# Lattice-based cryptography

Monika Trimoska

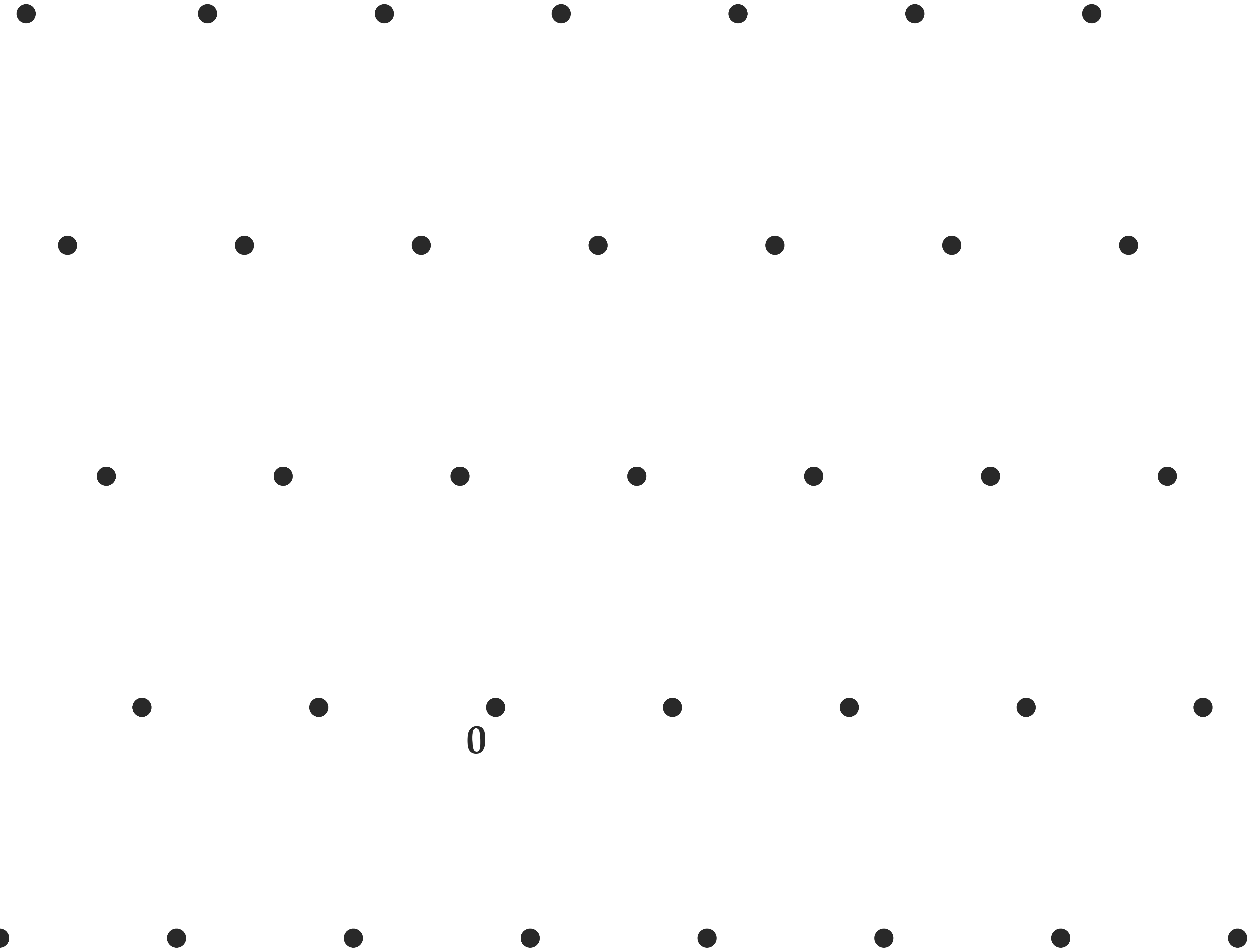
Selected Areas in Cryptology - Part 1

Spring, 2024

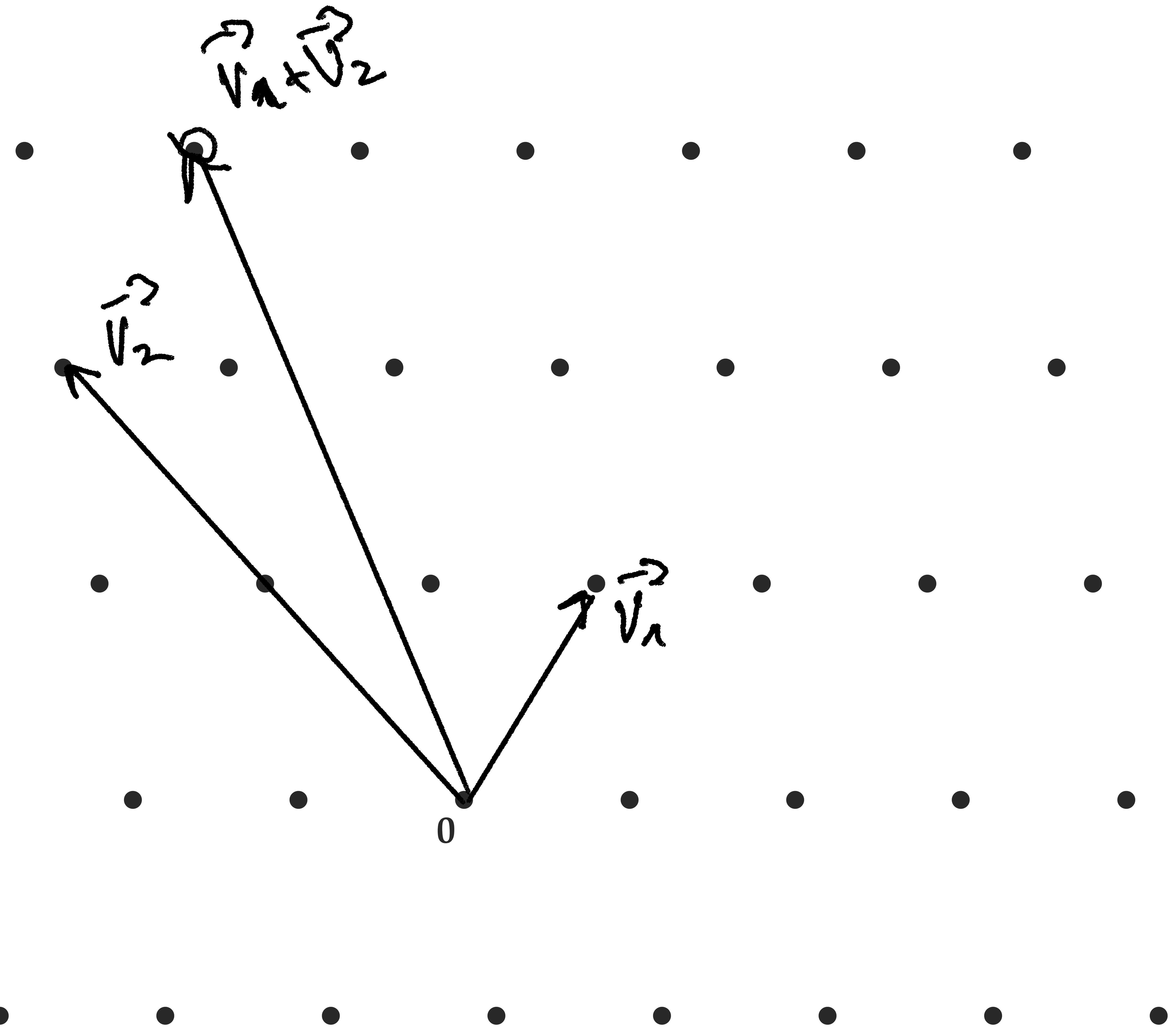
# Lattices

# What is a lattice?

A lattice $L \subset \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$.



# What is a lattice?




A lattice $L \subset \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$.

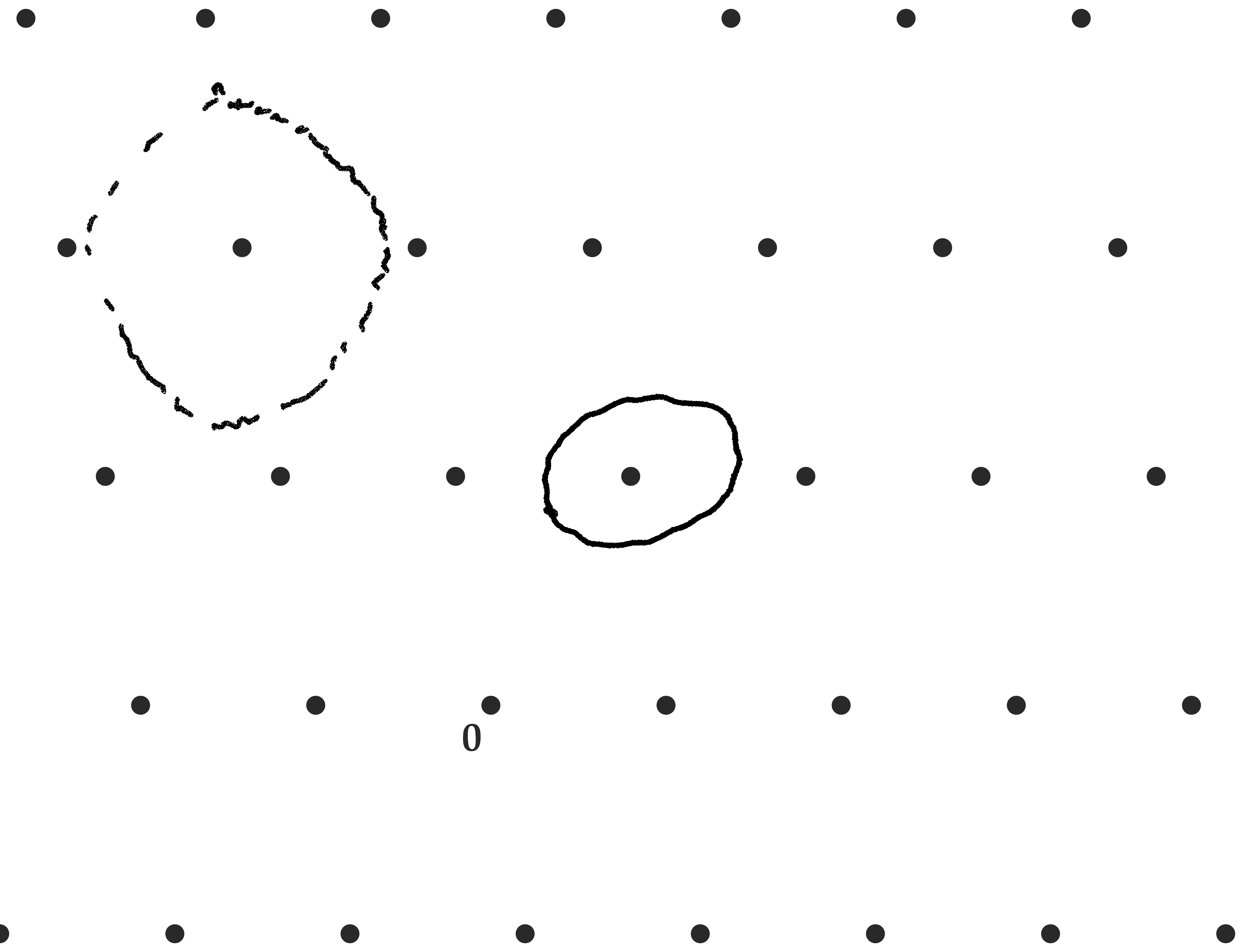
⟶ dots: points on the lattice $\mathbf{c} \in L$.

# What is a lattice?

A lattice $L \subset \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$.

dots: points on the lattice $\mathbf{c} \in L$.

for every $\mathbf{v} \in L$, there exists an open ball around $\mathbf{v}$ that contains no other elements from $L$.



# Codes and lattices

- Hard problems: finding low-weight codewords
- Hamming metric
- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$
- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low

- Hamming metric

- Working with $k$-dimension

- Structured codes with a de

**The syndrome decoding problem**

Given a syndrome $\mathbf{s} = \mathbf{He}$, find $\mathbf{e}$ such that $\mathrm{wt}(\mathbf{e}) \leq t$.

$$\overset{\mathbf{H}}{\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}} \overset{\mathbf{e}}{\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix}} = \overset{\mathbf{s}}{\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}$$

Find $\mathbf{e}$ of minimum weight.

# Codes and lattices

- Hard problems: finding low-weight codewords
- Hamming metric
- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$
- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric

- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords → Finding close lattice points

- Hamming metric

**Hamming metric**

For $\mathbf{x} \in \mathbb{F}_2^n$, the Hamming weight of $\mathbf{x}$ is the number of nonzero elements, aka.

$$\mathrm{wt}(\mathbf{x}) = |\{i \in \{1,\ldots,n\} \mid x_i \neq 0\}|.$$

- Working with $k$-dimens

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric

- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points
- Hamming metric $\longrightarrow$ Euclidean metric
- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$
- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric $\longrightarrow$ Euclidean metric

**Binary linear code**

An $[n, k]$ binary linear code $\mathscr{C}$ is a $k$-dimensional subspace of $\mathbb{F}_2^n$.

- Working with $k$-dimensi

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric $\longrightarrow$ Euclidean metric

- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points
- Hamming metric $\longrightarrow$ Euclidean metric
- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$ $\longrightarrow$ Working with full-rank lattices
- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords ⟶ Finding close lattice points

- Hamming metric ⟶ Eucl

$$\overset{\mathbf{H}}{\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}} \left( \overset{\mathbf{c}}{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}} + \overset{\mathbf{e}}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \right) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

- Working with $k$-dimensional code ... ith full-rank lattices

- Structured codes with a decoding algorithm

# Codes and lattices

- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric $\longrightarrow$ Euclidean metric

- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$ $\longrightarrow$ Working with full-rank lattices

- Structured codes with a decoding algorithm

# Codes and lattices

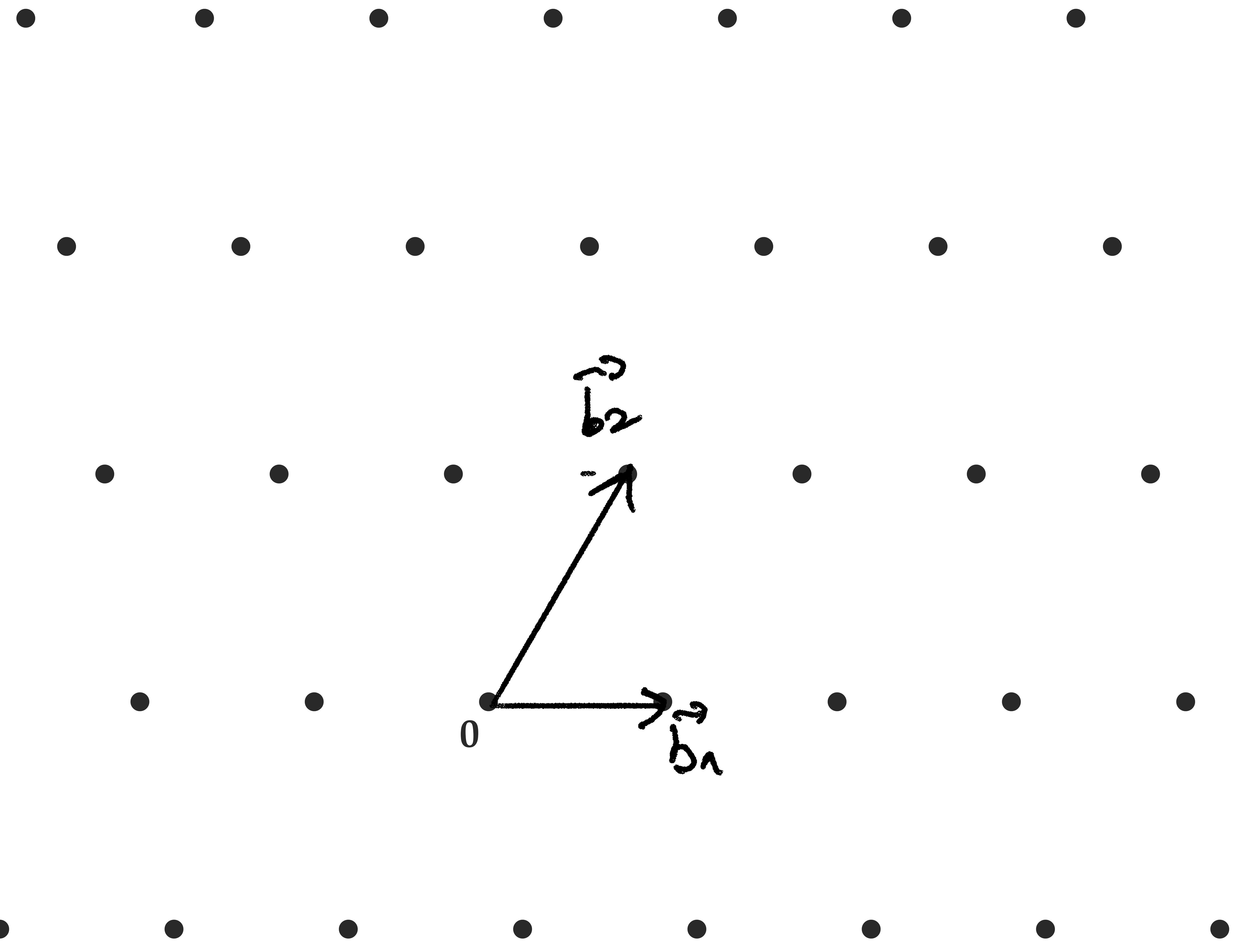
- Hard problems: finding low-weight codewords $\longrightarrow$ Finding close lattice points

- Hamming metric $\longrightarrow$ Euclidean metric

- Working with $k$-dimensional codes of length $n$ with $k$ smaller than $n$ $\longrightarrow$ Working with full-rank lattices

