Integer Programs meet Fixed-Parameter Tractability

Alexandra Lassota (TU/e)

Integer Programming

$$Ax \leq b, \; x \in \mathbb{Z}^n$$

Integer Programming

Introduction and Relevance

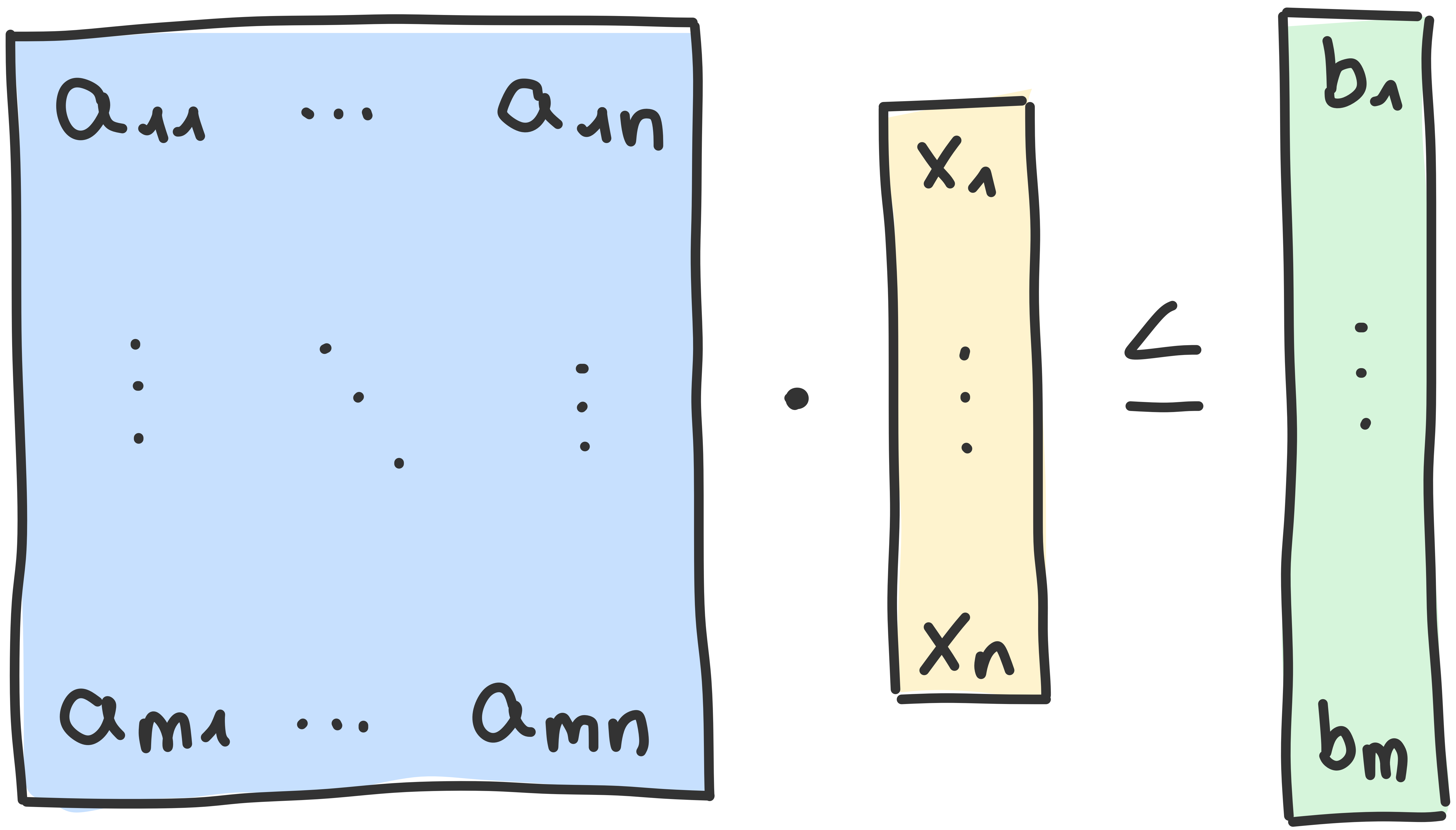
$$Ax \leq b, \; x \in \mathbb{Z}^n$$

Integer Programming

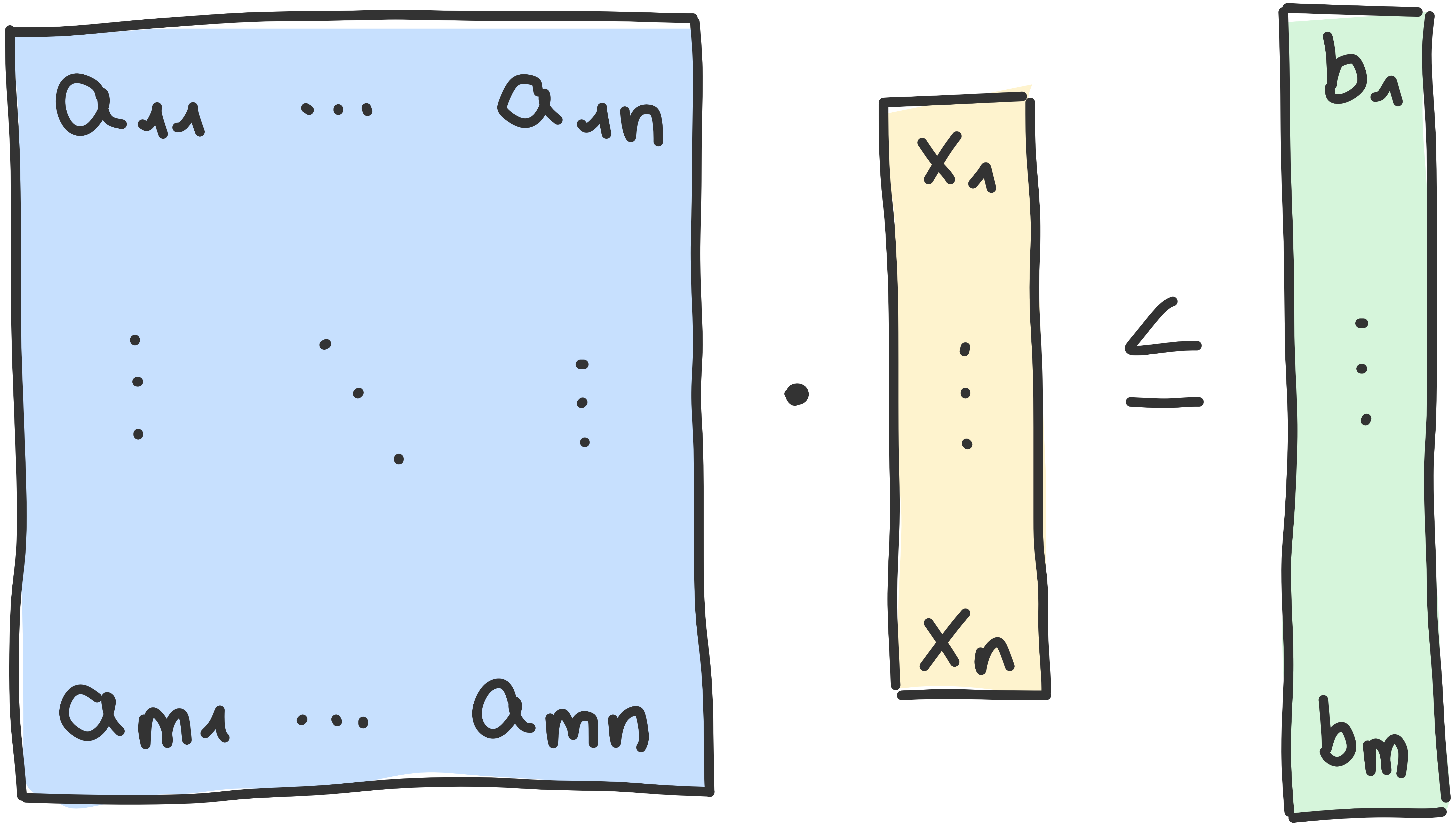
~~Introduction and Relevance~~

$$Ax \leq b,\ x \in \mathbb{Z}^n$$

Integer Programming



Integer Programming



NP-hard -> exponential blow-up in runtime

How do we deal the NP-hardness

How do we deal the NP-hardness

- work on solvers

How do we deal the NP-hardness

- work on solvers
- heuristics

How do we deal the NP-hardness

- work on solvers
- heuristics
- approximation algorithms

How do we deal the NP-hardness

- work on solvers
- heuristics
- approximation algorithms
- focussing on sub-classes of the problem

How do we deal the NP-hardness

- work on solvers
- heuristics
- approximation algorithms
- focussing on sub-classes of the problem
- fixed-parameter tractability

How do we deal the NP-hardness

- work on solvers
- heuristics
- approximation algorithms
- focussing on sub-classes of the problem
- fixed-parameter tractability

Fixed-Parameter Tractability

Given: Problem P with parameter k

Fixed-Parameter Tractability

Given: Problem P with parameter k

If we can solve any instance of P in time

$$f(k) \cdot \mathrm{poly}(|I|)$$

then P is in FPT/ P is fixed-parameter tractable w.r.t. k

Fixed-Parameter Tractability - Example

Given: $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$, $x \in \{0,1\}^n$ with parameter n

Fixed-Parameter Tractability - Example

Given: $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$, $x \in \{0,1\}^n$ with parameter n

Instance size: $O(nm \log(m\Delta))$

$\uparrow \|A\|_\infty$

Fixed-Parameter Tractability - Example

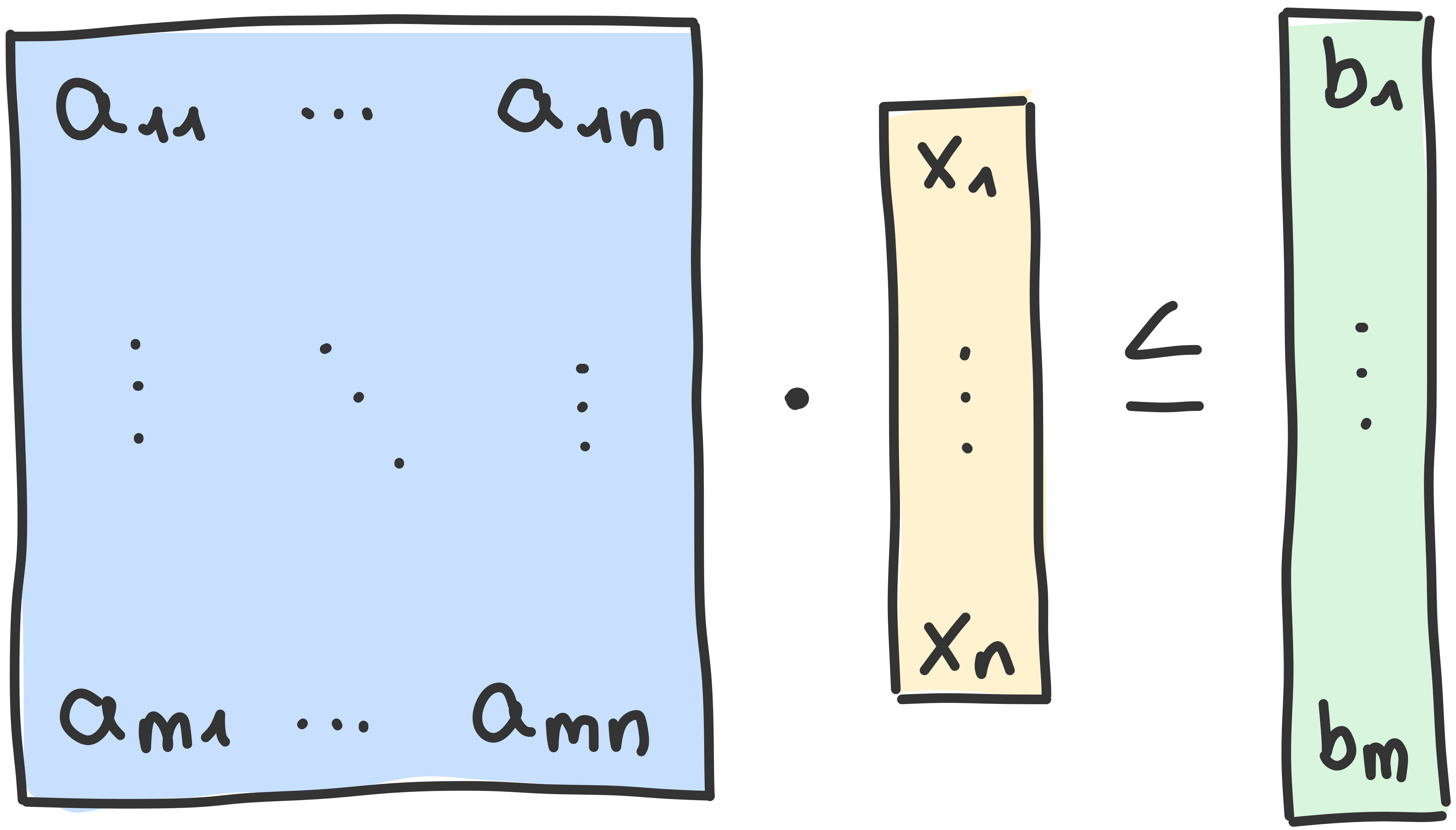
Given: $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$, $x \in \{0,1\}^n$ with parameter n

Instance size: $O(nm \log(m\Delta))$

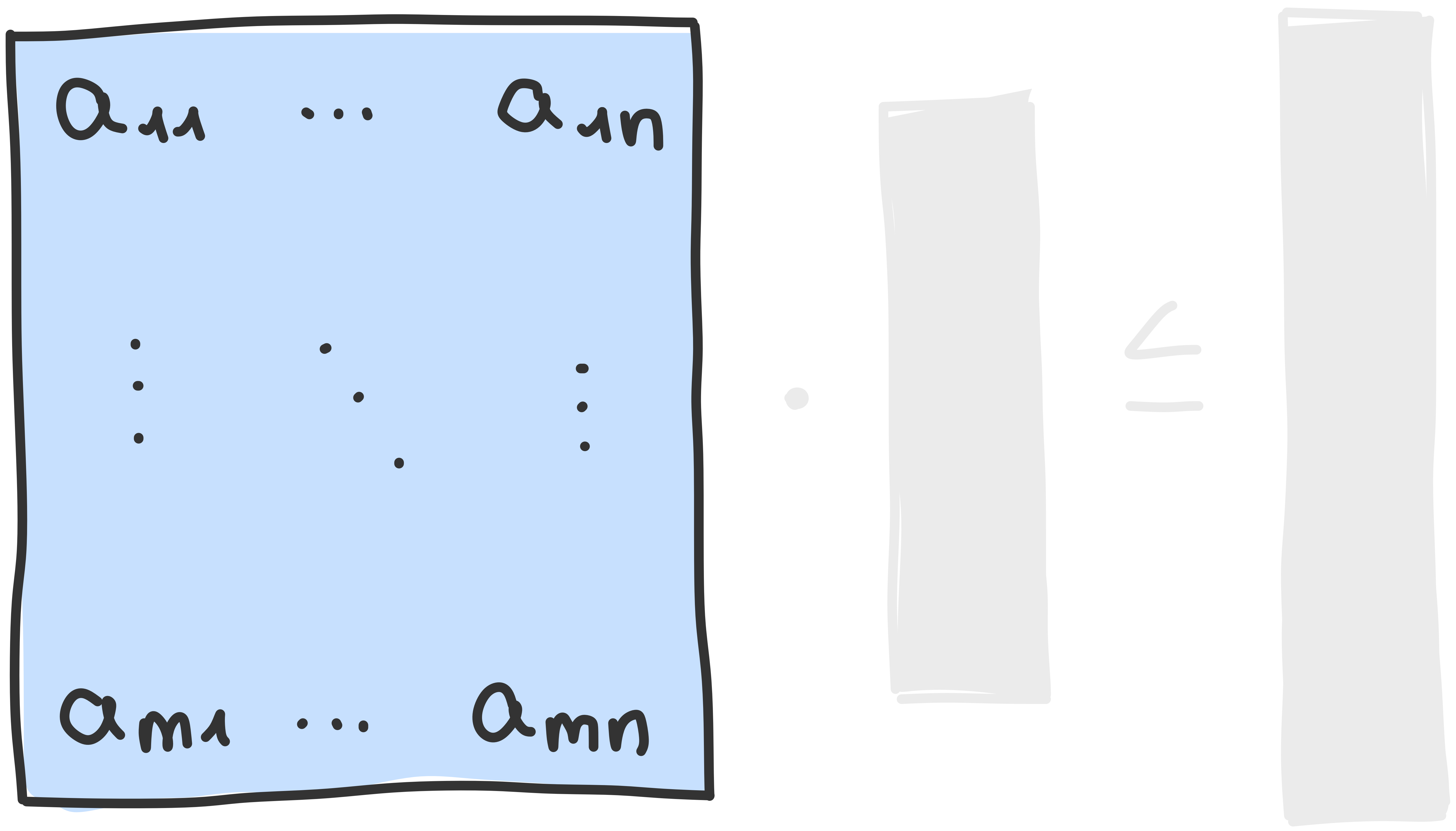
$\uparrow \|A\|_\infty$

Algorithm: Guess and test solution; $\underbrace{2^n}_{f(n)} \cdot \underbrace{O(nm)}_{poly(|I|)}$

