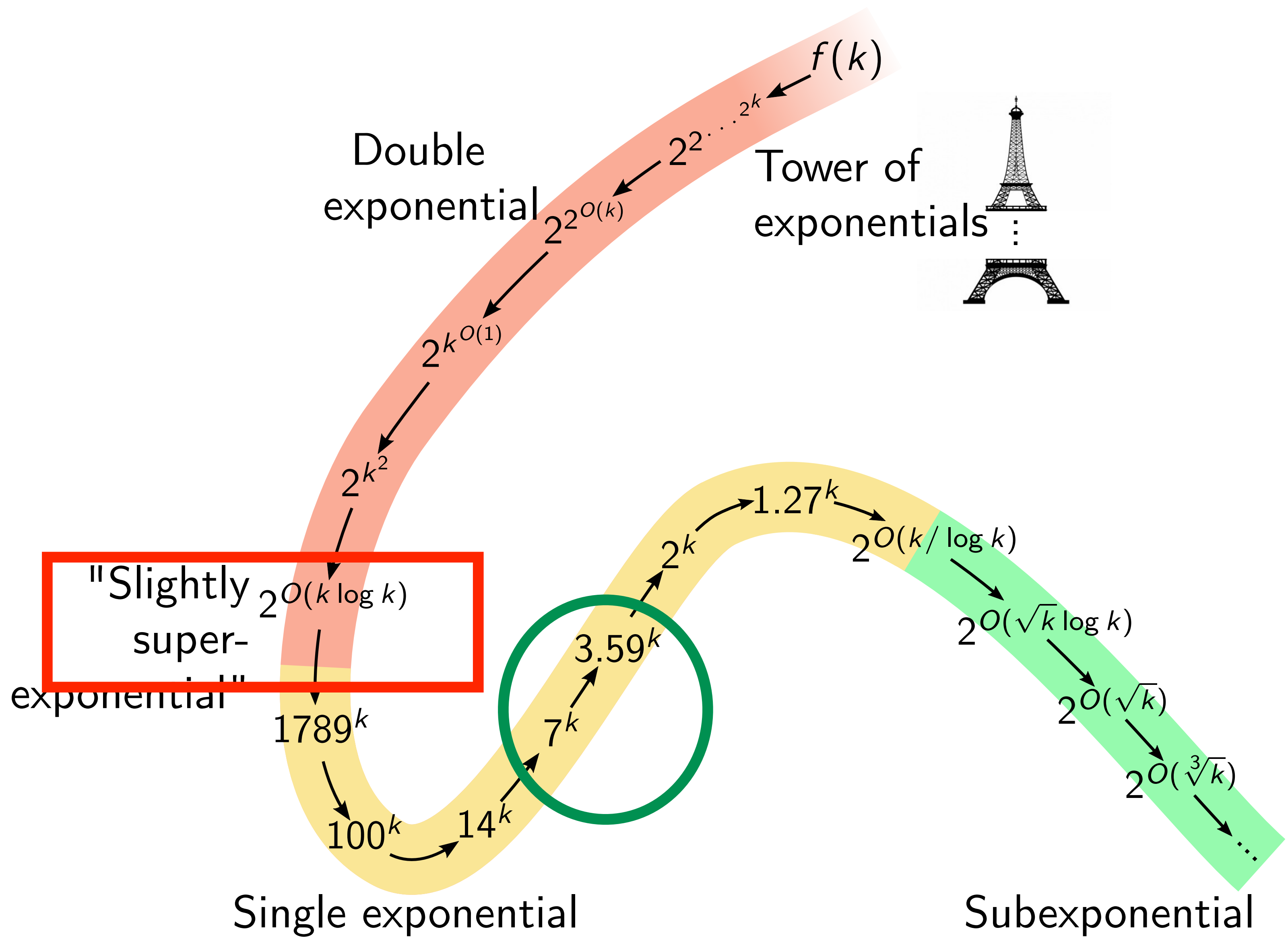
Input: A graph G , a positive integer k

Question: Does there exists a set of at most k vertices, say S, such that G-S is acyclic (forest)?

Lecture #1: $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$

(Branching: If minimum degeee is at least 3, then set of 3k largest-degree vertices contain a vertex of the solution.)

The race for better FPT algorithms



Goal: $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$

$5^k n^{\mathcal{O}(1)}$

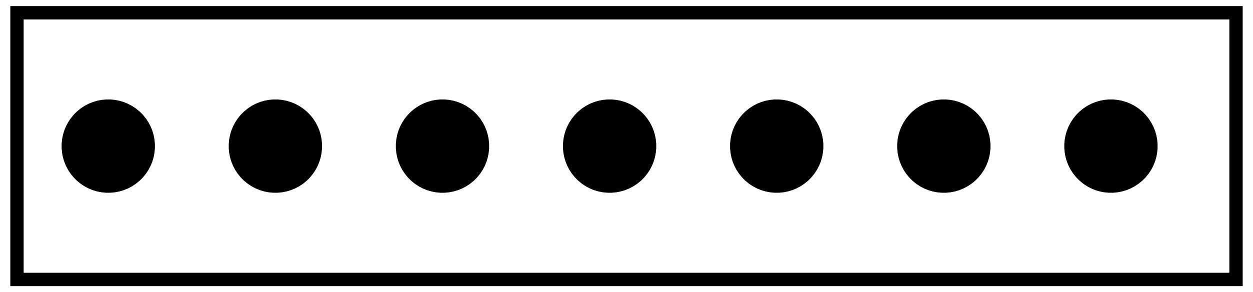
Compression Feedback Vertex Set

Input: A graph G , a positive integer k

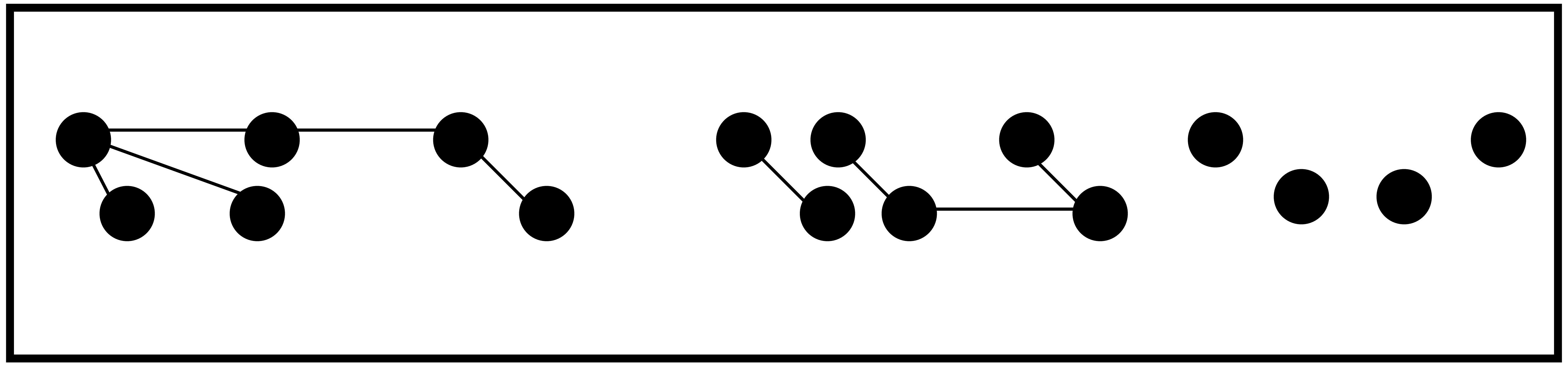
+ a solution X of size k+1 (a slightly large solution)

Question: Does there exists a set of at most k vertices, say S, such that G-S is acyclic (forest)?

Idea: X gives additional structure on the input graph.



X $|X|=k+1$



G - X

Forest

Algorithm:

Step 1: Guess the intersection of the solution S with X.

Say $S \cap X = X_S$, and $X \setminus S = X^*$.

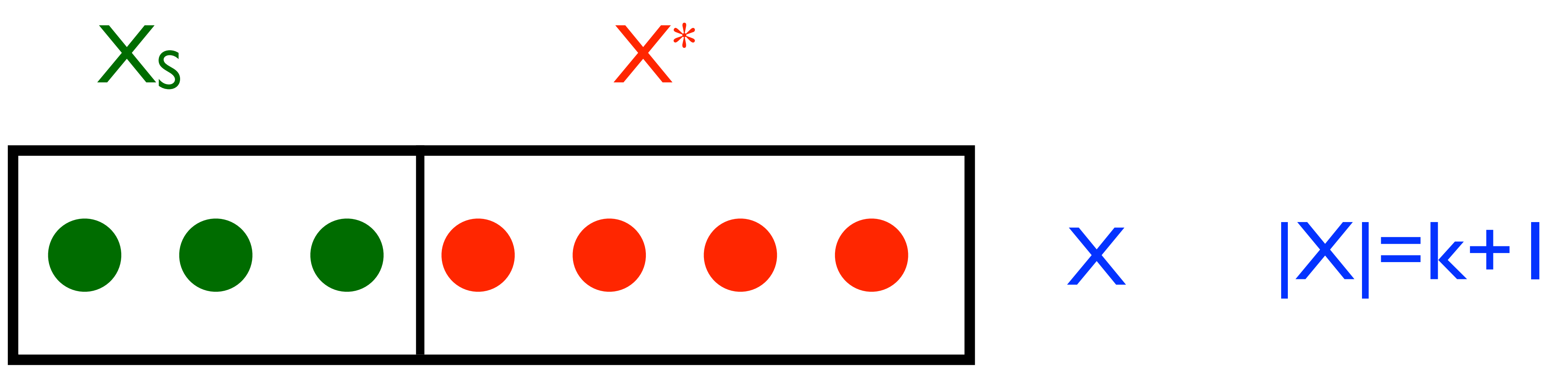
COMPRESSION FEEDBACK VERTEX SET

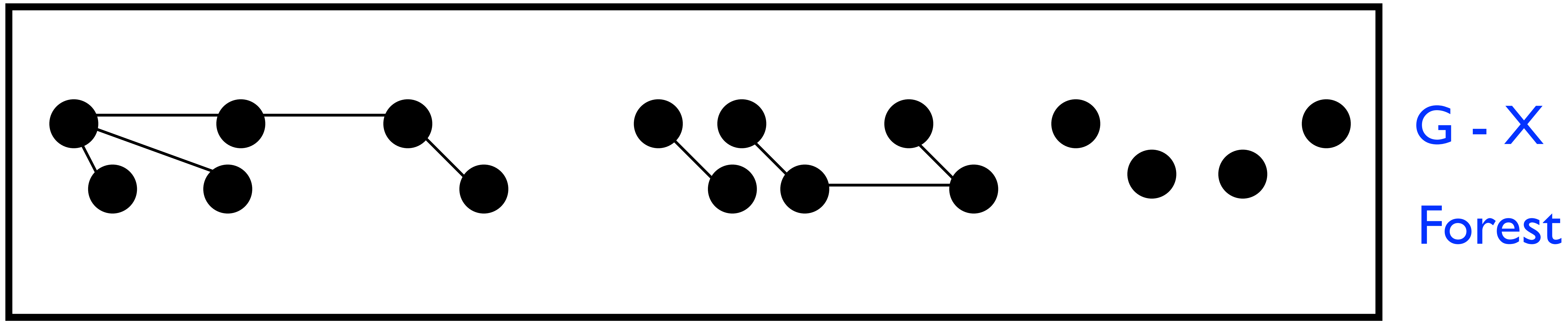
Input: A graph G , a positive integer k

+ a solution X of size k+1 (a slightly large solution)

Question: Does there exists a set of at most k vertices, say S, such that G-S is acyclic (forest)?