- Structured codes with a decoding algorithm $\longrightarrow$ Any lattice with a short basis

# Basis representation

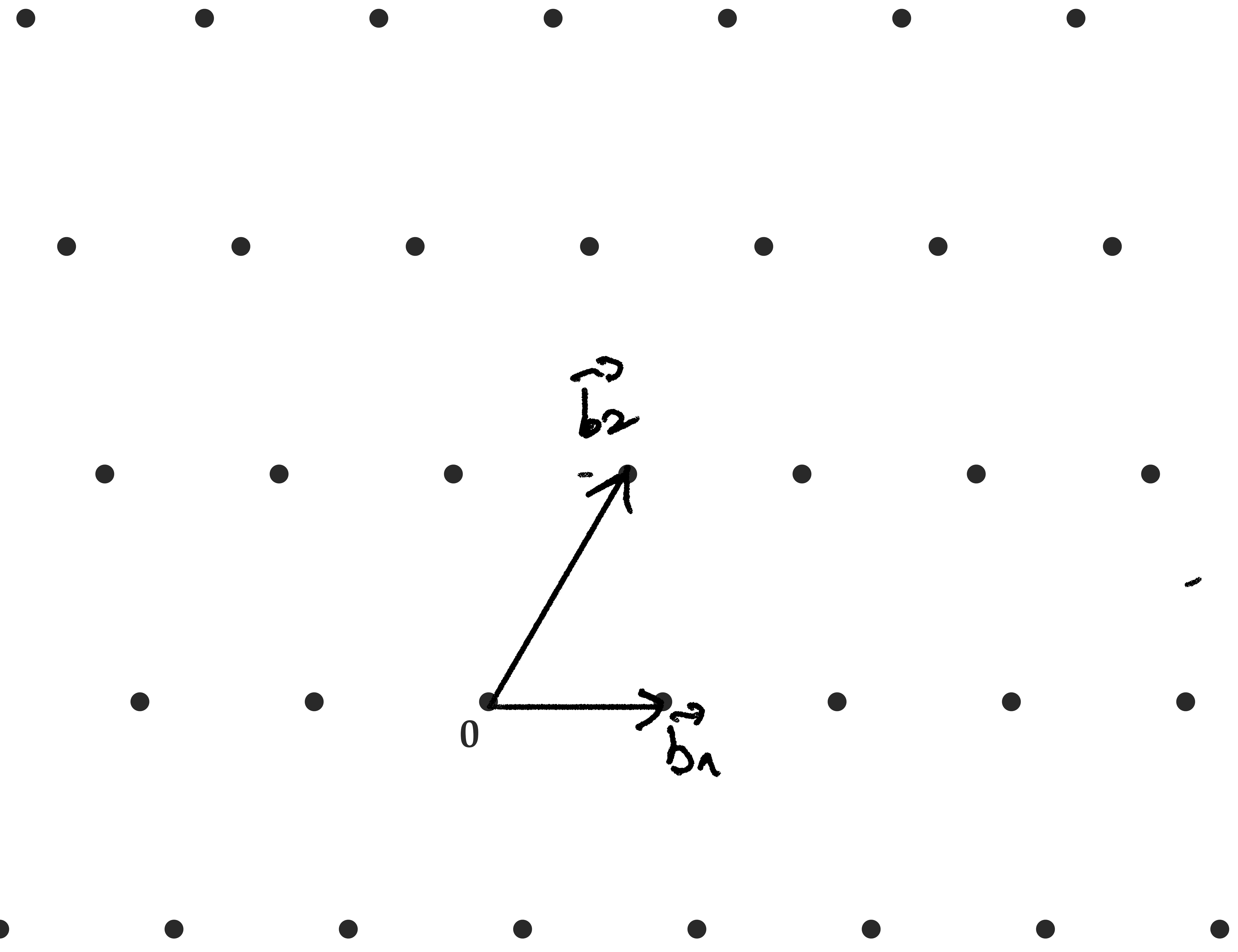



Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

rank

$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \}$$

# Basis representation



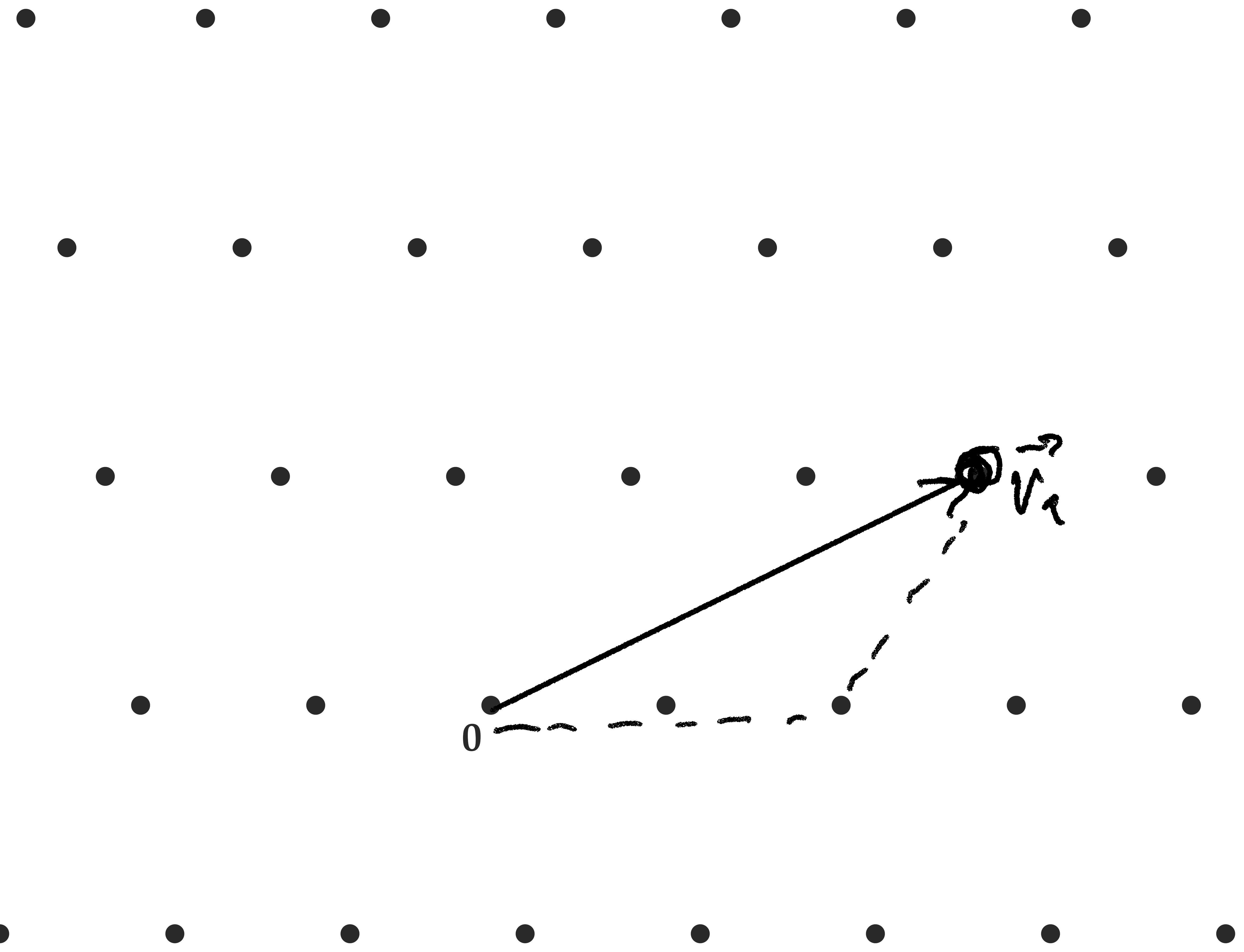

Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

rank

$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$$

# Basis representation




Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

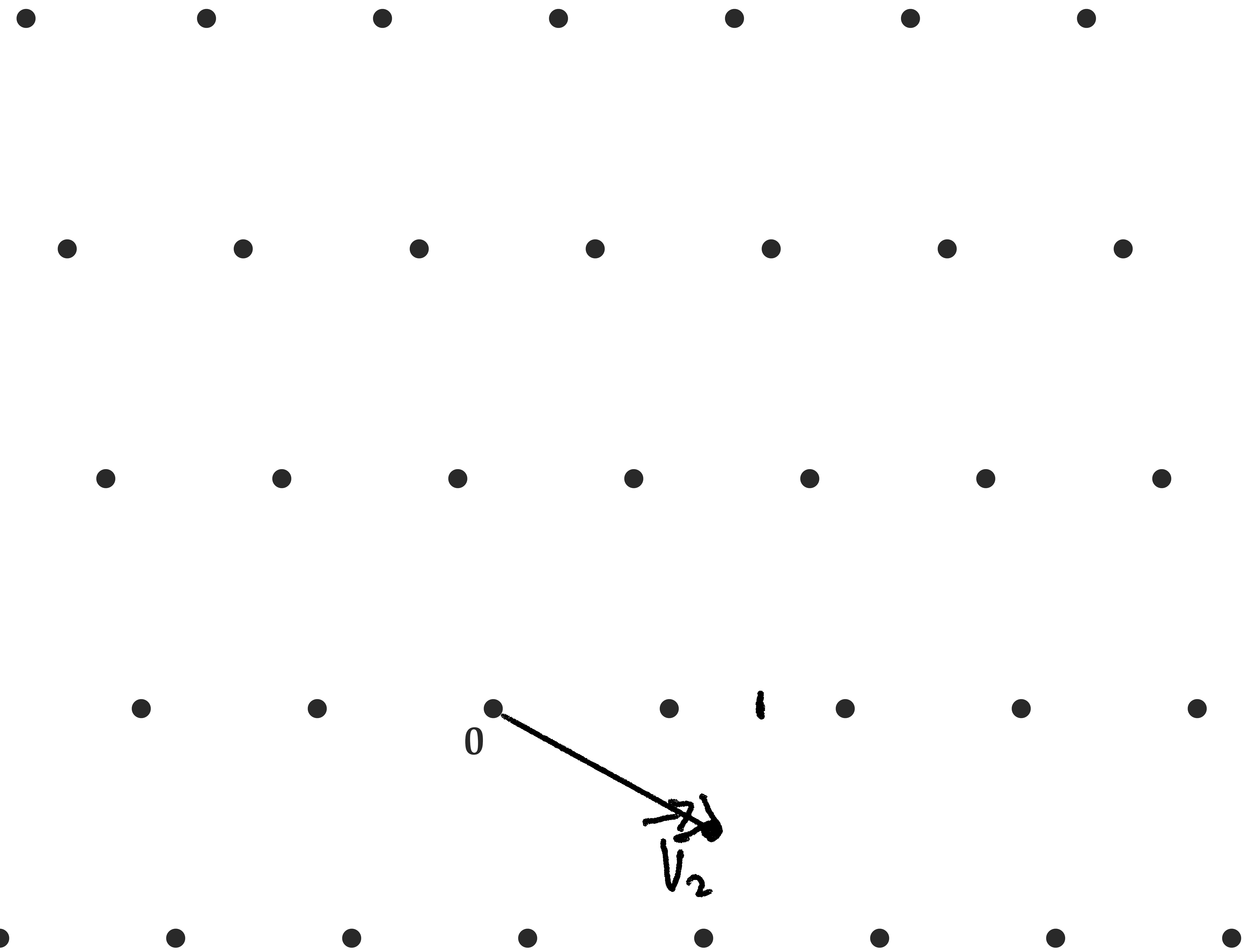
rank

$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$$

→ $\mathbf{v}_1 = 2\mathbf{b}_1 + \mathbf{b}_2$ is a lattice vector

# Basis representation




Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

rank

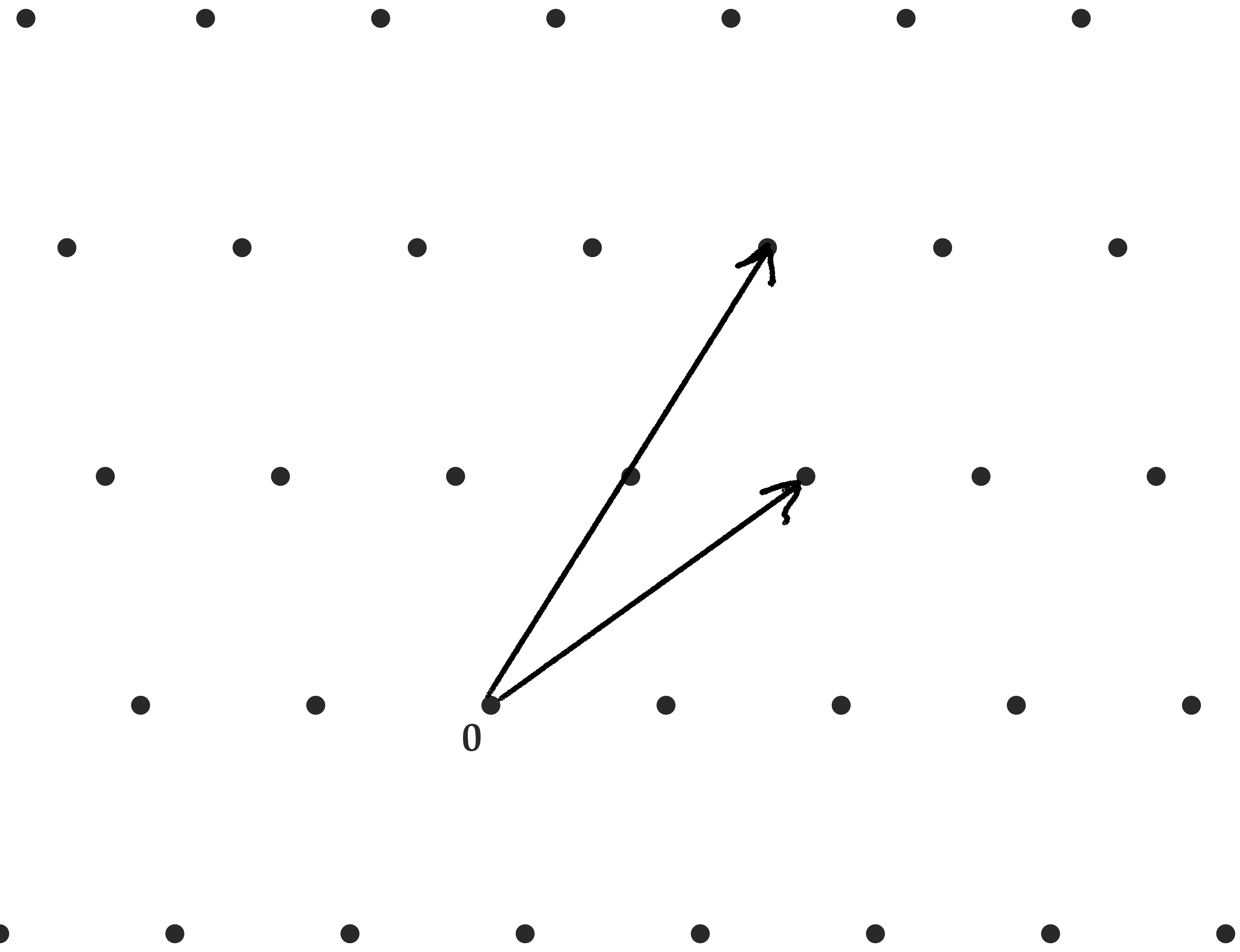
$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$$

→ $\mathbf{v}_1 = 2\mathbf{b}_1 + \mathbf{b}_2$ is a lattice vector

→ $\mathbf{v}_2 = 1.5\mathbf{b}_1 - 0.5\mathbf{b}_2$ is not a lattice vector

# Basis representation




Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

rank

$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$$

→ $\mathbf{v}_1 = 2\mathbf{b}_1 + \mathbf{b}_2$ is a lattice vector

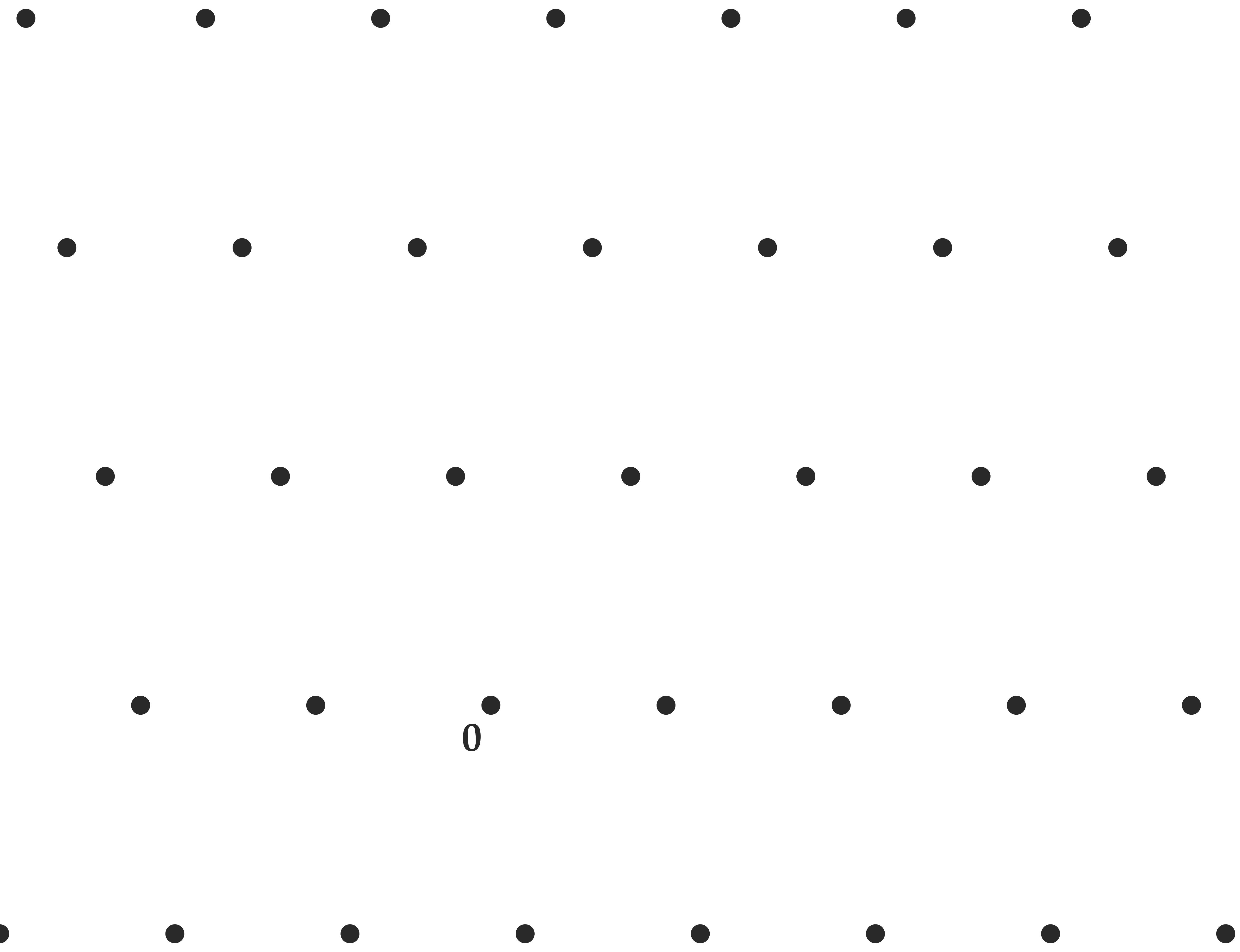
→ $\mathbf{v}_2 = 1.5\mathbf{b}_1 - 0.5\mathbf{b}_2$ is not a lattice vector

Another basis:

$$\mathbf{B}' = \mathbf{U} \cdot \mathbf{B}$$

with $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$, $\det(\mathbf{U}) = \pm 1$.