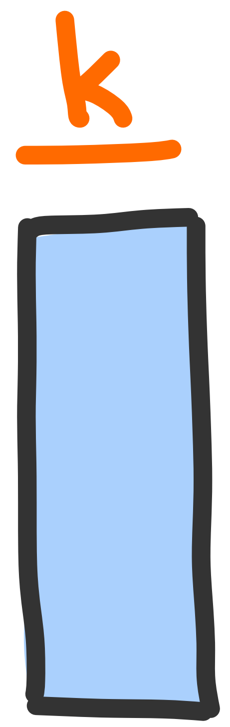
Integer Programming meets FPT



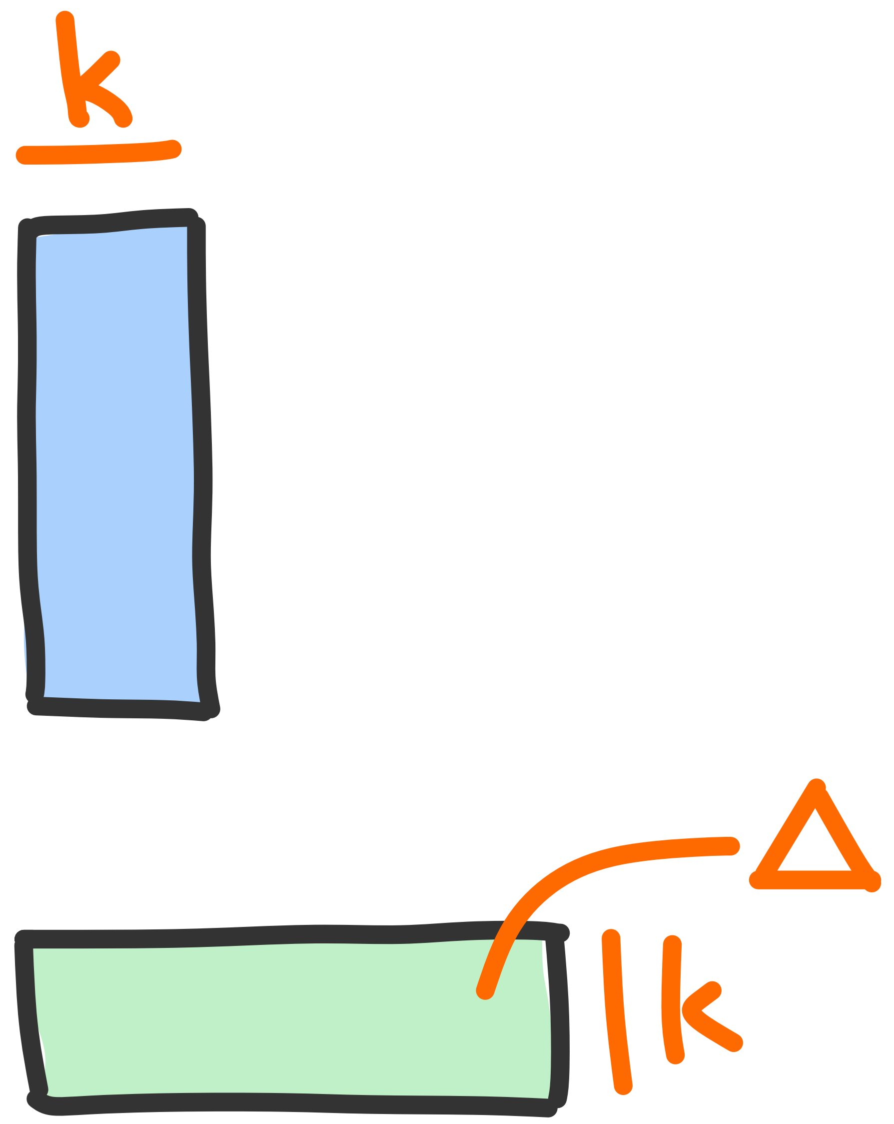
Integer Programming meets FPT



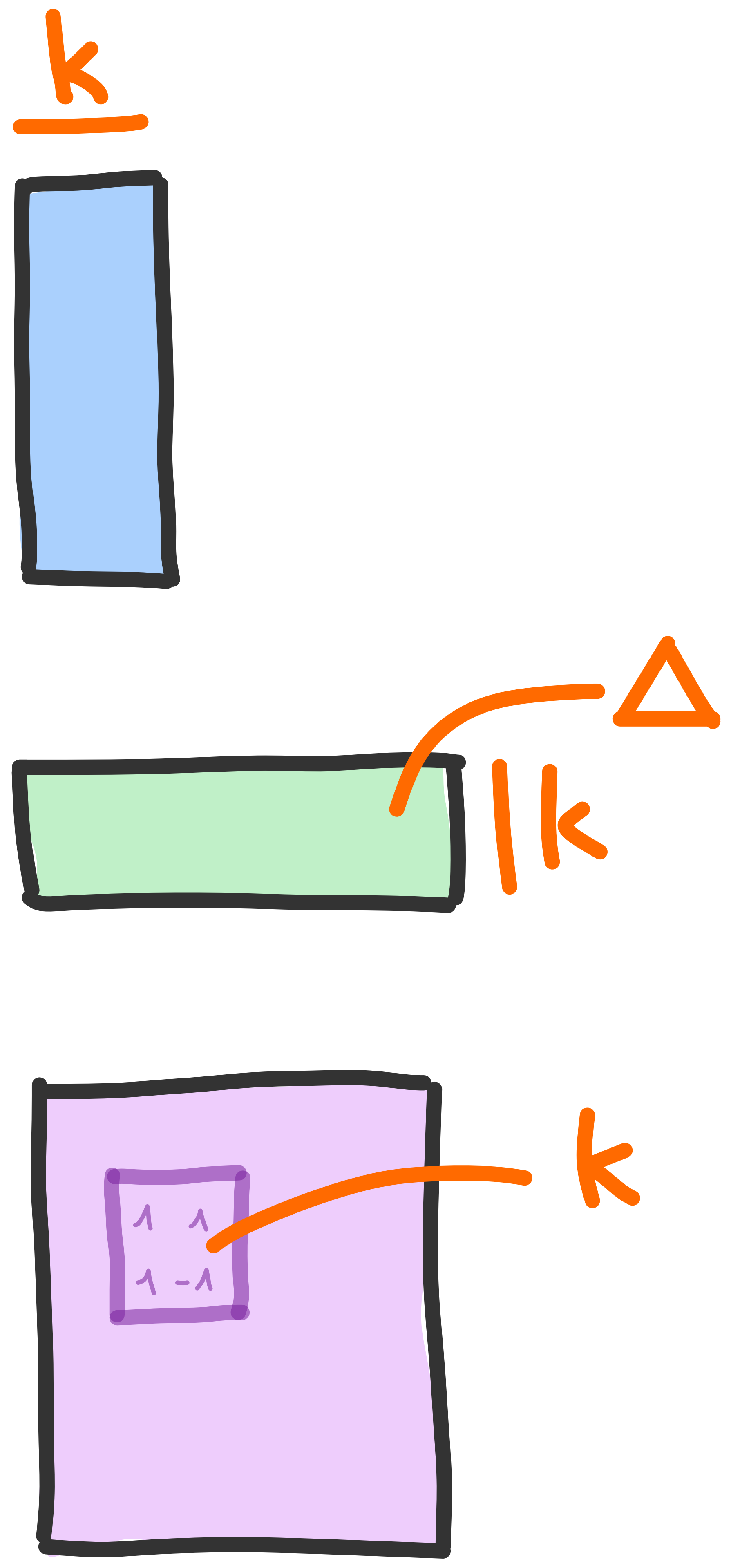
Integer Programming meets FPT



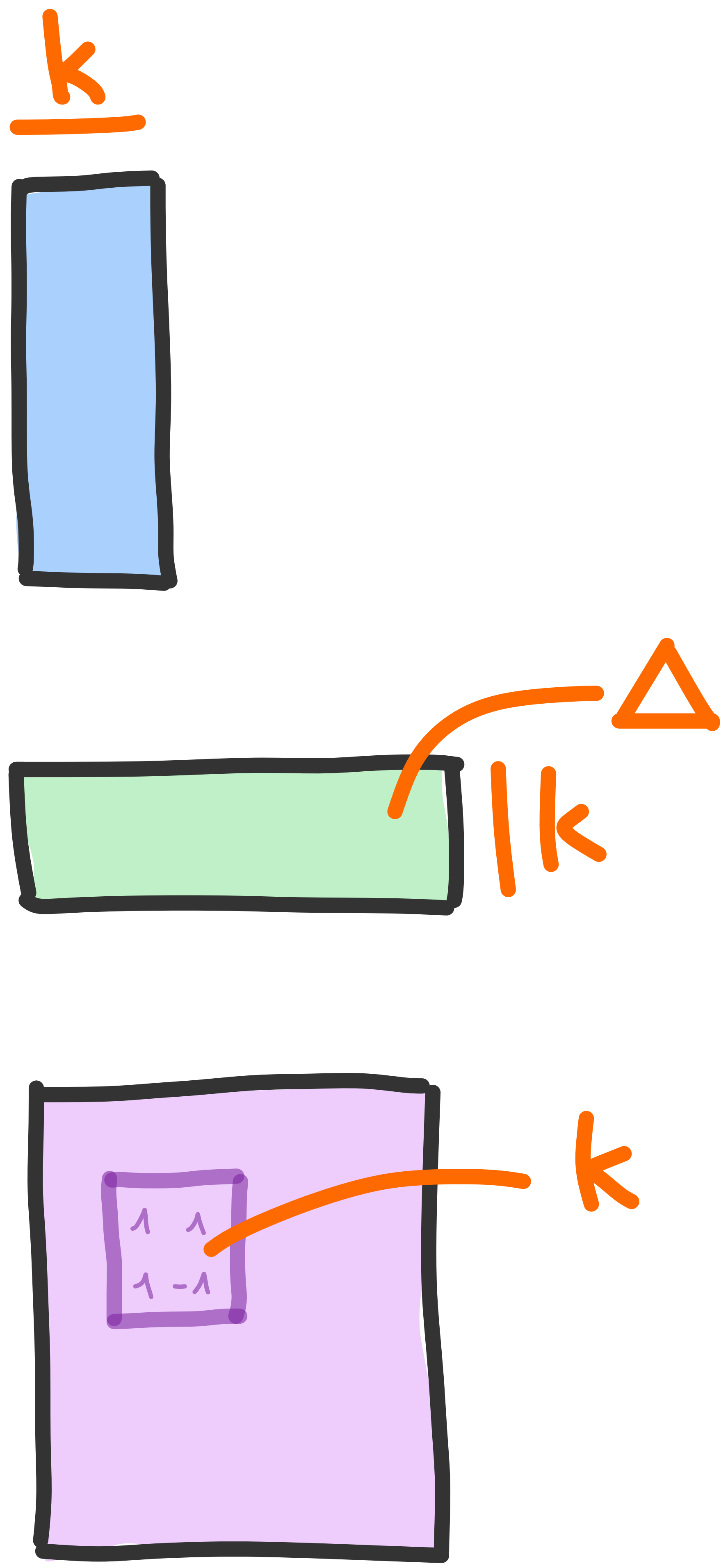
Integer Programming meets FPT

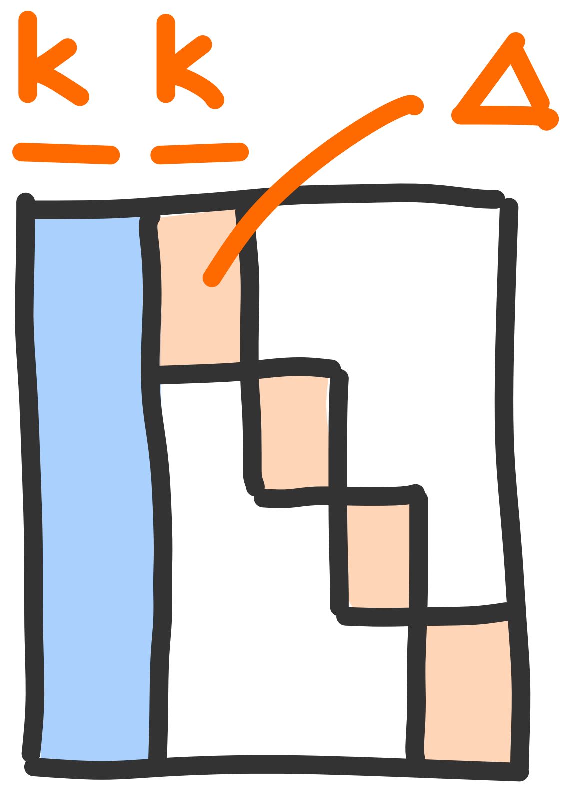


Integer Programming meets FPT

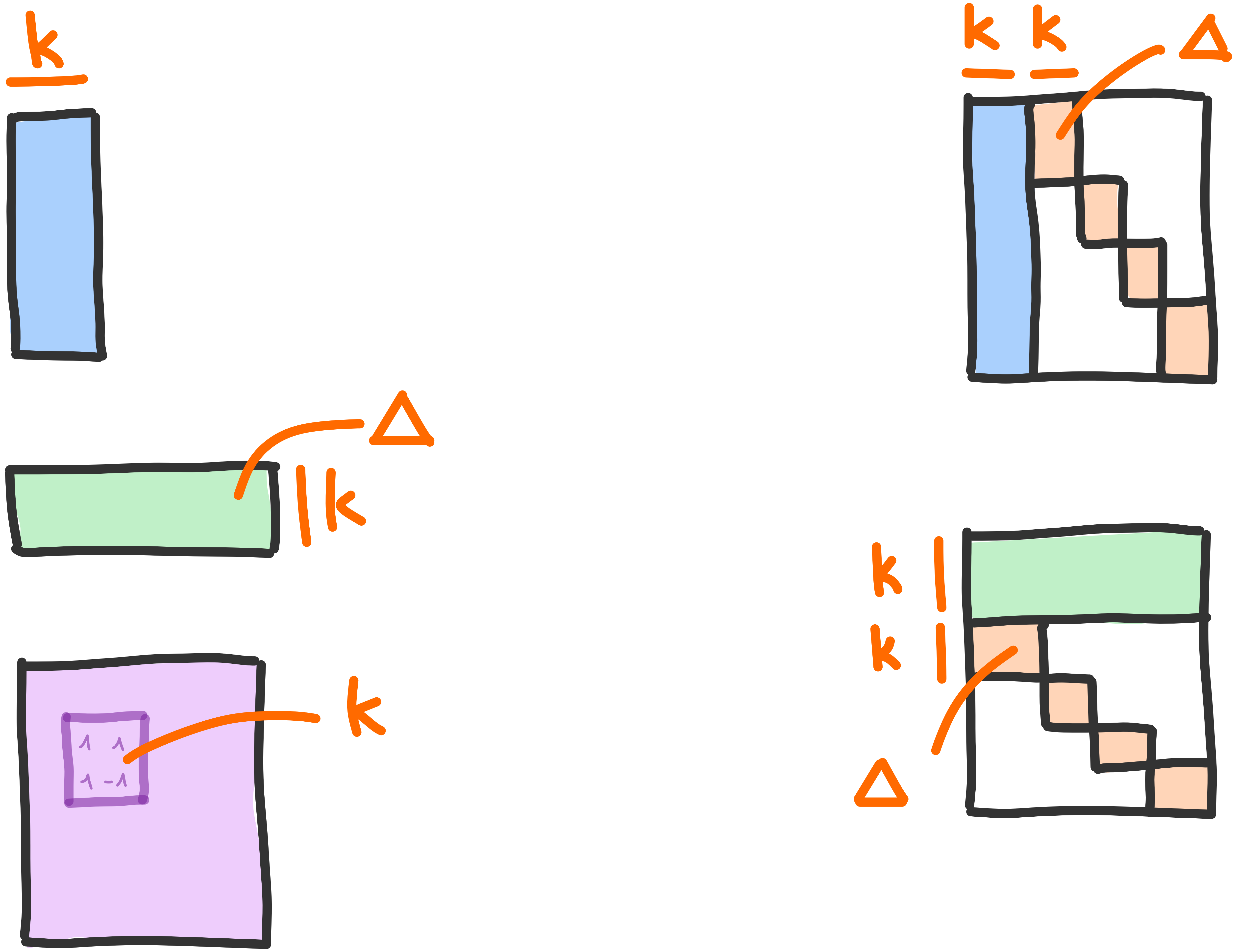


Integer Programming meets FPT

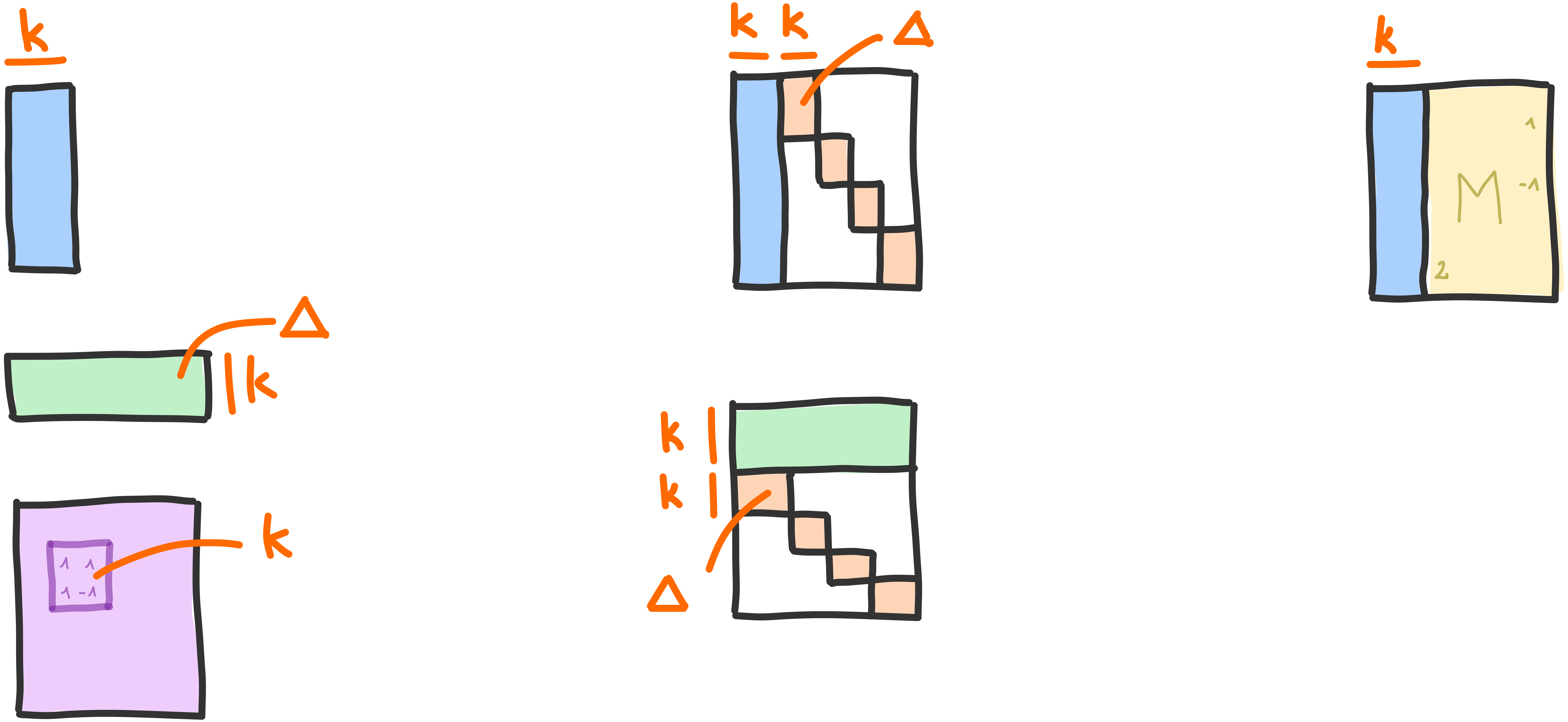




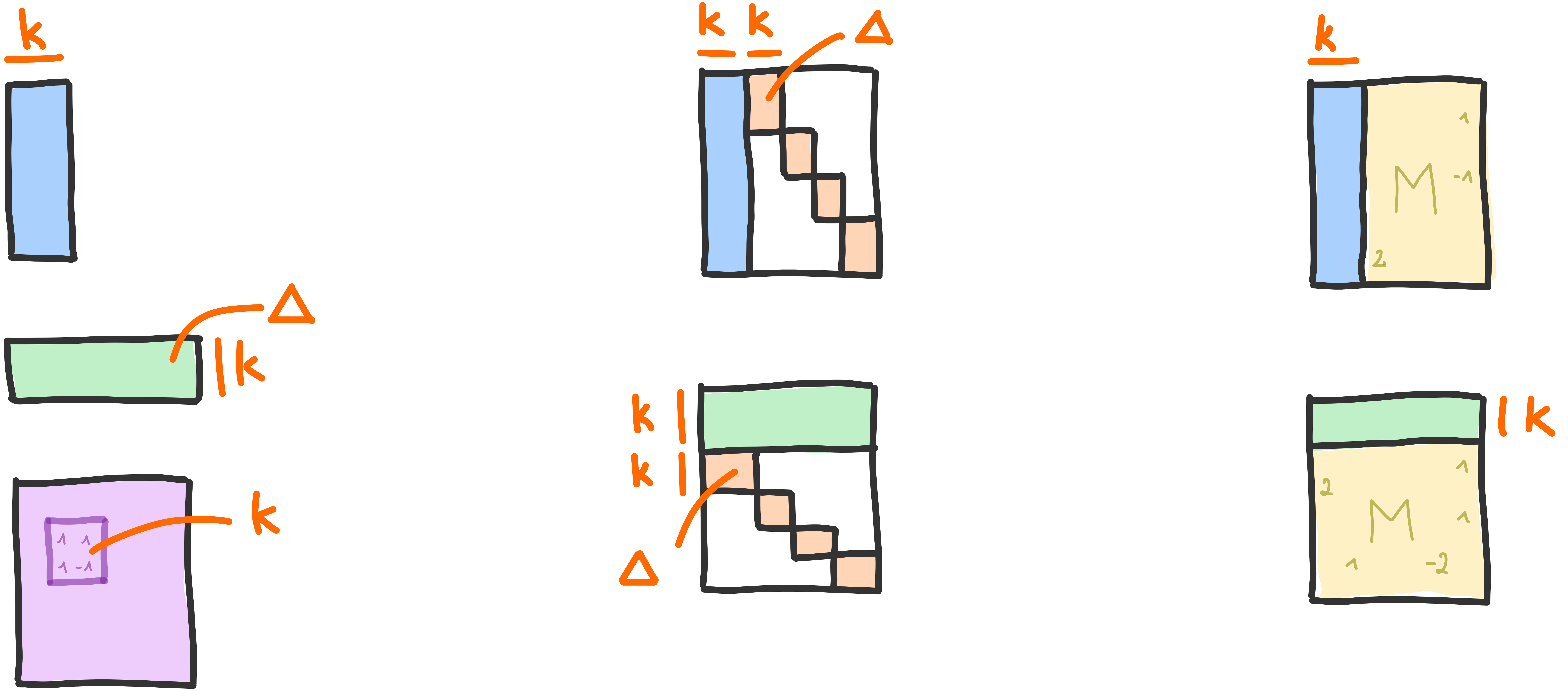
Integer Programming meets FPT



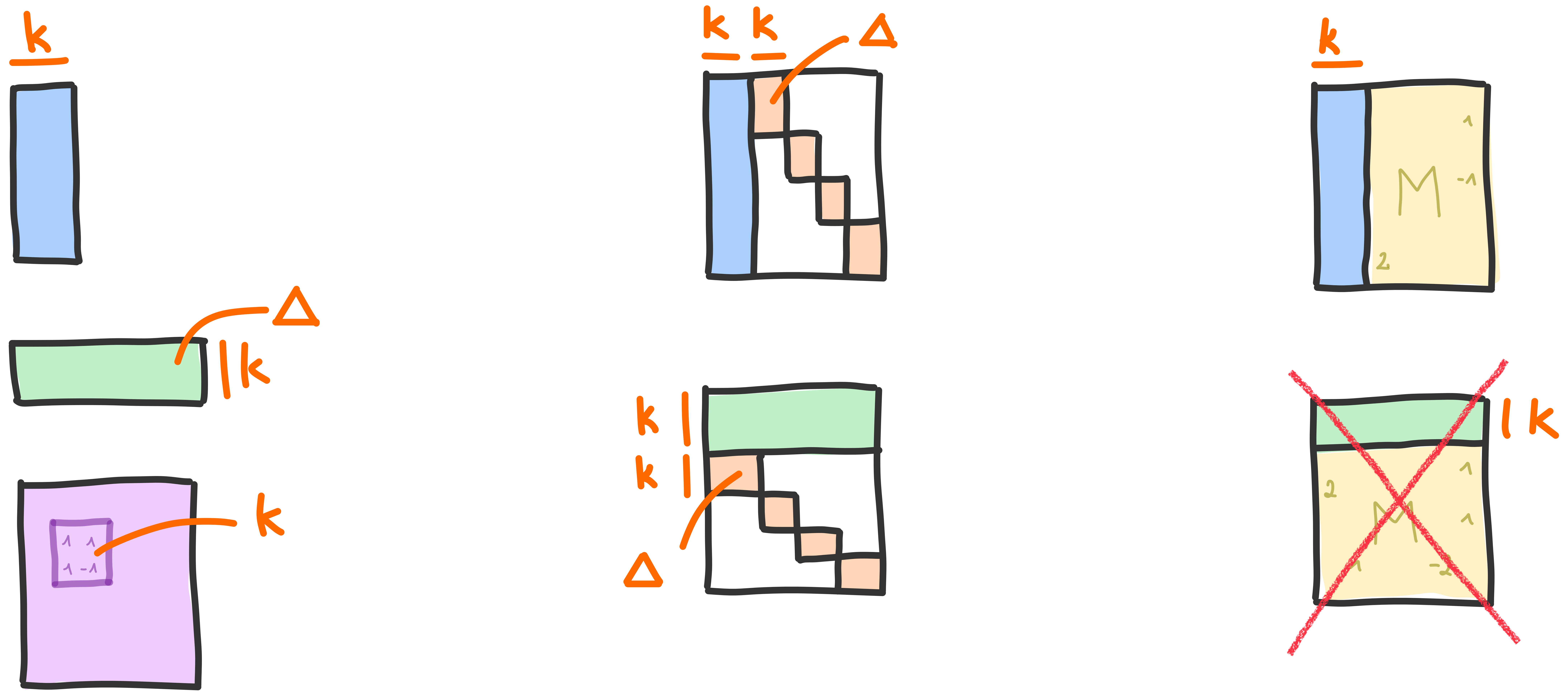
Integer Programming meets FPT



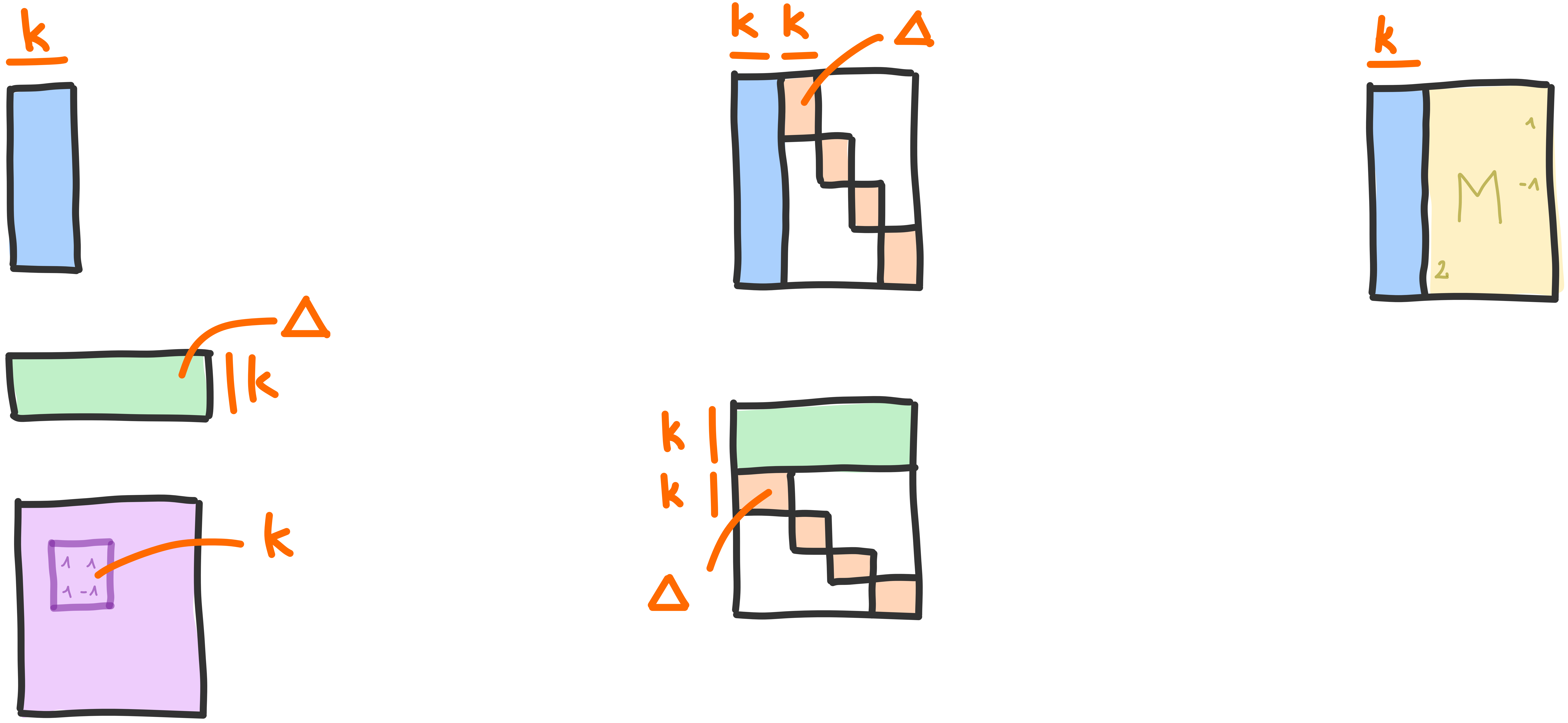
Integer Programming meets FPT



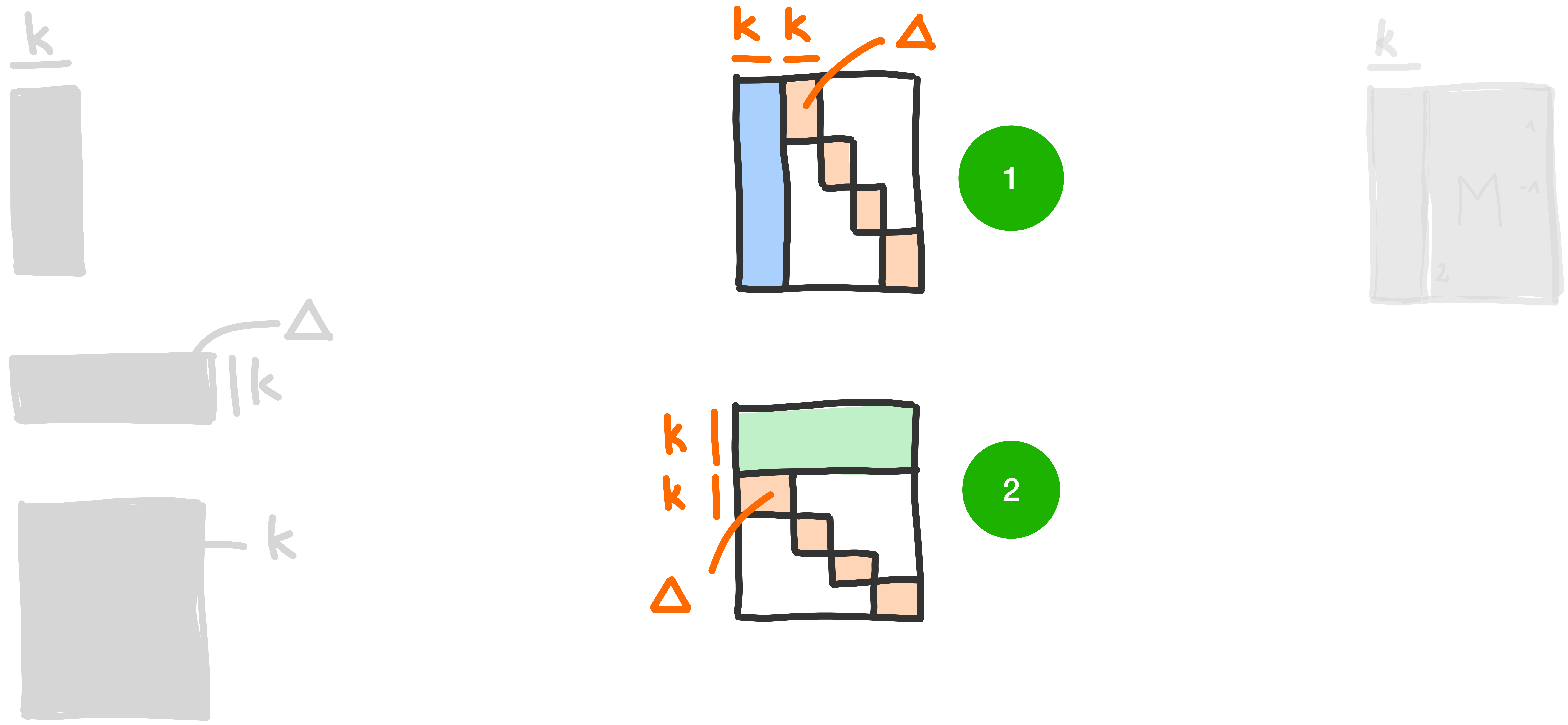
Integer Programming meets FPT



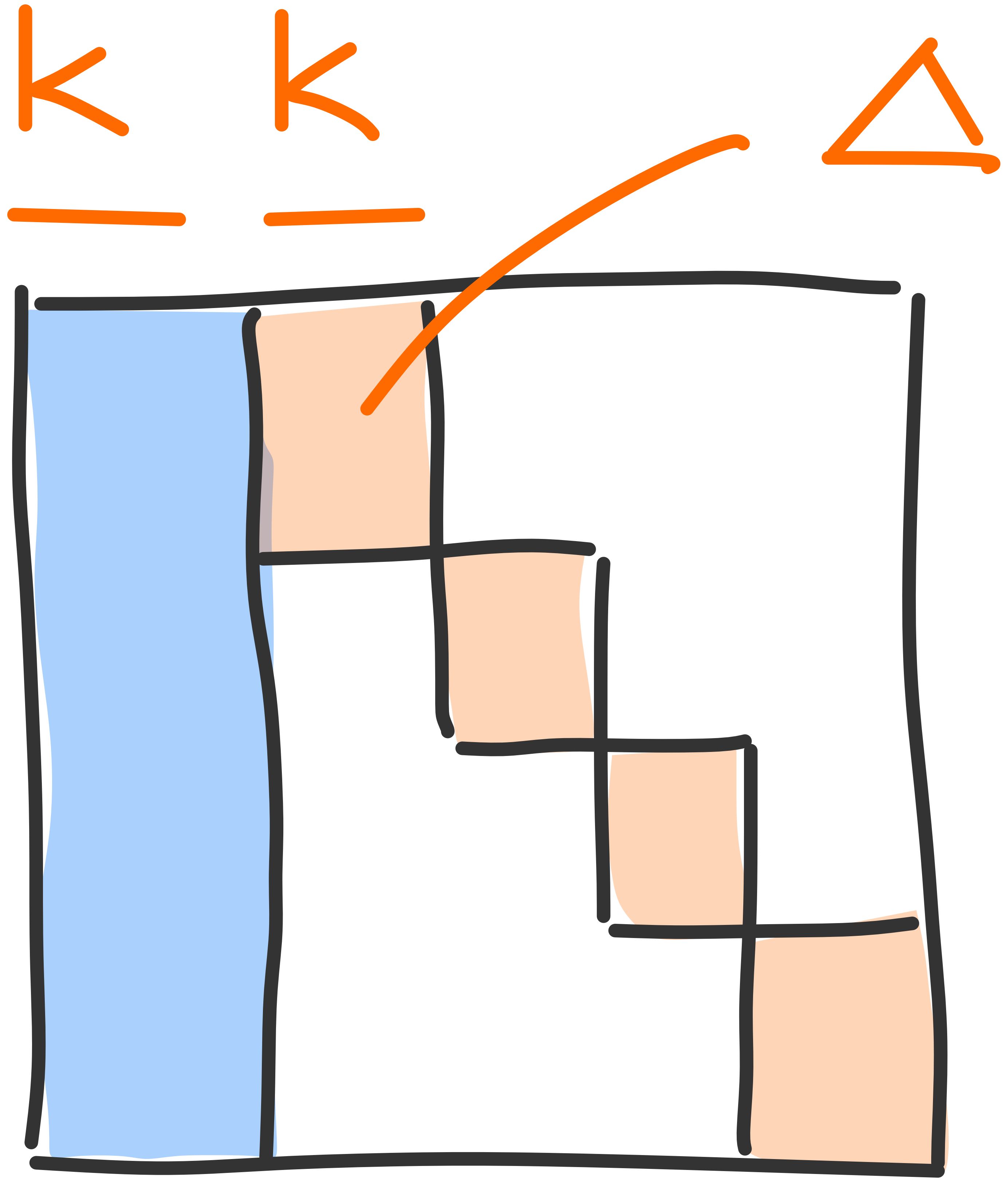
Integer Programming meets FPT



Integer Programming meets FPT - Agenda

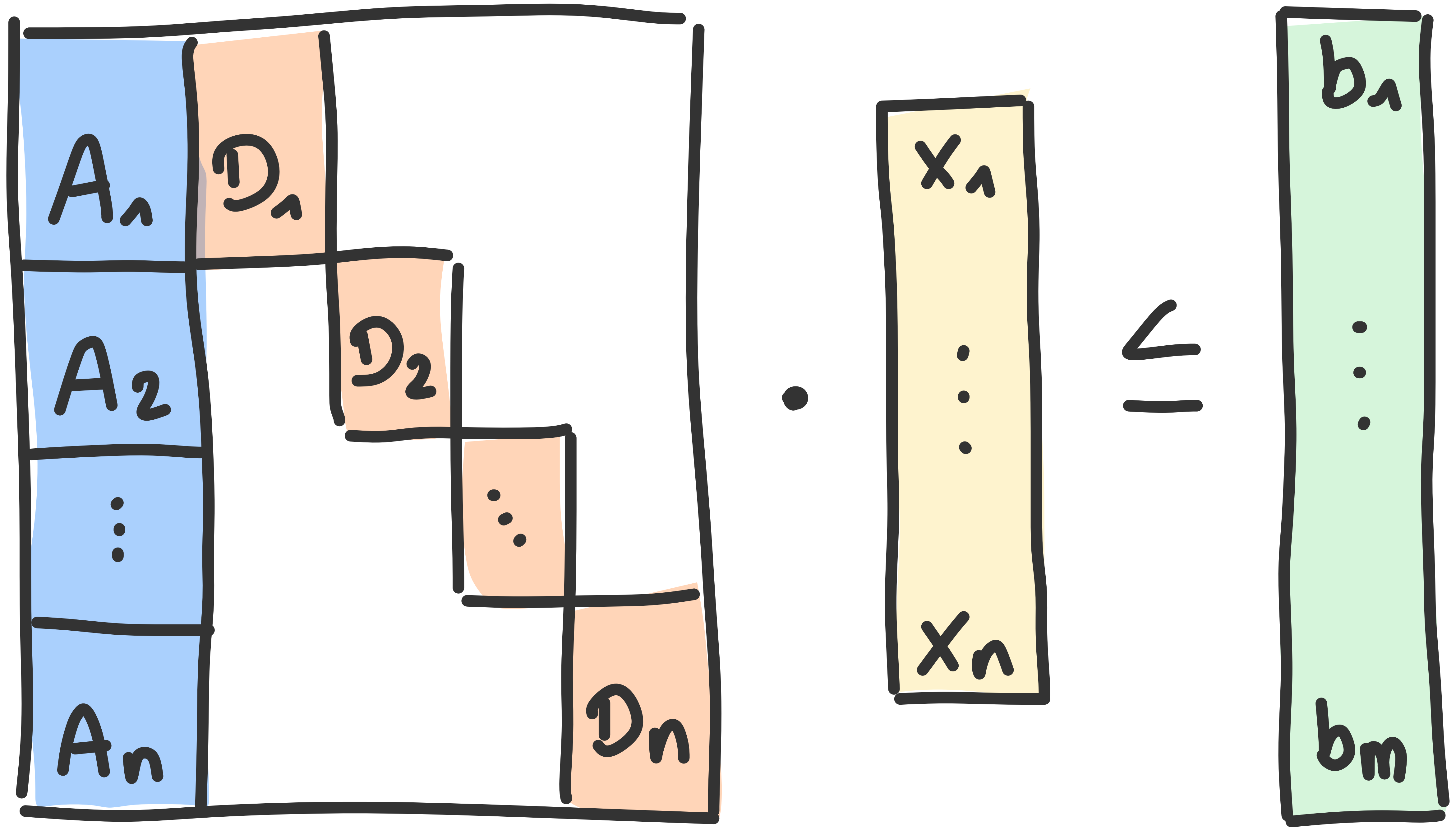


1

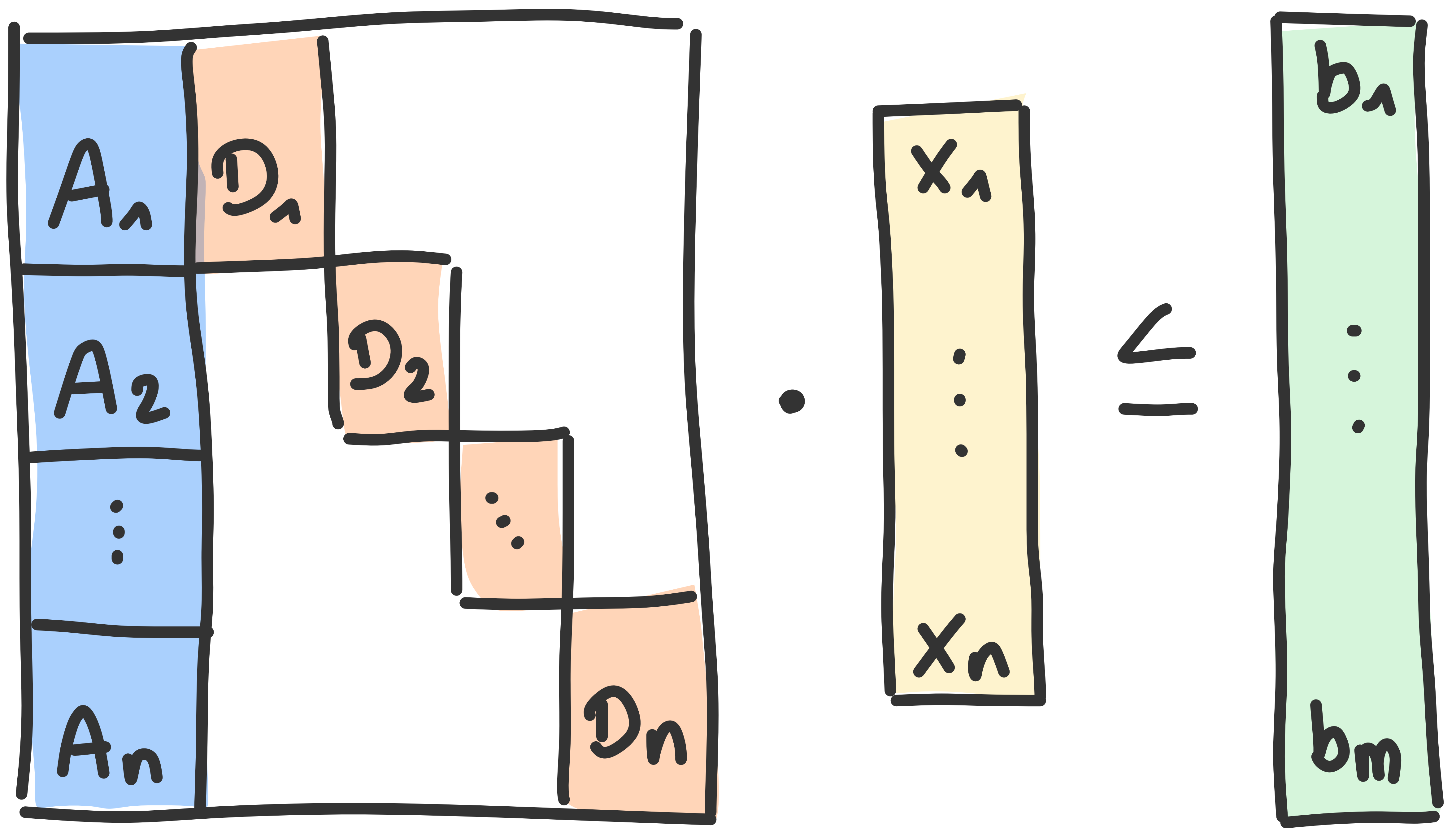