Idea: X gives additional structure on the input graph.





Algorithm:

Step 1: Guess the intersection of the solution S with X.

Say $S \cap X = X_S$, and $X \setminus S = X^*$.

Number of guesses/branches:

$$2^{|X|} \quad \text{or} \quad \sum_{i=0}^{k} \binom{|X|}{i} = \sum_{i=0}^{k} \binom{k+1}{i}$$

Step 2: Delete X_S from the graph.

Look for a solution of size $k-|X_S|$ that is disjoint from X^*.

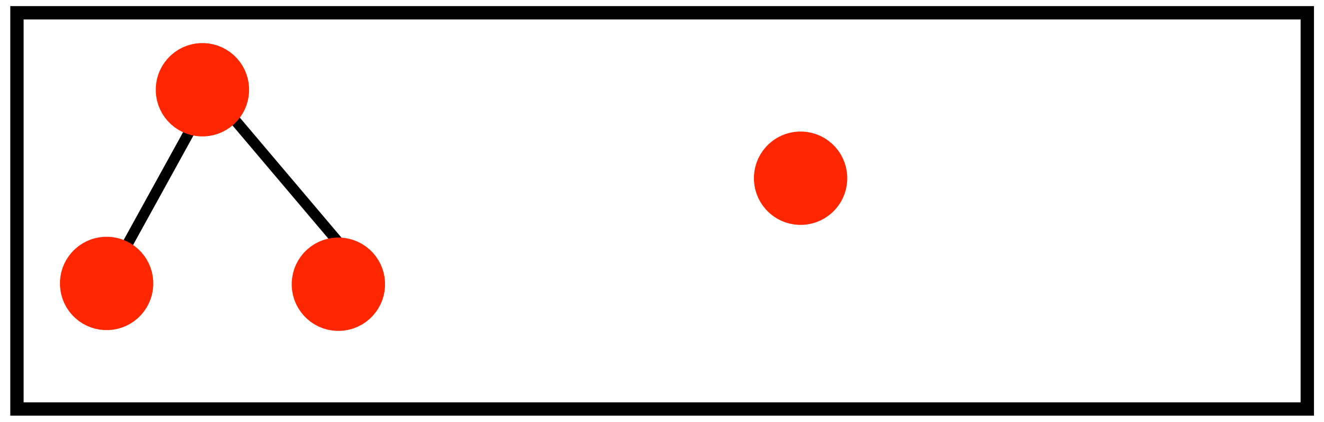
DISJOINT FEEDBACK VERTEX SET

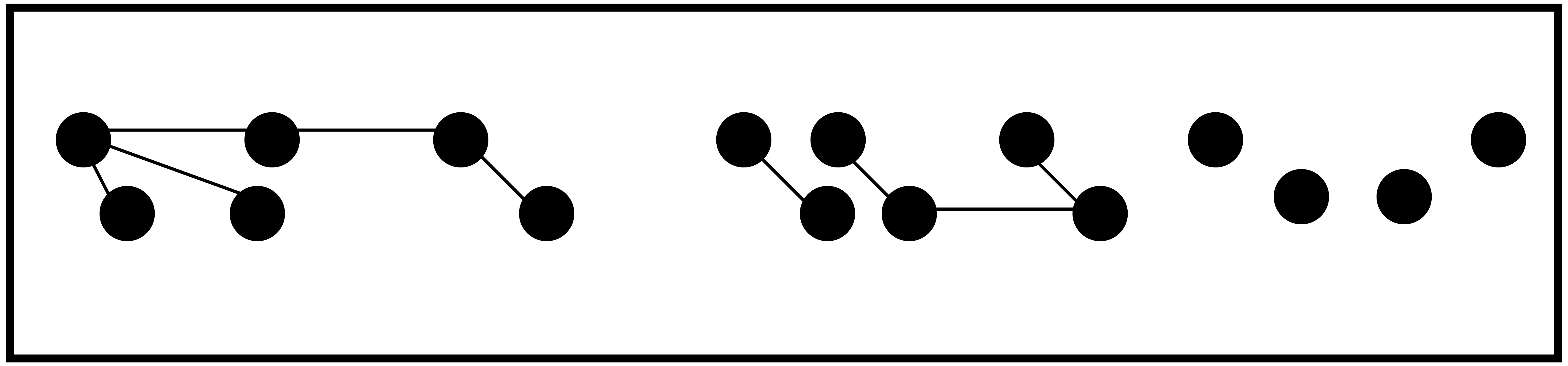
Input: A graph G, positive integers i and k such that k<i, a solution X^* of size i

Question: Does there exists a set S such that :

- G-S is acyclic?
- S is disjoint from X^*
- $|S| \leq k$

X^* Forest



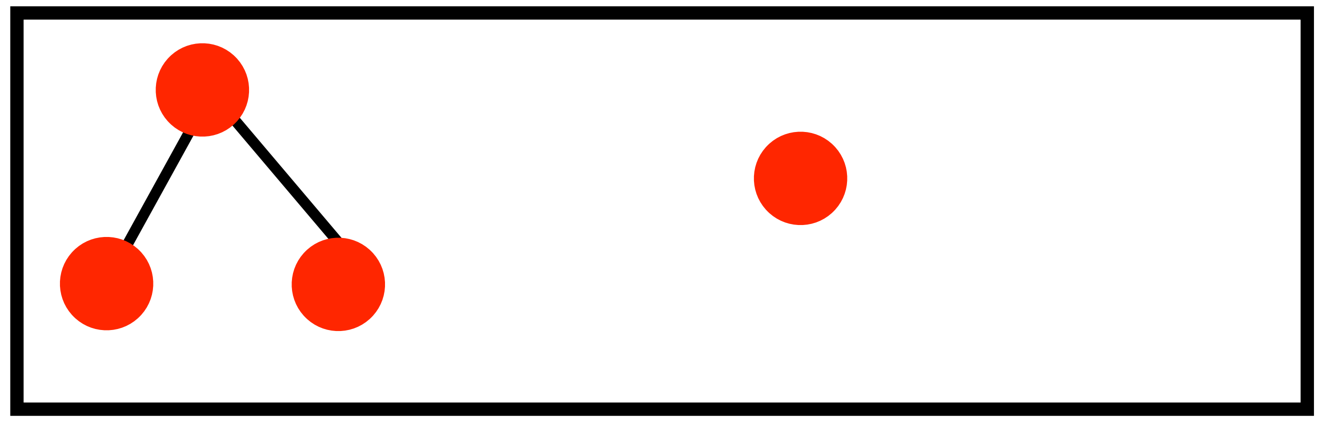


G - X^*

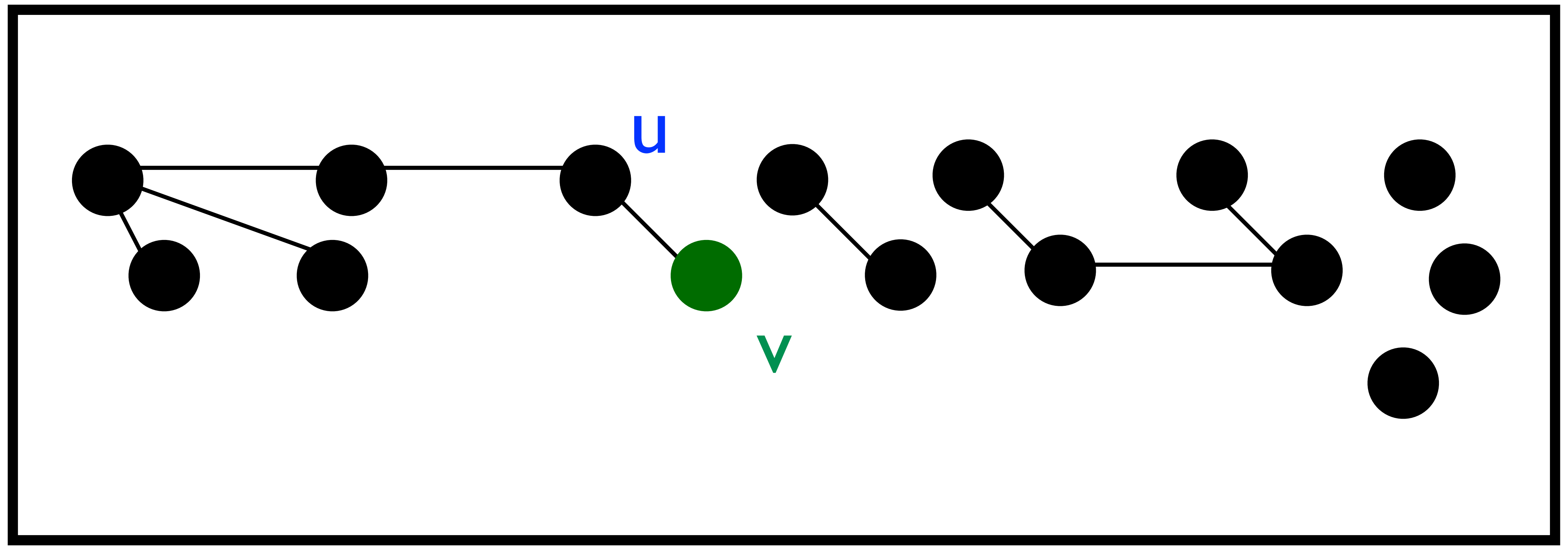
Forest

Observation: Graph induced on X^* is a forest. Otherwise, say No.

Solving Disjoint FVS



X^* Forest



$G - X^*$ Forest

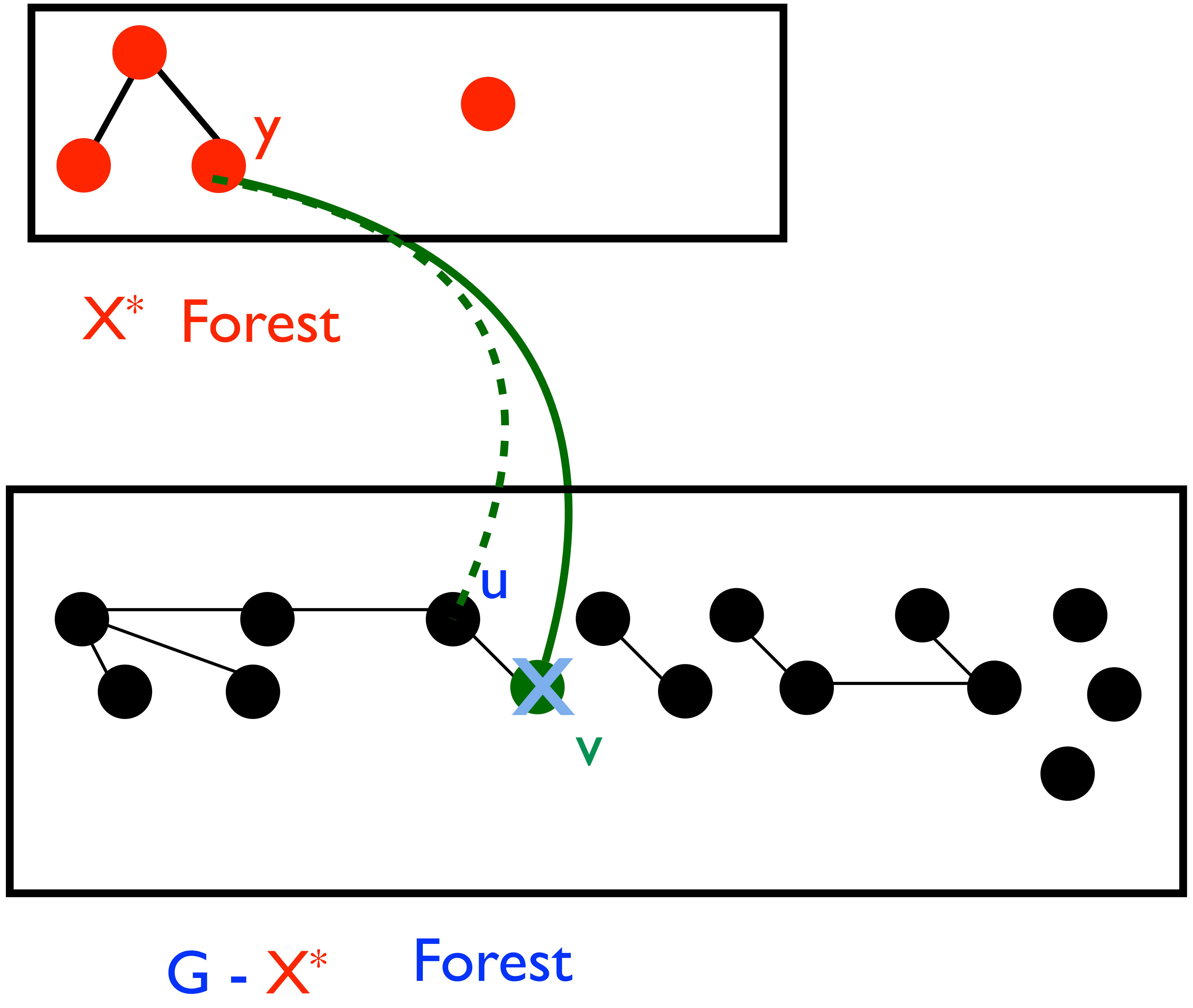
Algorithm:

Fix a leaf v of G - X^*. Let u be the unique parent of v in G - X^*.

1. If v has no neighbours in X^*, then delete v (because v does not participate in any cycle).

2.

Solving Disjoint FVS

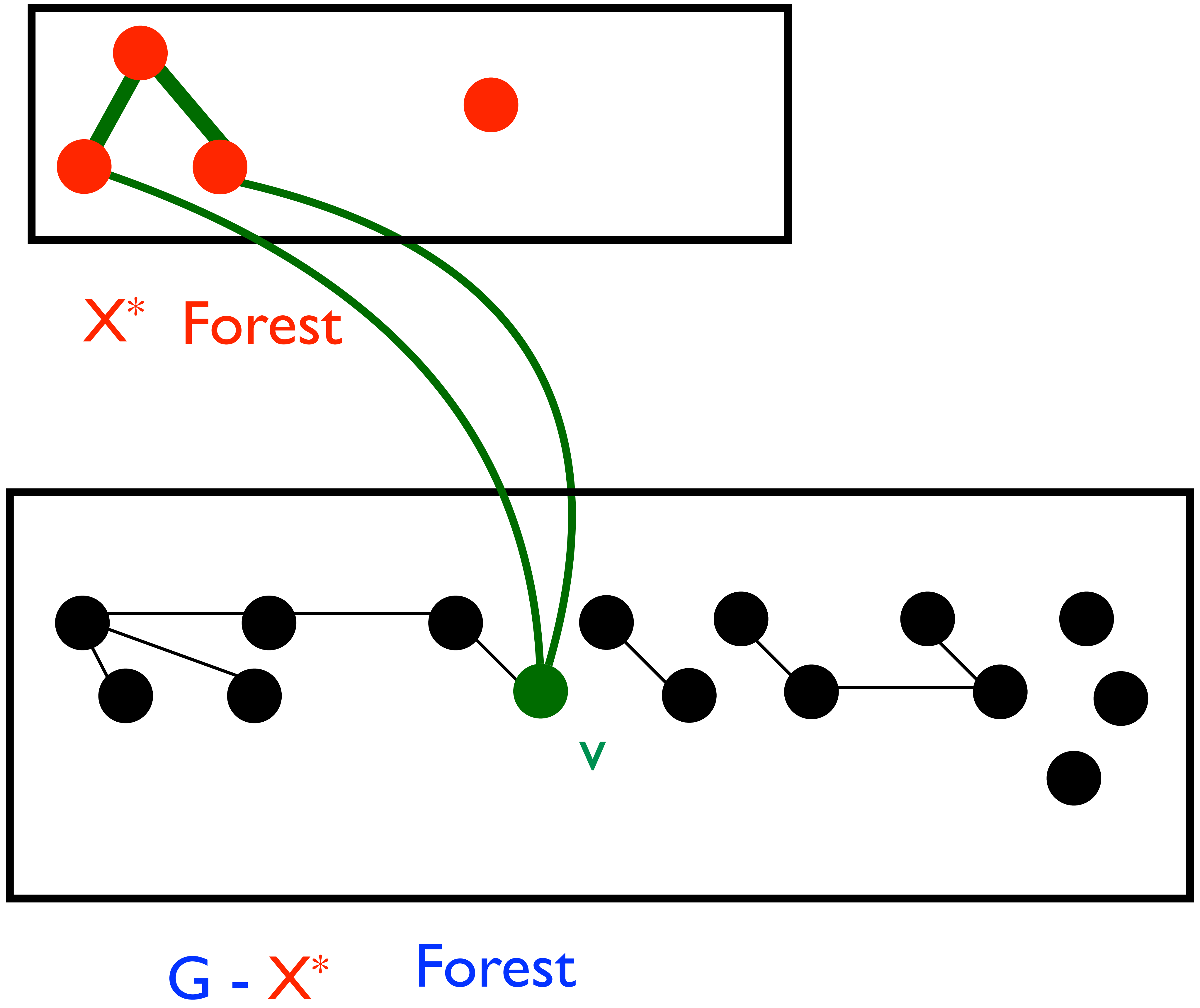


Algorithm:

Fix a leaf v of G - X*. Let u be the unique parent of v in G - X*.

1. If v has no neighbours in X*, then delete v (because v does not participate in any cycle).
2. If v has exactly one neighbour (say y) in X*, then delete v and and an edge between y and u (because any cycle that passes through v, also passes through u).
3.

Solving Disjoint FVS

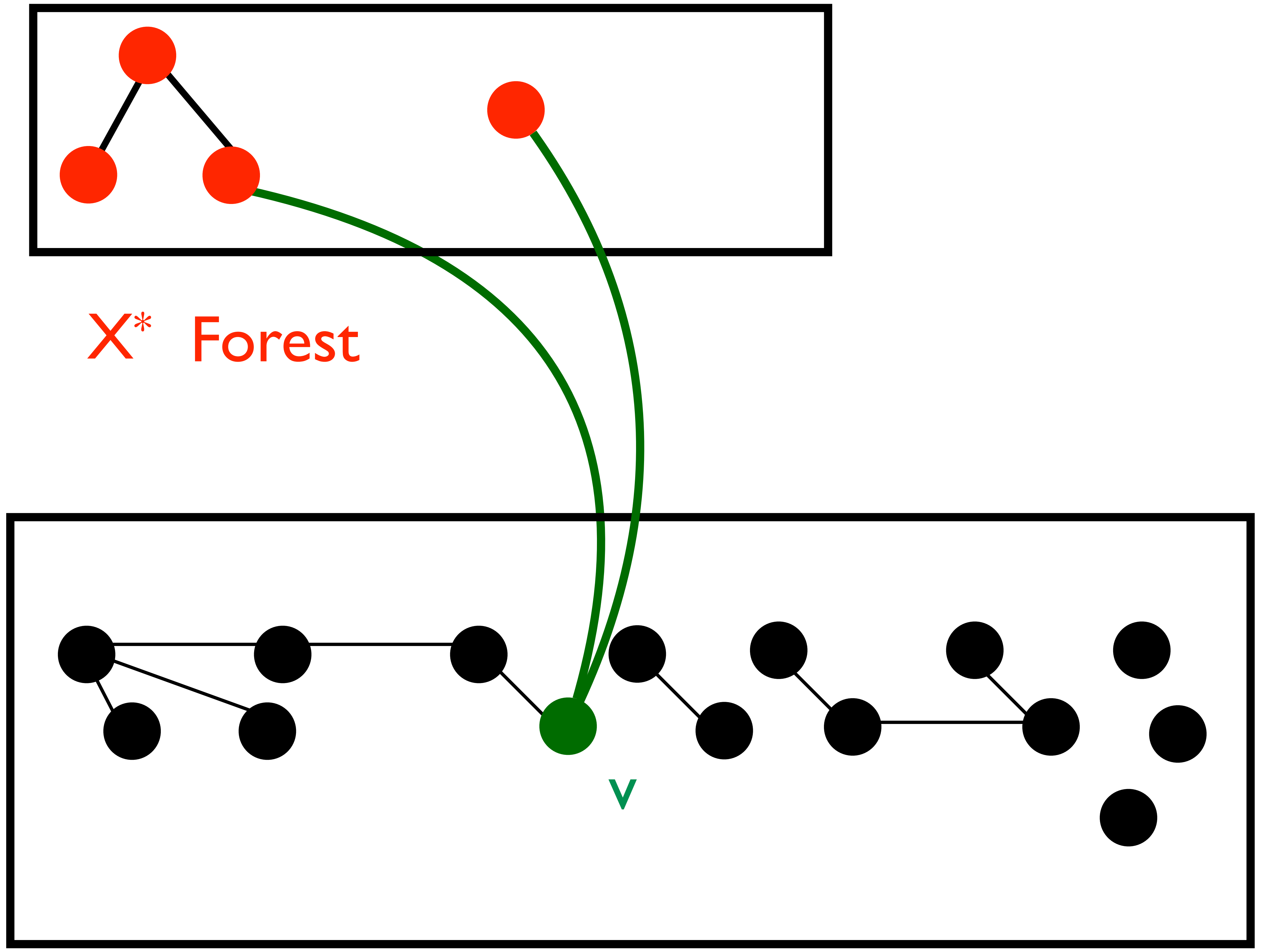


Algorithm:

Fix a leaf v of G - X^*. Let u be the unique parent of v in G - X^*.

1. If v has no neighbours in X^*, then delete v (because v does not participate in any cycle).
2. If v has exactly one neighbour (say y) in X^*, then delete v and and an edge between y and u (because any cycle that passes through v, also passes through u).
3. If v has at least two neighbours in the same tree of X^*, the pick v in the solution, delete it and decrease k by 1 (because there is a cycle all of whose vertices, except v, are in X^*).
4.