# Basis representation




Lattice basis: $n$ $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$

rank

$$L := \{ \sum_{i=1}^{n} a_i \mathbf{b}_i \,|\, a_i \in \mathbb{Z} \}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$$

→ $\mathbf{v}_1 = 2\mathbf{b}_1 + \mathbf{b}_2$ is a lattice vector

→ $\mathbf{v}_2 = 1.5\mathbf{b}_1 - 0.5\mathbf{b}_2$ is not a lattice vector

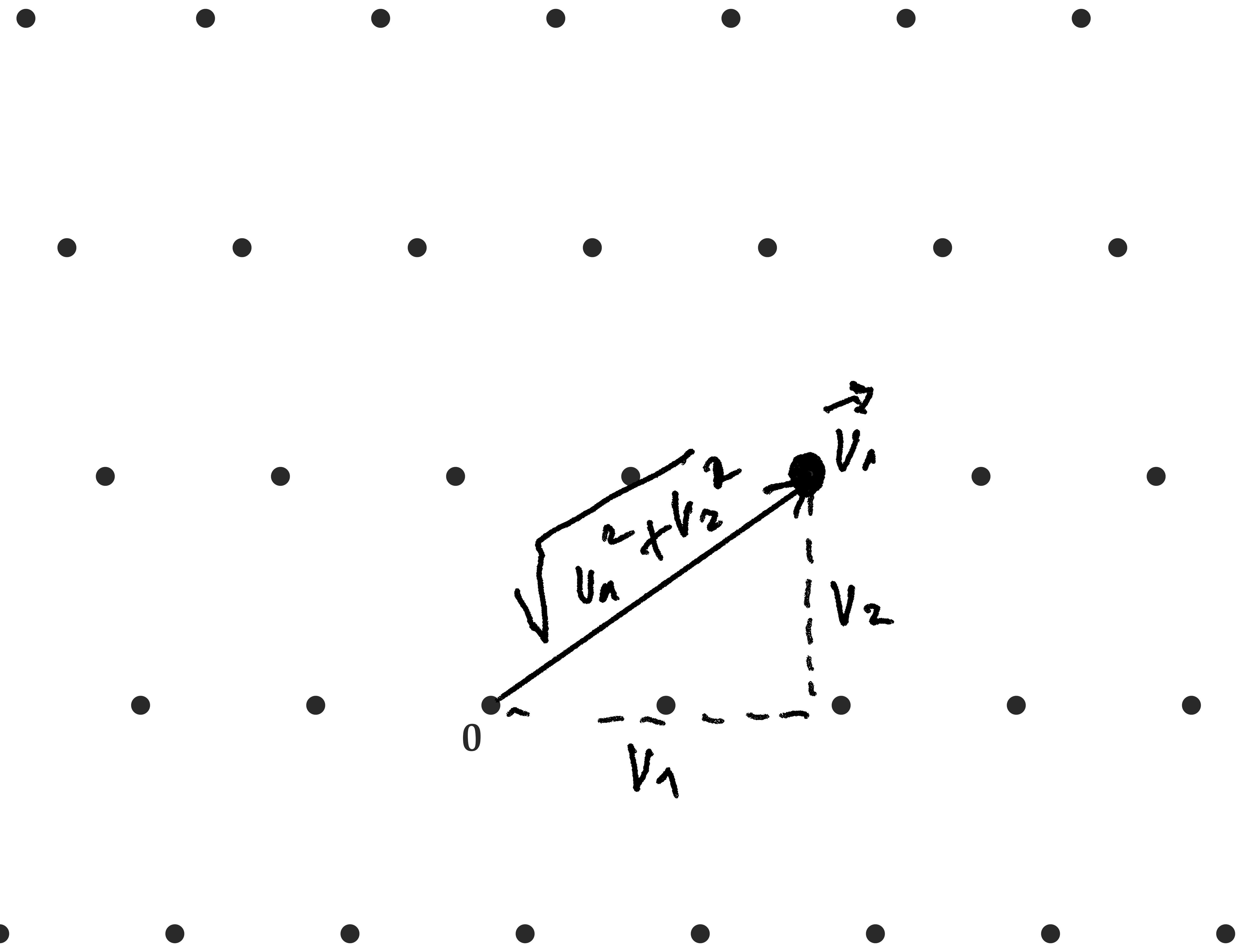
Another basis:

$$\mathbf{B}' = \mathbf{U} \cdot \mathbf{B}$$

with $\underbrace{\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})}$, $\det(\mathbf{U}) = \pm 1$.

→ infinitely many bases

# The Euclidean metric



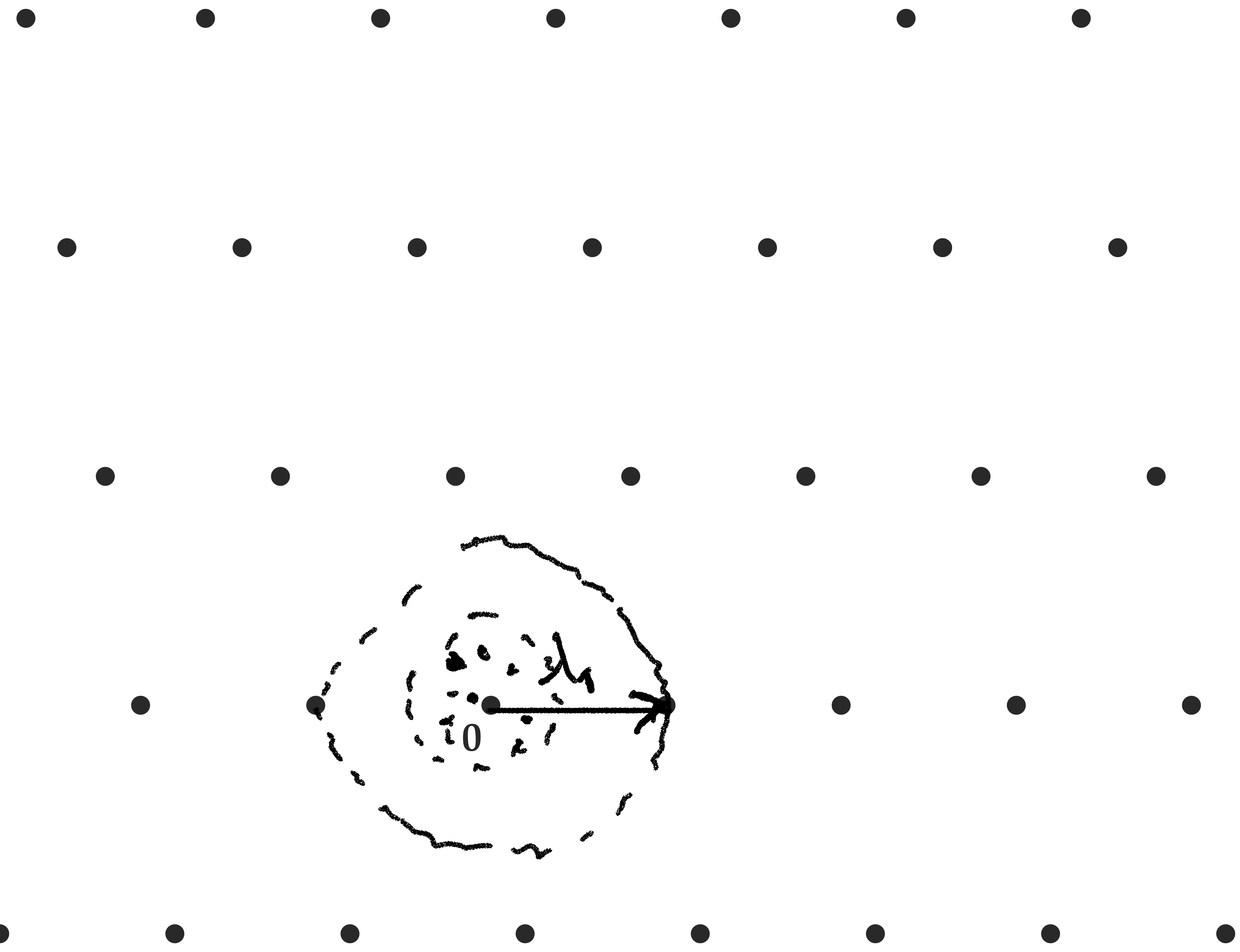

The Euclidean norm

$$||(v_1, \ldots, v_n)|| = \sqrt{v_1^2 + \ldots + v_n^2}$$

→ The Euclidean distance between $\mathbf{v}_1$ and $\mathbf{v}_2$

$$||(\mathbf{v}_1 - \mathbf{v}_2)||$$

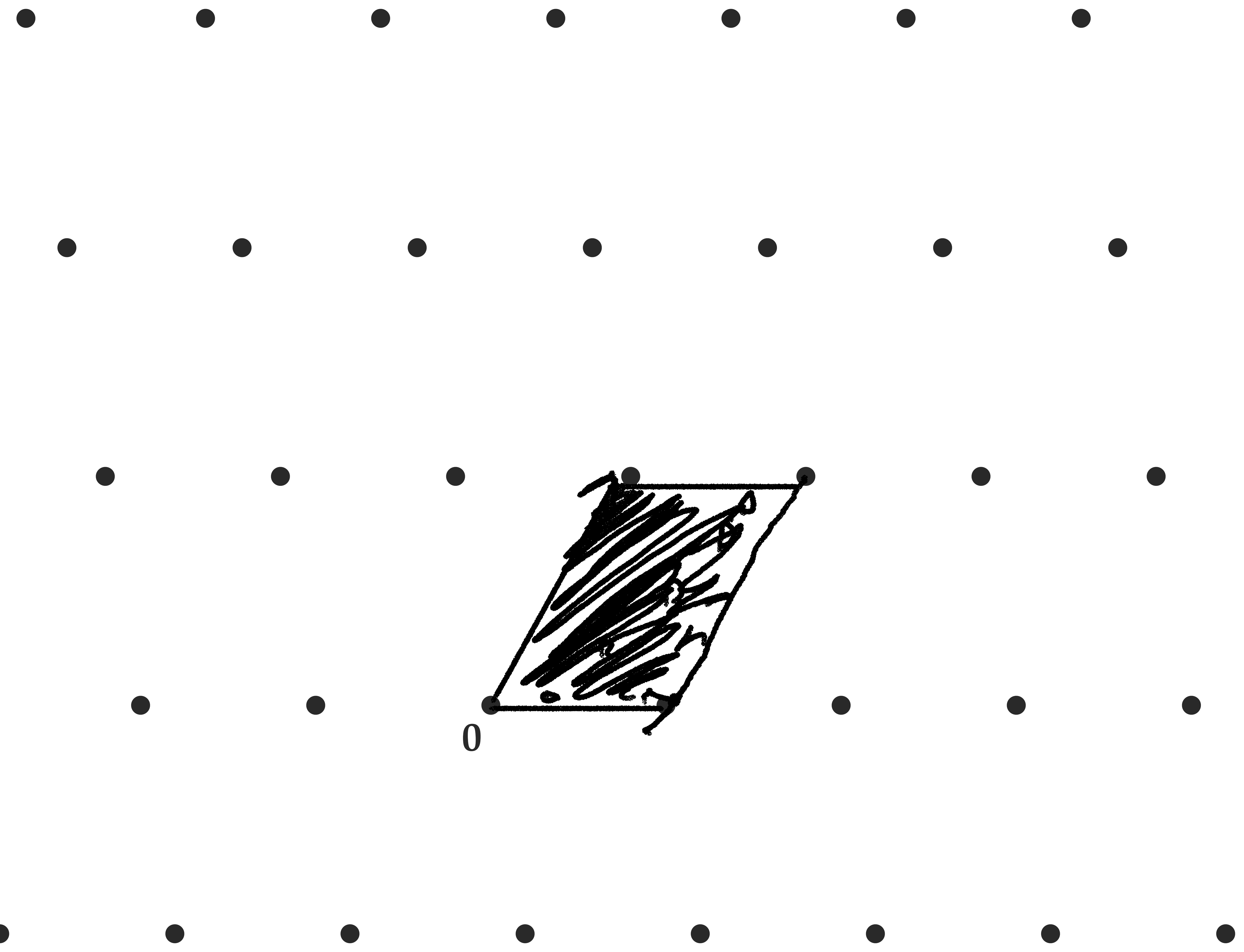
# The first minimum




The first minimum of a lattice $L$ is defined as the minimal norm of a nonzero lattice vector.

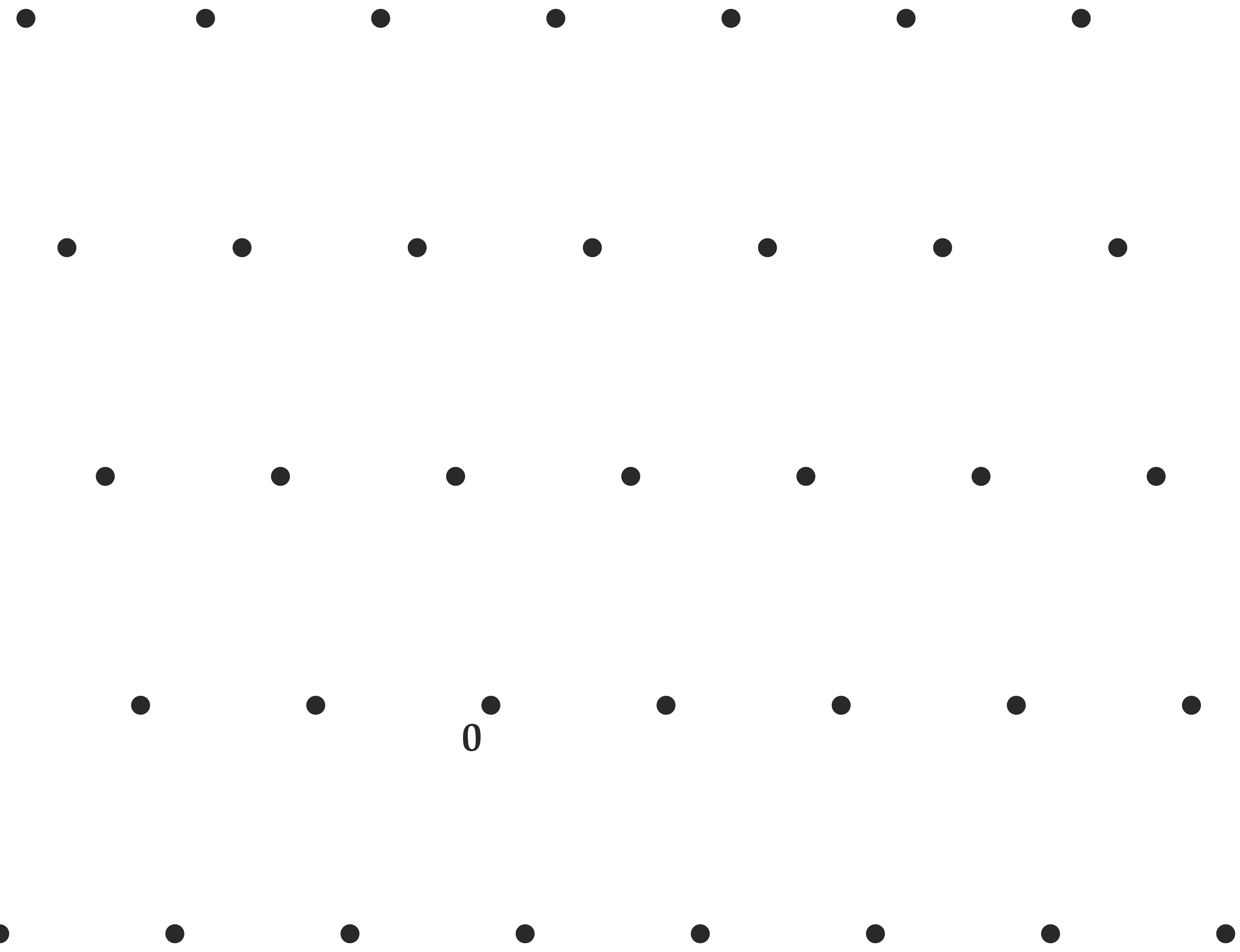
$$\lambda_1(L) = \min_{\mathbf{v} \in L \setminus \{0\}} ||\mathbf{v}||$$