2-stage stochastic IPs

2-stage stochastic IPs

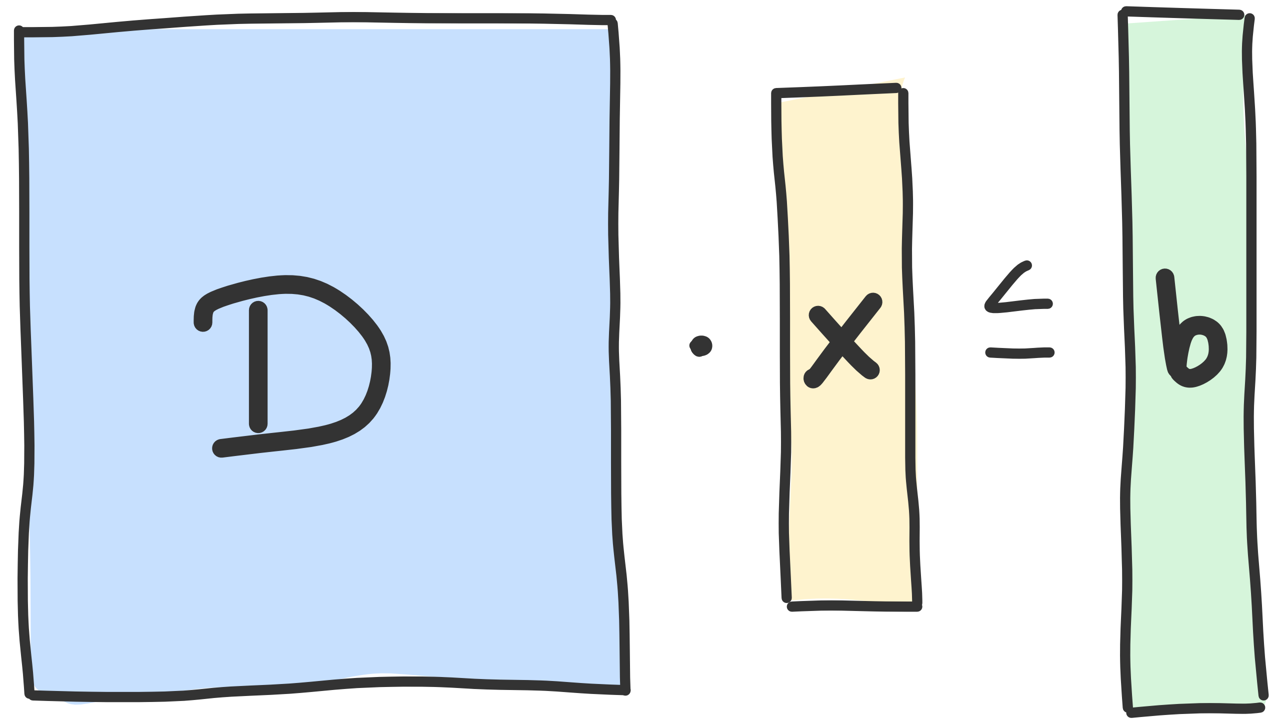


2-stage stochastic IPs

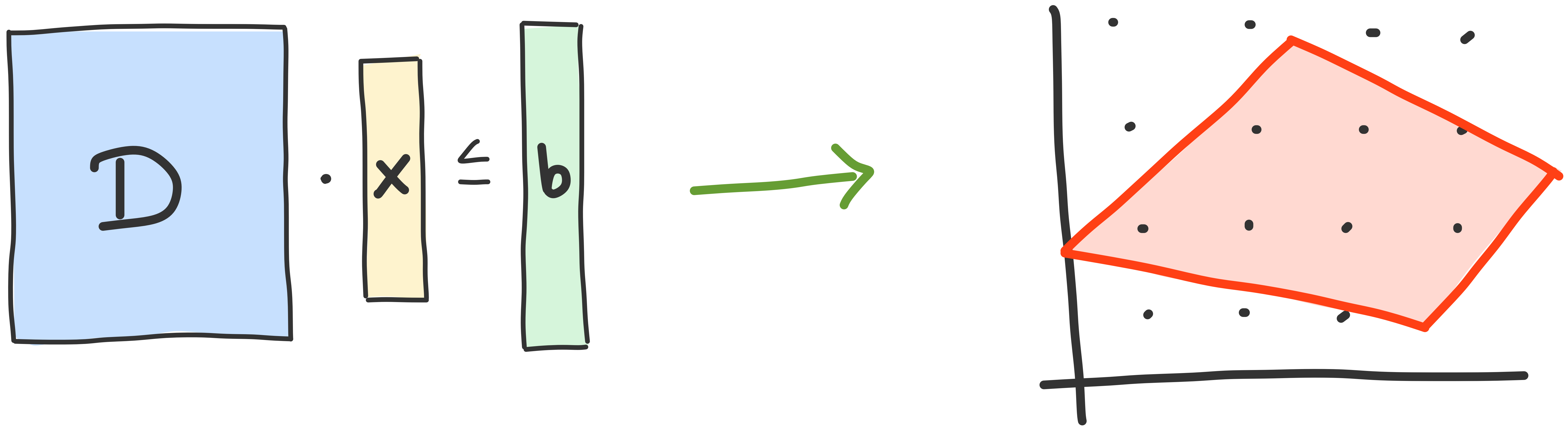


neural networks, worker scheduling, project planning, routing, facility location planning, …

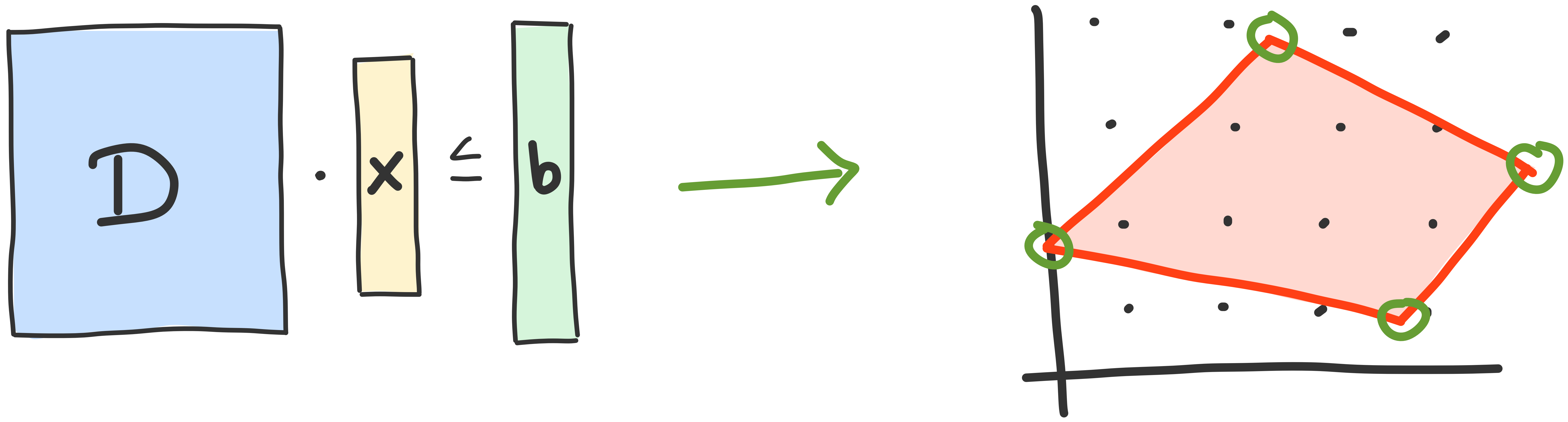
The Integer Hull



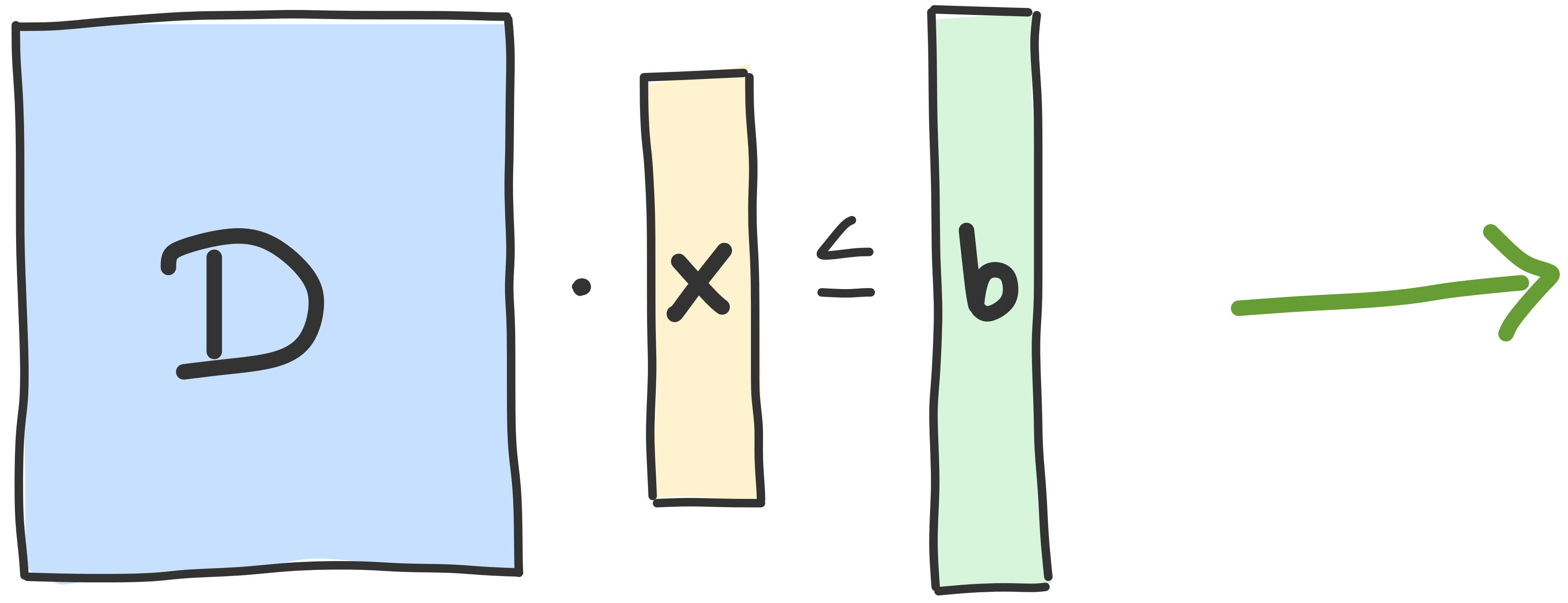
The Integer Hull

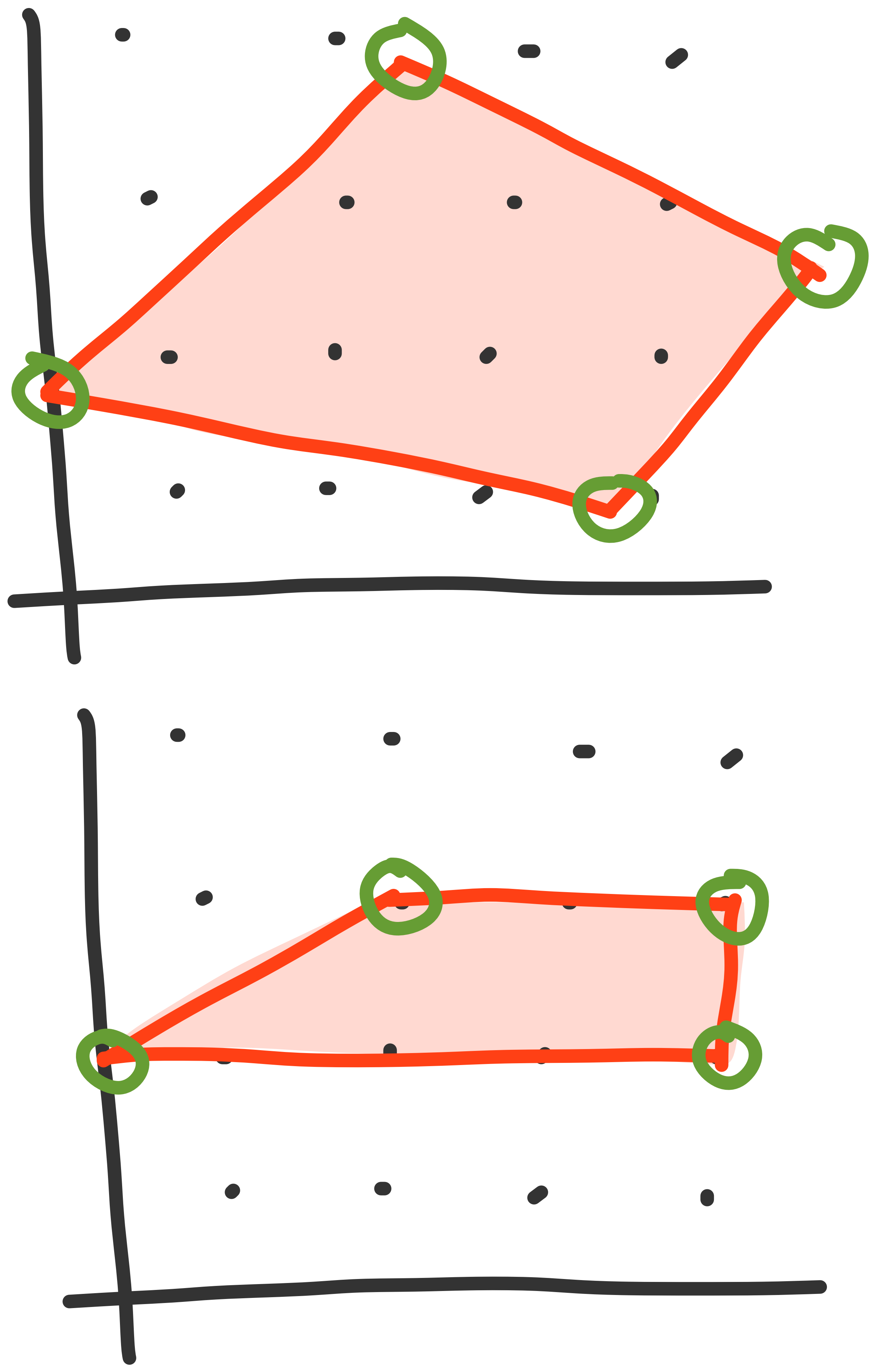


The Integer Hull

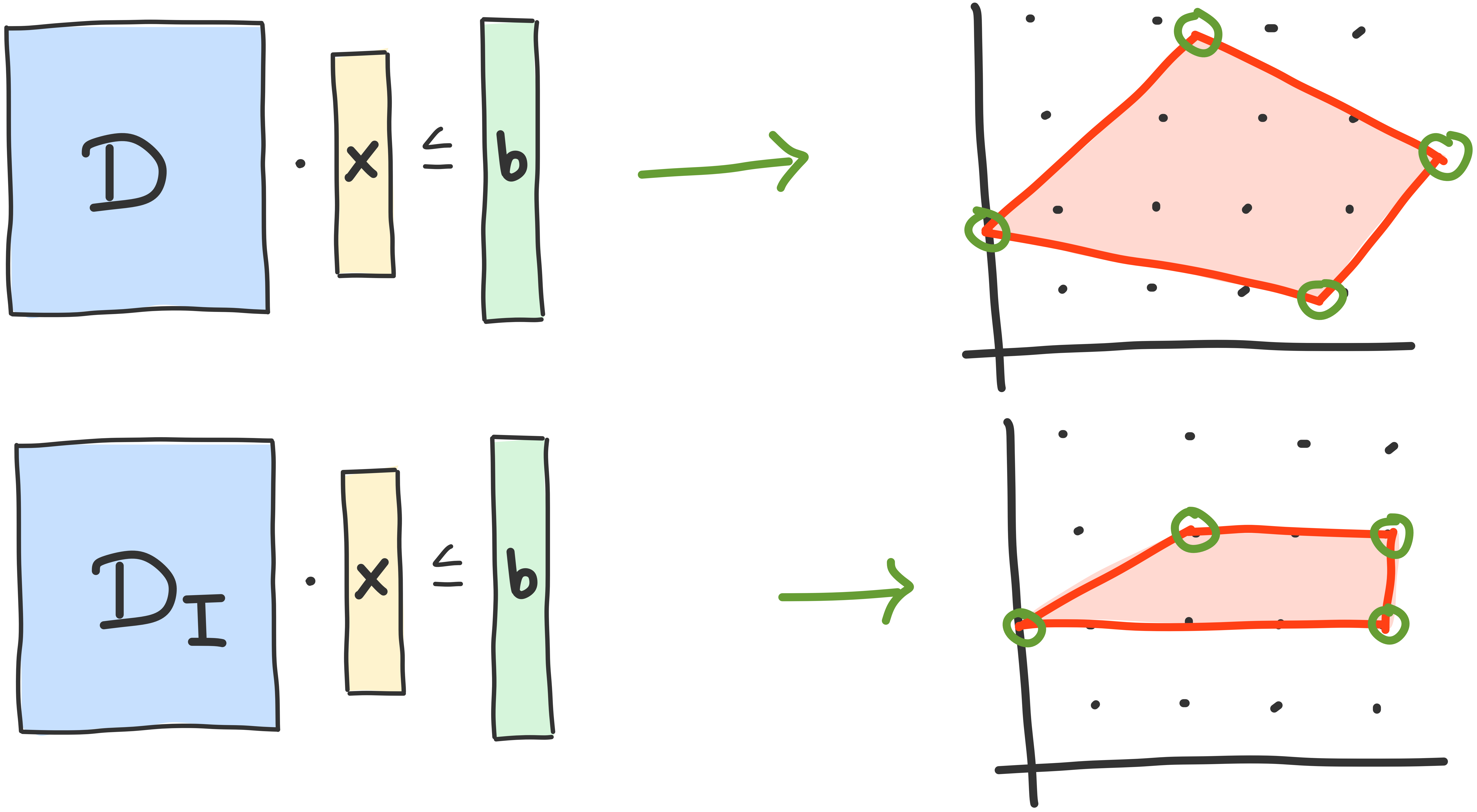


The Integer Hull

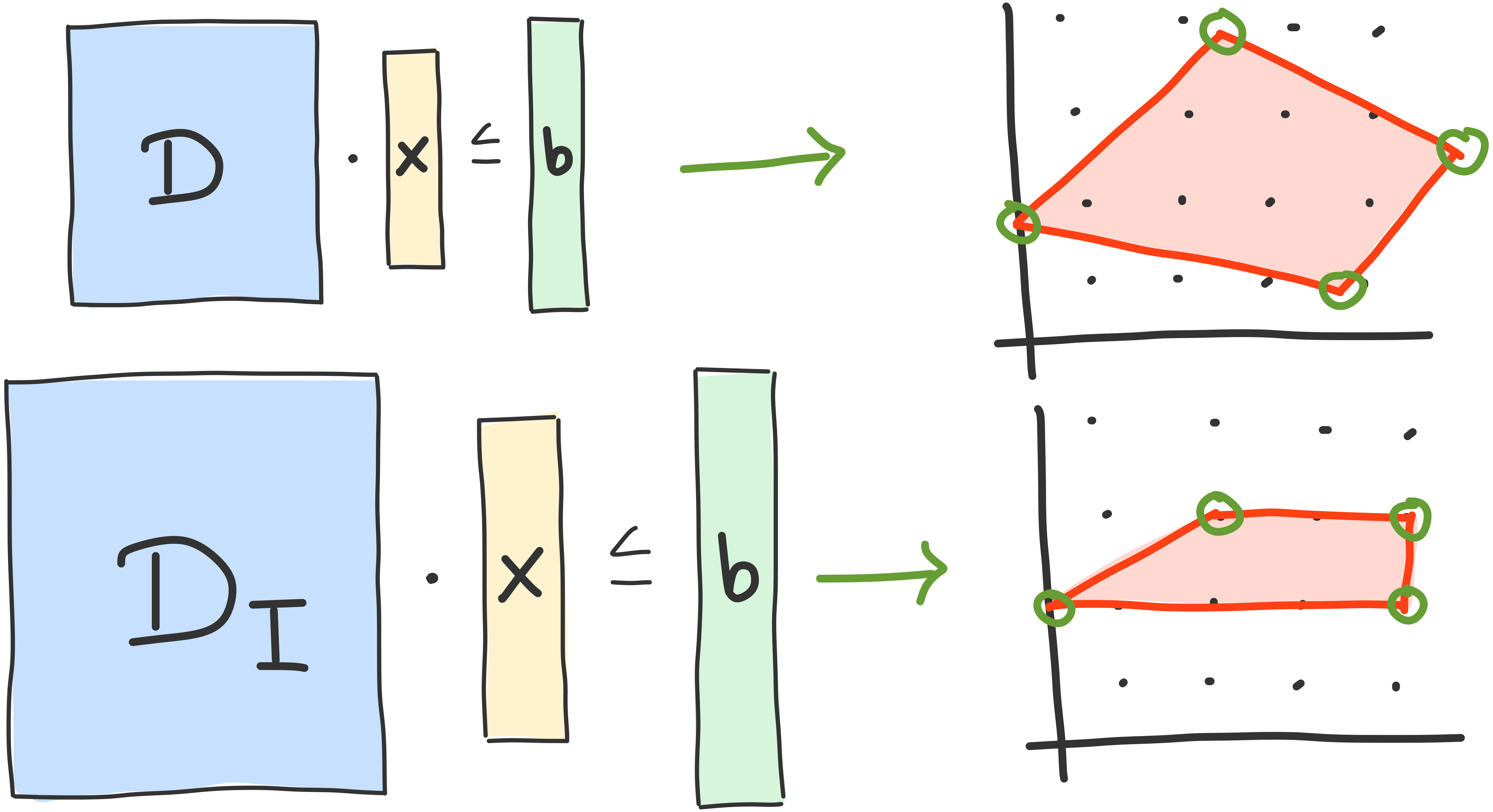




The Integer Hull



The Integer Hull



2-stage stochastic IPs

Theorem 1.* *Let $D \in \mathbb{Z}^{m\times k}$ with non-repeating rows and $\|A\|_\infty \leq \Delta$. There exists a $R \in \mathbb{N}$ depending on k and Δ such that, for each $r \in \{0,\dots,R-1\}^m$, there exist $B_r \in \mathbb{Z}^{m'\times k}$, $C_r \in \mathbb{Z}^{m'\times m}$ and $f_r \in \mathbb{Z}^{m'}$ such that the following holds:*