Solving Disjoint FVS



X* Forest

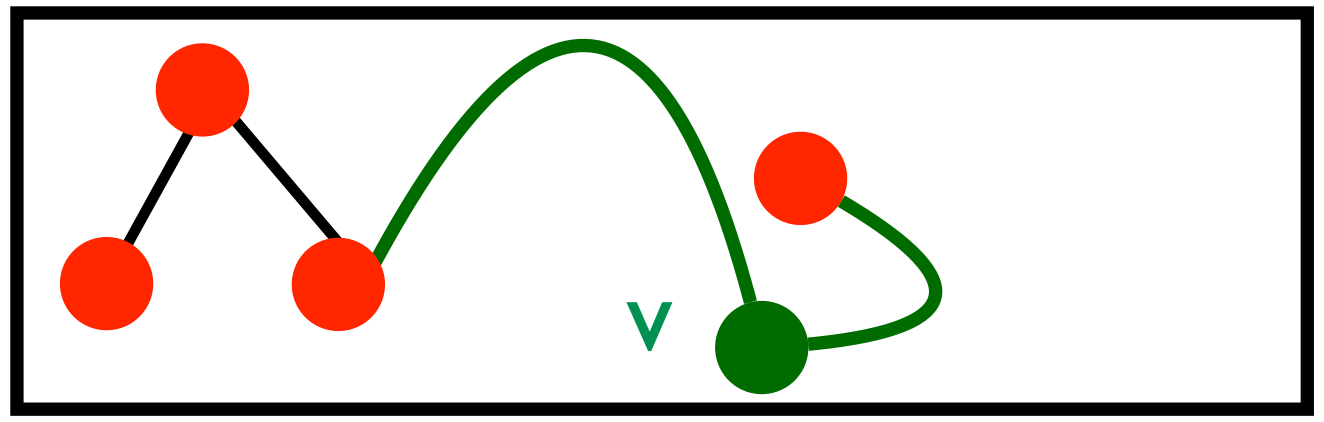
G - X* Forest

Algorithm:

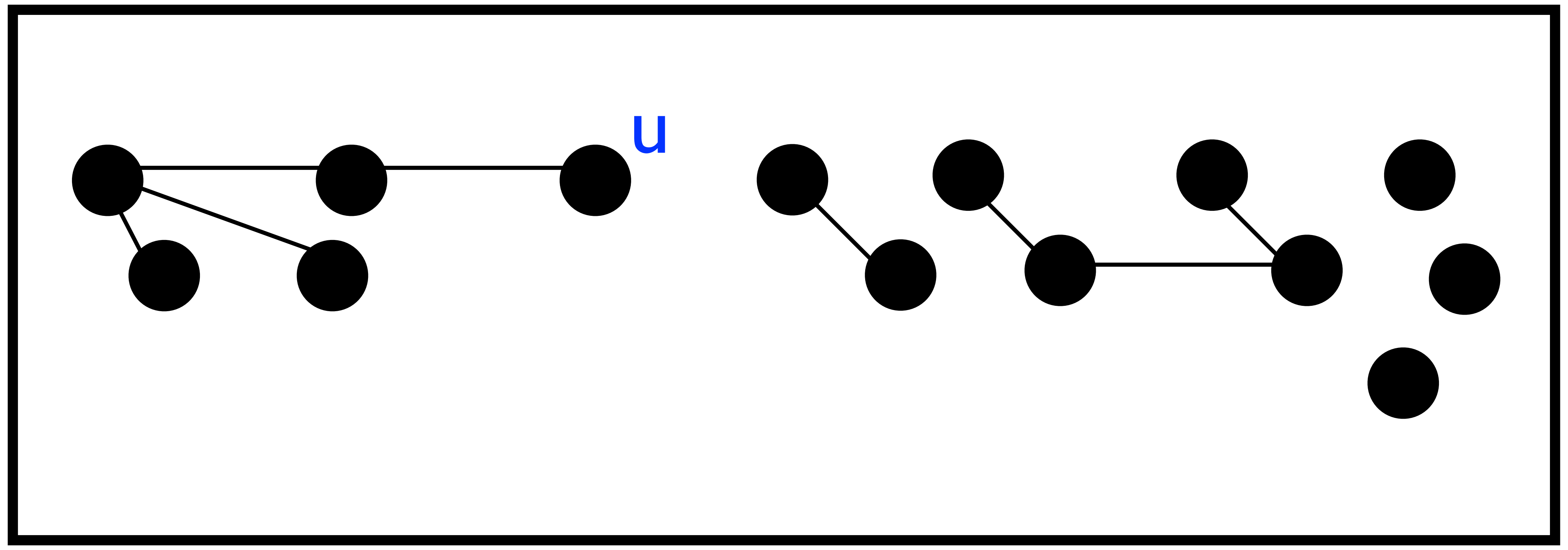
Fix a leaf v of G - X*. Let u be the unique parent of v in G - X*.

1. If v has no neighbours in X*, then delete v (because v does not participate in any cycle).
2. If v has exactly one neighbour (say y) in X*, then delete v and and an edge between y and u (because any cycle that passes through v, also passes through u).
3. If v has at least two neighbours in the same tree of X*, the pick v in the solution, delete it and decrease k by 1 (because there is a cycle all of whose vertices, except v, are in X*).
4. Otherwise, v has a neighbours in at least two different trees of X*. In this case, branch in the following two branches:
   a. Either v belongs to the solution. In this case, pick v in the solution, delete it, and decrease k by 1.
   b. Or v does not belong to the solution. In this case, update X* = X* ∪ v. In this case, the number of trees of X* decrease by 1 (because v has neighbours in at least two trees of X*).

Solving Disjoint FVS



X^* Forest



G - X^* Forest

Algorithm:

Fix a leaf v of G - X^*. Let u be the unique parent of v in G - X^*.

1. If v has no neighbours in X^*, then delete v (because v does not participate in any cycle).
2. If v has exactly one neighbour (say y) in X^*, then delete v and and an edge between y and u (because any cycle that passes through v, also passes through u).
3. If v has at least two neighbours in the same tree of X^*, the pick v in the solution, delete it and decrease k by 1 (because there is a cycle all of whose vertices, except v, are in X^*).
4. Otherwise, v has a neighbours in at least two different trees of X^*. In this case, branch in the following two branches:
    a. Either v belongs to the solution. In this case, pick v in the solution, delete it and decrease k by 1.
    b. Or v does not belong to the solution. In this case, update X^* = $X^* \cup v$. In this case, the number of trees of X^* decrease by 1 (because v has neighbours in at least two trees of X^*).

Solving Disjoint FVS

Fix a leaf v of G - X^*. Let u be the unique parent of v in G - X^*.

1. If v has no neighbours in X^*, then delete v (because v does not participate in any cycle).
2. If v has exactly one neighbour (say y) in X^*, then delete v and and an edge between y and u (because any cycle that passes through v, also passes through u).
3. If v has at least two neighbours in the same tree of X^*, the pick v in the solution, delete it and decrease k by 1 (because there is a cycle all of whose vertices, except v, are in X^*).
4. Otherwise, v has a neighbours in at least two different trees of X^*. In this case, branch in the following two branches:
   a. Either v belongs to the solution. In this case, pick v in the solution, delete it and decrease k by 1.
   b. Or v does not belong to the solution. In this case, update $X^* = X^* \cup v$. In this case, the number of trees of X^* decrease by 1 (because v has neighbours in at least two trees of X^*).

Running time:

Cases 1-3 are applicable in polynomial time.

Cases 4 is a branching step.

We are solving the following problem recursively.

Input: A graph G, an acyclic solution X^* such that the graph induced on X^* has t trees, and an integer k.

Question: Does there exists a solution of size at most k that is disjoint from X^*?

Case 4a: k drops by 1

Case 4b: t drops by 1

- Size of the branching tree is 2^{k+t}. Since we start with X^* of size i, t ≤ i. Also k ≤ i.
- Spend polynomial time on each node of the branching tree (check if Cases 1-3 are applicable).

Overall running time: $2^{k+t} n^{\mathcal{O}(1)} \leq 4^i n^{\mathcal{O}(1)}$

Disjoint Feedback Vertex Set

Input: A graph G, positive integers i and k such that k<i, a solution X^* of size i

Question: Does there exists a set S such that :

- G-S is acyclic?
- S is disjoint from X^*
- |S| ≤ k

Disjoint Feedback Vertex Set can be solved in $4^i n^{\mathcal{O}(1)}$ *time.*

Compression Feedback Vertex Set

Input: A graph G , a positive integer k

+ a solution X of size k+1

Question: Does there exists a set of at most k vertices, say S, such that G-S is acyclic (forest)?

Running time:

$$\sum_{i=0}^{k} \binom{k+1}{i} 4^i n^{\mathcal{O}(1)}$$

$$\leq (1+4)^k n^{\mathcal{O}(1)}$$ (Binomial theorem)

$$= 5^k n^{\mathcal{O}(1)}$$

Compression Feedback Vertex Set can be solved in $5^k n^{\mathcal{O}(1)}$ *time.*

Let Π be a vertex deletion to a hereditary property.

Definition

A graph property $\mathcal{P}$ is **hereditary** or **closed under induced subgraphs** if whenever $G \in P$, every induced subgraph of G is also in $\mathcal{P}$.

"removing a vertex does not ruin the property"
(e.g., triangle free, bipartite, planar)

If **DISJOINT-Π** can be solved in $g(i) \cdot n^{\mathcal{O}(1)}$

then **COMPRESSION-Π** can be solved in $\sum_{i=0}^{k} \binom{k+1}{i} g(i) \cdot n^{\mathcal{O}(1)}$

If **DISJOINT-Π** is FPT, then so is **COMPRESSION-Π**.

In fact, if **DISJOINT-Π** is solvable in $\alpha^i n^{O(1)}$ time, then **COMPRESSION-Π** is solvable in $(1+\alpha)^k n^{O(1)}$ time.

Solving FVS using COMPRESSION FVS: Iterative step

How to get a solution X of size k+1?

- Order the vertices of the graph arbitrarily, say $v_1, \ldots, v_n$.
- Let G_i be the graph induced on the first i vertices, $v_1, \ldots, v_i$.
- G_{k+2} has a trivial solution of size k+1 (take any k+1 vertices in the solution).
- Use **COMPRESSION FVS** to find a solution, say S_{k+2}, of size at most k for G_{k+2}.
  - If no such S_{k+2} exists, then report No, that is, there is no solution of size at most k for G (this is true acyclicity is a hereditary property).
  - Otherwise, $(S_{k+2} \cup v_{k+3})$ is an at most k+1 size solution for G_{k+3}.
- Repeat.

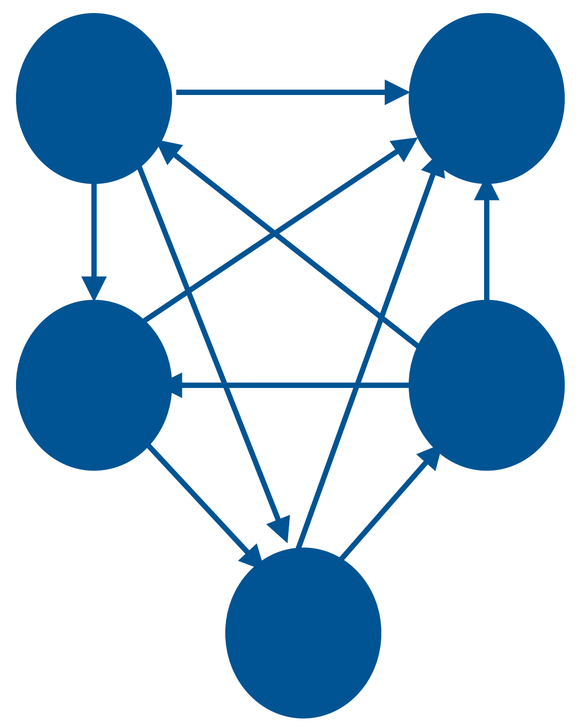
FEEDBACK VERTEX SET can be solved in $5^k n^{\mathcal{O}(1)}$ *time.*

If **Π** is a vertex deletion problem to a hereditary property, then if Compression-**Π** can be solved in $f(k)n^c$ time, then **Π** can be solved in $f(k)n^{c+1}$ time.

Another example: **FEEDBACK VERTEX SET IN TOURNAMENTS (FVST)**

- Directed Feedback Vertex Set is FPT (we will see later in the course). Also uses iterative compression and more advanced tools.
- For today, we focus our attention to the case when the input is a tournament.
- A tournament is a directed graph where there is exactly one arc between any pair of vertices.

Tournament



FVST

Input: A tournament D, a positive integer k

Question: Does there exists a set of at most k vertices, say S, such that D-S has no directed cycles?