# The fundamental parallelepiped




$$\mathscr{P}(\mathbf{B}) = [0,1)^n \cdot \mathbf{B}$$

# The fundamental parallelepiped

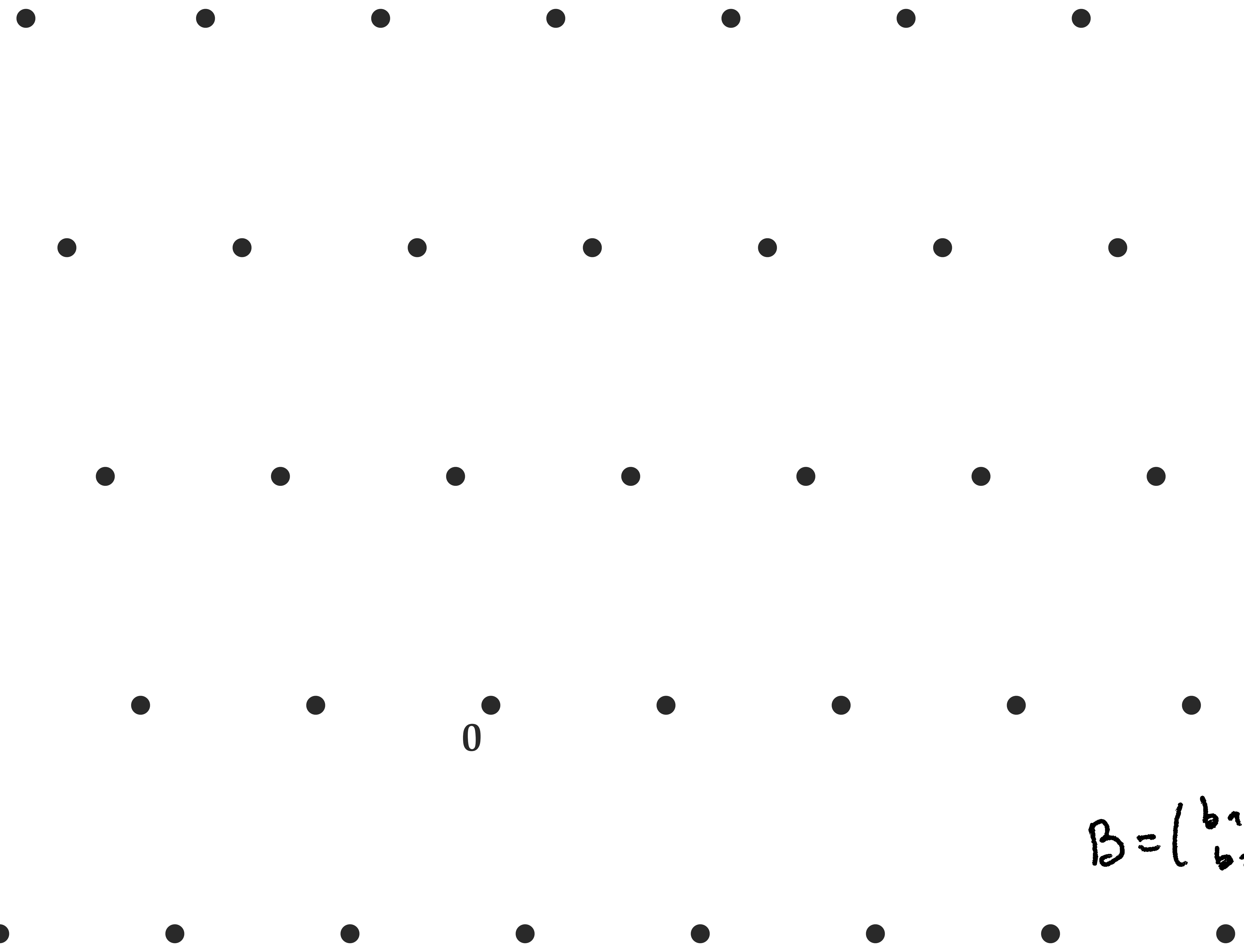



$\mathscr{P}(\mathbf{B}) = [0,1)^n \cdot \mathbf{B}$

Volume of a lattice

$\mathrm{vol}(L) = \mathrm{vol}(\mathscr{P}(\mathbf{B}))$

# The fundamental parallelepiped

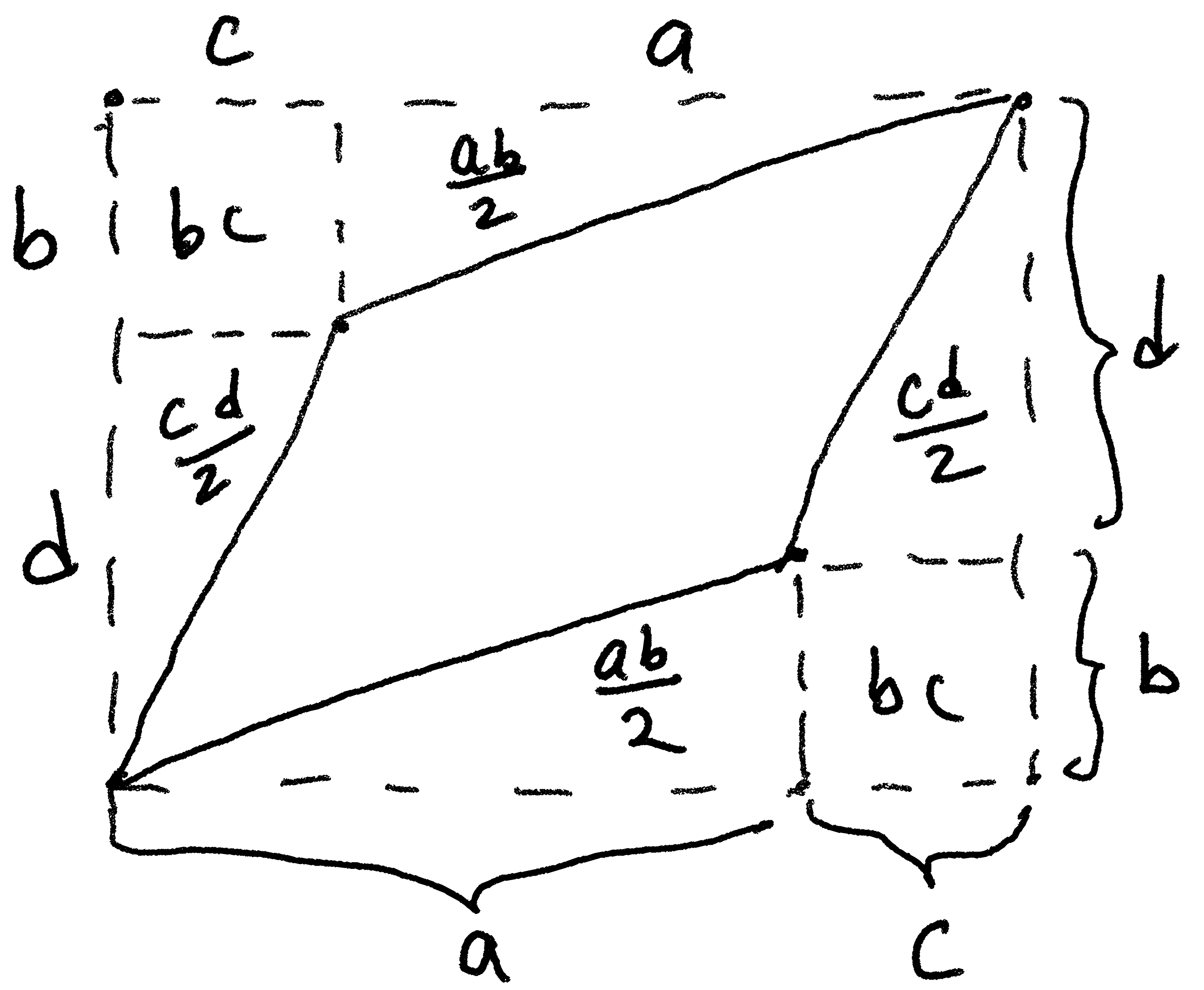



$\mathcal{P}(\mathbf{B}) = [0,1)^n \cdot \mathbf{B}$

Volume of a lattice

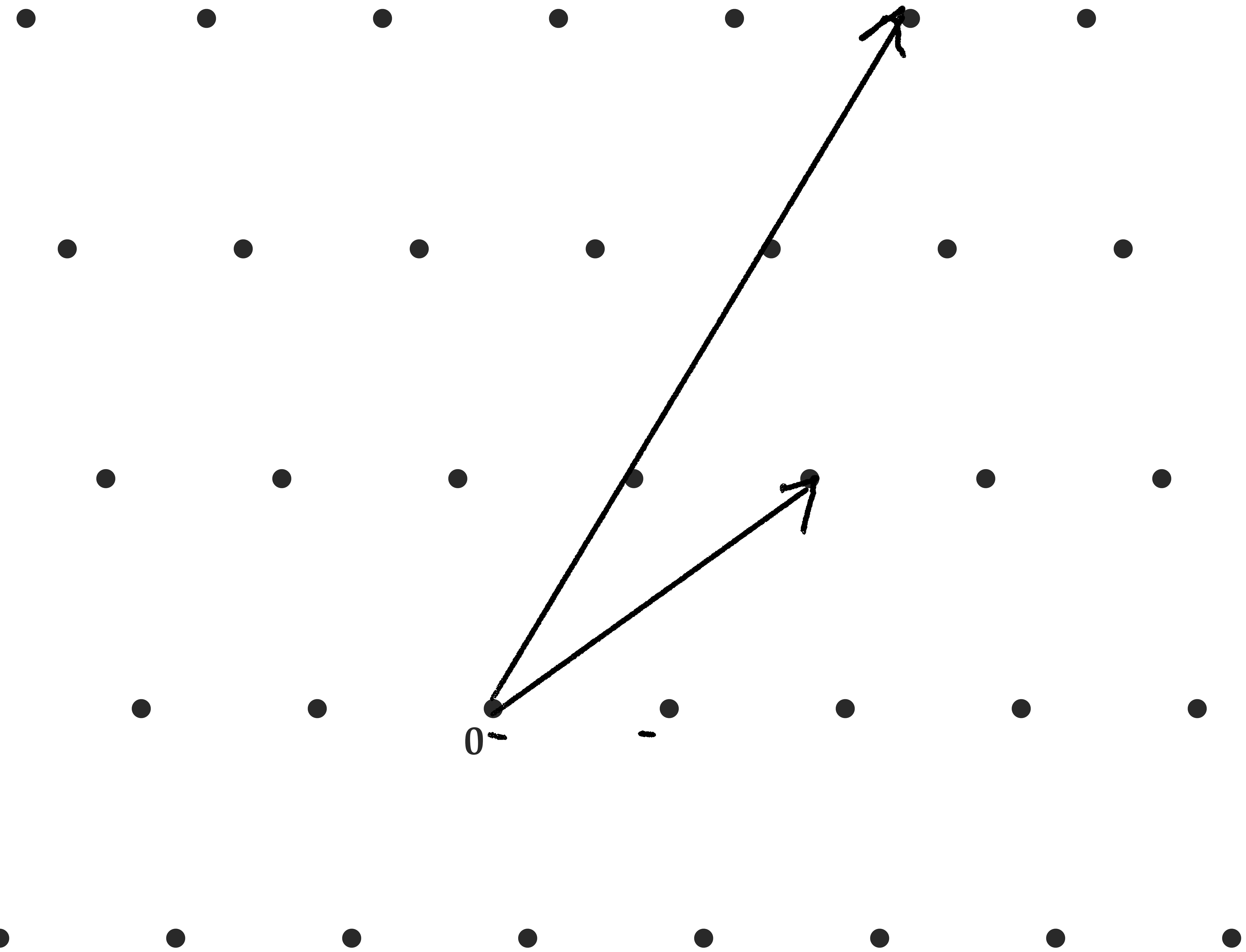
$$\mathrm{vol}(L) = \mathrm{vol}(\mathcal{P}(\mathbf{B})) = |\det(\mathbf{B})|$$