For each $b \in \mathbb{Z}^m$ with $b - r \in R \cdot \mathbb{Z}^m$, one has

$$\underbrace{P(b)_I}_{\text{integer hull}} = \{x \in \mathbb{R}^k \colon B_r x \leq f_r + C_r b\}\,.$$

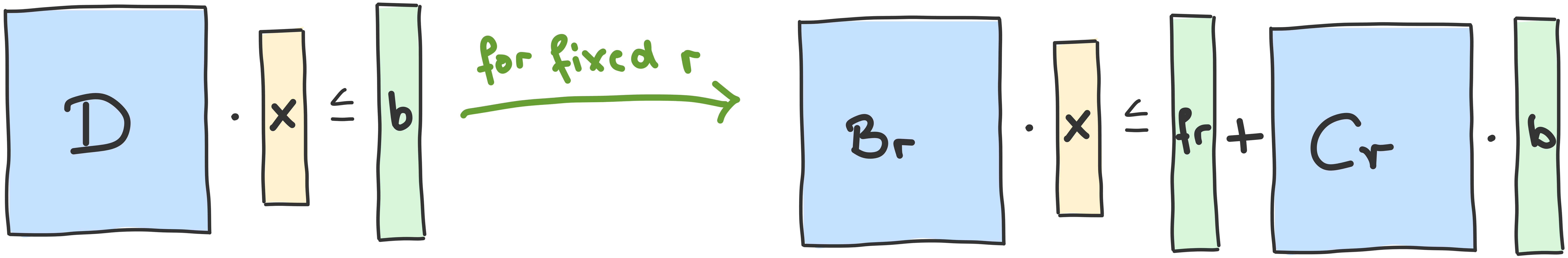
* "A parameterized linear formulation of the integer hull" by F. Eisenbrand and T. Rothvoss, 2025

2-stage stochastic IPs

Theorem 1.* *Let $D \in \mathbb{Z}^{m\times k}$ with non-repeating rows and $\|A\|_\infty \le \Delta$. There exists a $R \in \mathbb{N}$ depending on k and Δ such that, for each $r \in \{0,\dots,R-1\}^m$, there exist $B_r \in \mathbb{Z}^{m'\times k}$, $C_r \in \mathbb{Z}^{m'\times m}$ and $f_r \in \mathbb{Z}^{m'}$ such that the following holds:*

For each $b \in \mathbb{Z}^m$ with $b - r \in R \cdot \mathbb{Z}^m$, one has

$$P(b)_I = \{x \in \mathbb{R}^k \colon B_r x \le f_r + C_r b\}.$$



* “A parameterized linear formulation of the integer hull” by F. Eisenbrand and T. Rothvoss, 2025

2-stage stochastic IPs

Theorem 1.* *Let* $D \in \mathbb{Z}^{m \times k}$ *with non-repeating rows and* $\|A\|_\infty \leq \Delta$*. There exists a* $R \in \mathbb{N}$ *depending on* k *and* Δ *such that, for each* $r \in \{0, \ldots, R-1\}^m$*, there exist* $B_r \in \mathbb{Z}^{m' \times k}$*,* $C_r \in \mathbb{Z}^{m' \times m}$ *and* $f_r \in \mathbb{Z}^{m'}$ *such that the following holds:*

For each $b \in \mathbb{Z}^m$ *with* $b - r \in R \cdot \mathbb{Z}^m$*, one has*

$$P(b)_I = \{x \in \mathbb{R}^k \colon B_r x \leq f_r + C_r b\} .$$

There is an fpt-sized description of the integer hull of $Dx \leq b$ (for fixed r)

* "A parameterized linear formulation of the integer hull" by F. Eisenbrand and T. Rothvoss, 2025

2-stage stochastic IPs

Theorem 1. *Let* $D \in \mathbb{Z}^{m \times k}$ *with non-repeating rows and* $\|A\|_\infty \leq \Delta$. *There exists a* $R \in \mathbb{N}$ *depending on* k *and* Δ *such that, for each* $r \in \{0, \ldots, R-1\}^m$, *there exist* $B_r \in \mathbb{Z}^{m' \times k}$, $C_r \in \mathbb{Z}^{m' \times m}$ *and* $f_r \in \mathbb{Z}^{m'}$ *such that the following holds:*

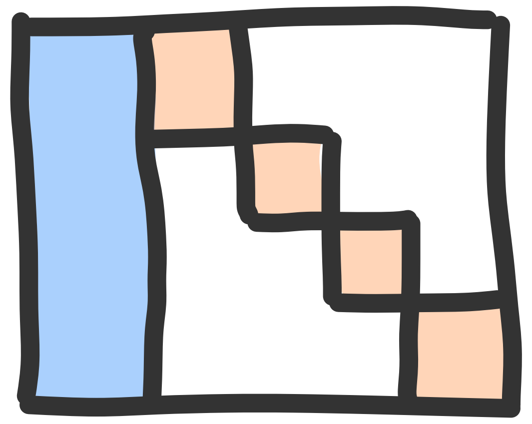
For each $b \in \mathbb{Z}^m$ *with* $b - r \in R \cdot \mathbb{Z}^m$, *one has*

$$P(b)_I = \{x \in \mathbb{R}^k \colon B_r x \leq f_r + C_r b\} .$$

Furthermore, the number $R \in \mathbb{N}$, *the matrices* B_r *and* C_r, *as well as the vector* f_r *can be computed in time depending on* Δ *and* k *only.*

"A parameterized linear formulation of the integer hull" by F. Eisenbrand and T. Rothvoss, 2025

The Algorithm



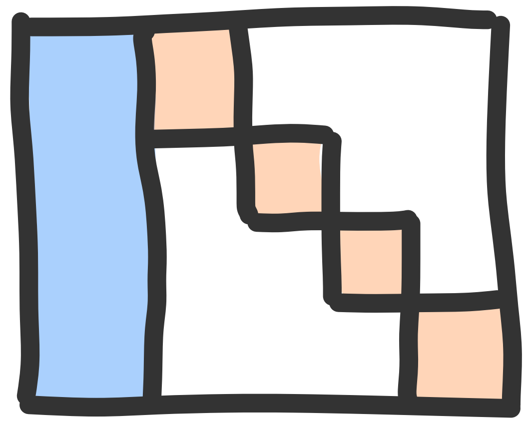
1. Compute R and guess remainder r for all D_i

The Algorithm

1. Compute R and guess remainder r for all D_i

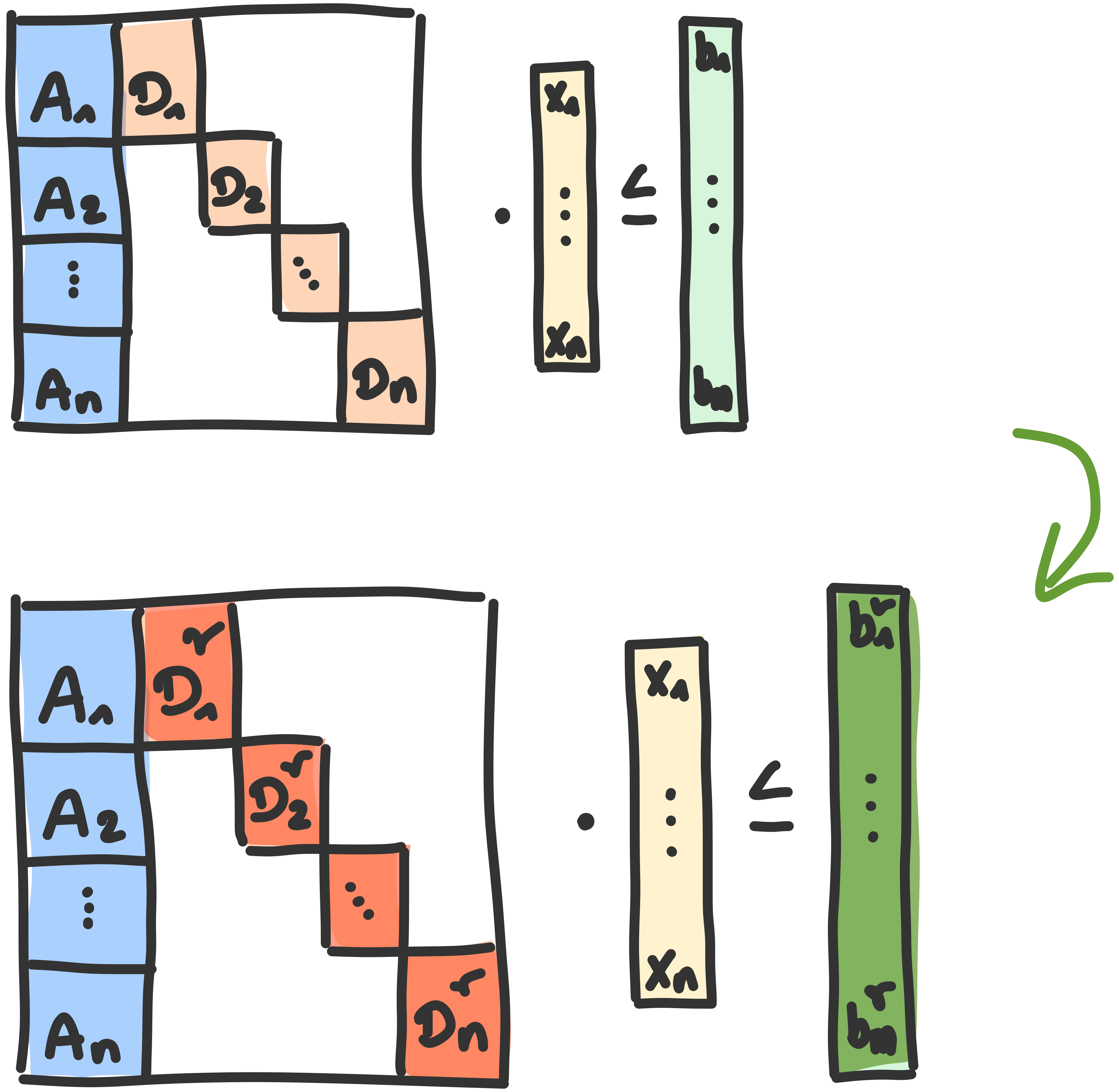
2. Compute integer hull description of each D_i and replace them

The Algorithm

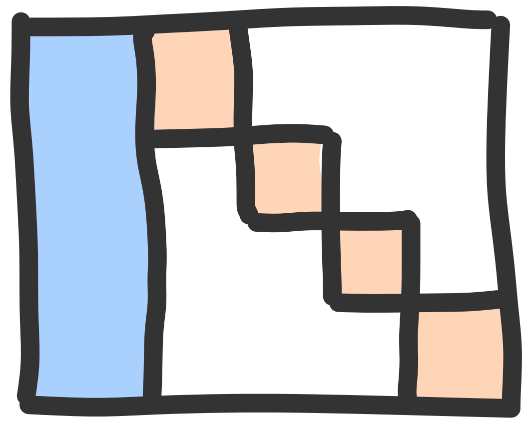


1. Compute R and guess remainder r for all D_i

2. Compute integer hull description of each D_i and replace them



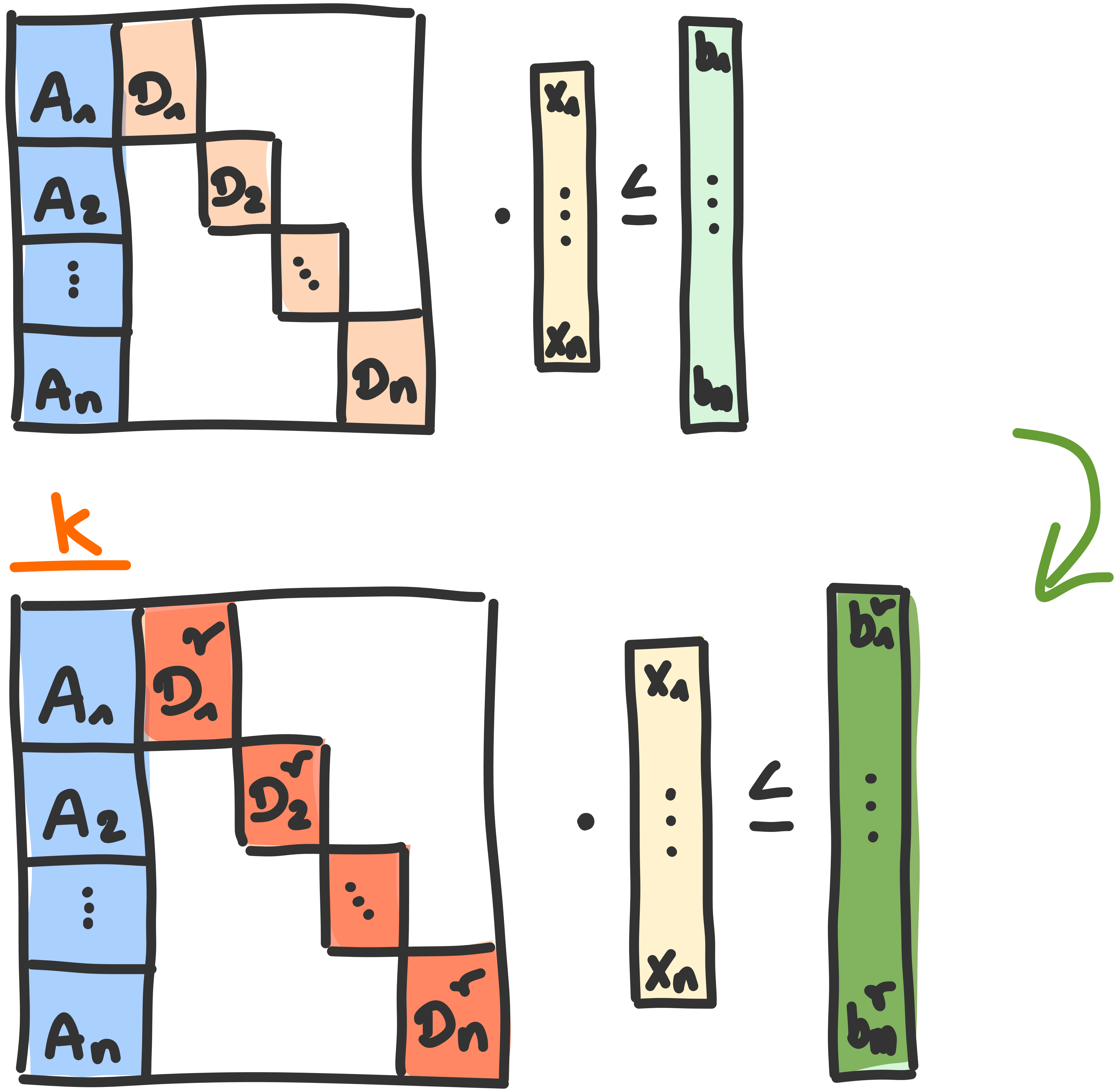
The Algorithm

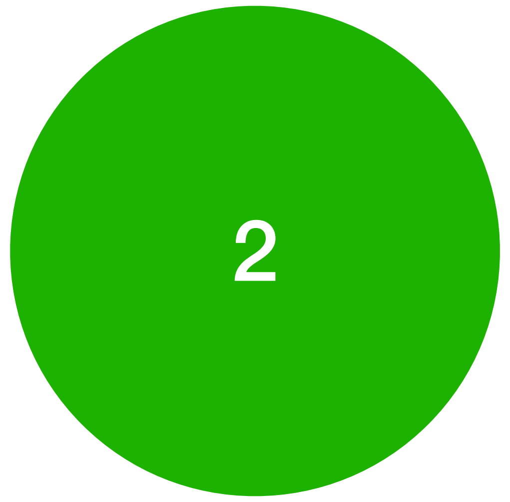


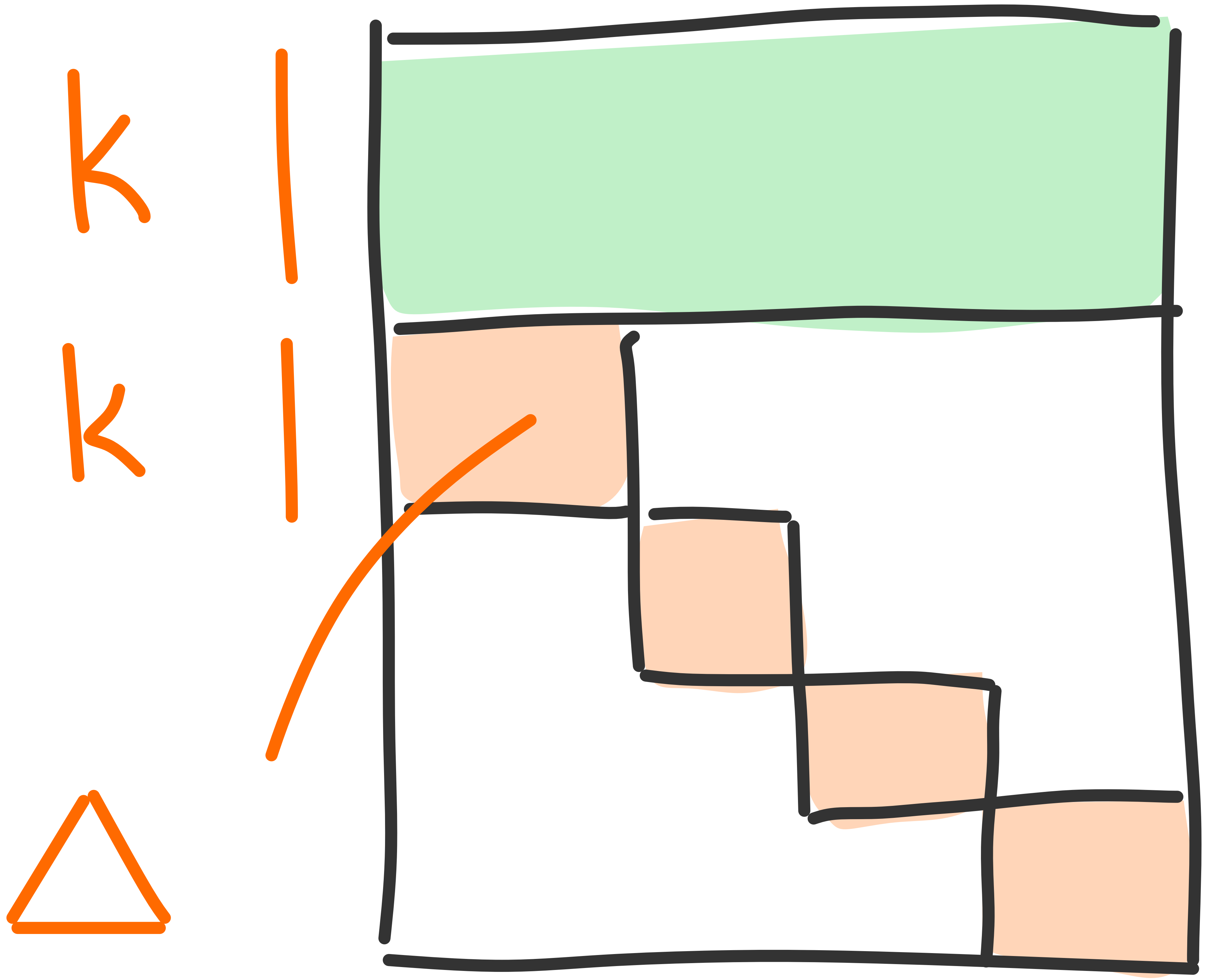
1. Compute R and guess remainder r for all D_i