NP-hard

A tournament D has a directed cycle if and only if D has a directed triangle (cycle on 3 vertices). (Exercise)
Therefore, FVST can be solved in $3^k n^{O(1)}$ using branching.

Goal: $2^k \cdot n^{\mathcal{O}(1)}$

DISJOINT FVST

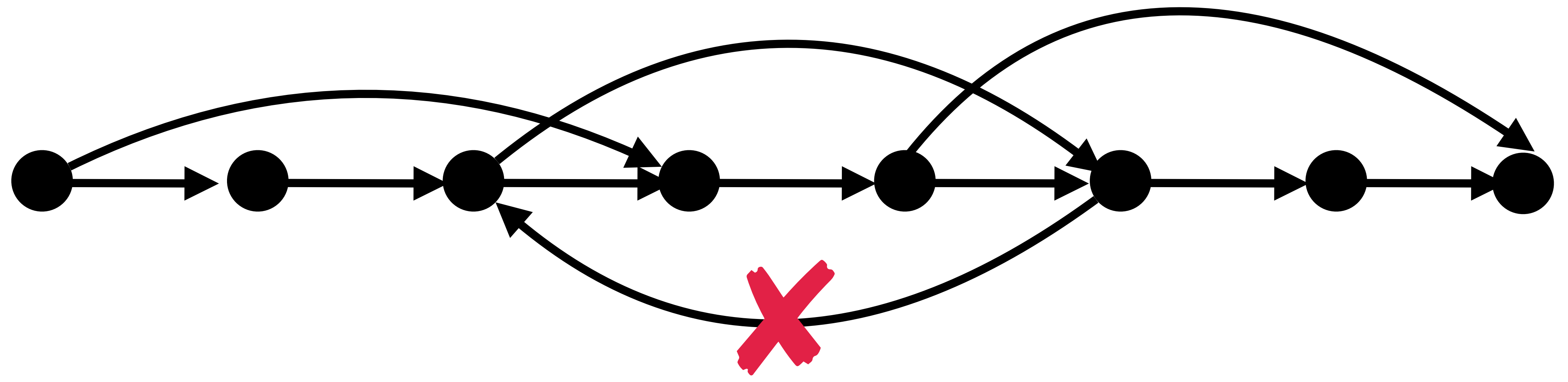
Input: A tournament D, positive integers i and k such that k<i, a solution X^* of size i

Question: Does there exists a set S such that :

- S is disjoint from X^*
- $|S| \leq k$
- G-S has no directed cycles?

Enough to show that **DISJOINT FVST** is solvable in: polynomial time

A topological ordering of an acyclic directed graph D is an ordering of the vertices of D, say σ, say that, for any u,v, such that $\sigma(u) > \sigma(v)$, (u,v) is not an arc of D (no backward arcs).



Exercise

If D is a tournament then the following are equivalent:

D is acyclic,

D has no directed triangles,

D has a unique topological ordering.

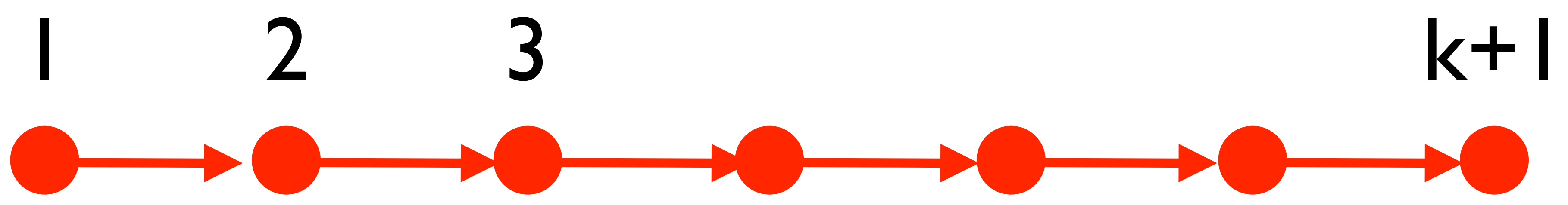
Solving Disjoint FVST

Observation: Graph induced on X^* is acyclic. Otherwise, say No.

Let ρ be the unique topological ordering of X^*. Let σ be the unique topological ordering of $A=V(D)-X^*$.

X^*

Vertices of X^* ordered according to ρ

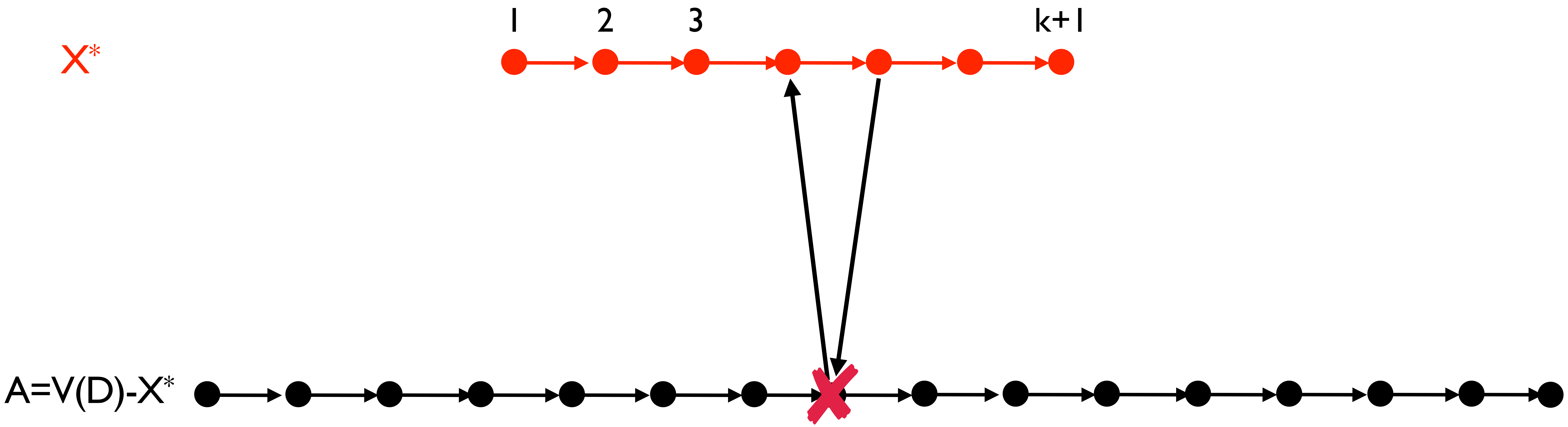


The vertices of X^* appear in the same ordering in the final topological ordering.

$A=V(D)-X^*$

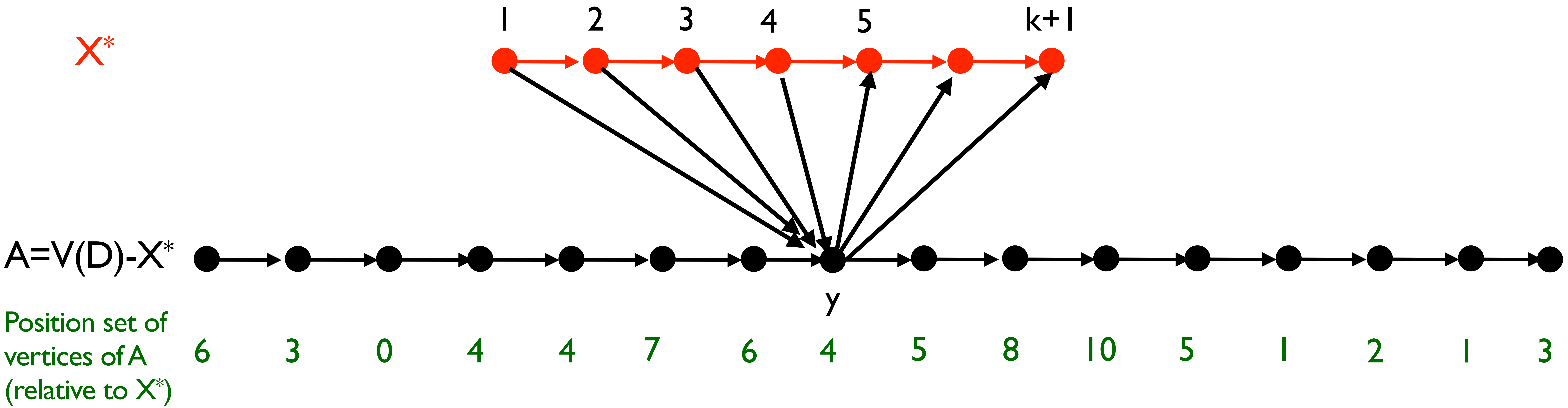
Vertices of A ordered according to σ

Solving Disjoint FVST



Reduction rule: If there exists a directed triangle two of whose vertices belong to X*, then pick its intersection with A in the solution.

Solving Disjoint FVST

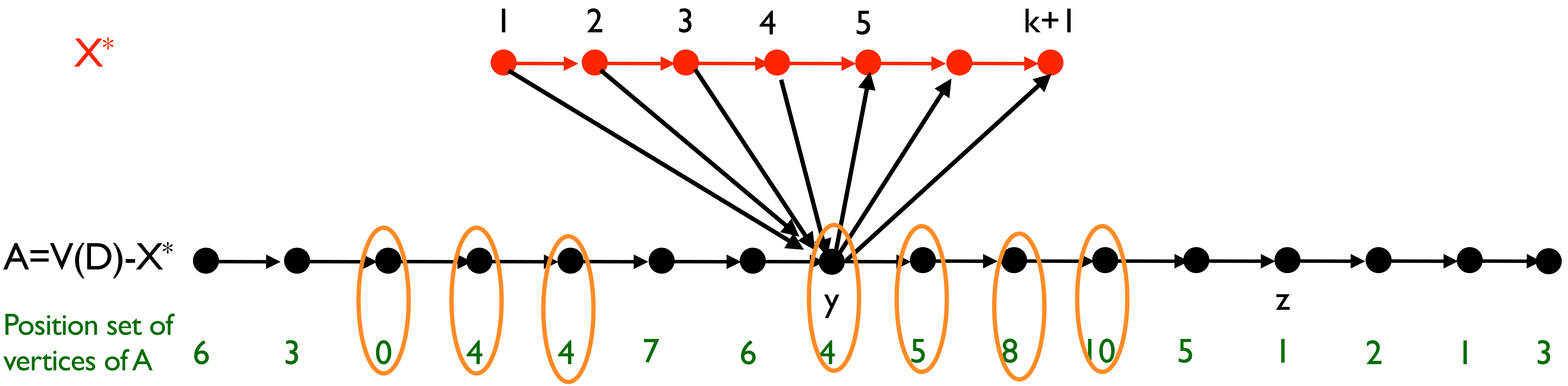


For each $y \in A$, posn(y)= largest $i \in X^*$, such that i is an in-neighbour of y.

If y has no in-neighbour in X^* then posn(y) =0.

(posn(y) tells the relative position of y w.r.t. to the vertices of X^* in the final ordering).