$$B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$



# Hard problems

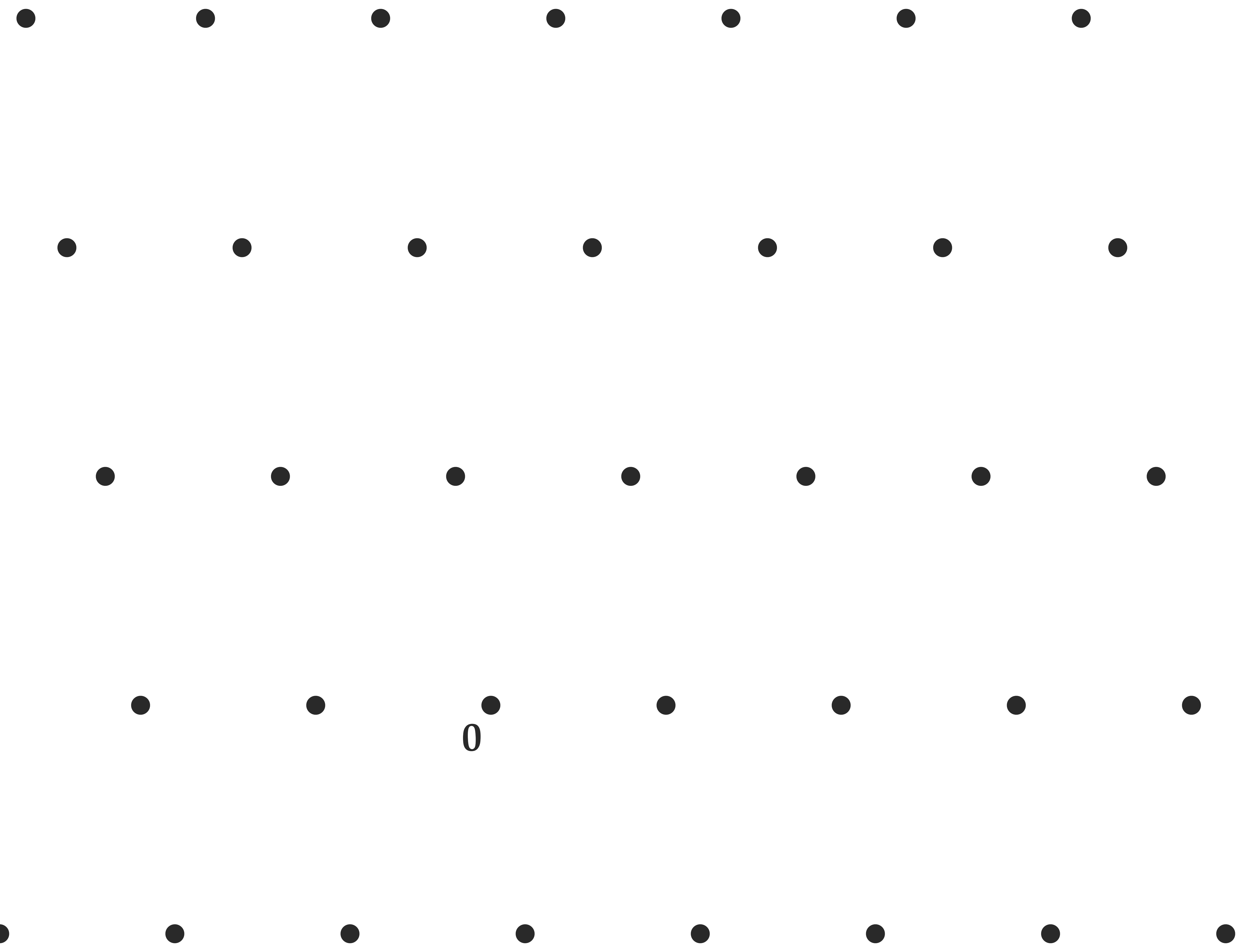
# Shortest vector problem




**The Shortest Vector Problem (SVP)**

**Input:** an arbitrary basis $\mathbf{B}$ of a lattice L.
**Question:** Find a shortest vector $\mathbf{v} \in L$.

# Shortest vector problem

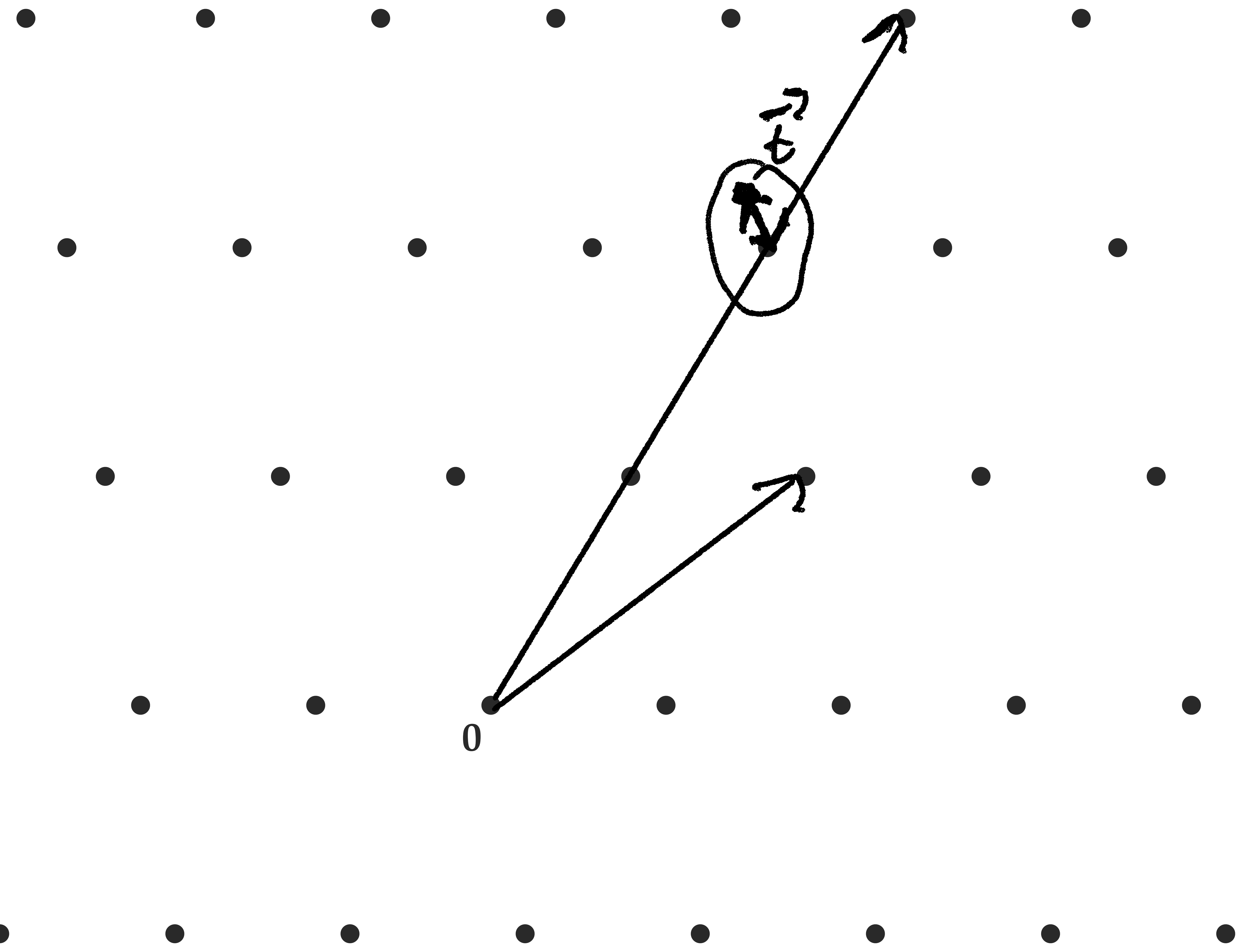



**The Shortest Vector Problem (SVP)**

**Input:** an arbitrary basis $\mathbf{B}$ of a lattice L.
**Question:** Find a shortest vector $\mathbf{v} \in L$.

$$||\mathbf{v}|| = \lambda_1(L)$$

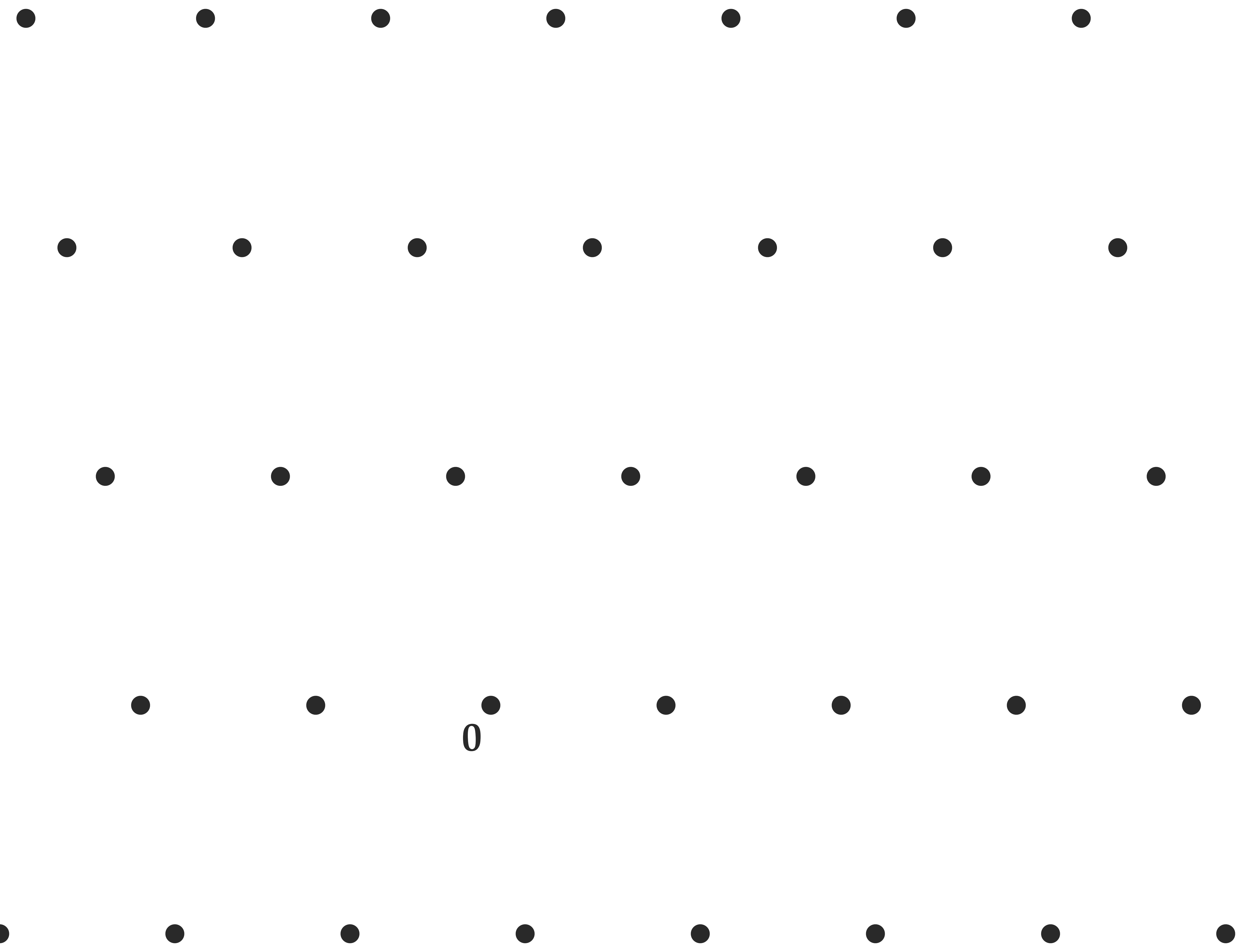
# Closest vector problem




**The Closest Vector Problem (CVP)**

**Input:** an arbitrary basis $\mathbf{B}$ of a lattice L and a target vector $\mathbf{t} \in \mathbb{R}^n$.
**Question:** Find a lattice vector $\mathbf{v} \in L$ that is closest to $\mathbf{t}$.

# Closest vector problem




**The Closest Vector Problem (CVP)**

**Input:** an arbitrary basis $\mathbf{B}$ of a lattice L and a target vector $\mathbf{t} \in \mathbb{R}^n$.
**Question:** Find a lattice vector $\mathbf{v} \in L$ that is closest to $\mathbf{t}$.

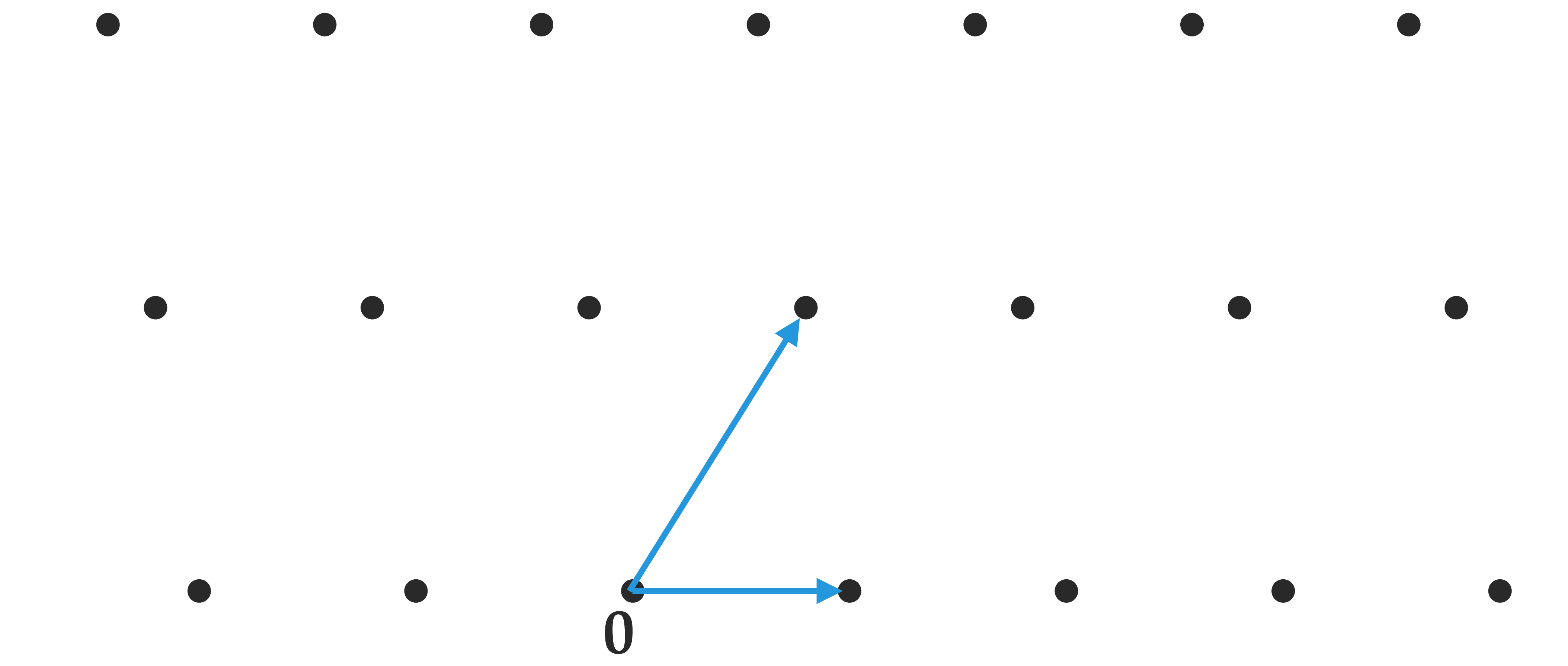
$$||\mathbf{t} - \mathbf{v}|| = \mathrm{dist}(L, \mathbf{t}) = \frac{1}{2}\lambda_1(L)$$

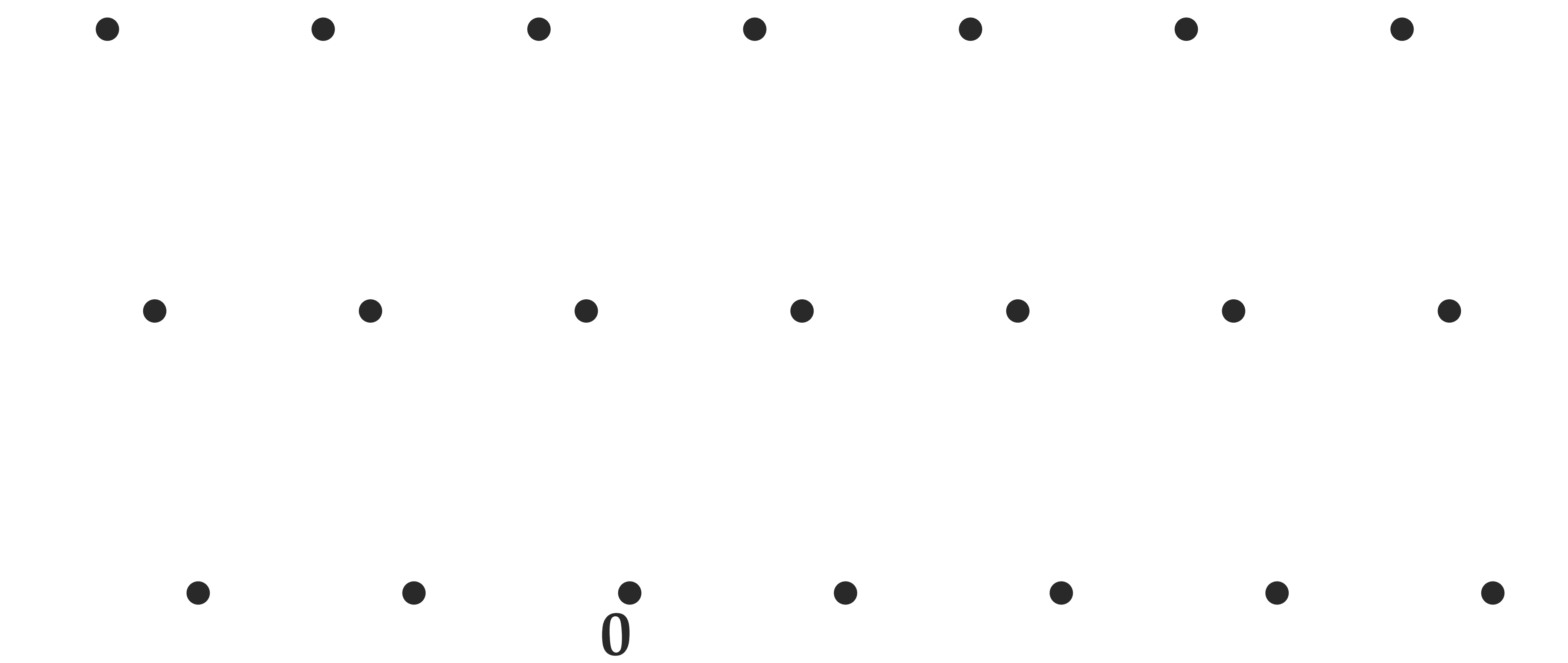
# Hardness in practice

In practice:

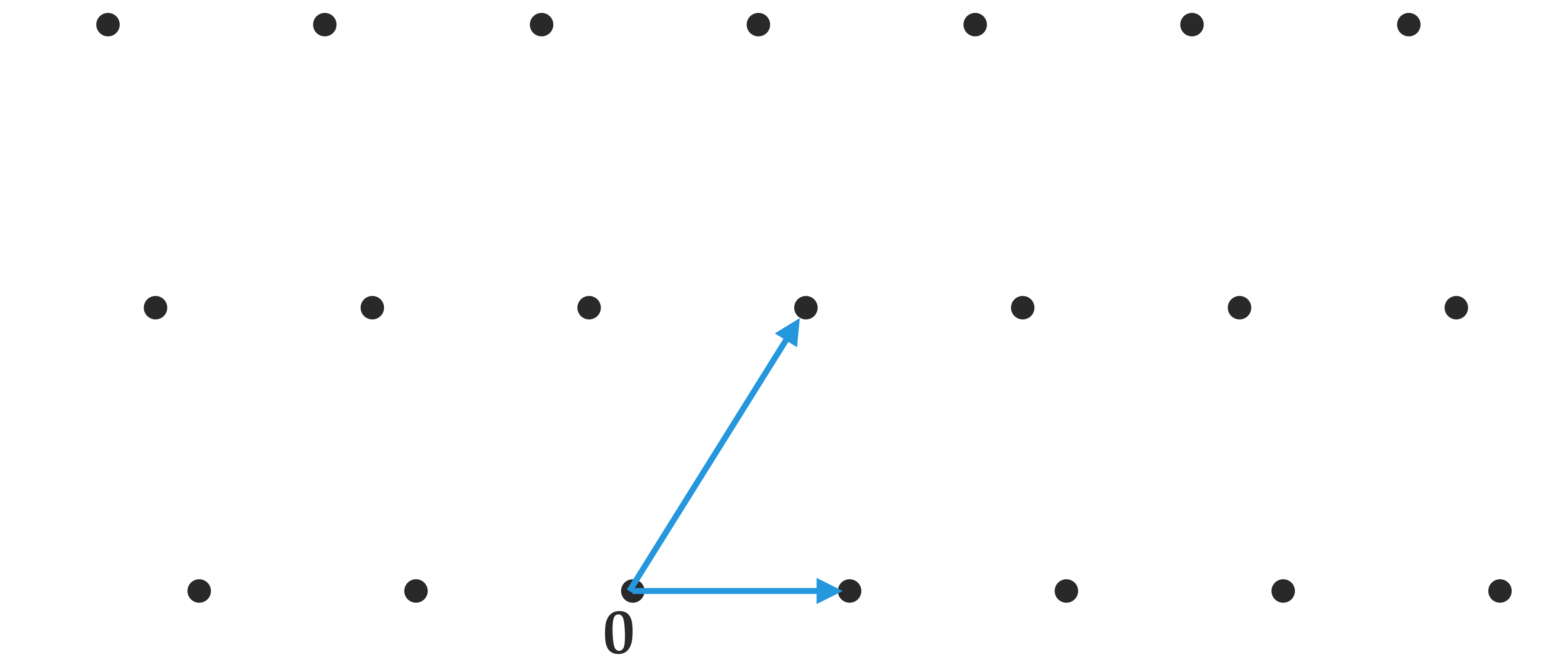
- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 60$ or $n = 80 \rightsquigarrow$ a few minutes on a personal laptop
- up to $n = 180 \rightsquigarrow$ few weeks on a big computer with good code
- from $n = 400$ to $n = 1000 \rightsquigarrow$ cryptography