2. Compute integer hull description of each D_i and replace them

3. Solve the MIP

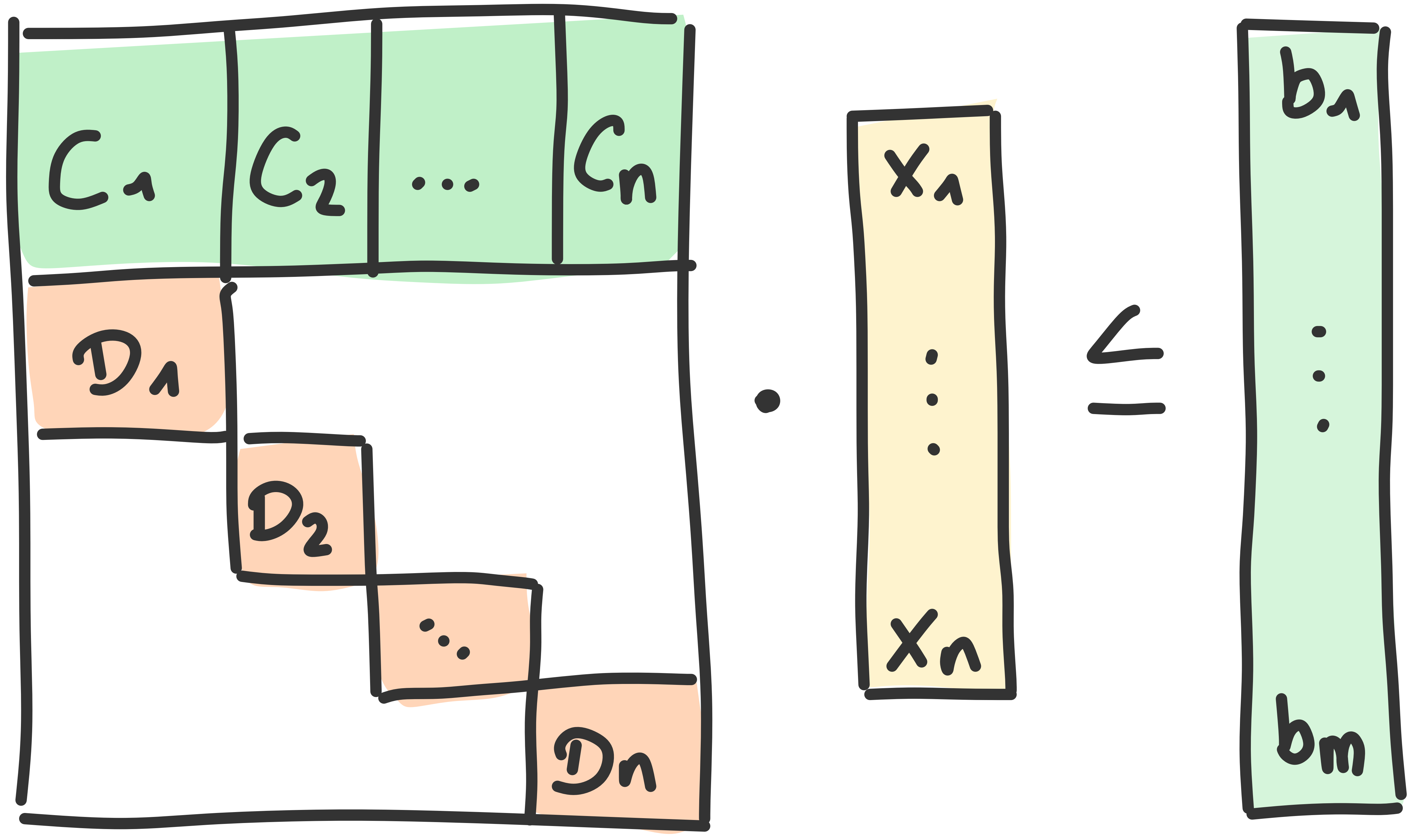




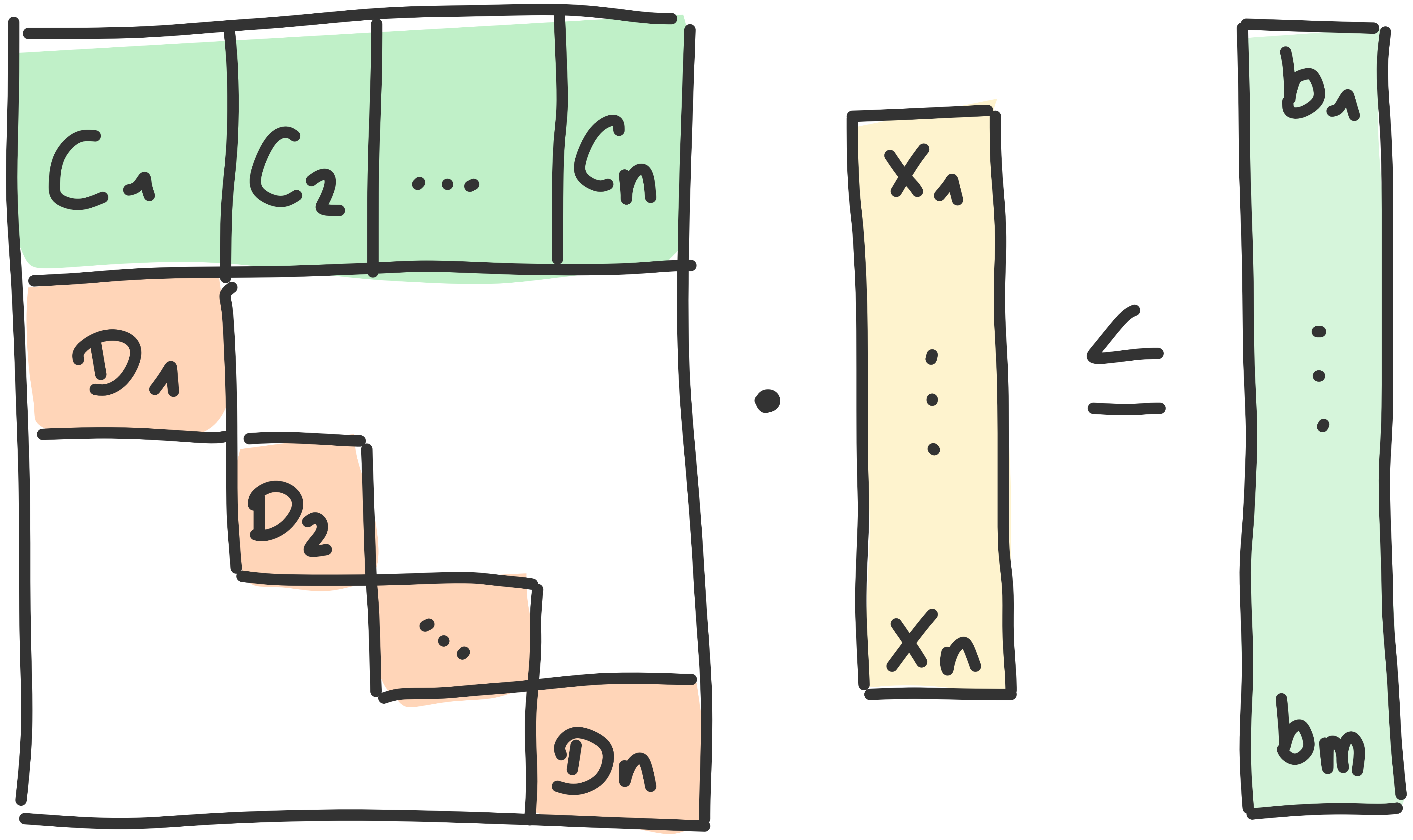


n-fold IPs

n-fold IPs

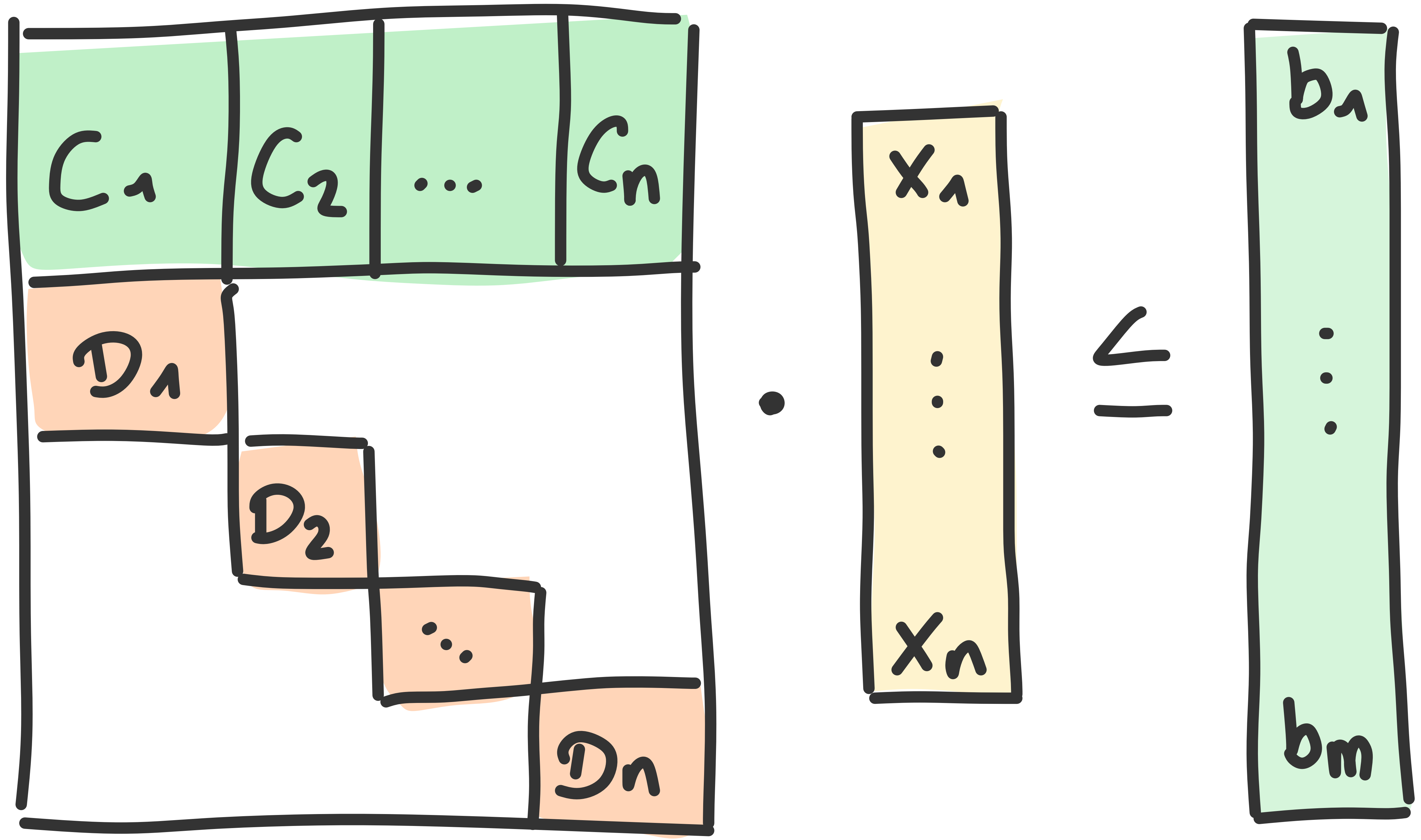


n-fold IPs

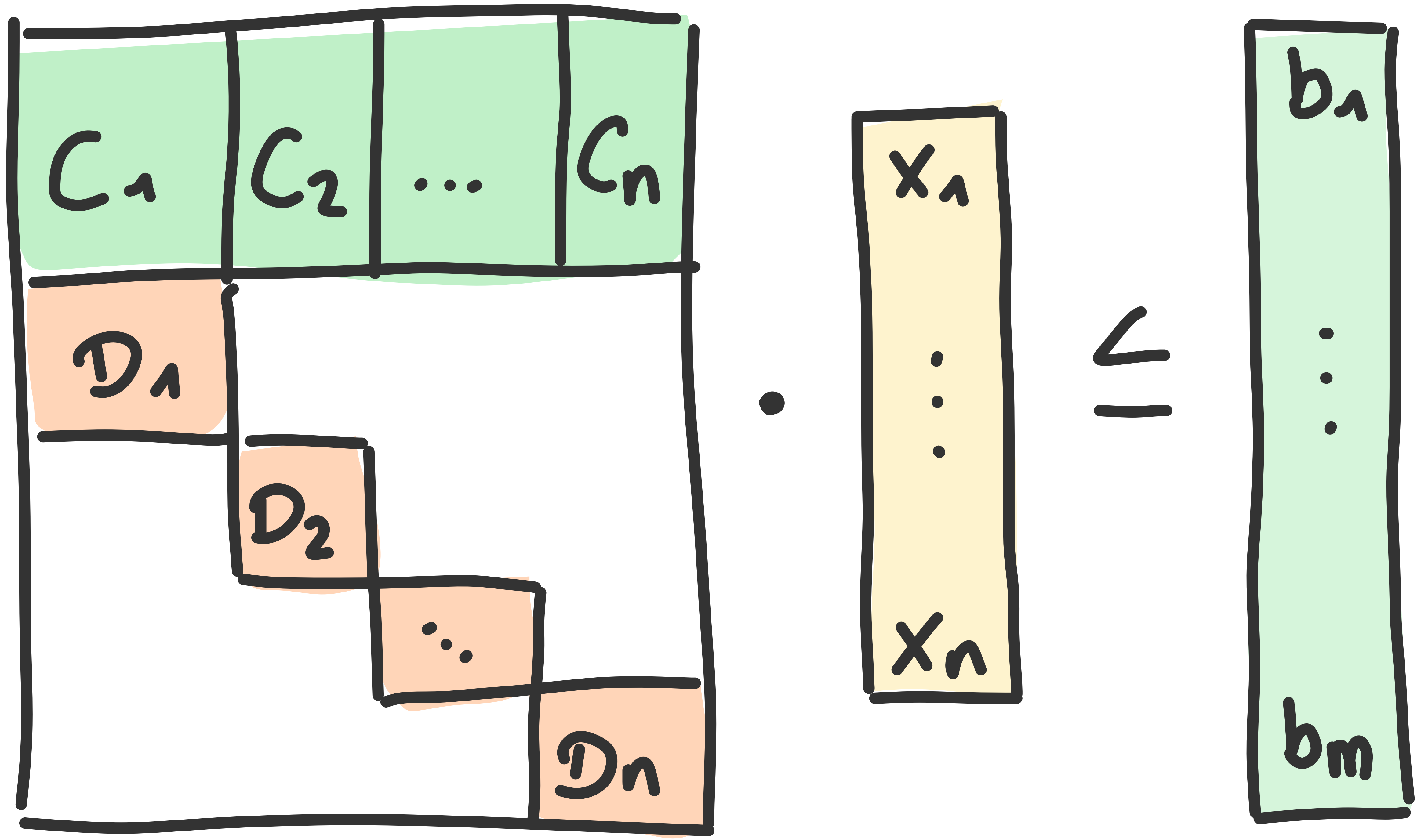


scheduling, knapsack-like problems, string problems, social choice, …

n-fold IPs

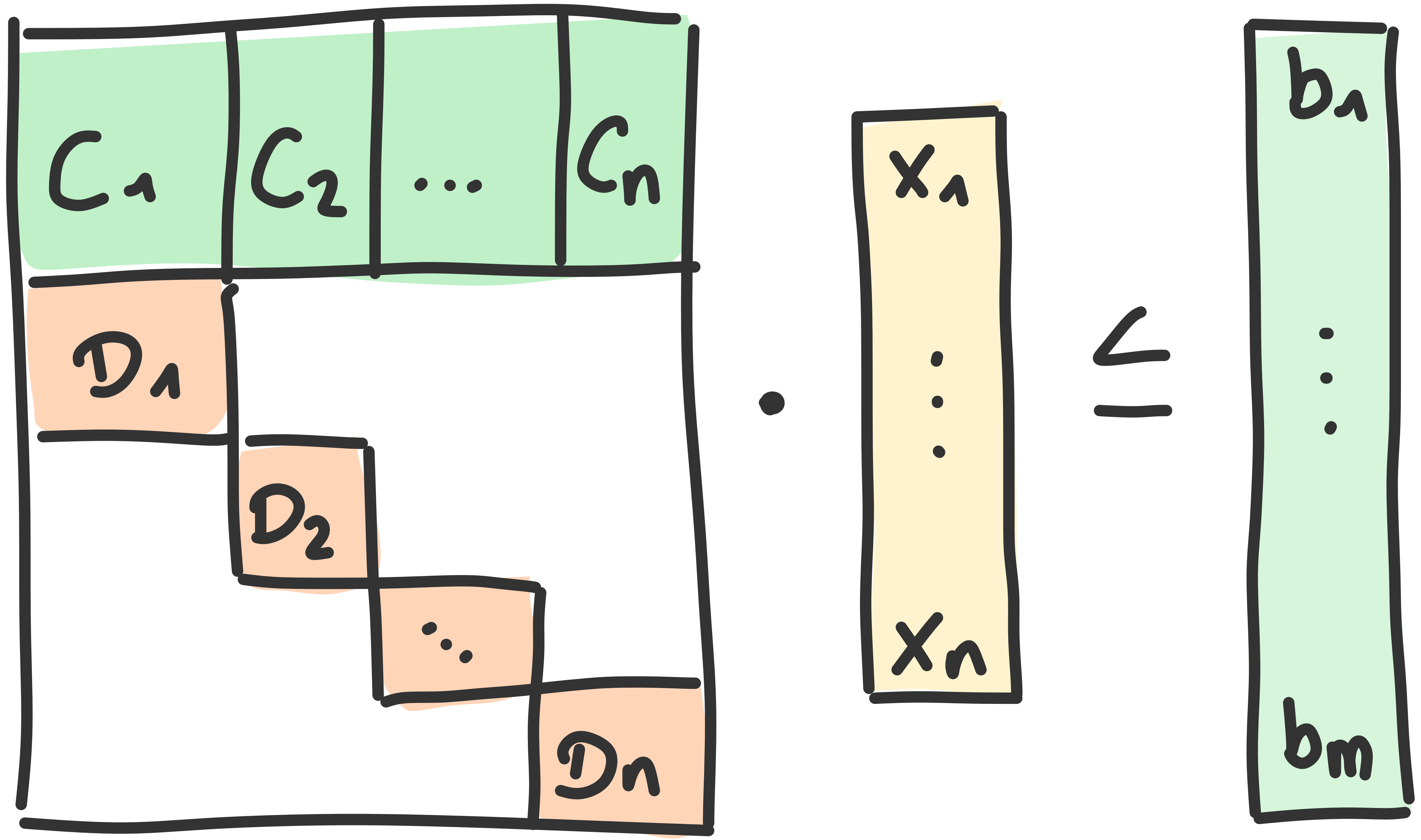


n-fold IPs



If dimension is small, we are done

n-fold IPs



If dimension is small, we are done

If it is large, there are only few types of diagonal blocks

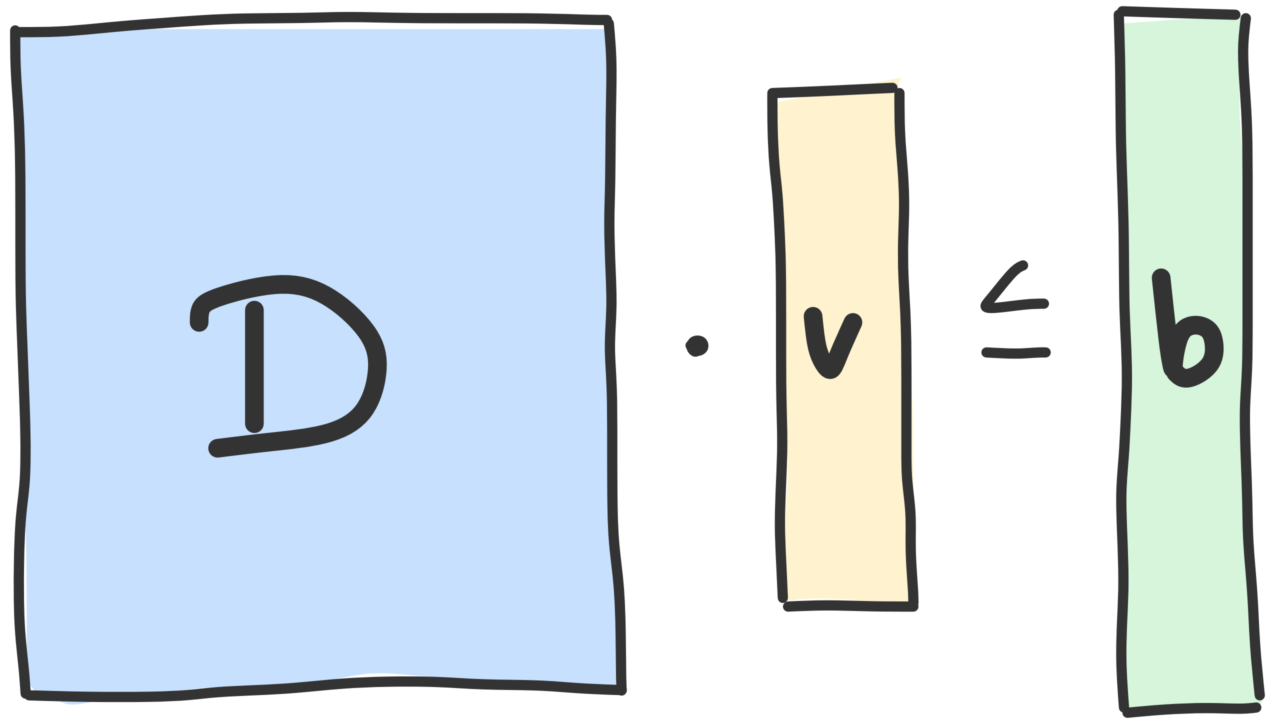
n-fold IPs

$\exists b', b''$ **sign-compatible to** b **s.t.** $b = b' + b''$ **and** $\forall v \in \mathbb{Z}^y_{\geq 0}$ **with**

$D_i v = b$ **there exist** v', v'' **with 1.** $v = v' + v''$ **2.** $D_i v' = b'$ **3.** $D_i v'' = b''$

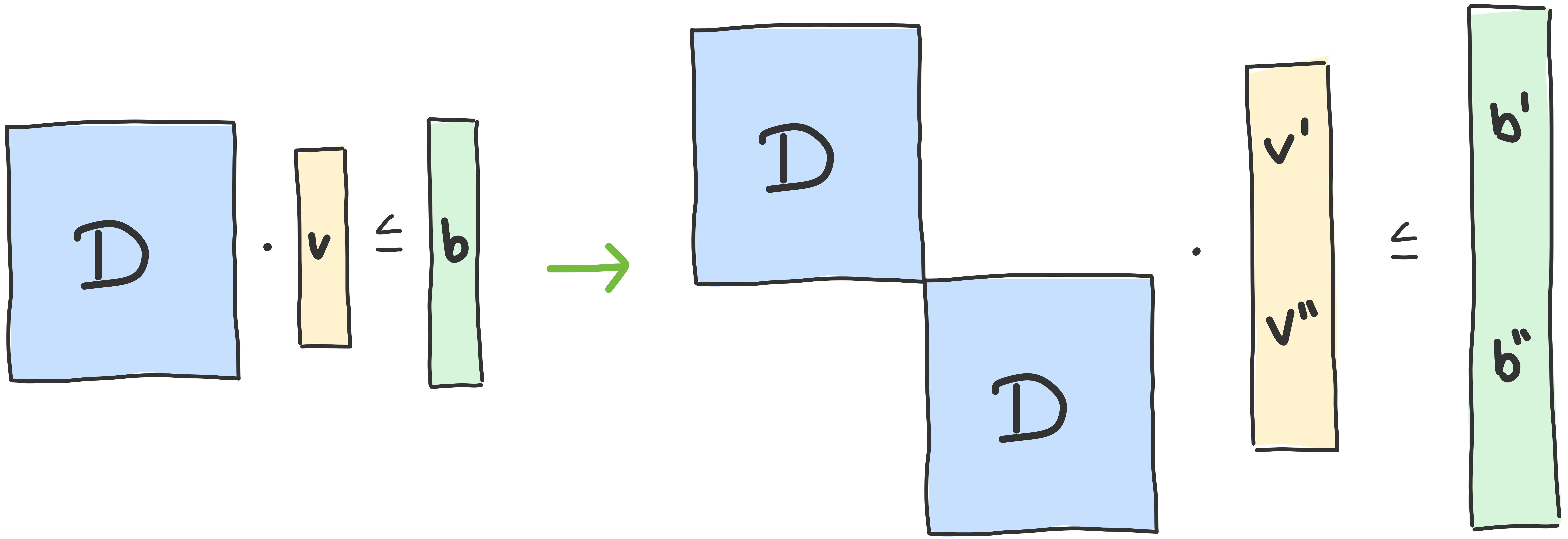
n-fold IPs

$\exists b', b''$ **sign-compatible to** b **s.t.** $b = b' + b''$ **and** $\forall v \in \mathbb{Z}^y_{\geq 0}$ **with** $D_i v = b$ **there exist** v', v'' **with 1.** $v = v' + v''$ **2.** $D_i v' = b'$ **3.** $D_i v'' = b''$



n-fold IPs

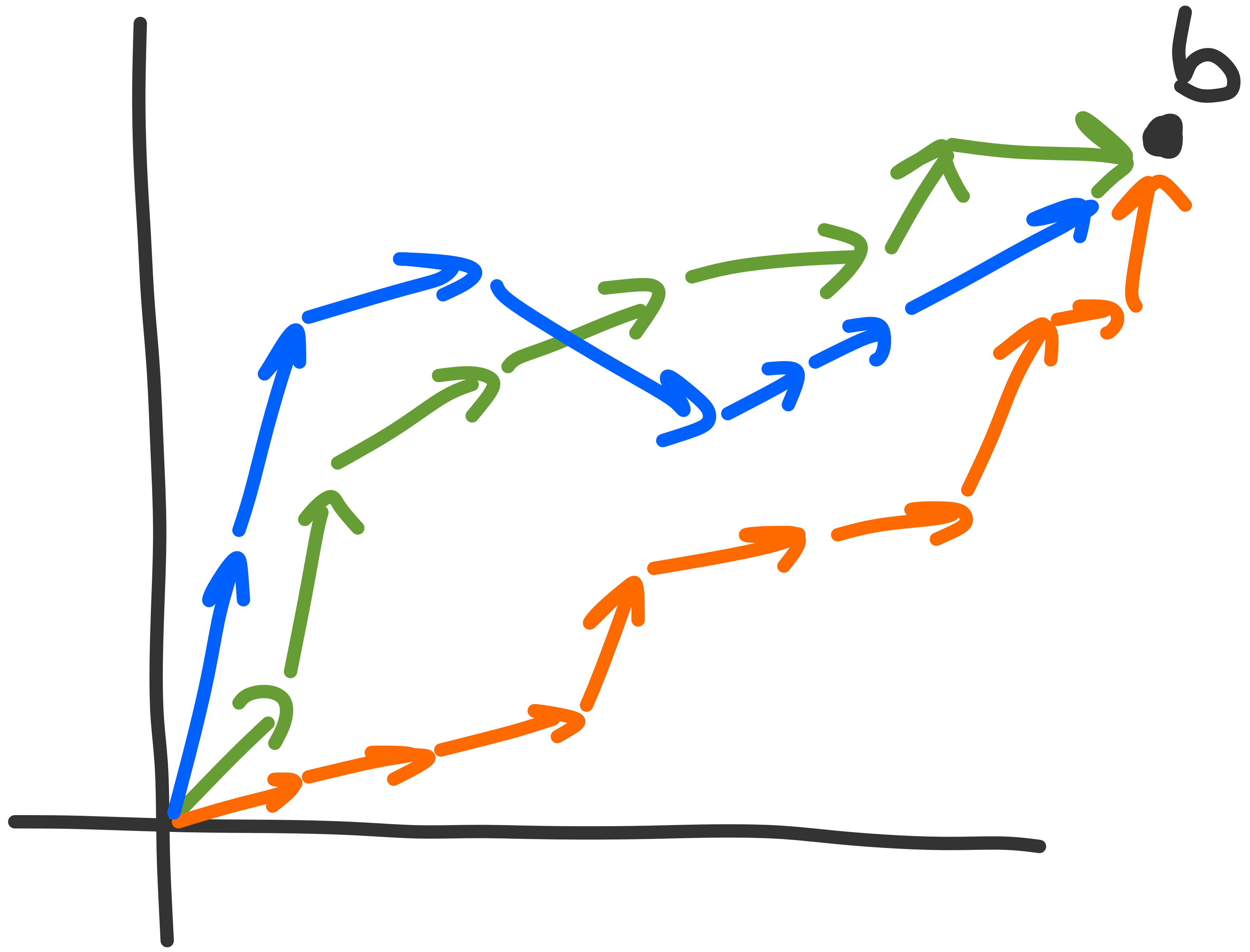
$\exists b', b''$ sign-compatible to b s.t. $b = b' + b''$ and $\forall v \in \mathbb{Z}^{y}_{\geq 0}$ with $D_i v = b$ there exist v', v'' with 1. $v = v' + v''$ 2. $D_i v' = b'$ 3. $D_i v'' = b''$



n-fold IPs - based on Klein Lemma

$\exists b', b''$ **sign-compatible to** b **s.t.** $b = b' + b''$ **and** $\forall v \in \mathbb{Z}^y_{\geq 0}$ **with**

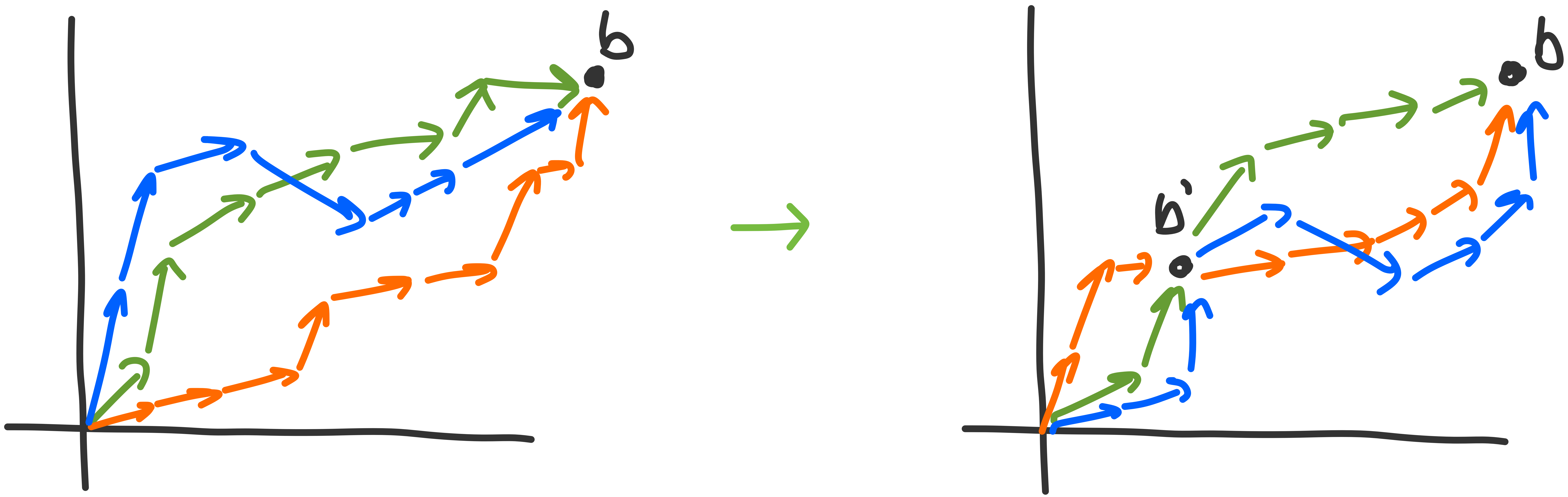
$D_i v = b$ **there exist** v', v'' **with 1.** $v = v' + v''$ **2.** $D_i v' = b'$ **3.** $D_i v'' = b''$



n-fold IPs - based on Klein Lemma

$\exists b', b''$ **sign-compatible to** b **s.t.** $b = b' + b''$ **and** $\forall v \in \mathbb{Z}^y_{\geq 0}$ **with**

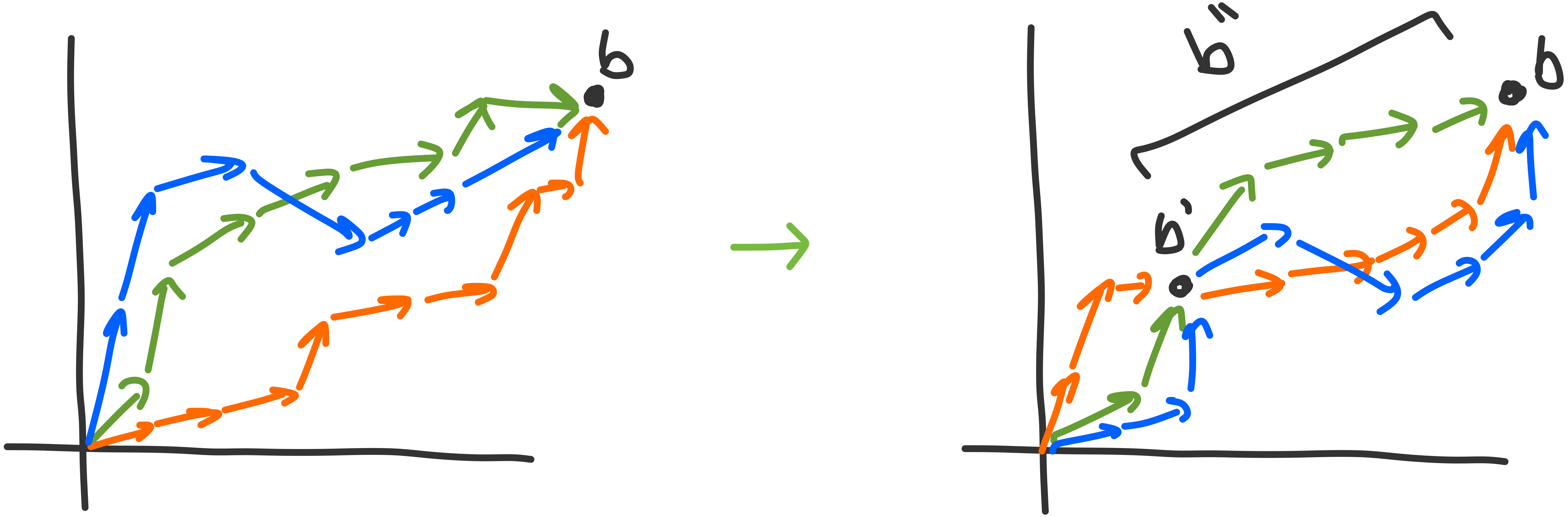
$D_i v = b$ **there exist** v', v'' **with 1.** $v = v' + v''$ **2.** $D_i v' = b'$ **3.** $D_i v'' = b''$



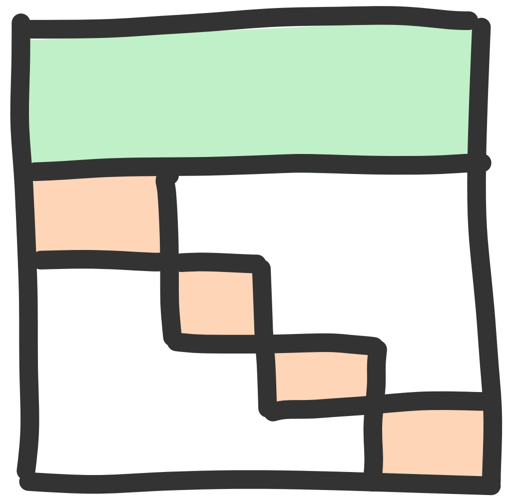
n-fold IPs - based on Klein Lemma

$\exists b', b''$ **sign-compatible to** b **s.t.** $b = b' + b''$ **and** $\forall v \in \mathbb{Z}^y_{\geq 0}$ **with**