Solving Disjoint FVST



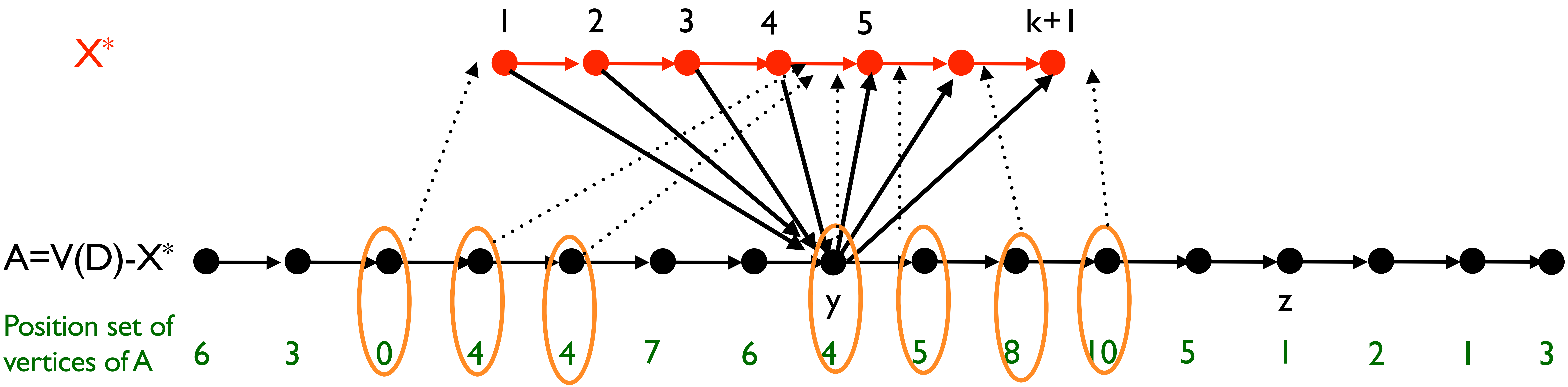
Goal: Find a maximum sized subset A, say $W \subseteq A$, such that $D[X^* \cup W]$ is acyclic.

Consider the vertices of A that are not in the solution.

Can y and z both not be in solution when y appears before z in the ordering of A? No (posn(y) > posn(z)).

Goal: Find a longest non-decreasing subsequence in the sequence of position set of A.

Solving Disjoint FVST



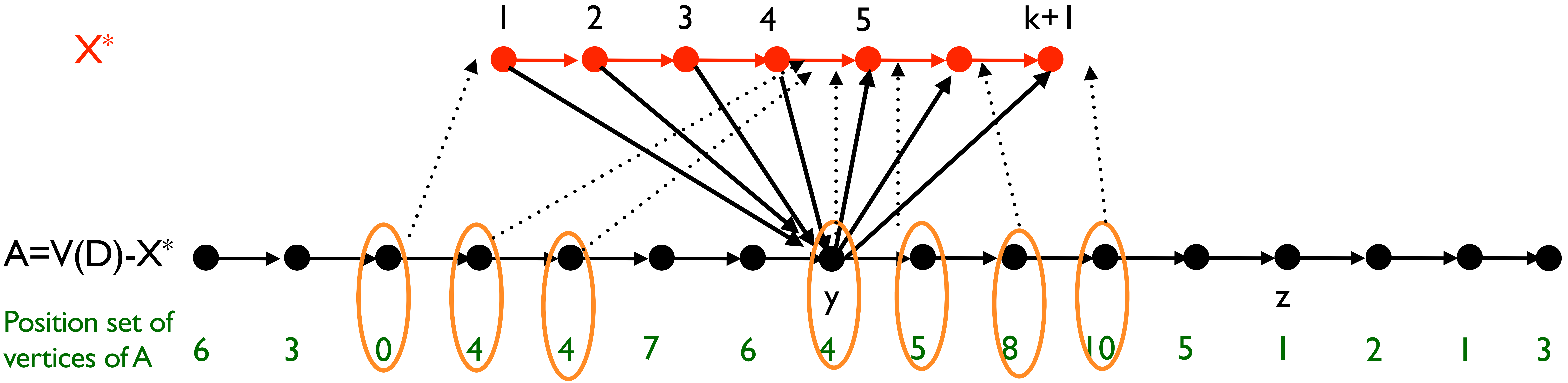
Goal: Find a maximum sized subset A, say $W \subseteq A$, such that $D[X^* \cup W]$ is acyclic.

Consider the vertices of A that are not in the solution.

Can y and z both not be in solution when y appears before z in the ordering of A? No (posn(y) > posn(z)).

Goal: Find a longest non-decreasing subsequence in the sequence of position set of A.

Solving Disjoint FVST



Goal: Find a maximum sized subset A, say $W \subseteq A$, such that $D[X^* \cup W]$ is acyclic.

Consider the vertices of A that are not in the solution.

Can y and z both not be in solution when y appears before z in the ordering of A? No (posn(y) > posn(z)).

Goal: Find a longest non-decreasing subsequence in the sequence of position set of A.

This can be done in polynomial time using standard dynamic programming. (Exercise)

Another example: ODD CYCLE TRANSVERSAL (OCT)

OCT

Input: A graph G , a positive integer k

Question: Does there exists a set of at most k vertices, say S, such that G-S has no odd length cycle (bipartite)?

Goal:

$3^k \cdot n^{\mathcal{O}(1)}$

$\mathcal{O}(3^k \cdot kn(n+m))$

DISJOINT OCT

Input: A graph G, positive integers i and k such that k<i, a solution X^* of size i

Question: Does there exists a set S such that :

- S is disjoint from X^*
- $|S| \leq k$
- G-S is bipartite?

Enough to show that **DISJOINT OCT** is solvable in:

$2^i \cdot n^{\mathcal{O}(1)}$

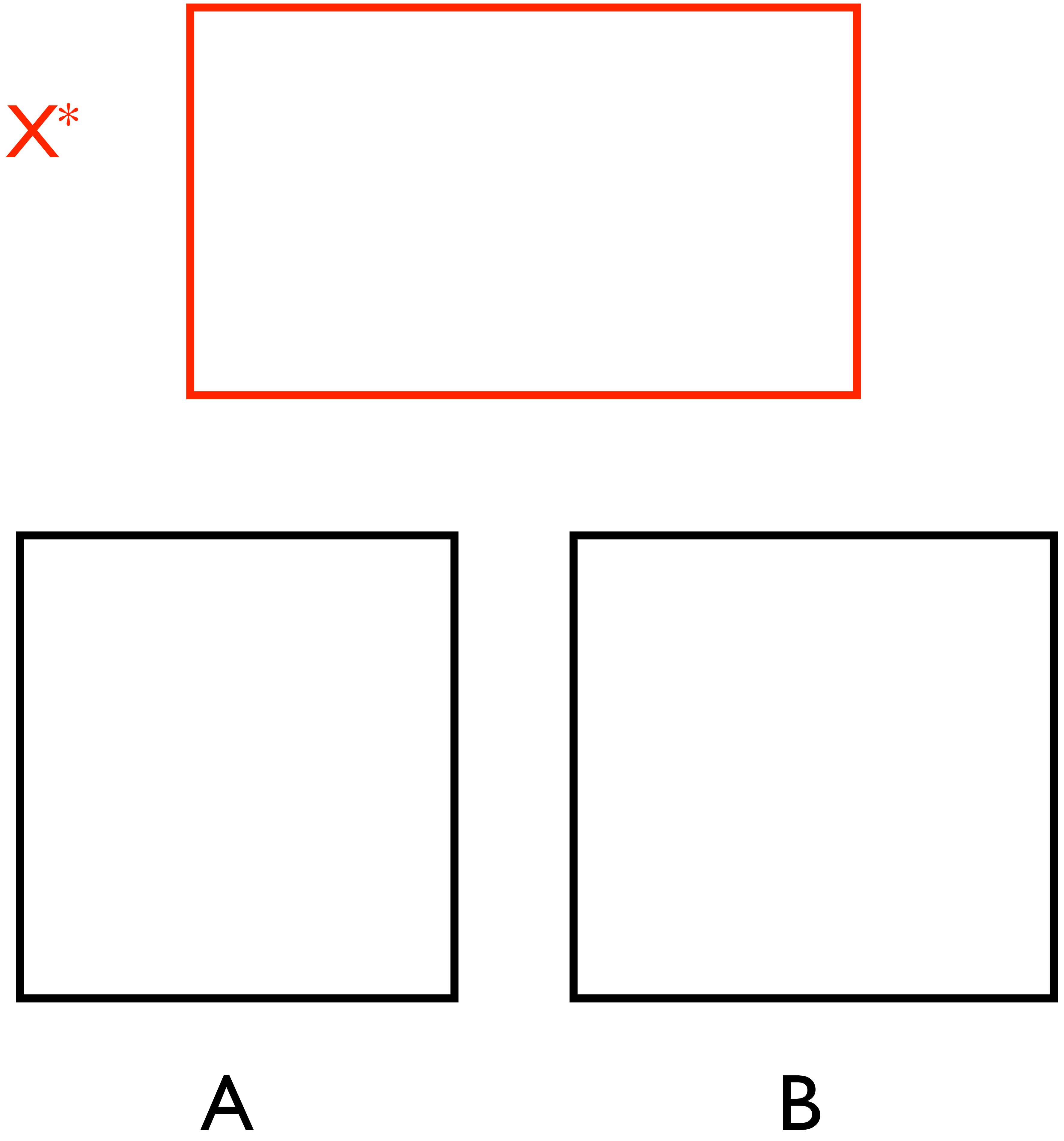
$\mathcal{O}(2^i \cdot k(n+m))$

Solving Disjoint OCT

Observation: Graph induced on X^* is bipartite. Otherwise, say No.

Since G-X^* is bipartite, let (A,B) be a bipartition of G-X^*.

Let (L,R) be a solution bipartition that is a bipartition of G-S.