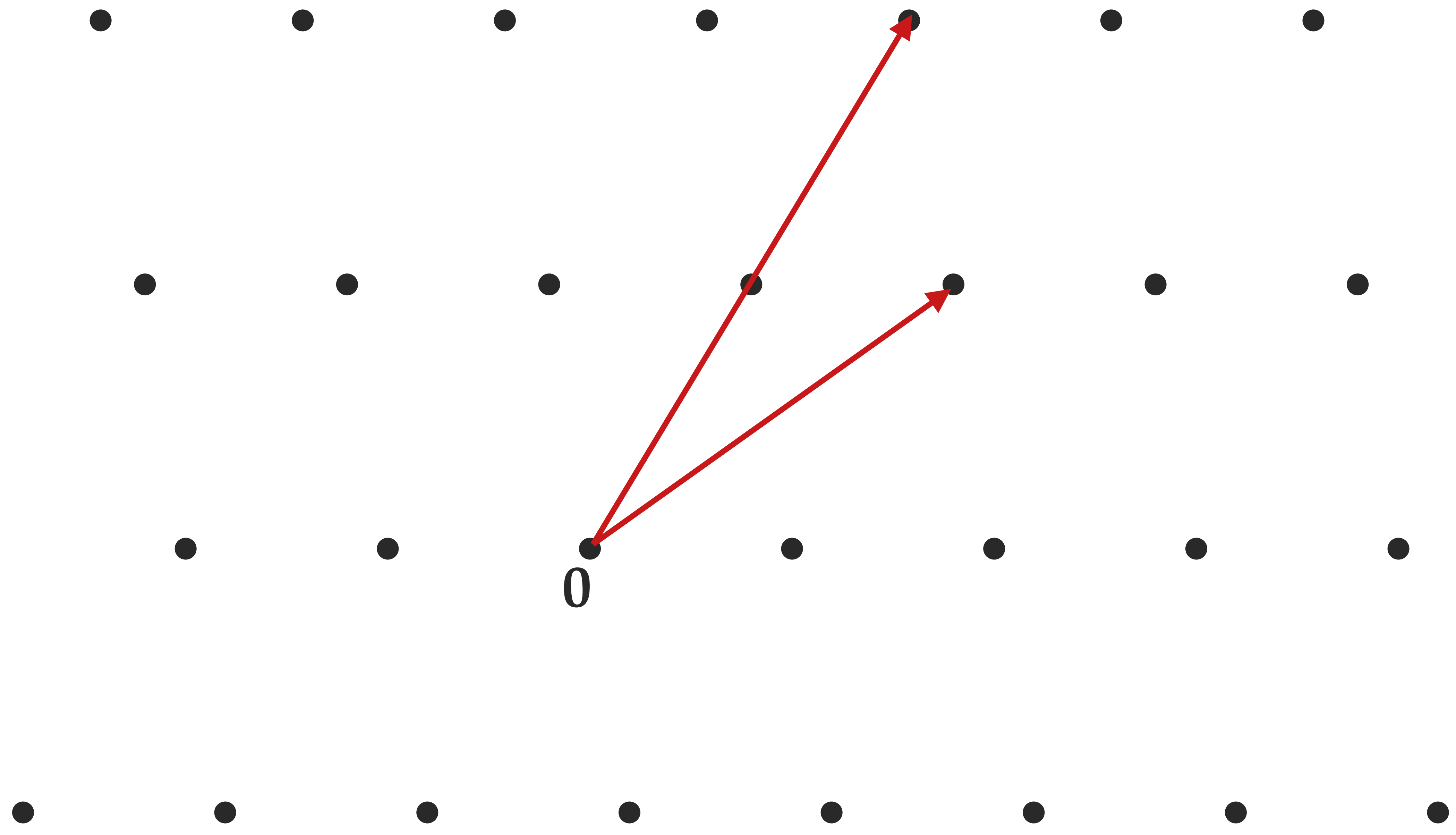
# Good basis VS bad basis



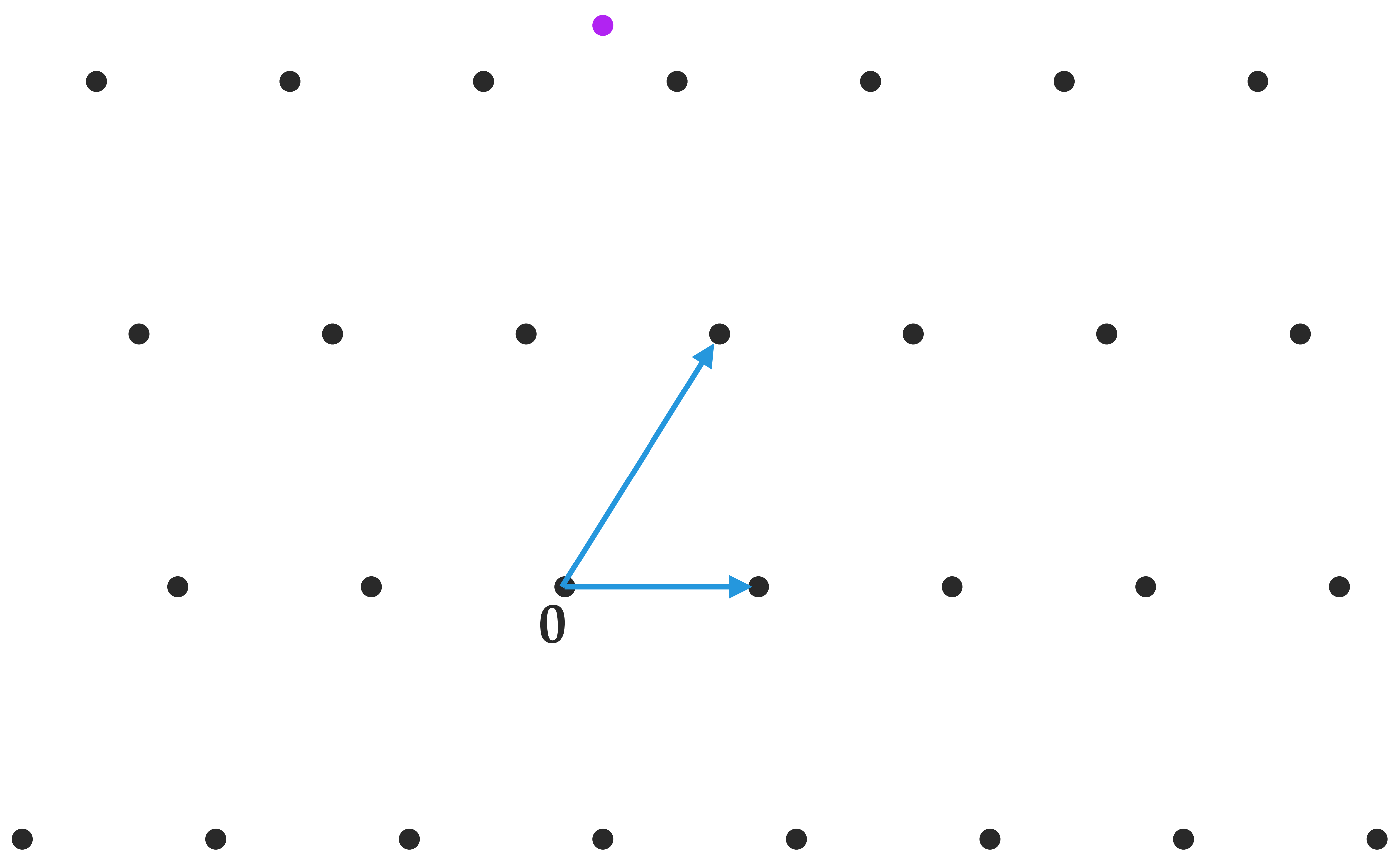
# Good basis VS bad basis



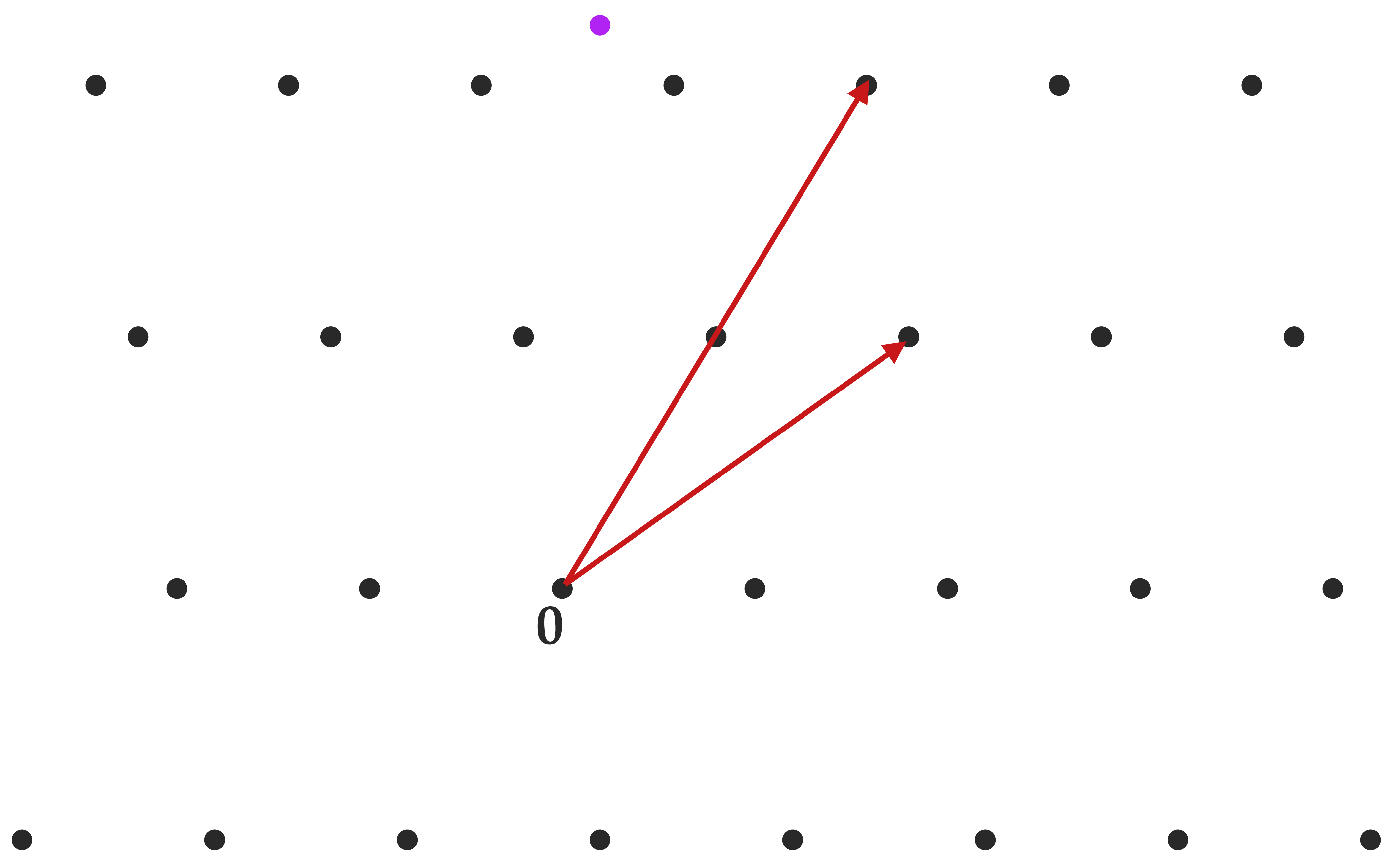



# Good basis VS bad basis

# Good basis VS bad basis





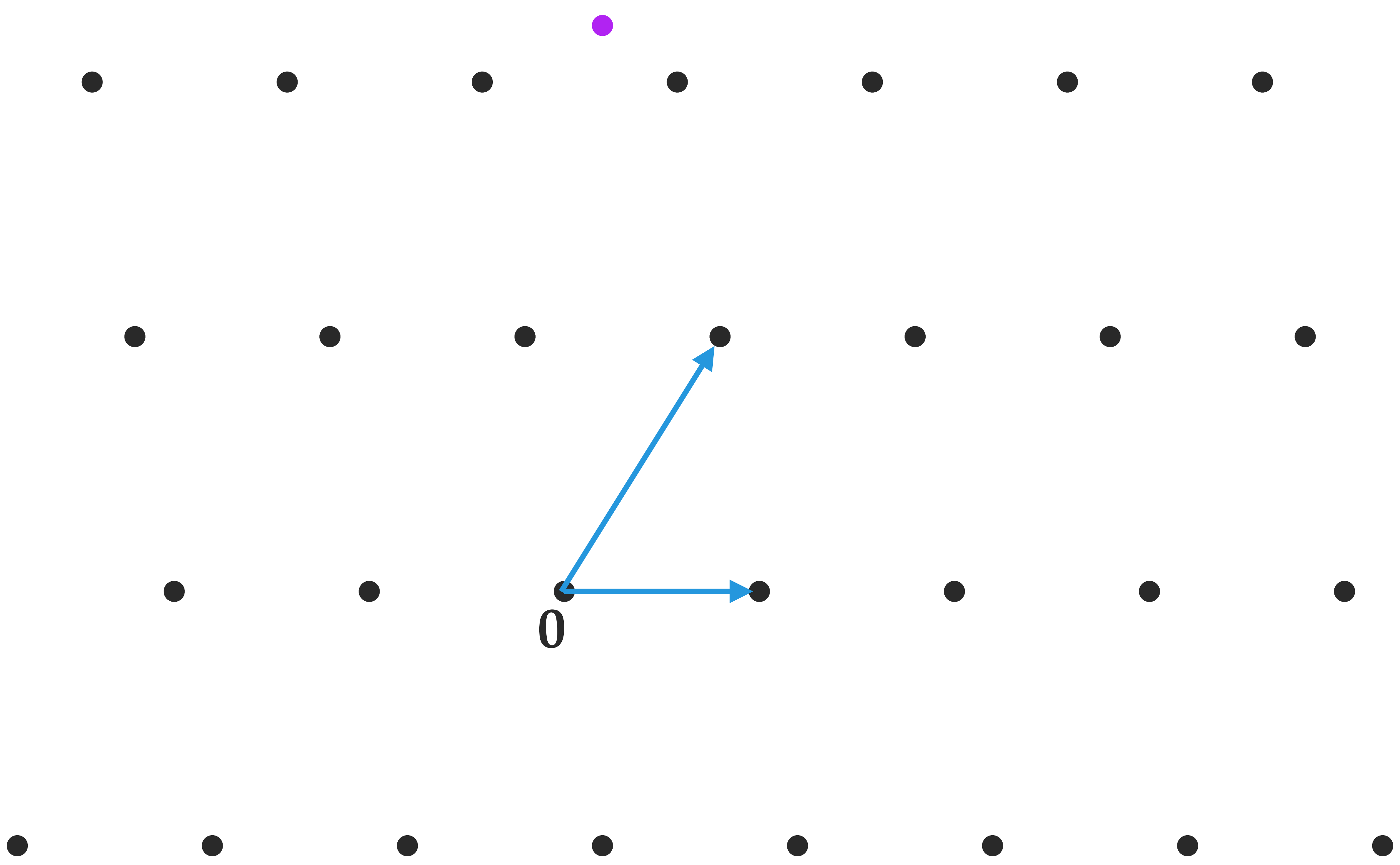



Good basis

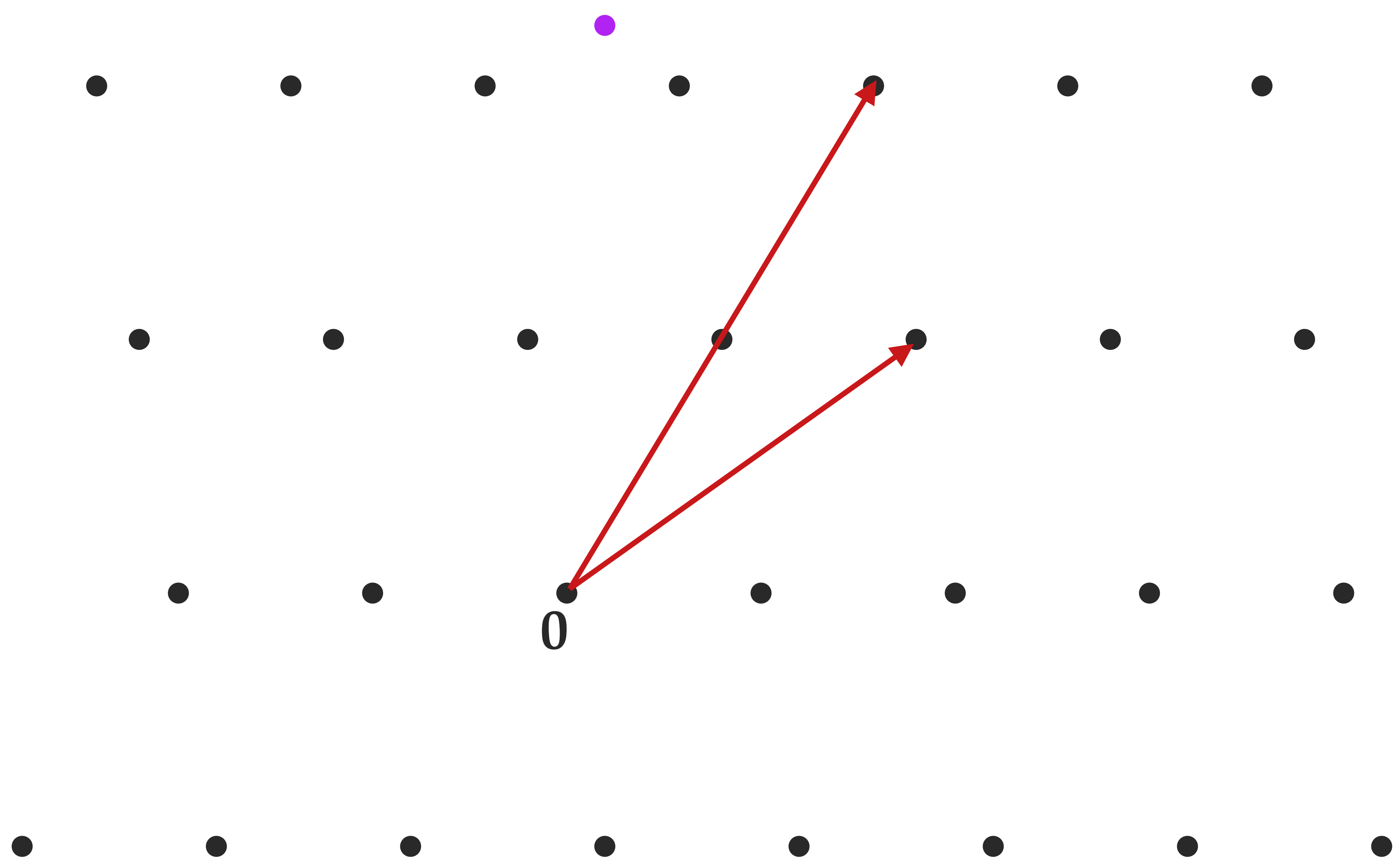
CVP input $\mathbf{t} = (1,11)$

Bad basis

# Good basis VS bad basis




Hard problem



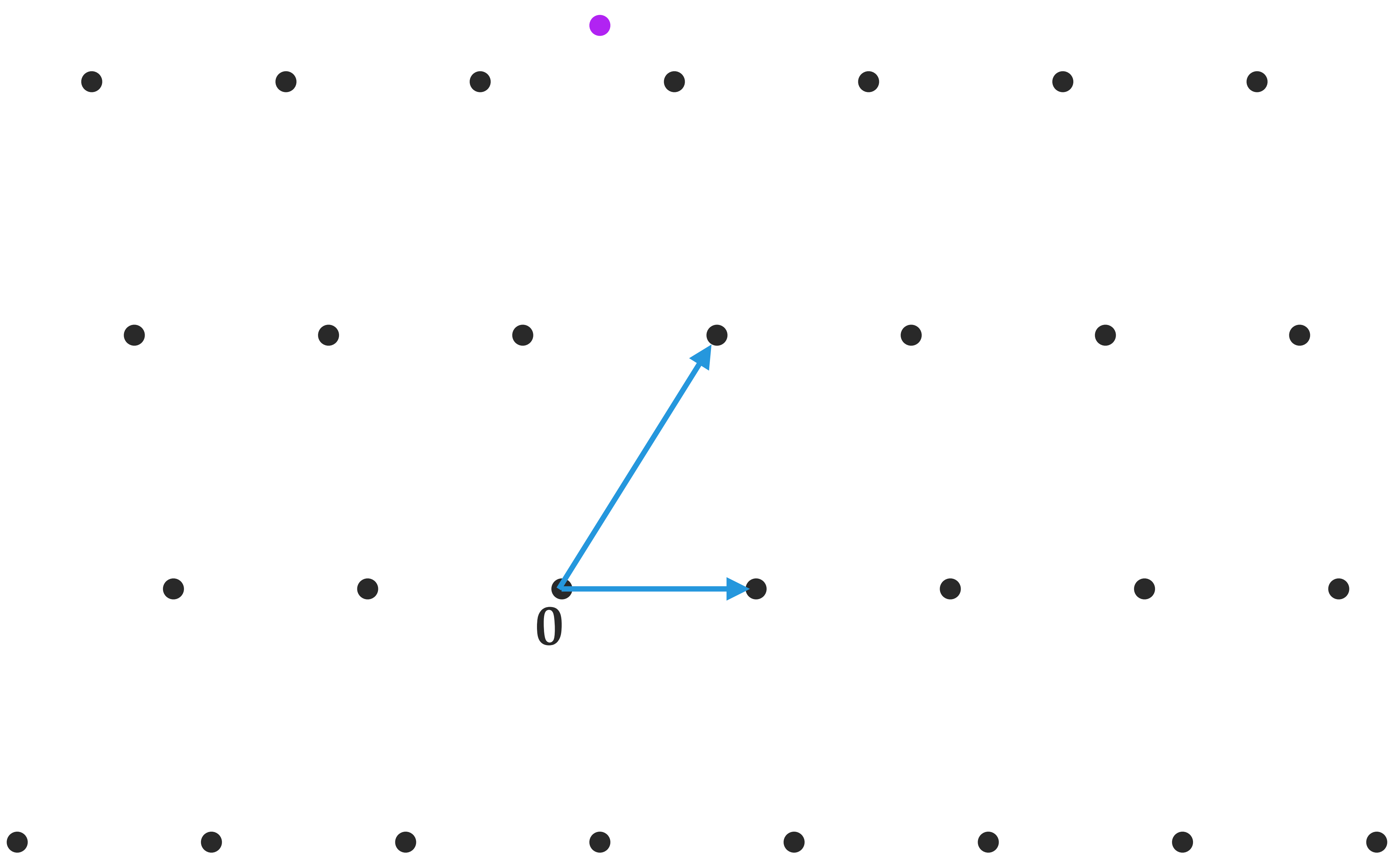

Good basis

CVP input $\mathbf{t} = (1,11)$

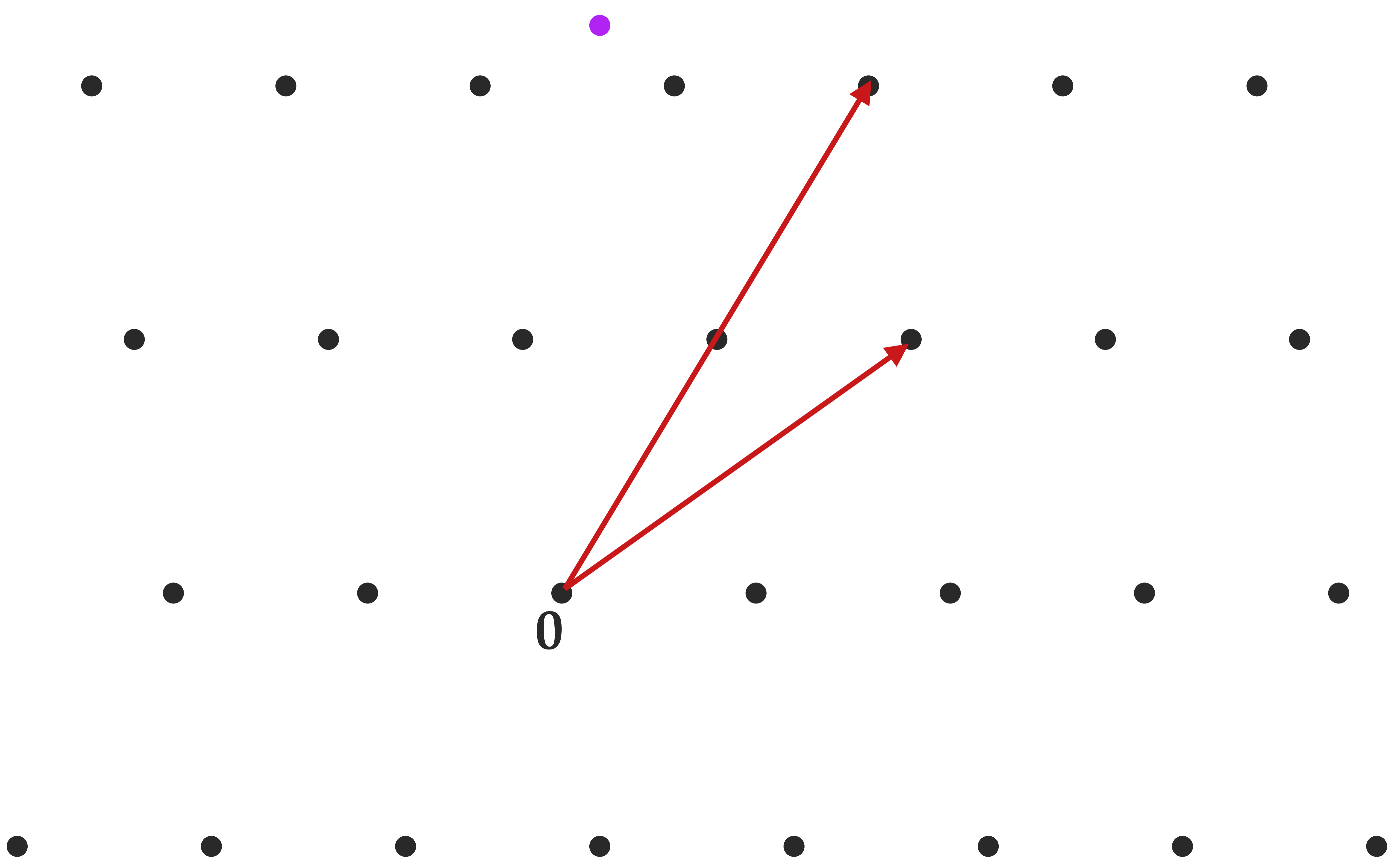