$D_i v = b$ **there exist** v', v'' **with 1.** $v = v' + v''$ **2.** $D_i v' = b'$ **3.** $D_i v'' = b''$

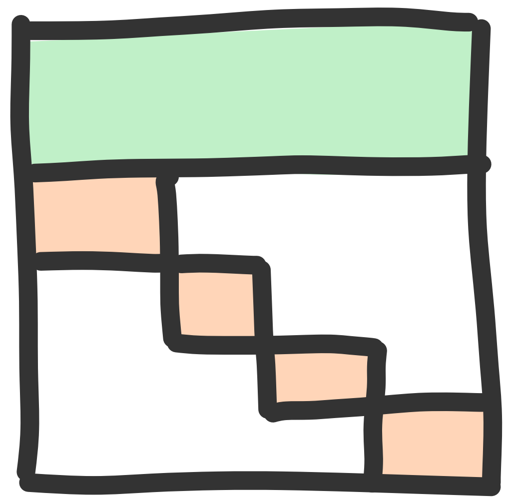


The Algorithm

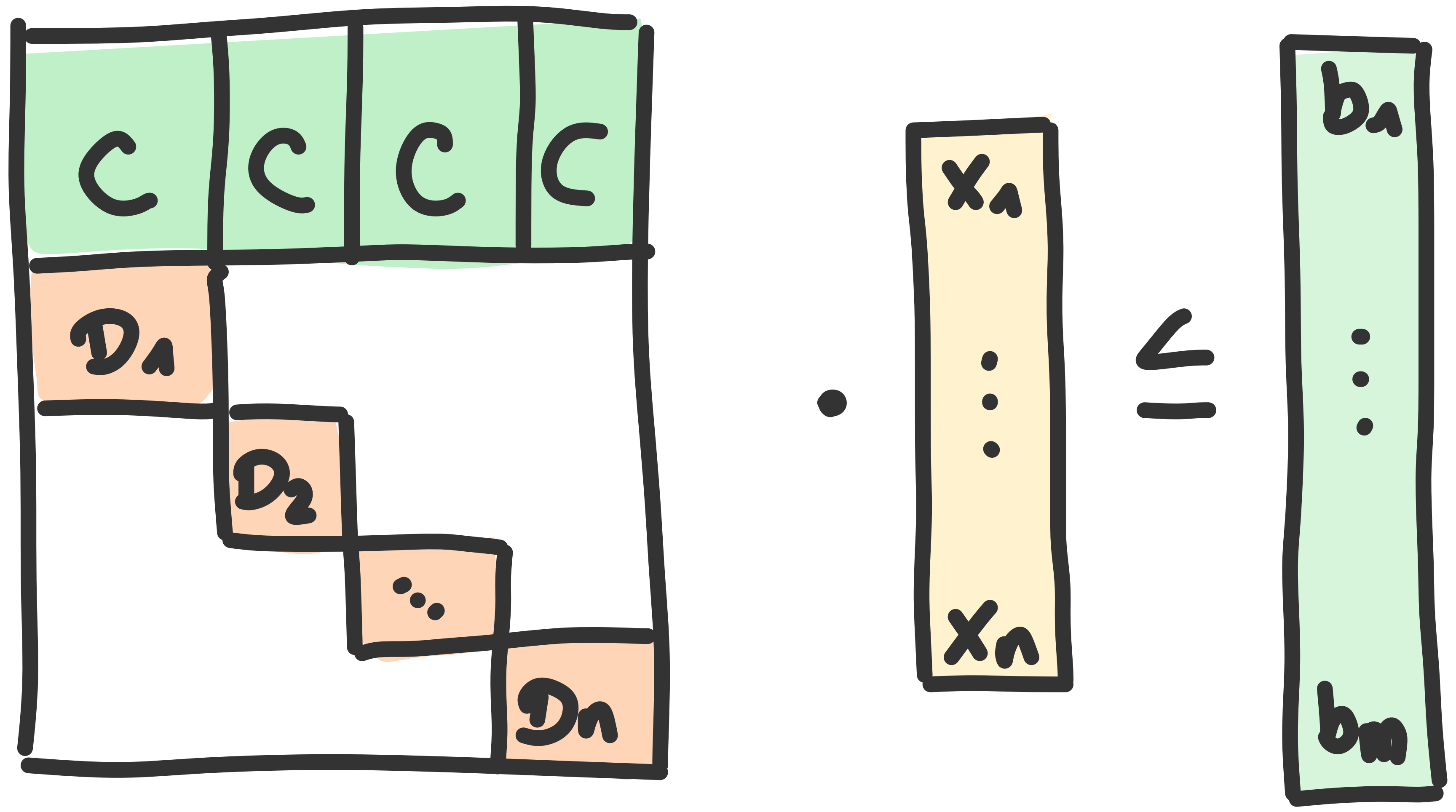


1. Breaking the bricks

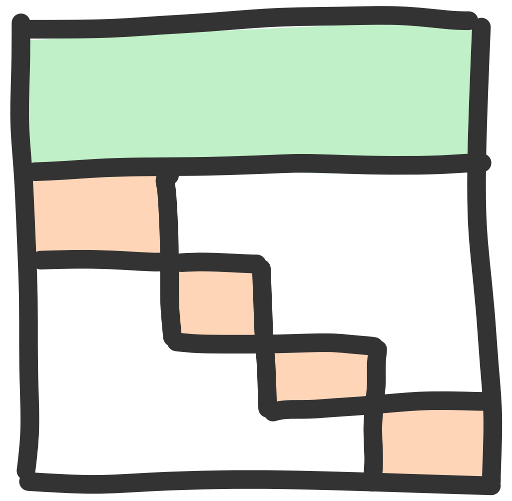
The Algorithm



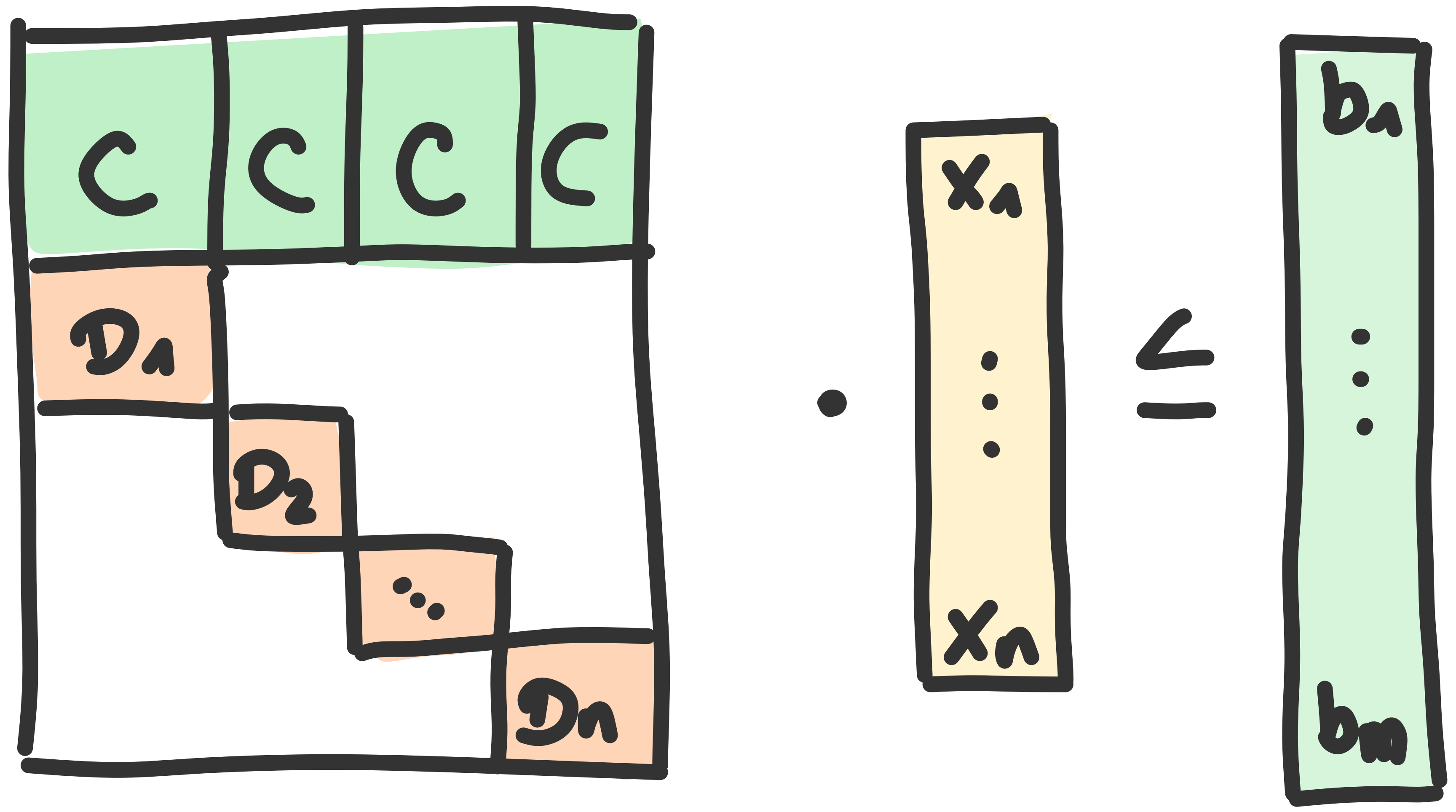
1. Breaking the bricks

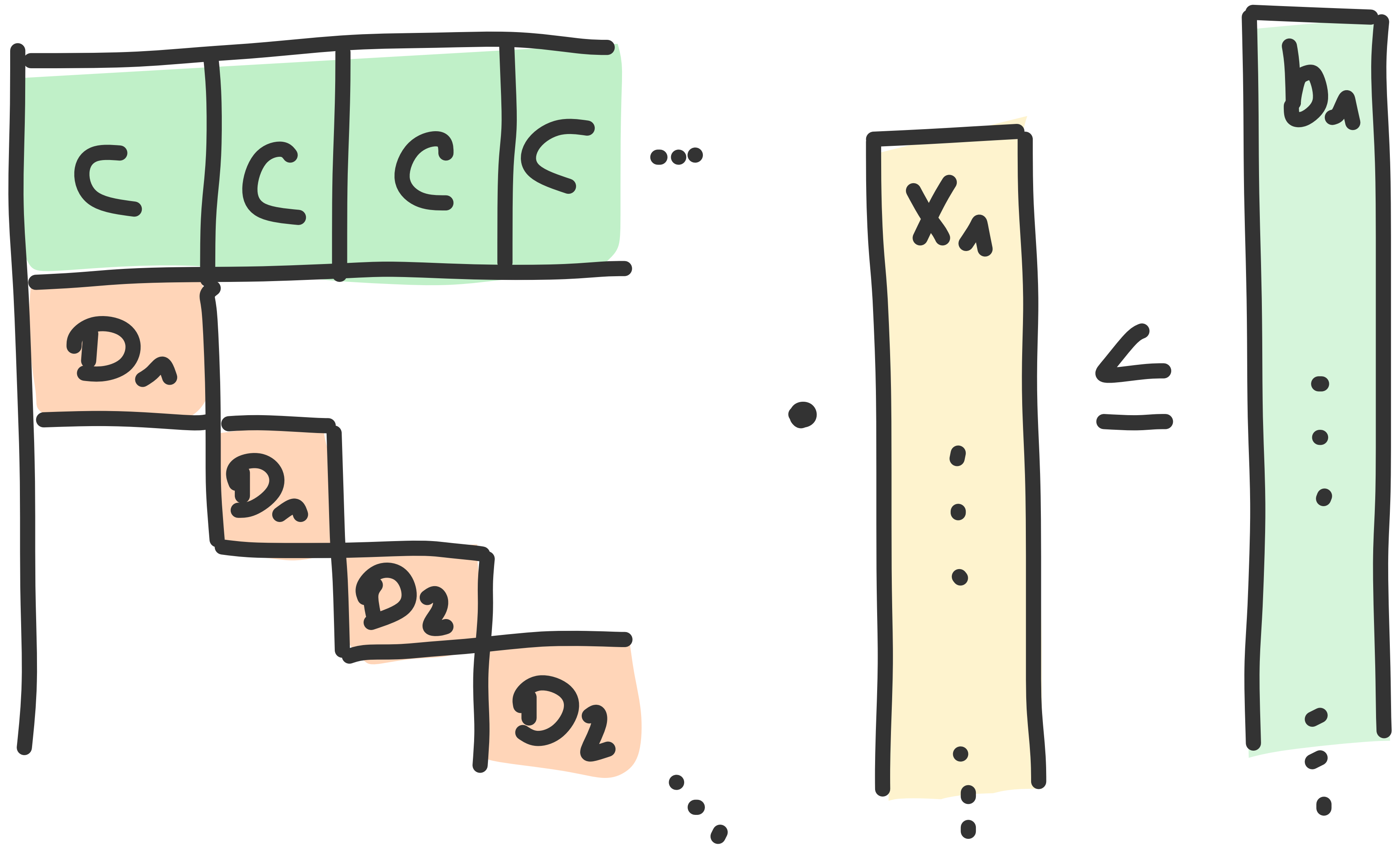


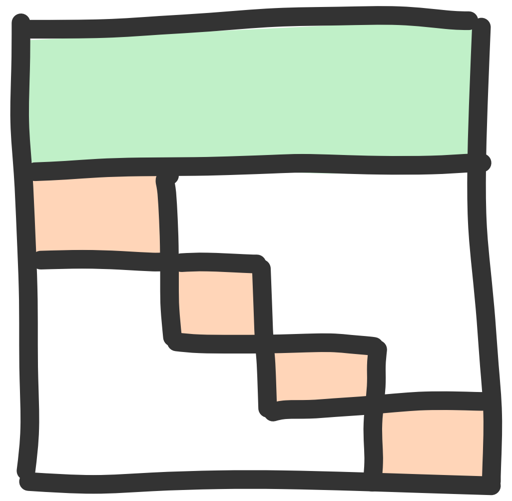
The Algorithm



1. Breaking the bricks

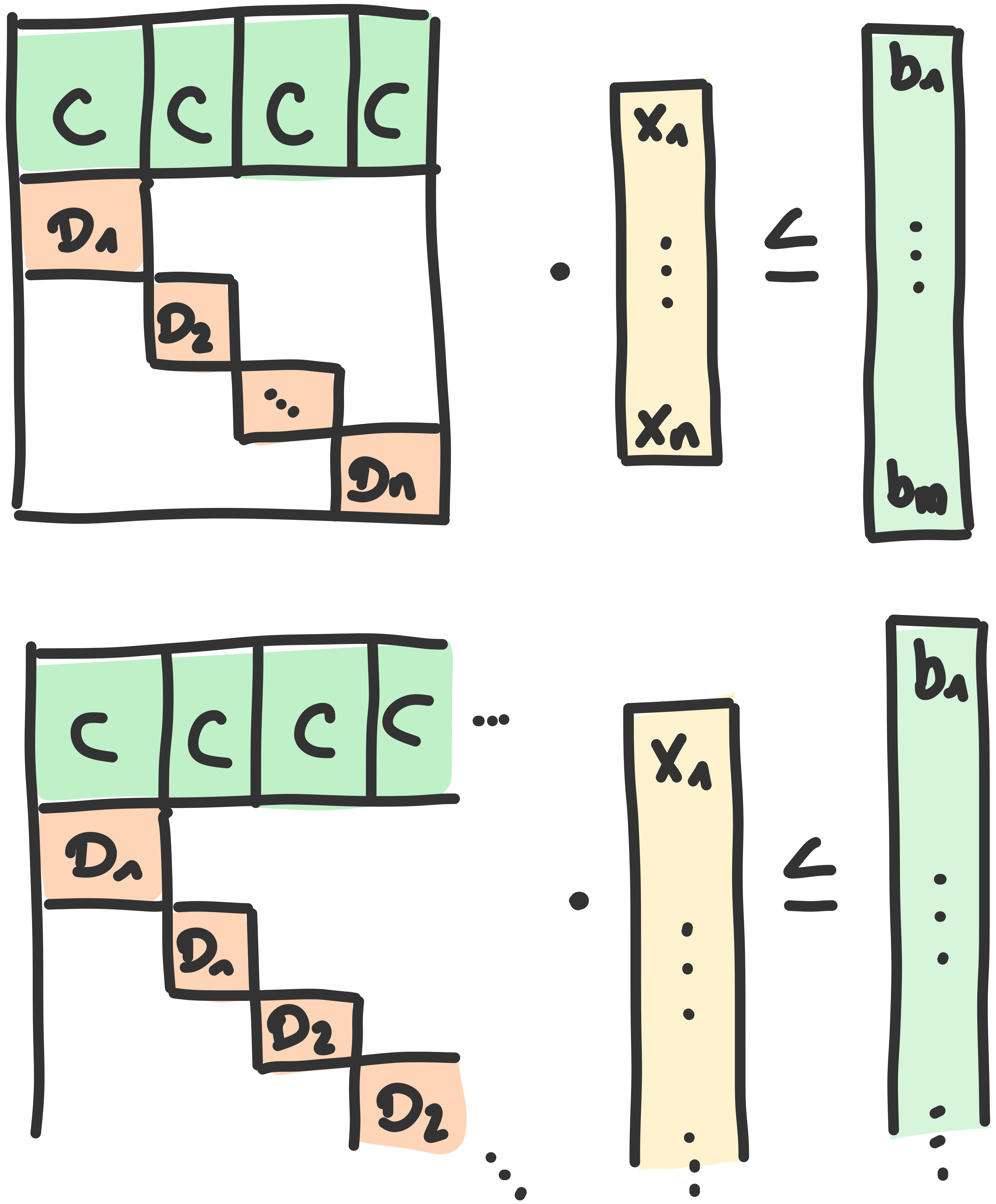




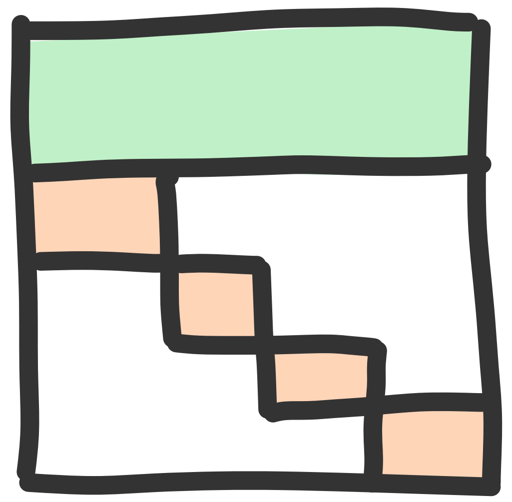


The Algorithm

1. Breaking the bricks

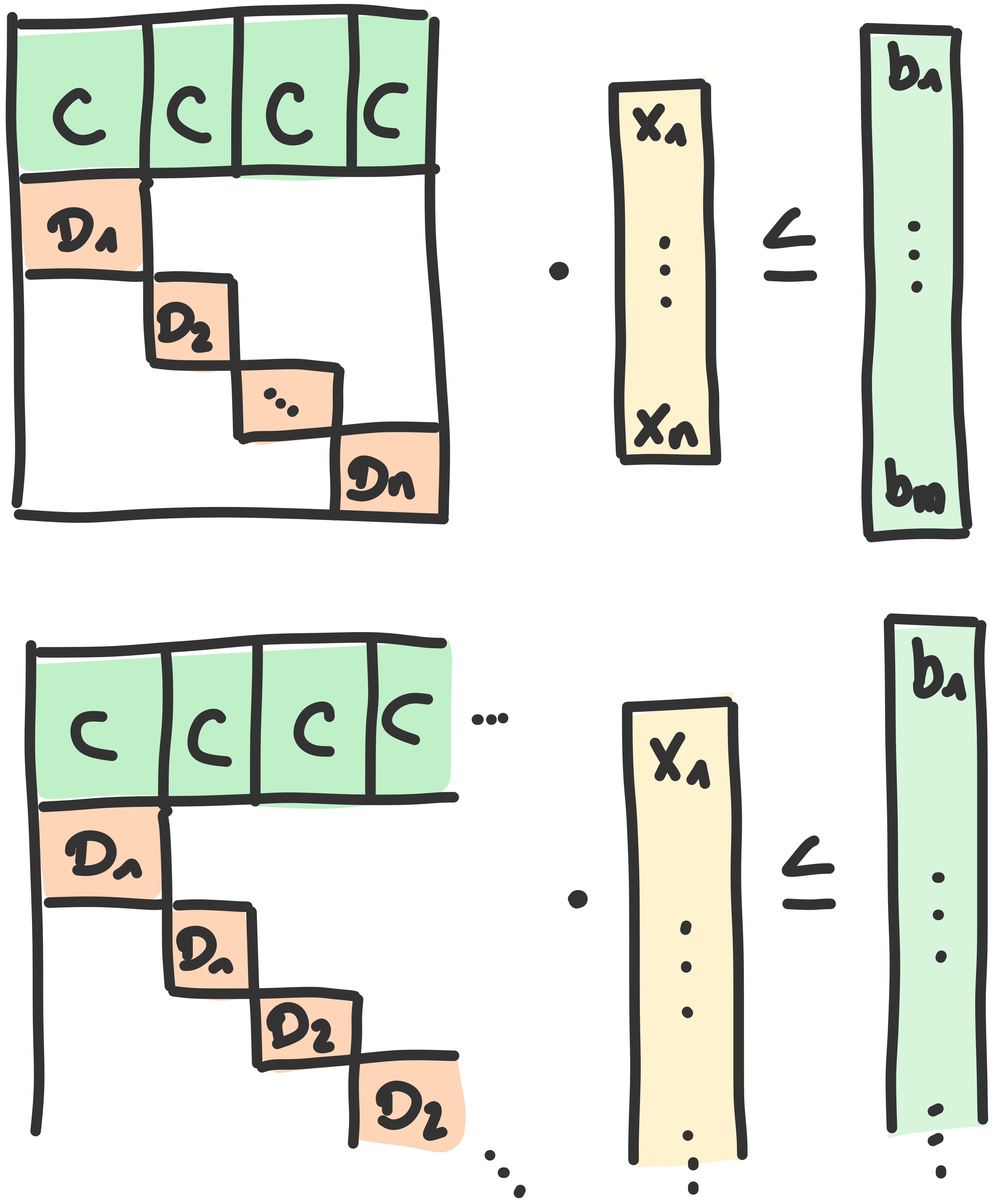


The Algorithm

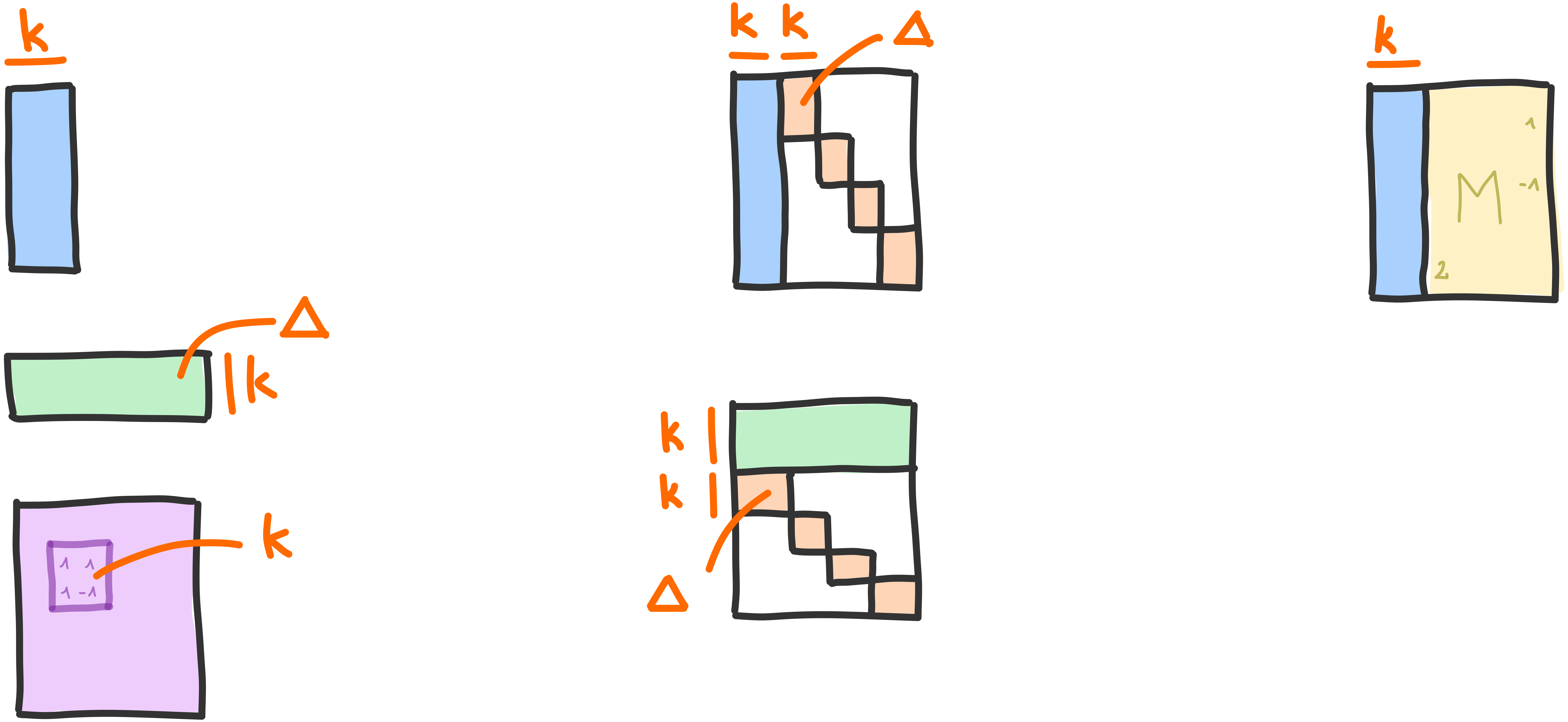


1. Breaking the bricks

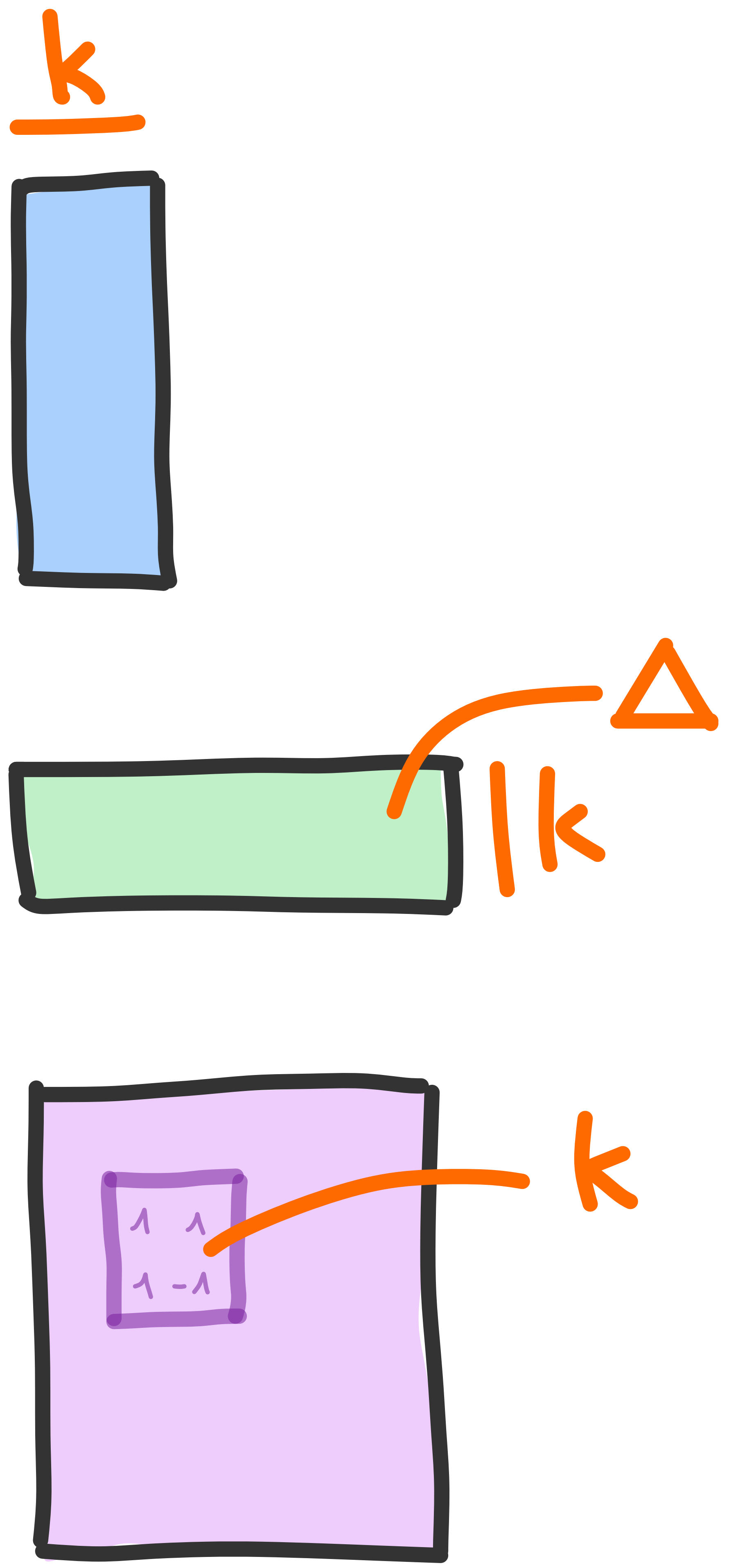
2. We can deal easily with the high-multiplicity instance :)