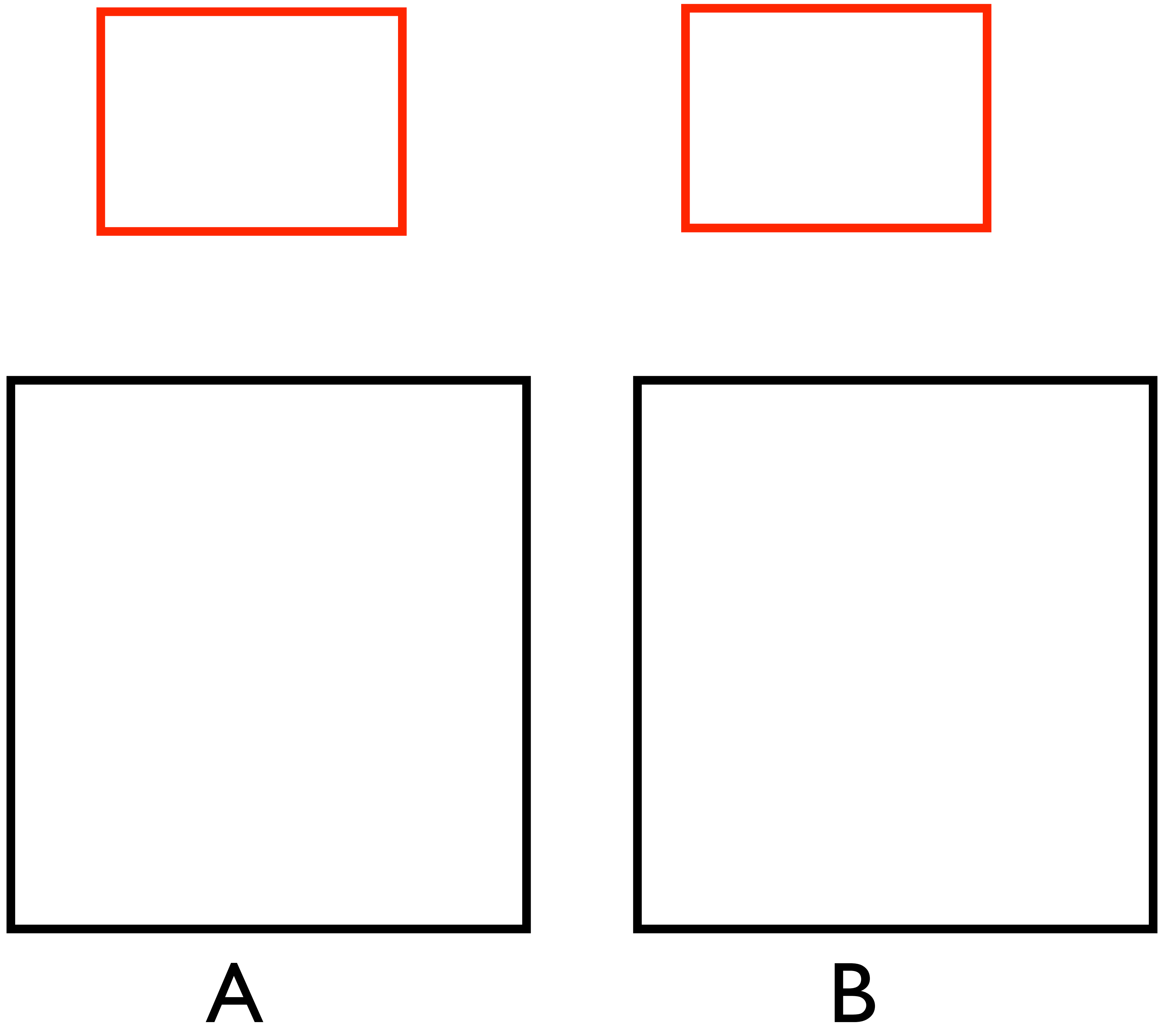


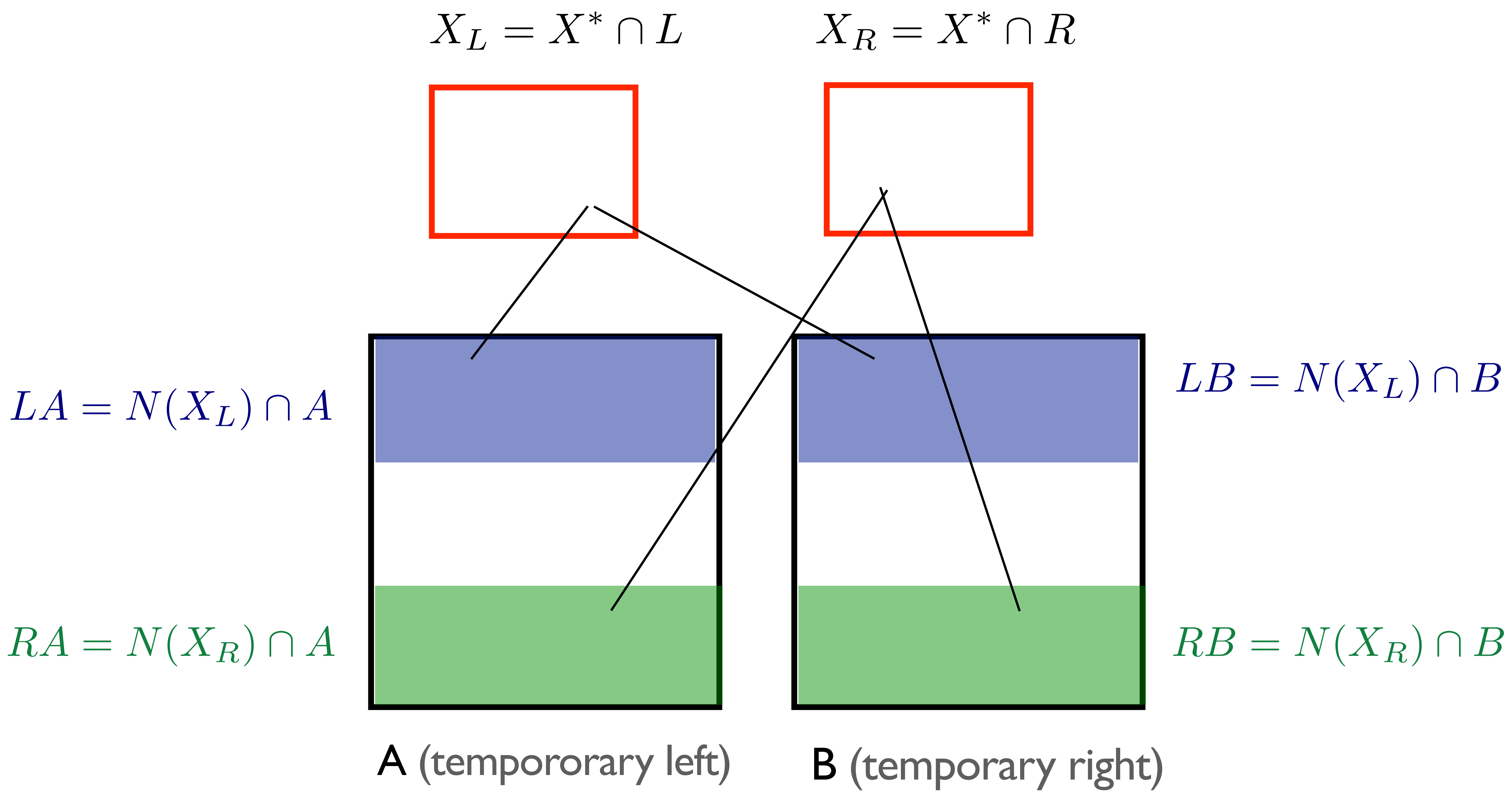
Step 1: Guess the partition of X^* into L and R.

$$2^{|X^*|} = 2^{k+1}$$

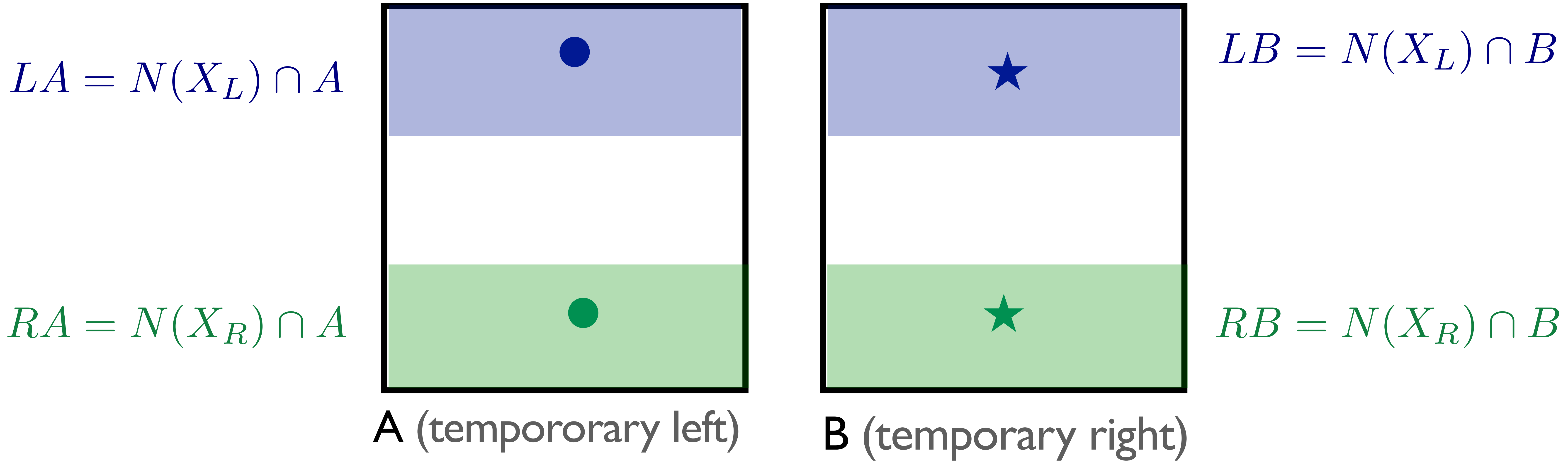
$X_L = X^* \cap L$

$X_R = X^* \cap R$





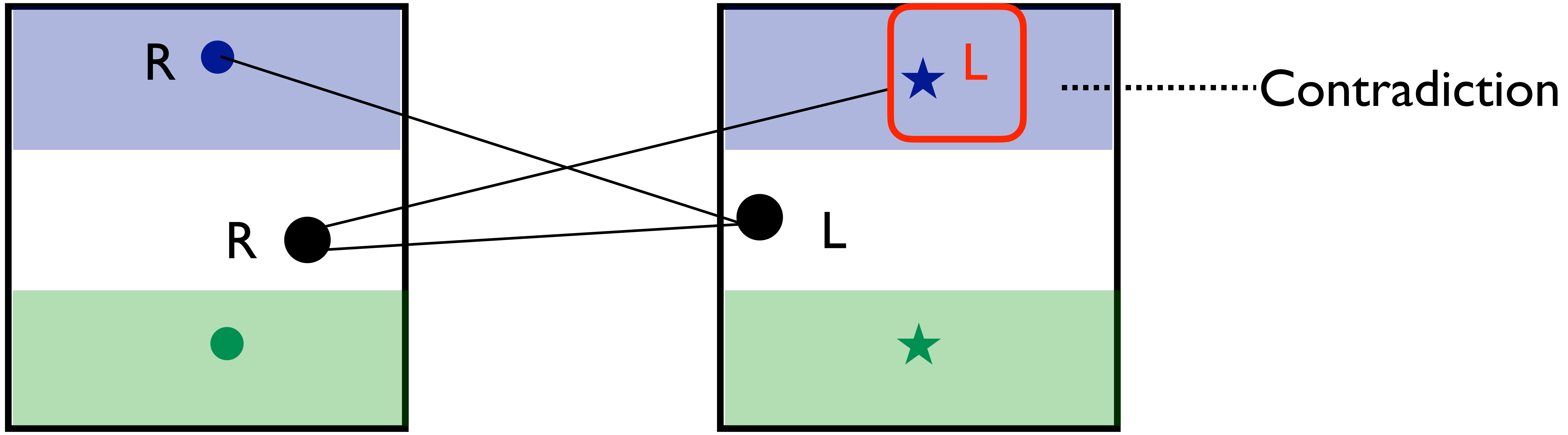
Revised goal: Find a set S ⊆ A ∪ B of at most k vertices such that G-S has a bipartition (L,R) where the blue vertices (LA ∪ LB) goes to the right side (R) and the green vertices (RA ∪ RB) goes to the left side (L).



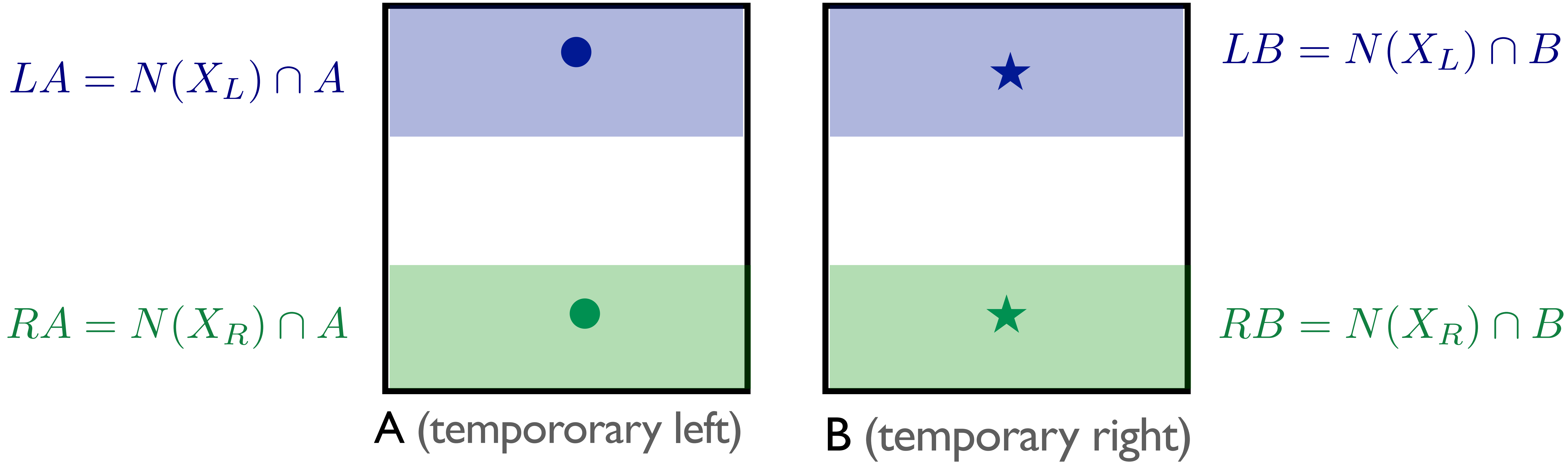
Revised goal: Find a set S ⊆ A ∪ B of at most k vertices such that G-S has a bipartition (L,R) where the blue vertices (LA ∪ LB) goes to the right side (R) and the green vertices (RA ∪ RB) goes to the left side (L).