Bad basis

$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

# Good basis VS bad basis




Hard problem




Good basis

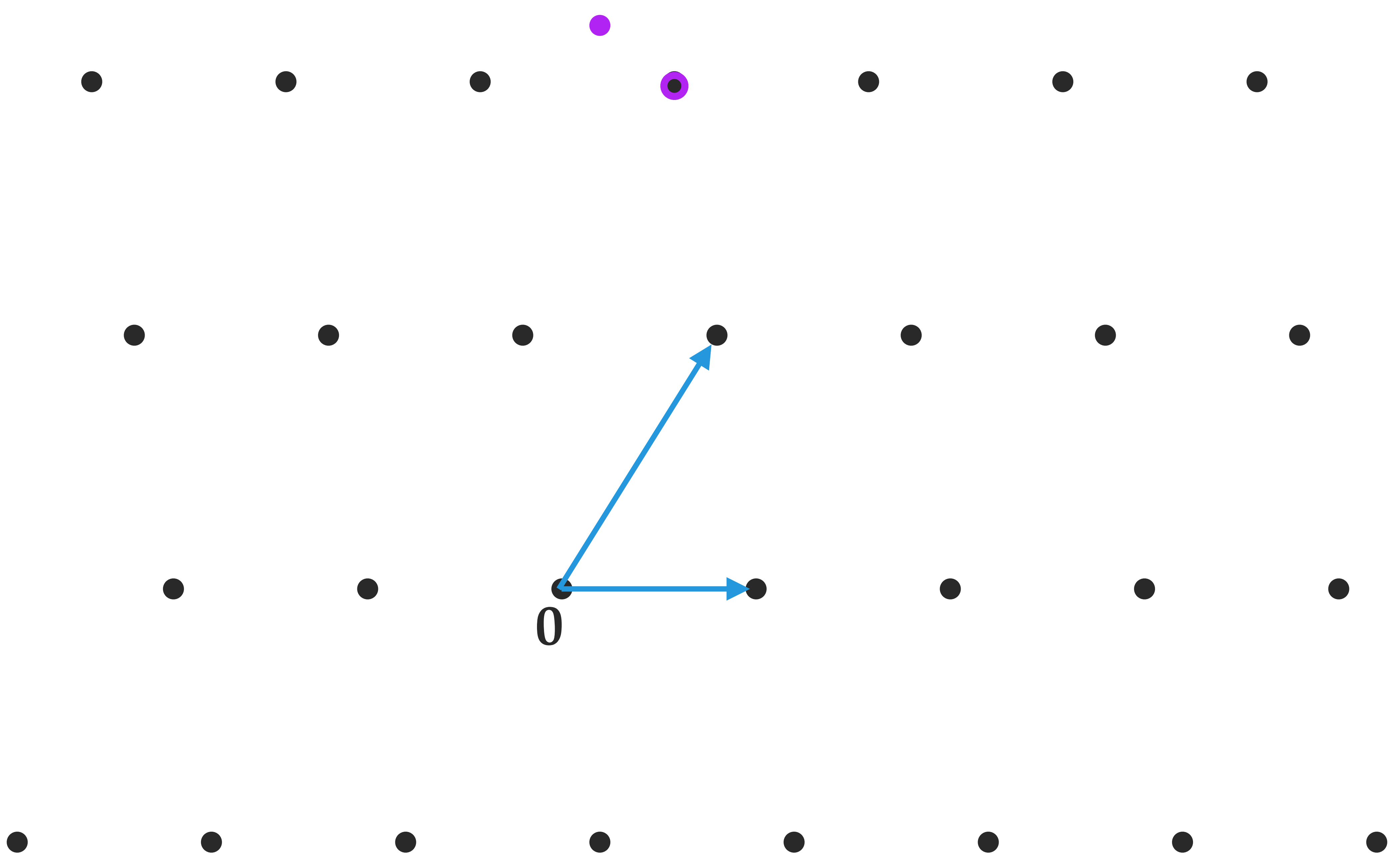
CVP input $\mathbf{t} = (1,11)$

Bad basis

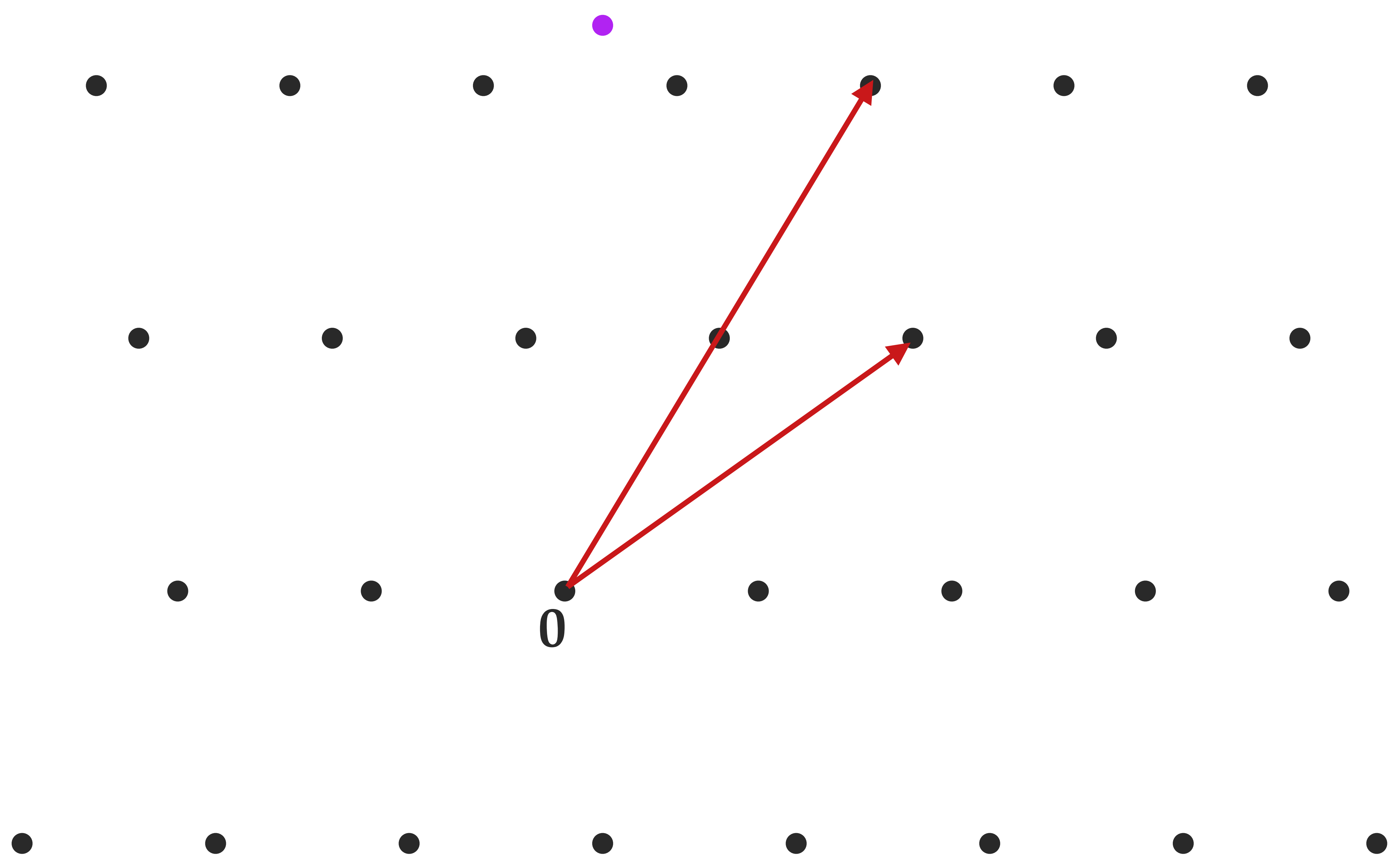
$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

# Good basis VS bad basis




Hard problem




Good basis

CVP input $\mathbf{t} = (1,11)$

Bad basis

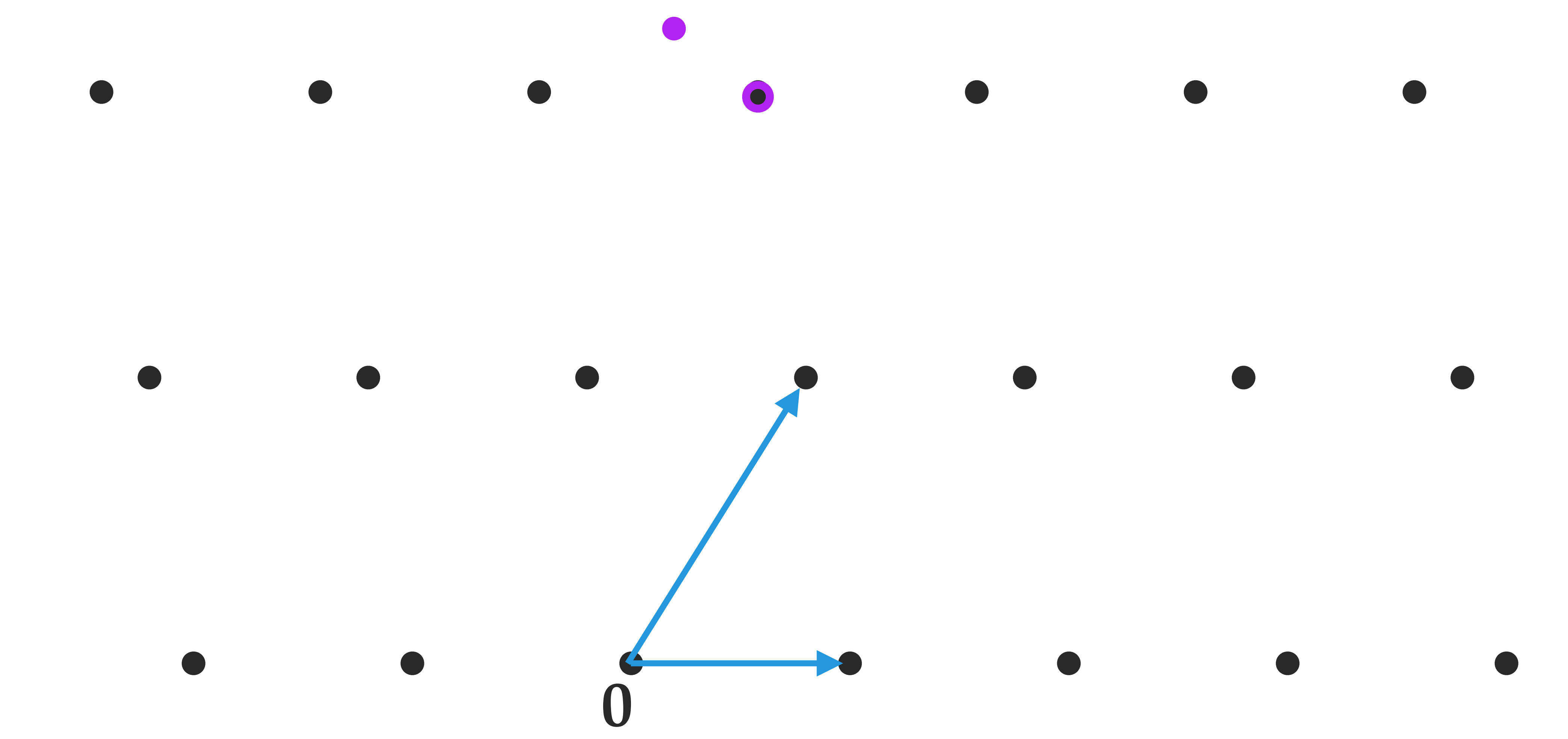
$$(\lambda_1, \lambda_2) \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