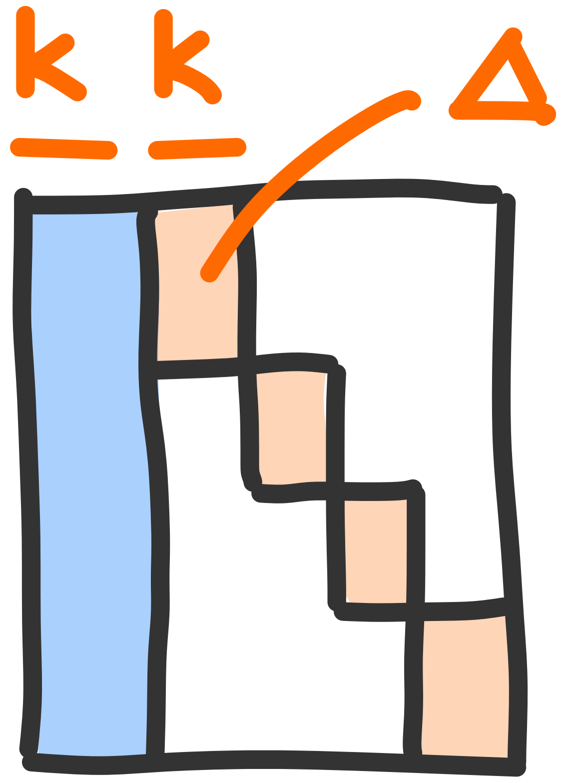


Integer Programming meets FPT

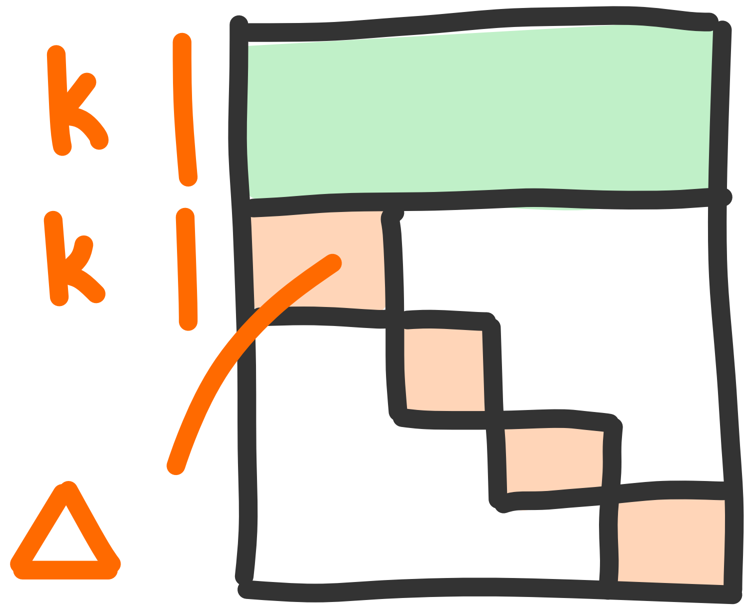


Integer Programming meets FPT

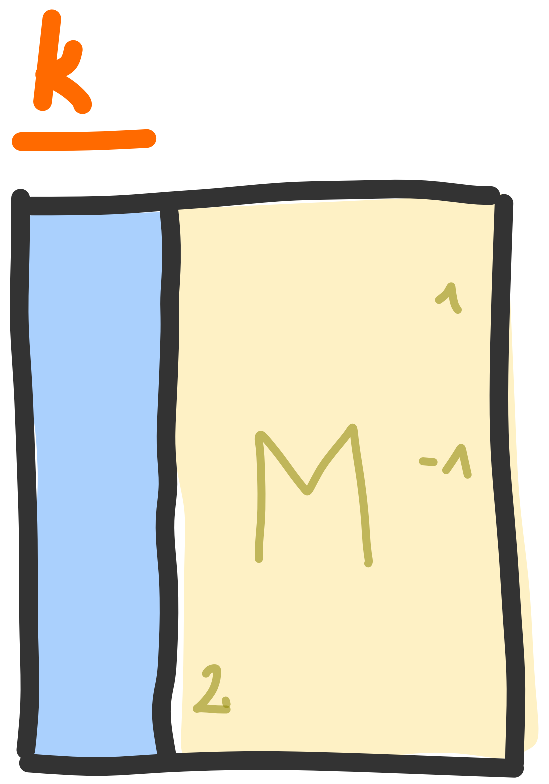




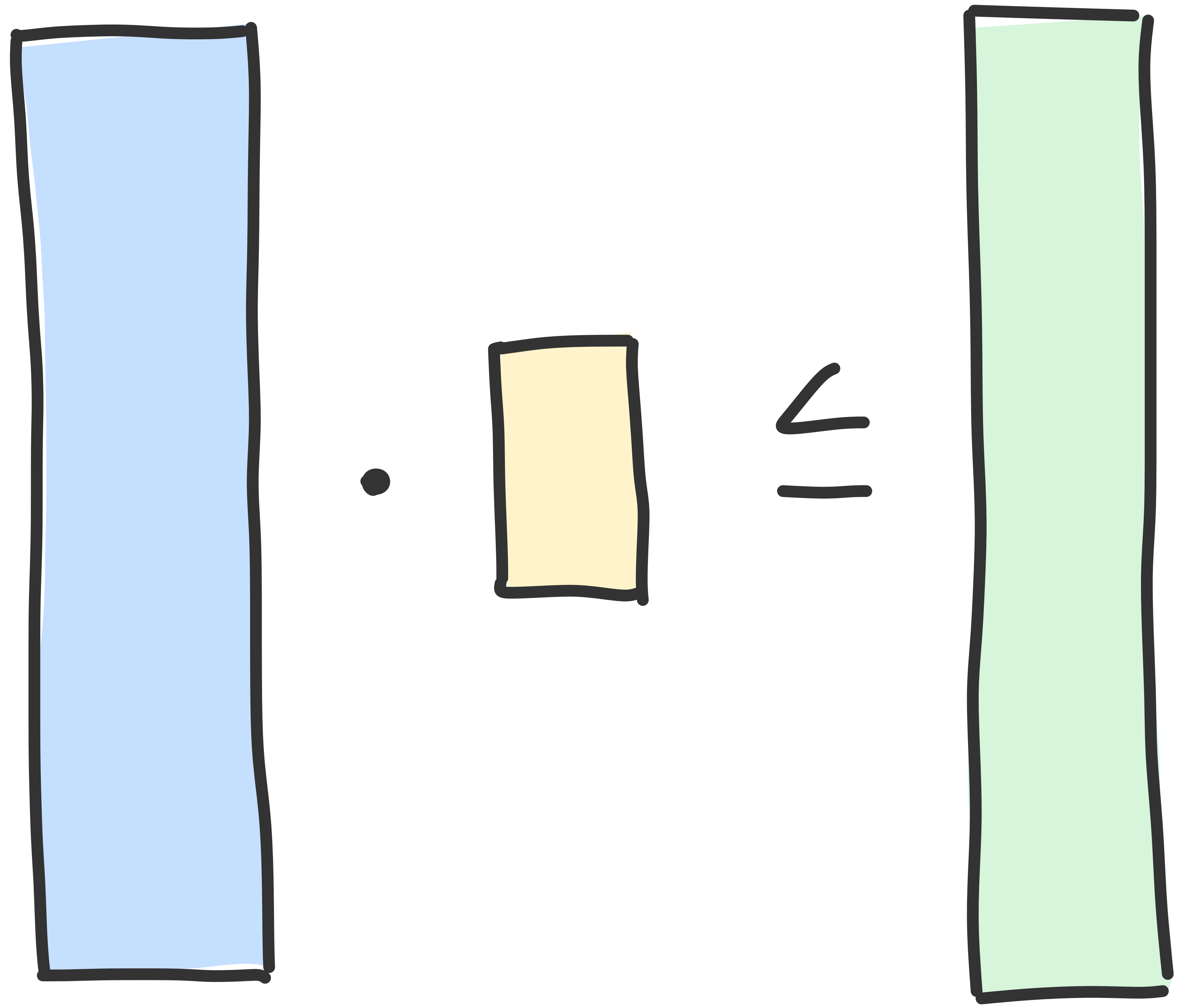
Integer
Hull



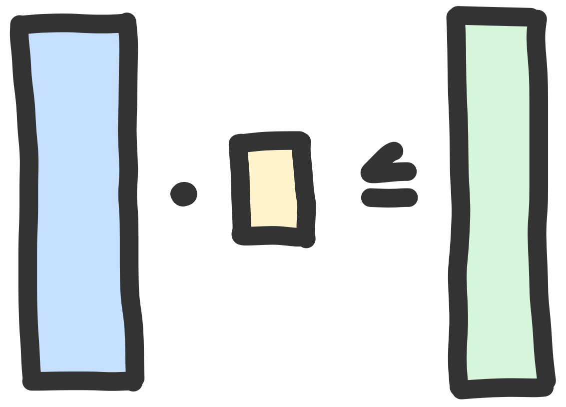
Brick
Decomposition



Lenstra-type Integer Programs



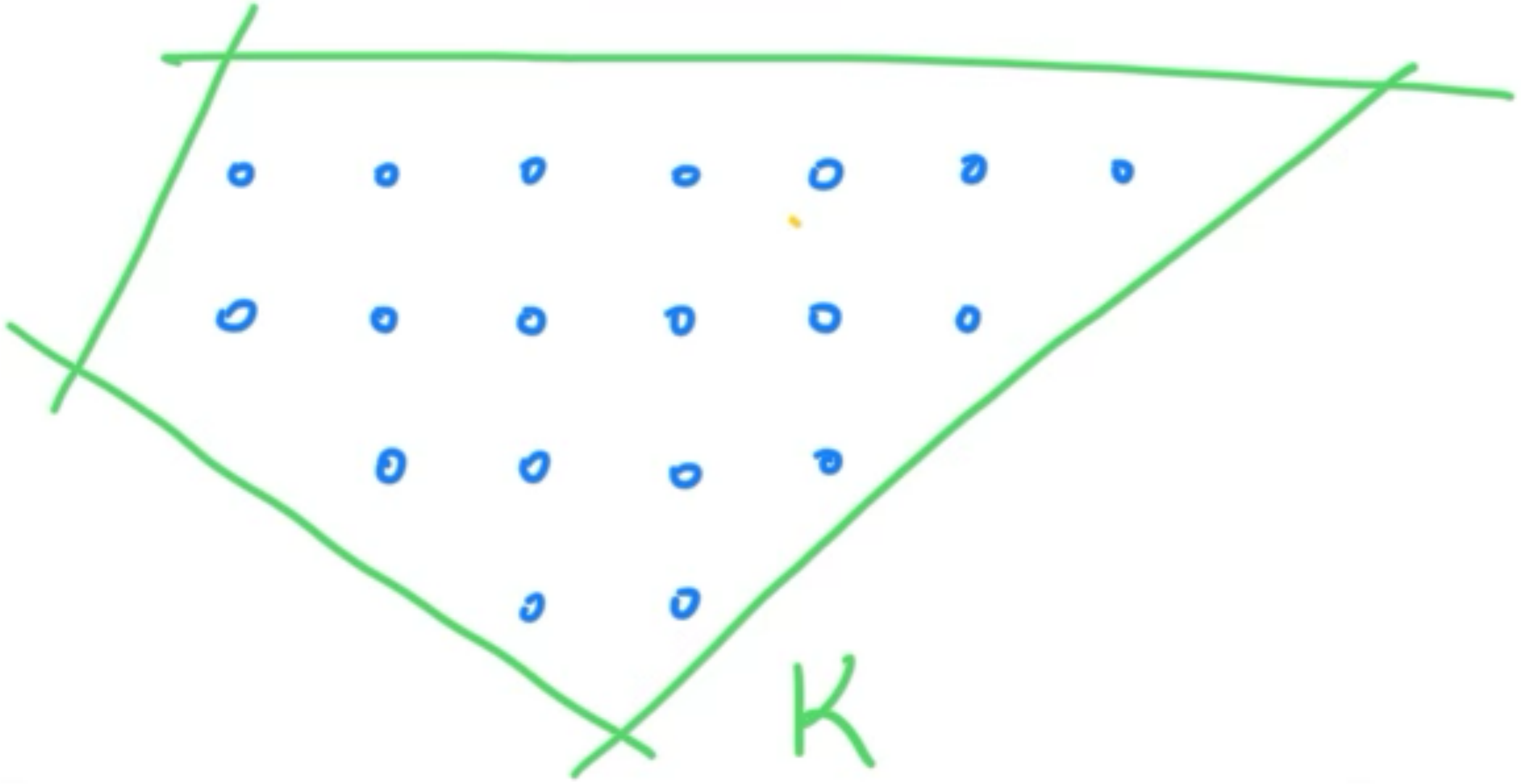
Lenstra-type Integer Programs

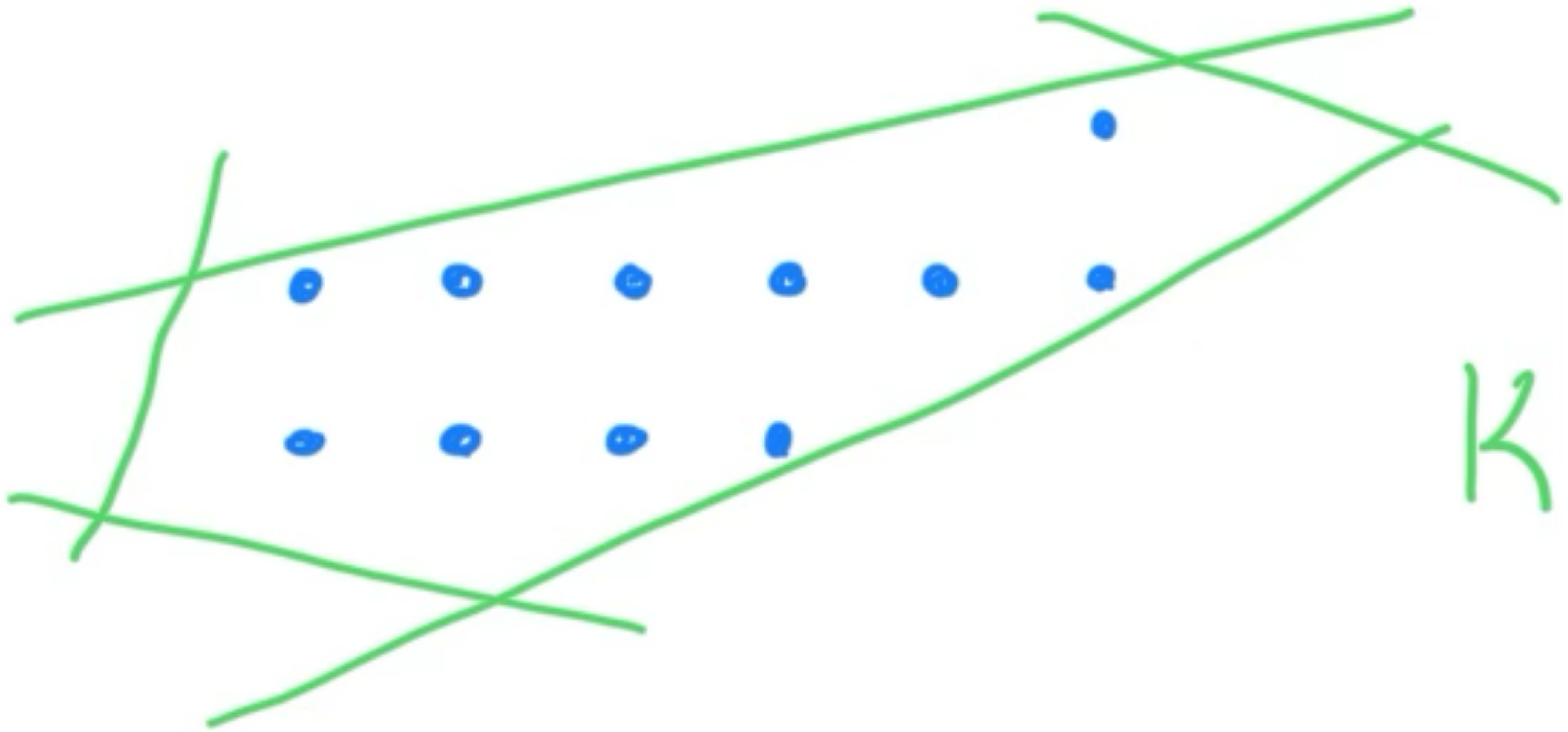


2^{n^3}	Lenstra ‘83
$2^n n^{2.5n}$	Kannan ‘87
$2^n n^n$	Dadush, Peikert, Vempala ‘11, Dadush ‘12
$log(n)^n$	Rothvoss, Reis ‘23

Lenstra-type Integer Programs

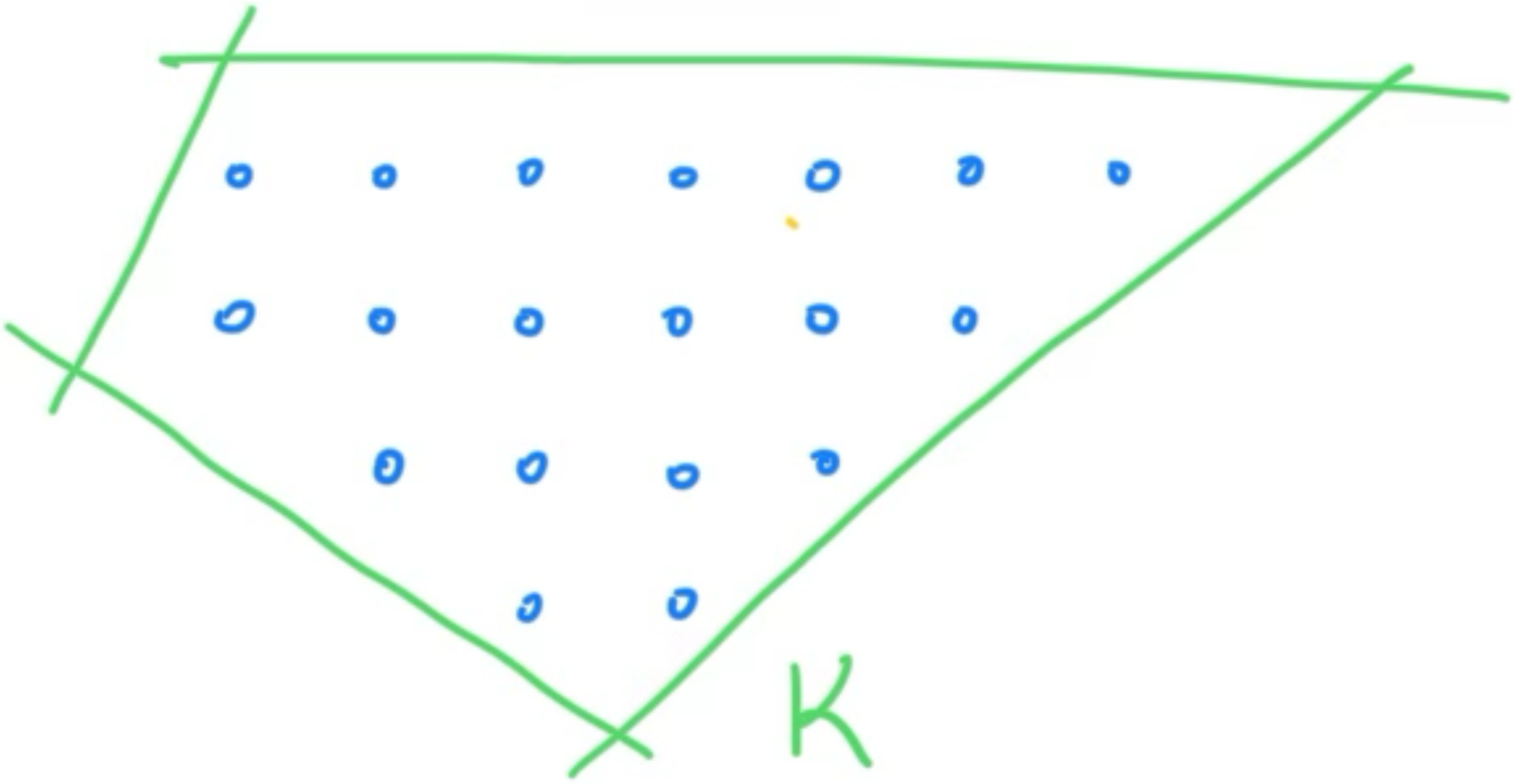
1. Decide whether K is “fat” or “flat”





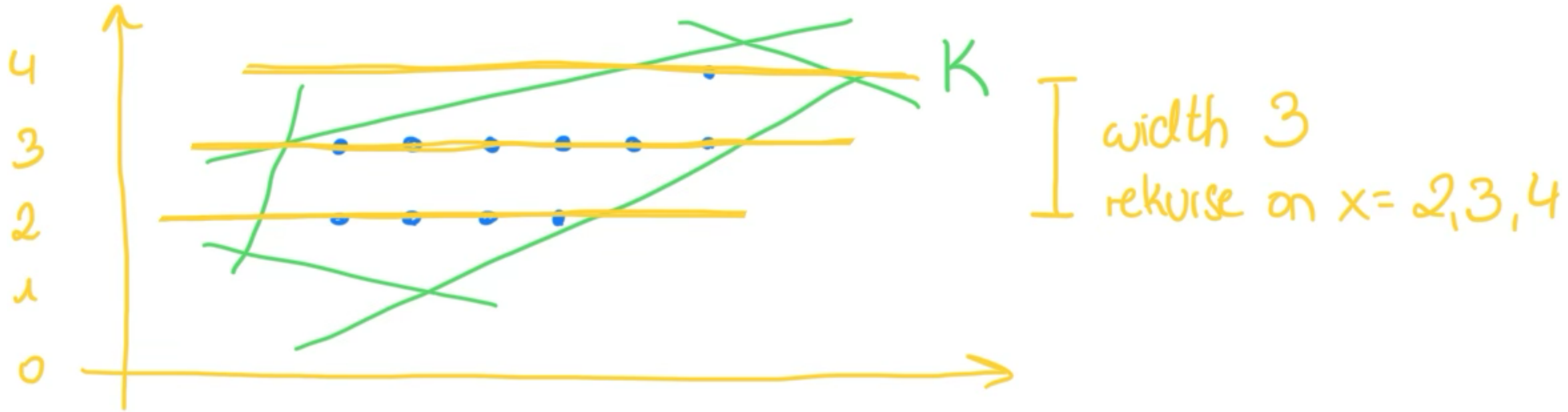
Lenstra-type Integer Programs

1.1. If "fat": easy to find solution

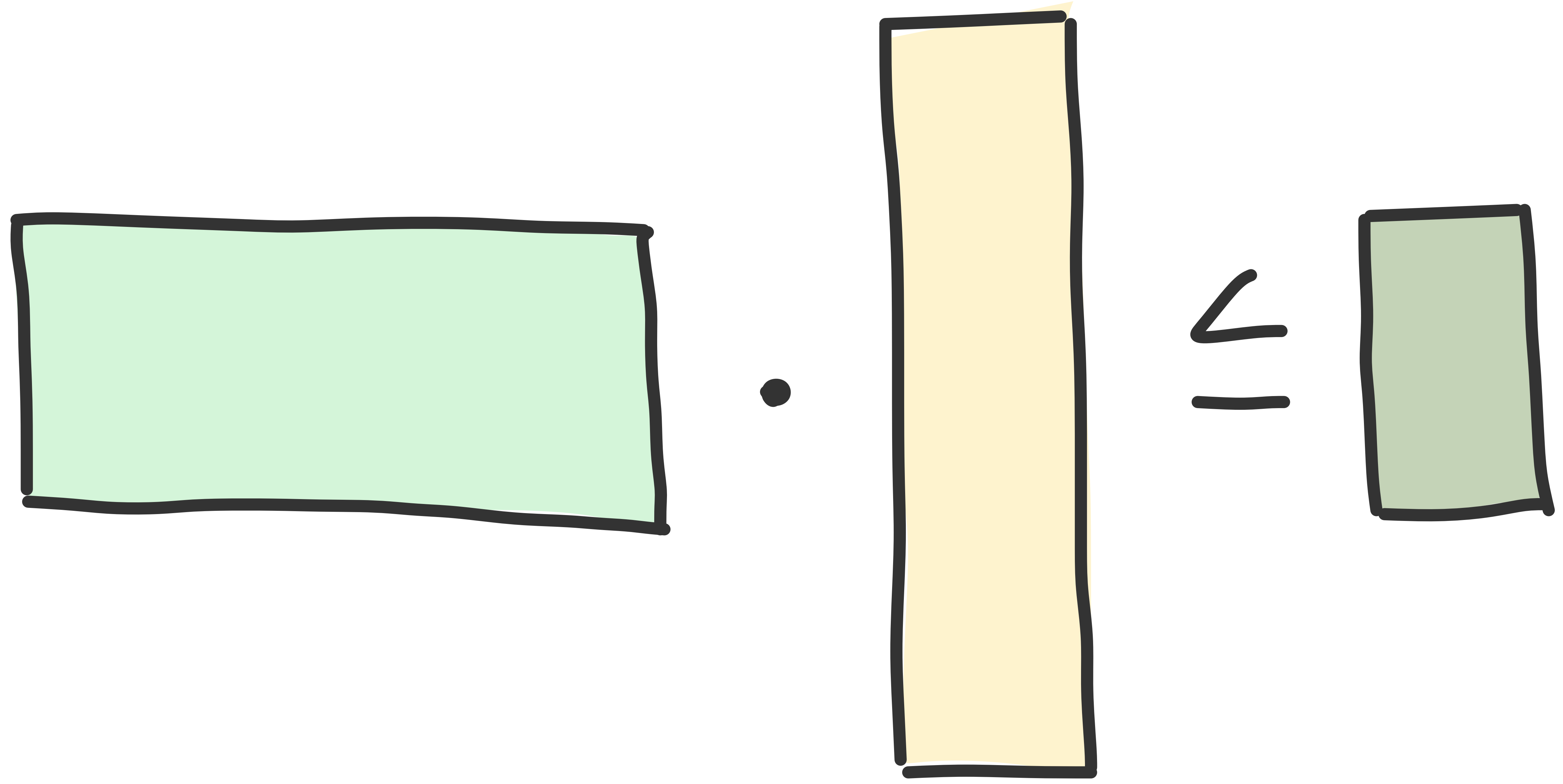


Lenstra-type Integer Programs

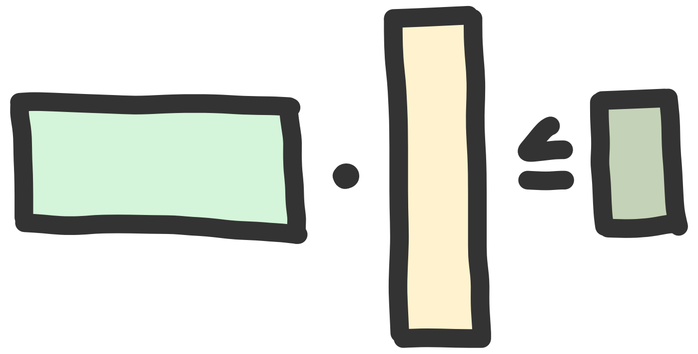
1.2. If “flat”: find flat dimension and decompose