Observation (necessary condition): A solution S is a (LA ∪ RB)-(LB ∪ RA) separator.

Suppose there is a (LA,LB)-path in L ∪ R.



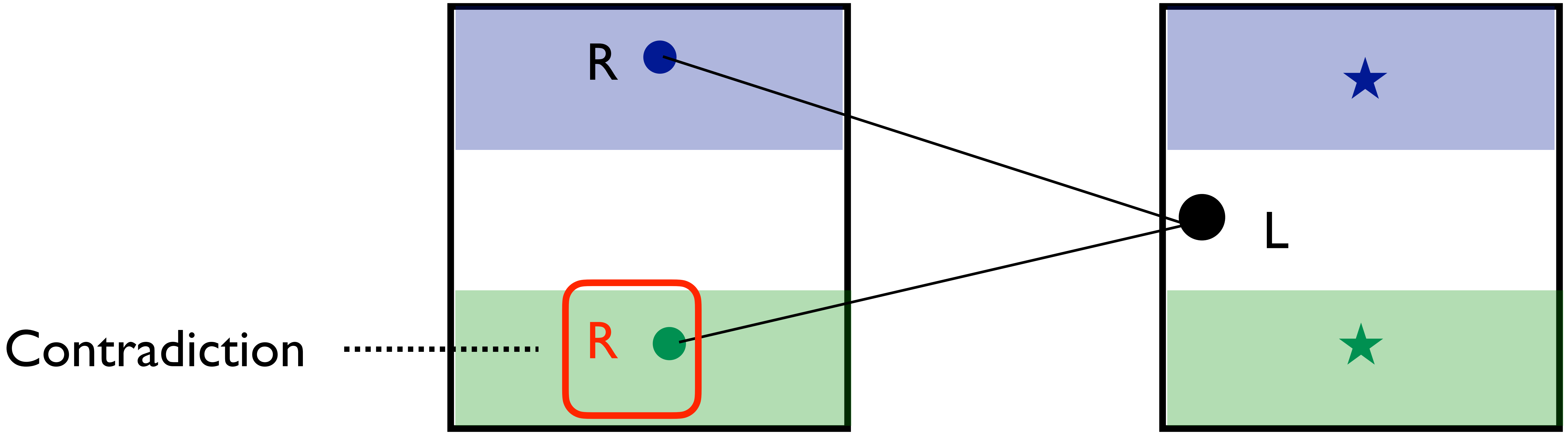
Therefore, no (LA,LB)-path. Similarly, no (RB,RA)-path.



Revised goal: Find a set S ⊆ A ∪ B of at most k vertices such that G-S has a bipartition (L,R) where the blue vertices (LA ∪ LB) goes to the right side (R) and the green vertices (RA ∪ RB) goes to the left side (L).

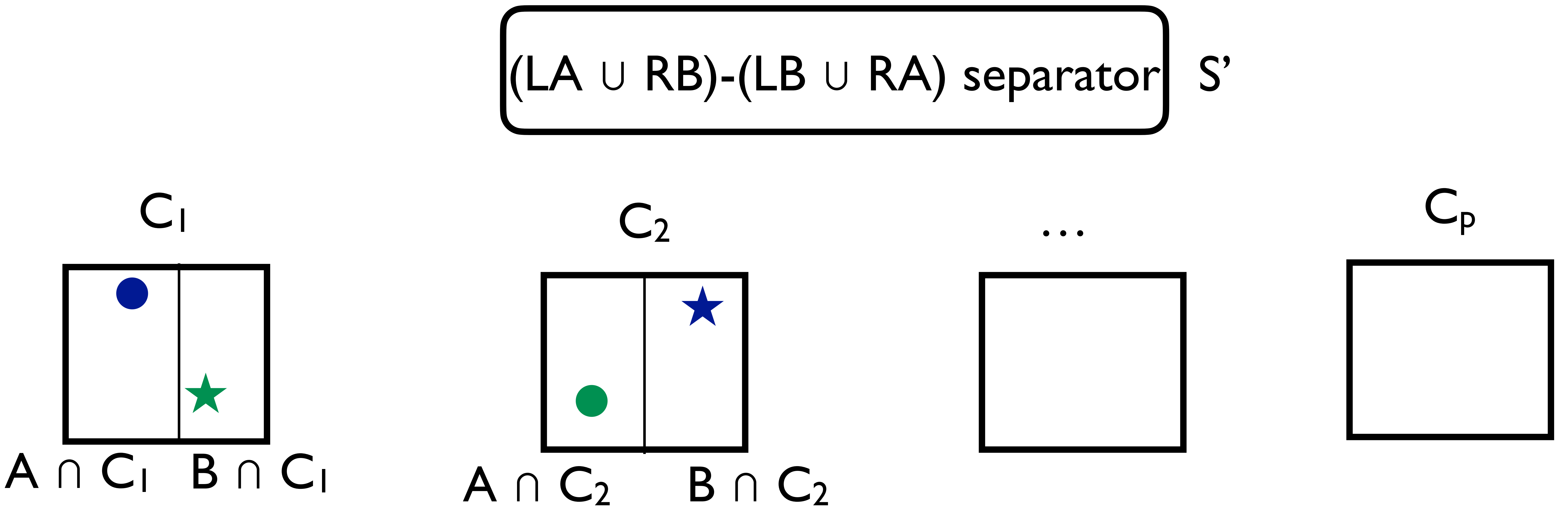
Observation (necessary condition): A solution S is a (LA ∪ RB)-(LB ∪ RA) separator.

Suppose there is a (LA,RA)-path in L ∪ R.



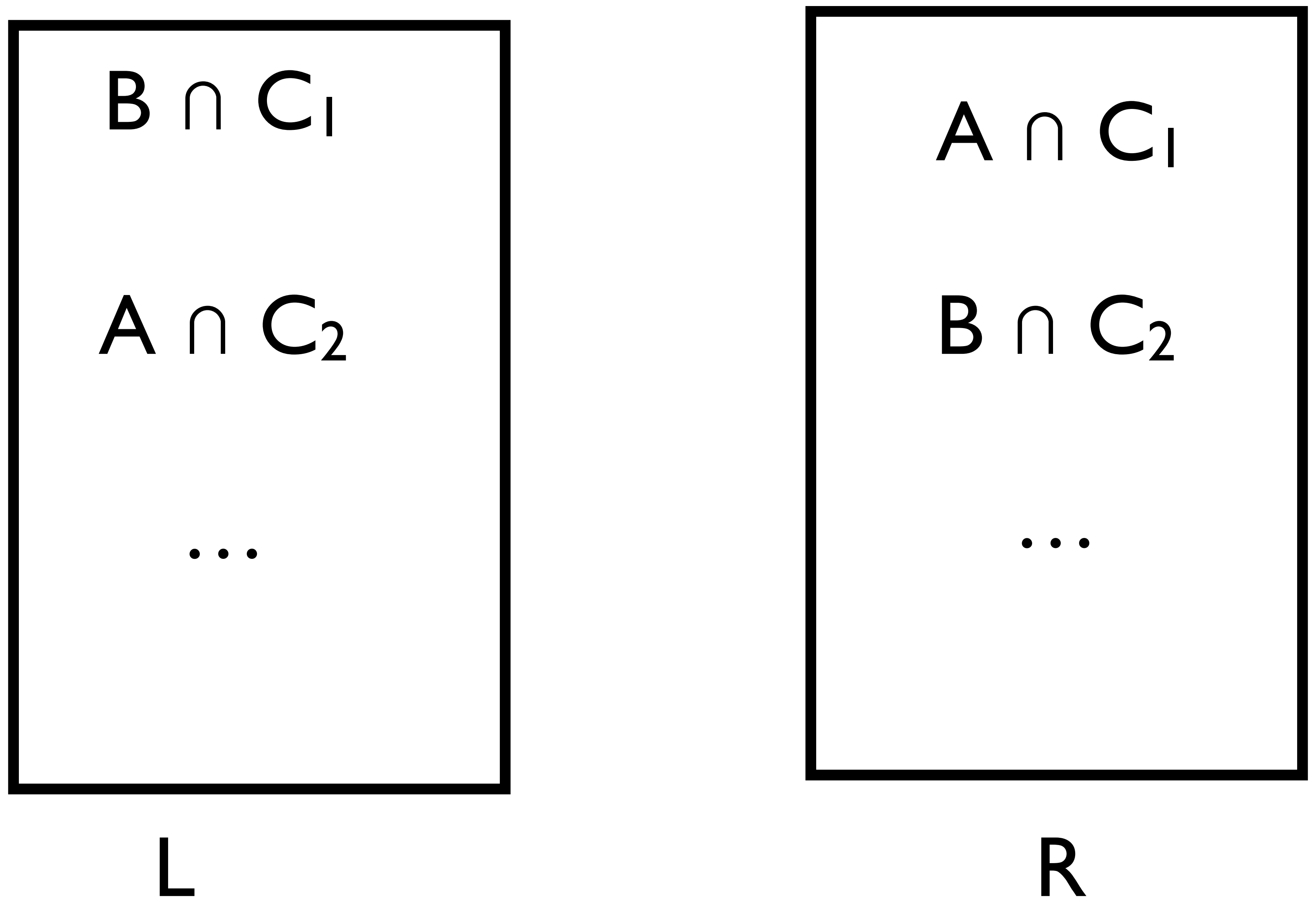
Therefore, no (LA,RA)-path. Similarly, no (RB,LB)-path.

Observation (sufficient condition): Any (LA ∪ RB)-(LB ∪ RA) separator is a solution.



C_i are connected components after removal of S’

Output bipartition of the union of C_i s



Claim: S is a solution if and only if S is an (LA ∪ RB)-(LB ∪ RA) separator size at most k.

Such a set S can be found using k rounds of Ford-Fulkerson is O(k (n+m)) time.

Disjoint OCT

Input: A graph G, positive integers i and k such that k<i, a solution X^* of size i

Question: Does there exists a set S such that :

- S is disjoint from X^*
- $|S| \leq k$
- G-S is bipartite?

Overall algorithm:

- Guess the partition of X* into L and R. $2^{|X^*|} = 2^i$
- Compute sets LA,LB,RA,RB. $O(n+m)$
- Find an (LA ∪ RB)-(LB ∪ RA) separator $O(k(n+m))$

Overall running time: $\mathcal{O}(2^i \cdot k(n+m))$

Odd cycle transversal can be solved in $\mathcal{O}(3^k \cdot kn(n+m))$ time.