Papademitriou-type Integer Programs



Papademitriou-type Integer Programs



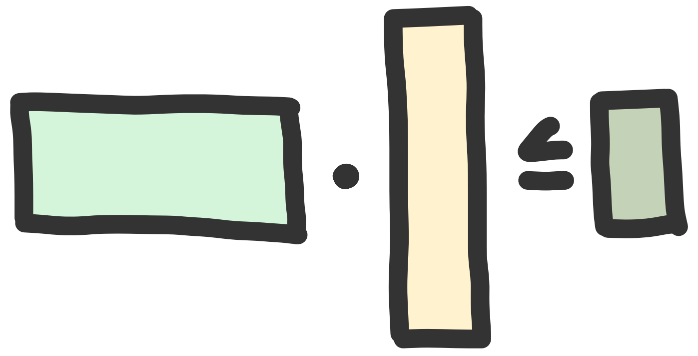
$n^{2m+2} \cdot (m\Delta + m||b||^{\infty})^{(m+1)(2m+1)}$ Papadimitriou '81

$n \cdot (m\Delta)^{2m} \cdot ||b||_1^2$ Eisenbrand, Weismantel '18

$O(nm) \cdot (m\Delta)^{2m} \cdot \log(||b||_{\infty})$ Jansen, Rohwedder '19

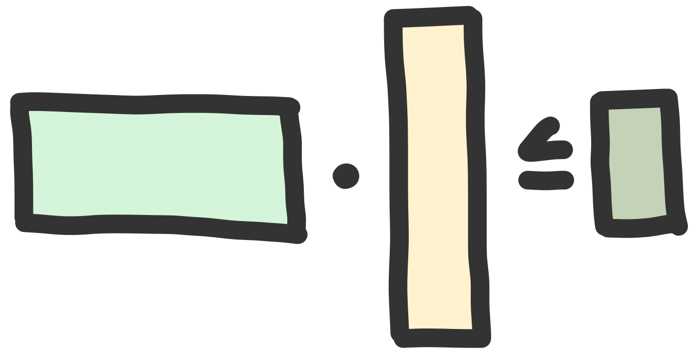
$O(nm) \cdot (\sqrt{m}\Delta)^{2m}$ Jansen Rohwedder '22

Papademitriou-type Integer Programs



Graver elements: inclusion—wise minimal ($\sqsubseteq$) kernel elements

$\sqsubseteq$: $x_i \leq y_i \ \ \forall i$
and sign—compatible

Papademitriou-type Integer Programs 

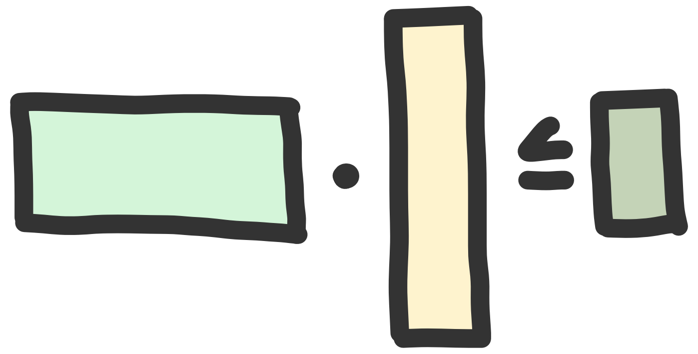
Graver elements: inclusion—wise minimal ($\sqsubseteq$) kernel elements

$\sqsubseteq$: $x_i \leq y_i \quad \forall i$

and sign—compatible

$$\begin{pmatrix} 3 \\ 5 \\ -3 \end{pmatrix} \sqsubseteq \begin{pmatrix} 5 \\ 5 \\ -4 \end{pmatrix} \qquad \begin{pmatrix} 3 \\ -2 \\ 4 \end{pmatrix} \not\sqsubseteq \begin{pmatrix} 5 \\ 0 \\ 2 \end{pmatrix}$$

Papademitriou-type Integer Programs



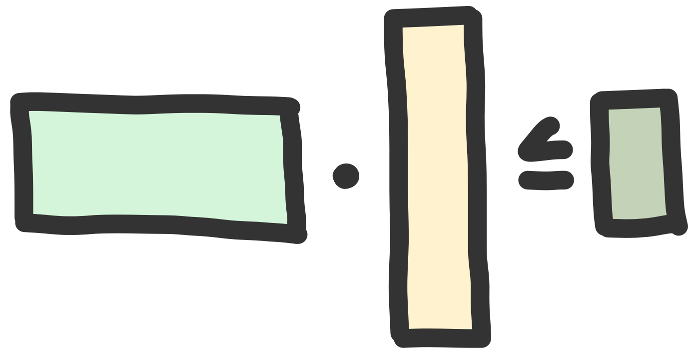
Graver elements: inclusion—wise minimal (⊑) kernel elements

$\sqsubseteq : x_i \leq y_i \quad \forall i$

and sign—compatible

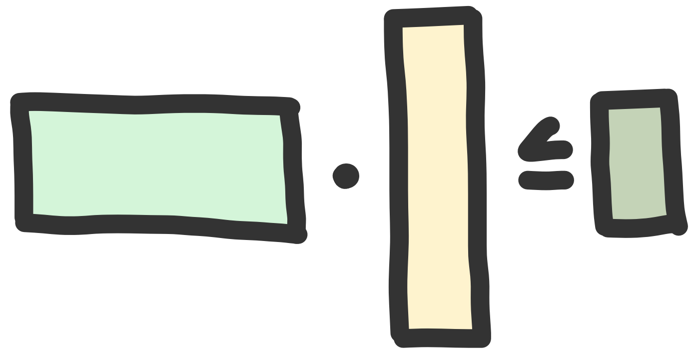
$$\begin{pmatrix} 3 \\ 5 \\ -3 \end{pmatrix} \sqsubseteq \begin{pmatrix} 5 \\ 5 \\ -4 \end{pmatrix} \qquad \begin{pmatrix} 3 \\ -2 \\ 4 \end{pmatrix} \not\sqsubseteq \begin{pmatrix} 5 \\ 0 \\ 2 \end{pmatrix}$$

Papademitriou-type Integer Programs

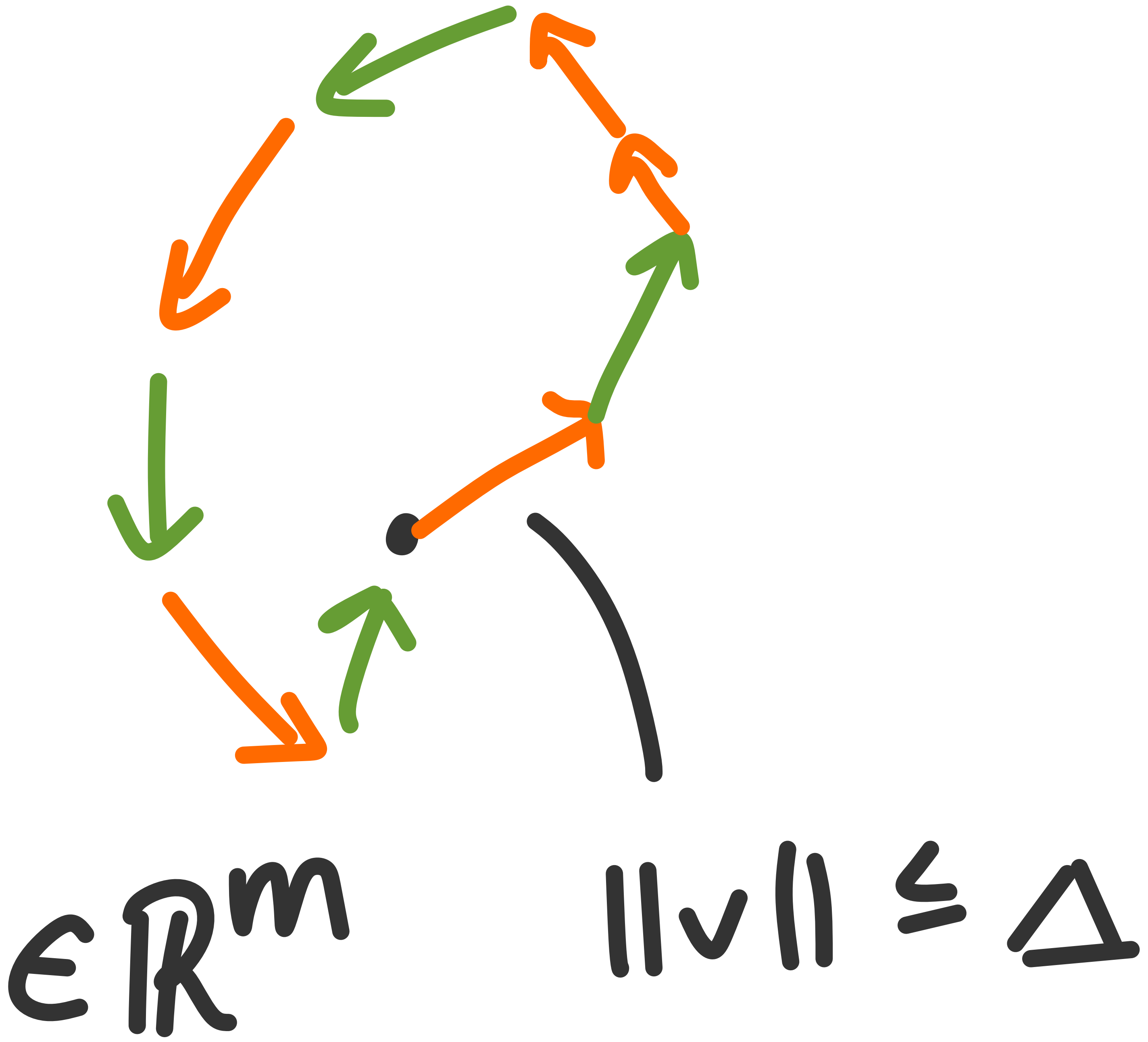


$||g||_\infty \leq \Delta^{f(k)}$ **via Steinitz lemma**

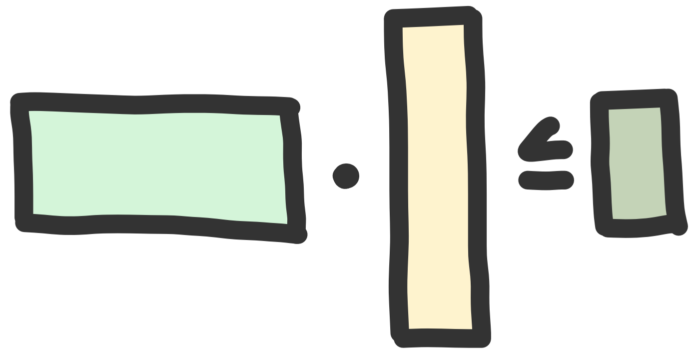
Papademitriou-type Integer Programs



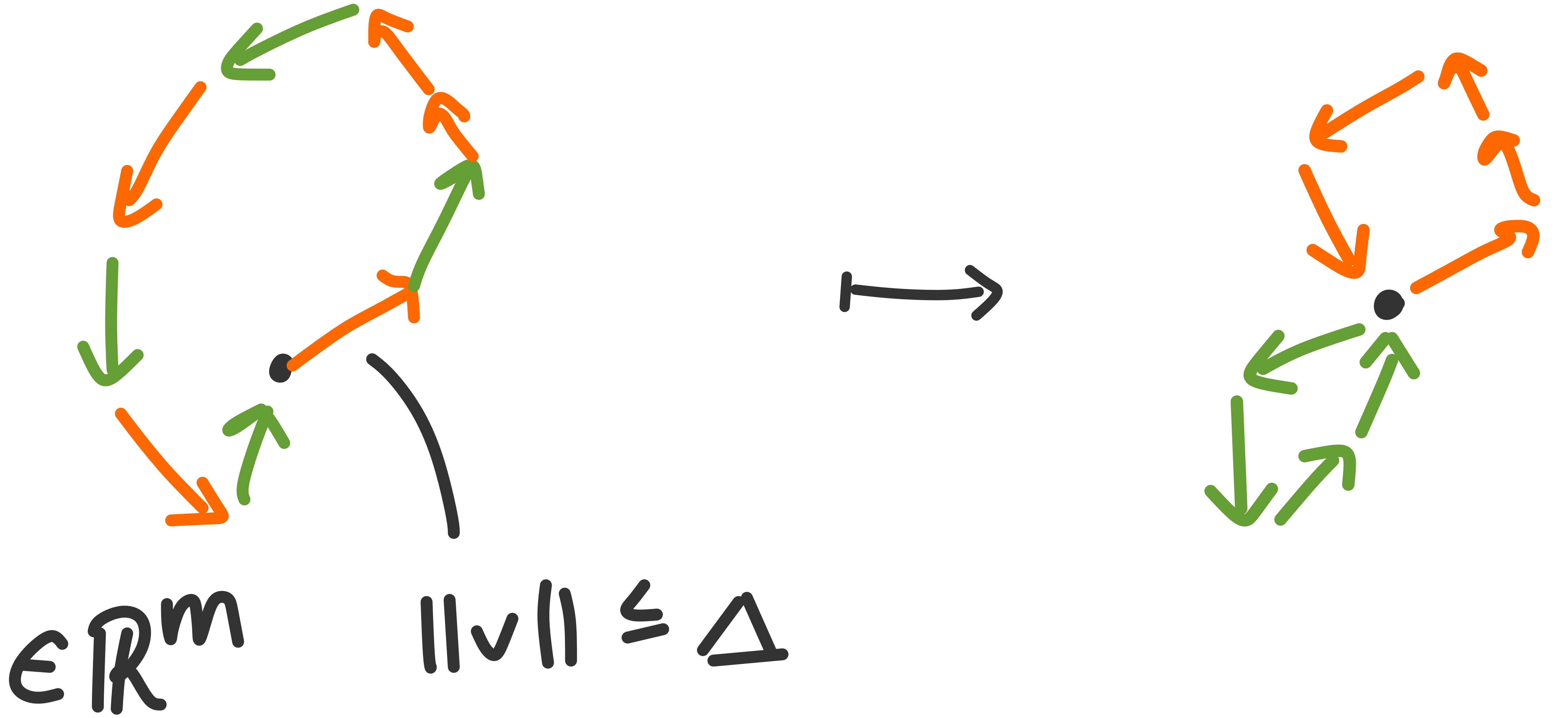
$||g||_\infty \leq \Delta^{f(k)}$ **via Steinitz lemma**



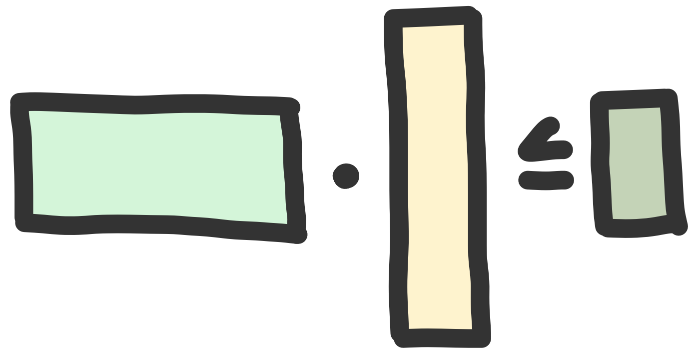
Papademitriou-type Integer Programs



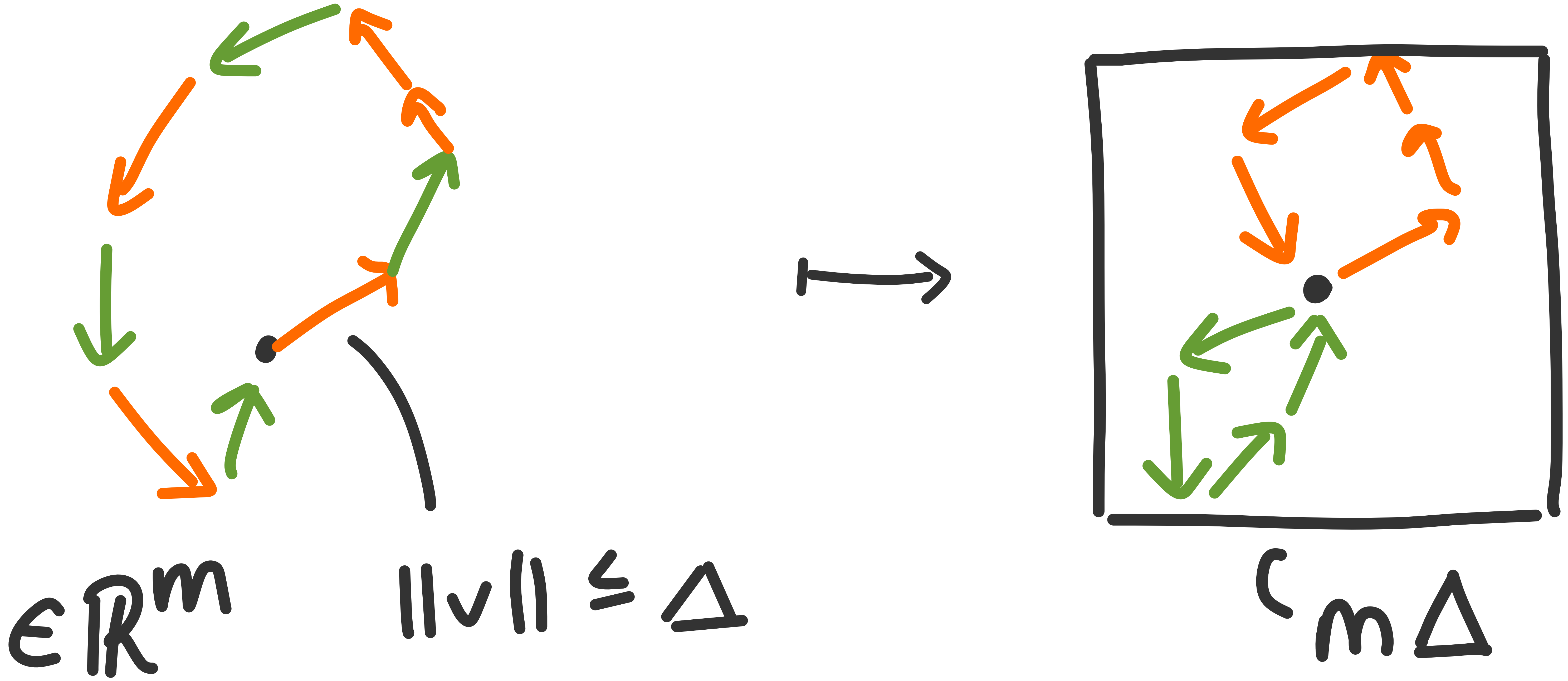
$||g||_\infty \leq \Delta^{f(k)}$ **via Steinitz lemma**



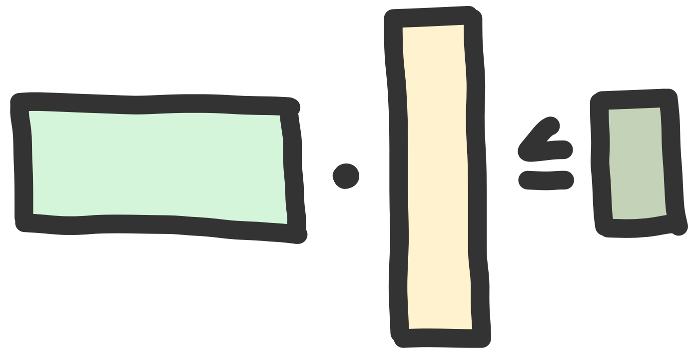
Papademitriou-type Integer Programs



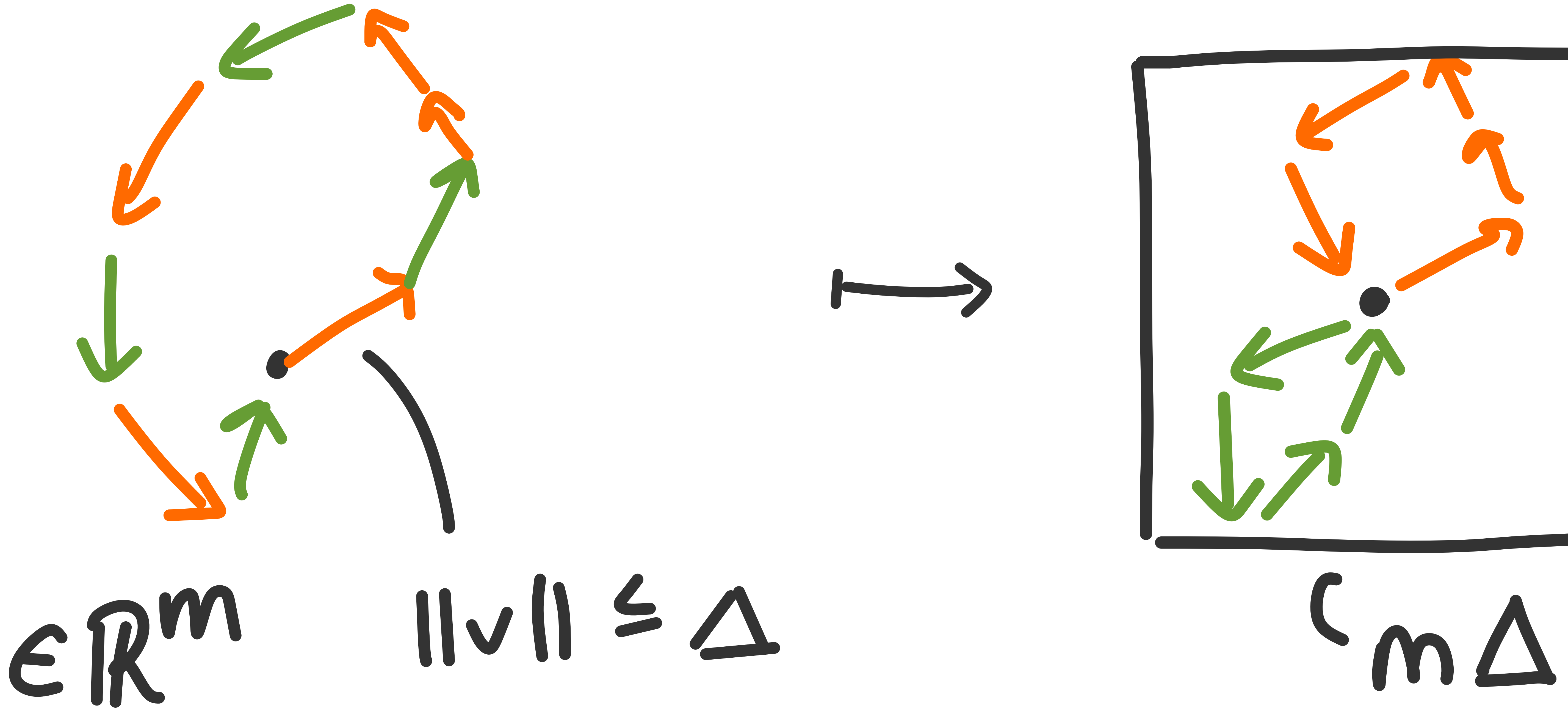
$||g||_\infty \leq \Delta^{f(k)}$ **via Steinitz lemma**



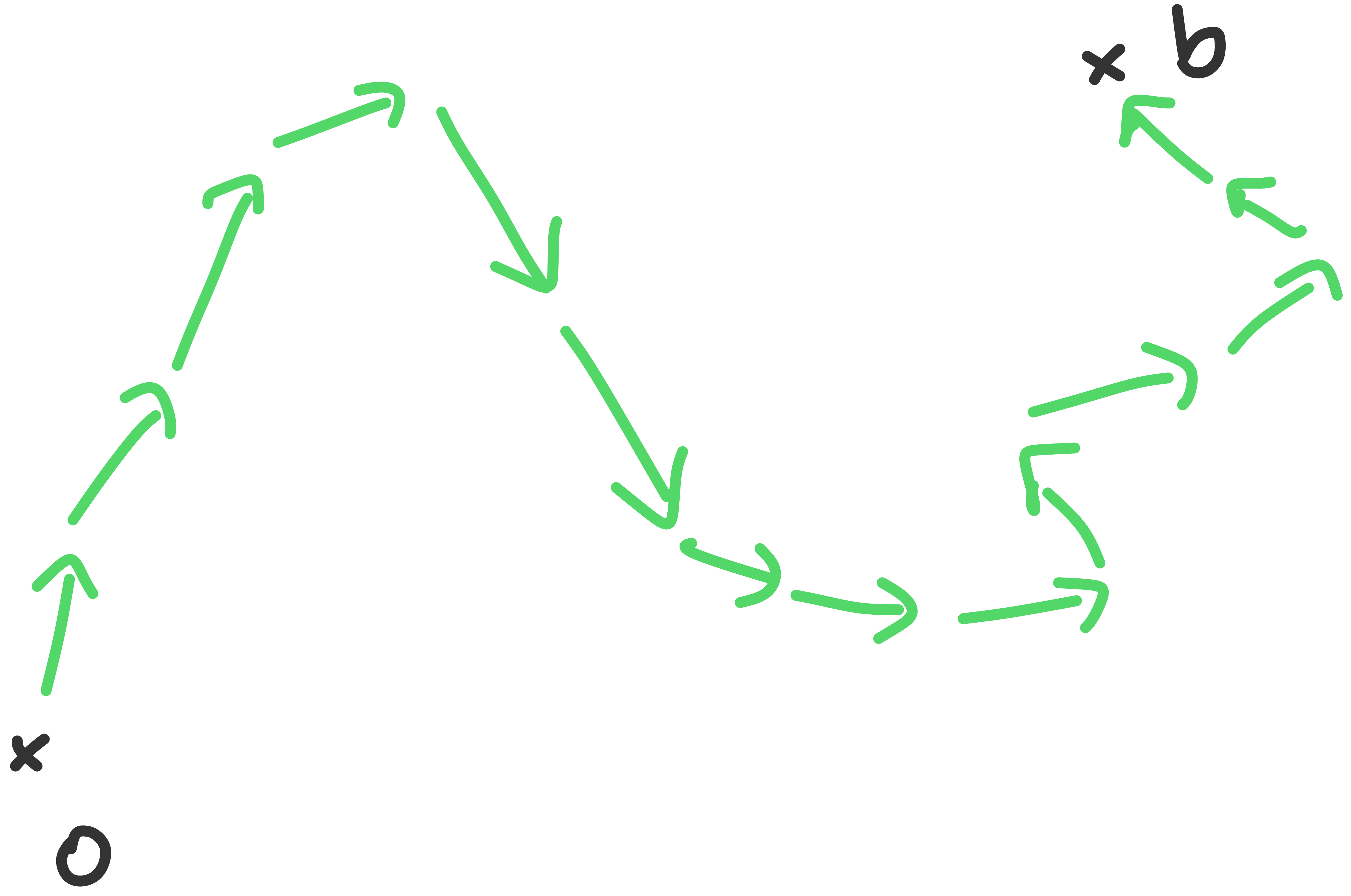
Papademitriou-type Integer Programs



$||g||_\infty \leq \Delta^{f(k)}$ **via Steinitz lemma**



Papademitriou-type Integer Programs



Papademitriou-type Integer Programs