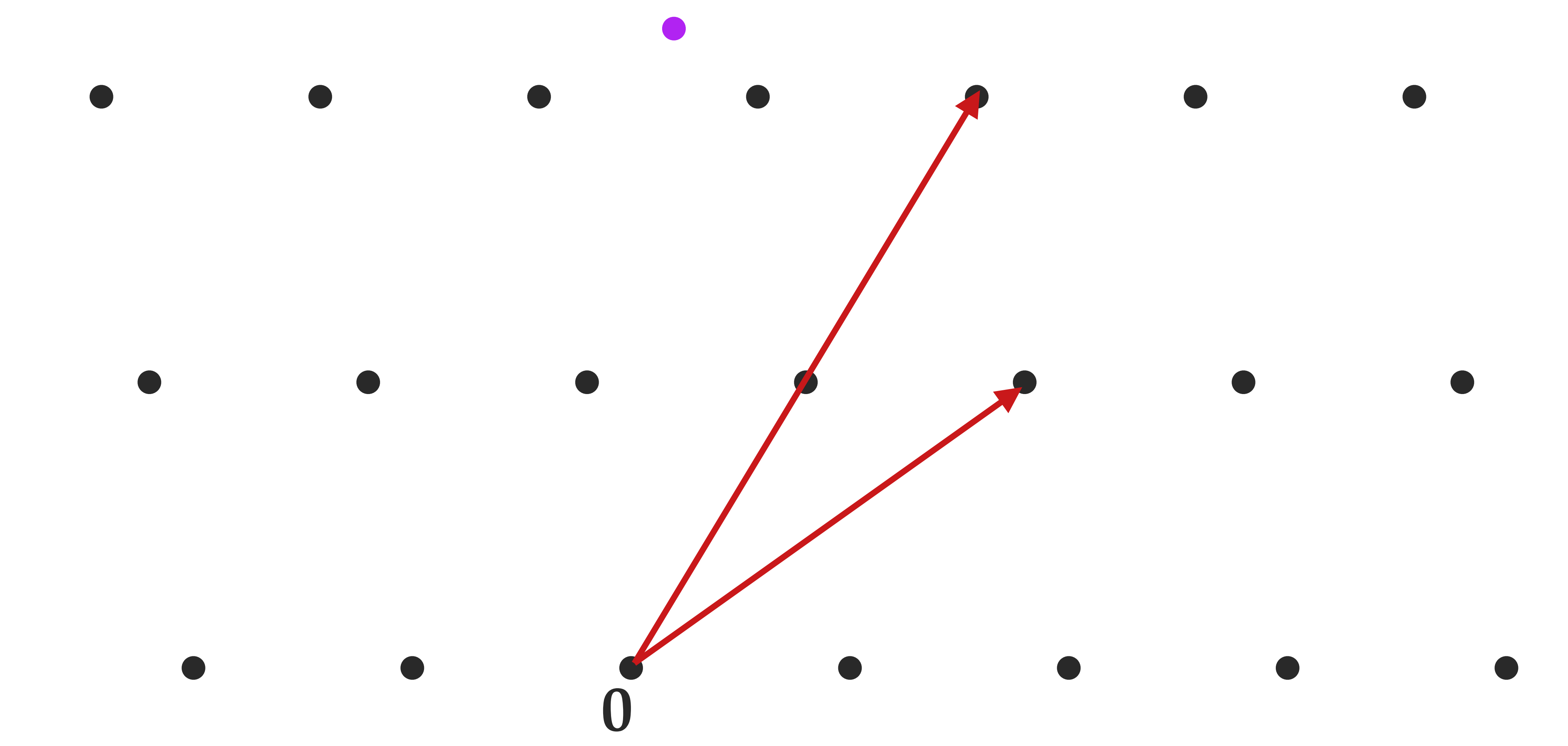
$$\Big\downarrow \text{rounding}$$

$$\mathbf{c} = -1\mathbf{b}_1 + 2\mathbf{b}_2$$

# Good basis VS bad basis




Hard problem




Good basis

CVP input $\mathbf{t} = (1,11)$

Bad basis

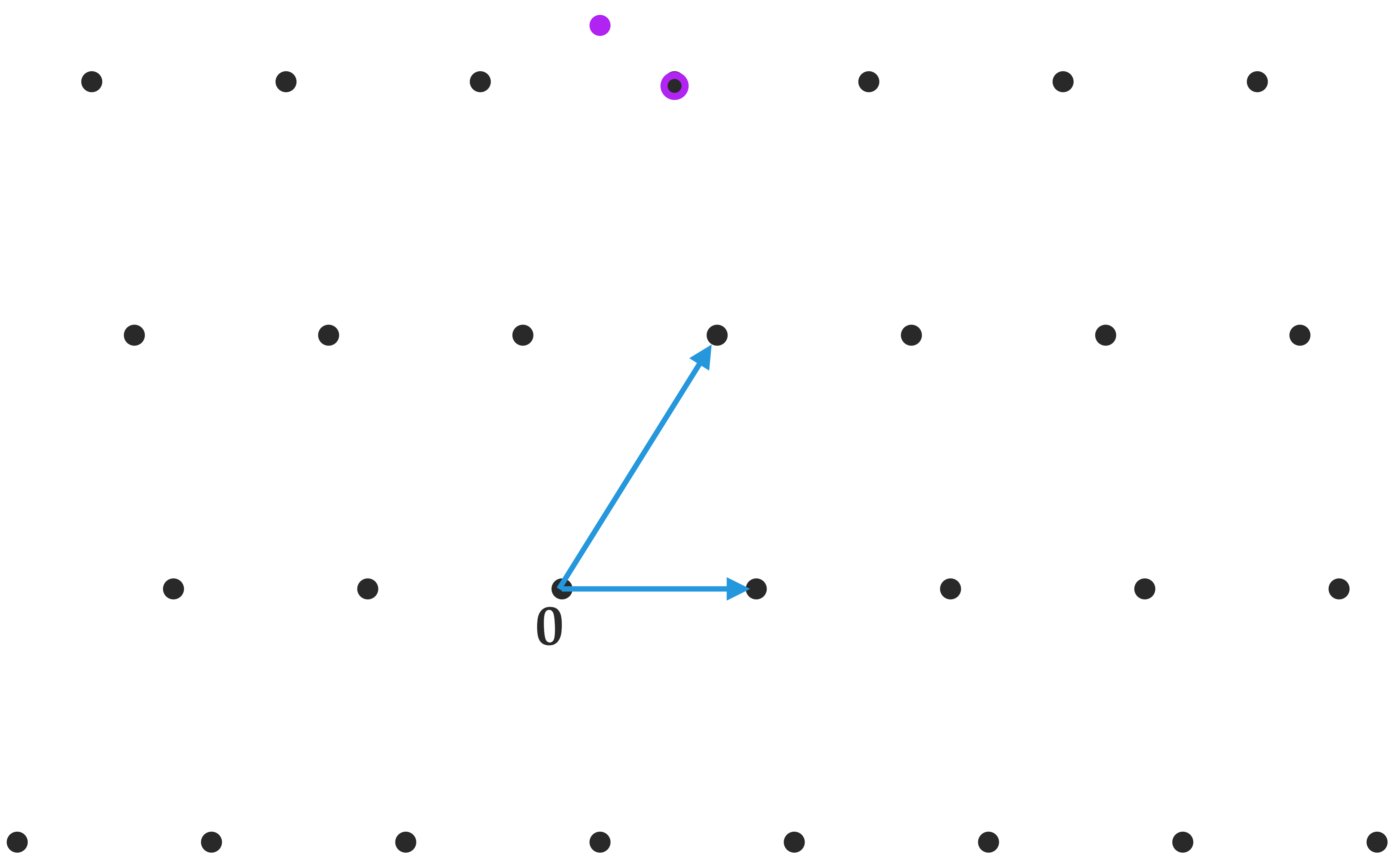
$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

$$\mathbf{t} \ = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

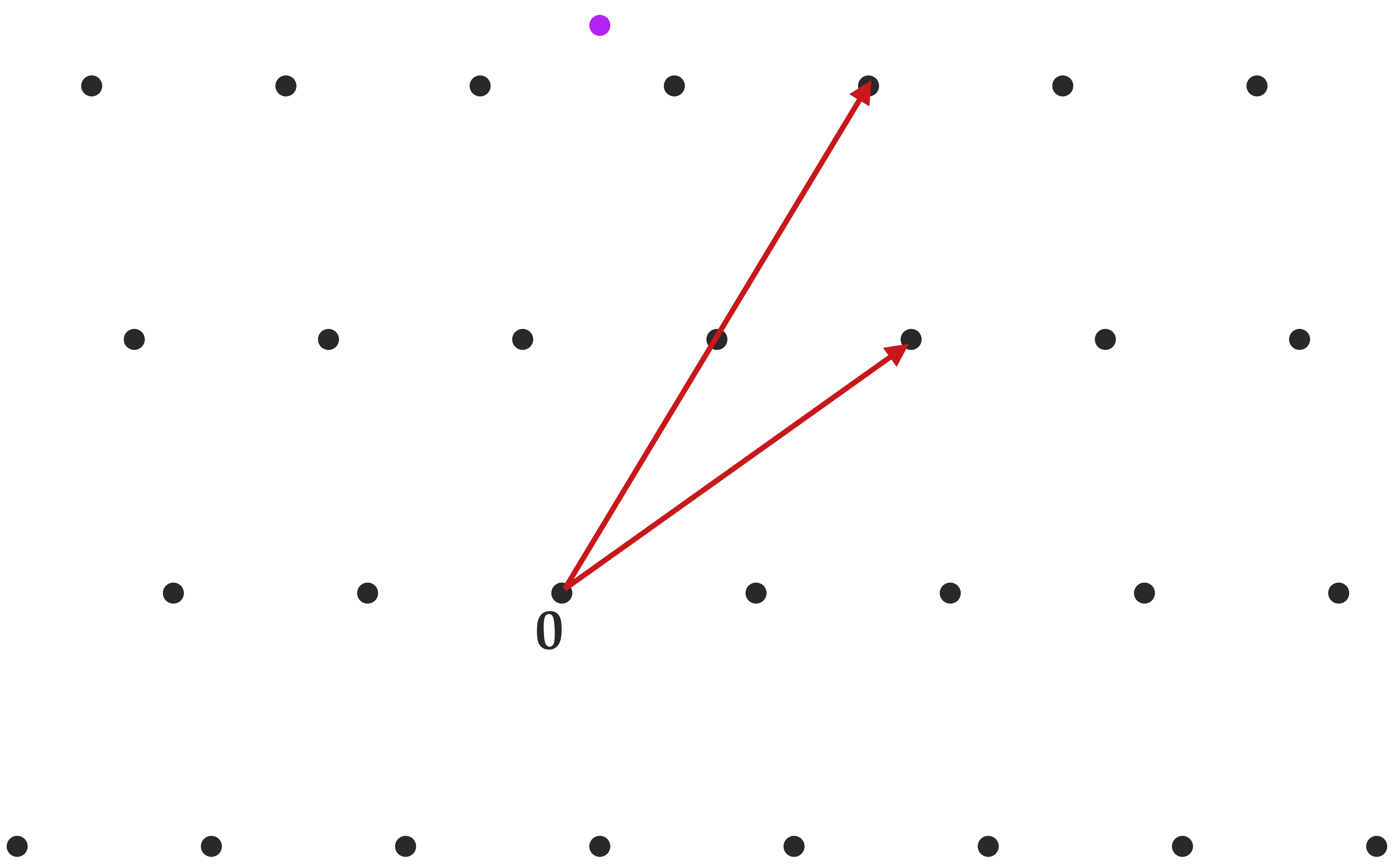
rounding

$$\mathbf{c} \ = -1\mathbf{b}_1 + 2\mathbf{b}_2 \quad \checkmark$$

# Good basis VS bad basis




Hard problem




Good basis

$$(\lambda_1, \lambda_2)\begin{pmatrix}4 & 0\\ 3 & 5\end{pmatrix} = (1,11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

$\downarrow$ rounding

$$\mathbf{c} = -1\mathbf{b}_1 + 2\mathbf{b}_2 \quad ✓$$

CVP input $\mathbf{t} = (1,11)$

Bad basis

$$(\lambda_1, \lambda_2)\begin{pmatrix}7 & 5\\ 6 & 10\end{pmatrix} = (1,11)$$

# Good basis VS bad basis


Good basis

CVP input $\mathbf{t} = (1,11)$

Bad basis

$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

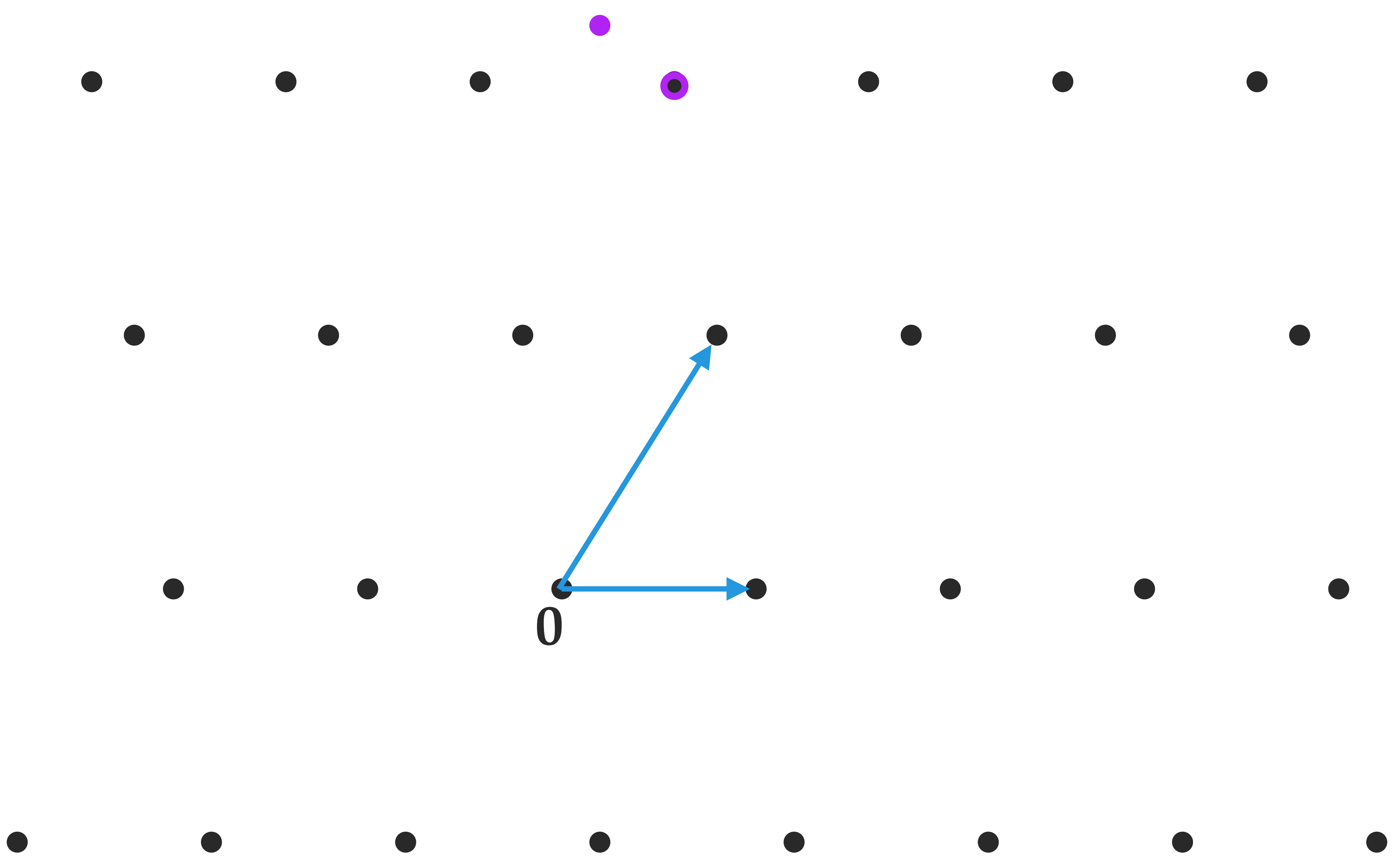
$\downarrow$ rounding

$$\mathbf{c} = -1\mathbf{b}_1 + 2\mathbf{b}_2 \quad ✓$$

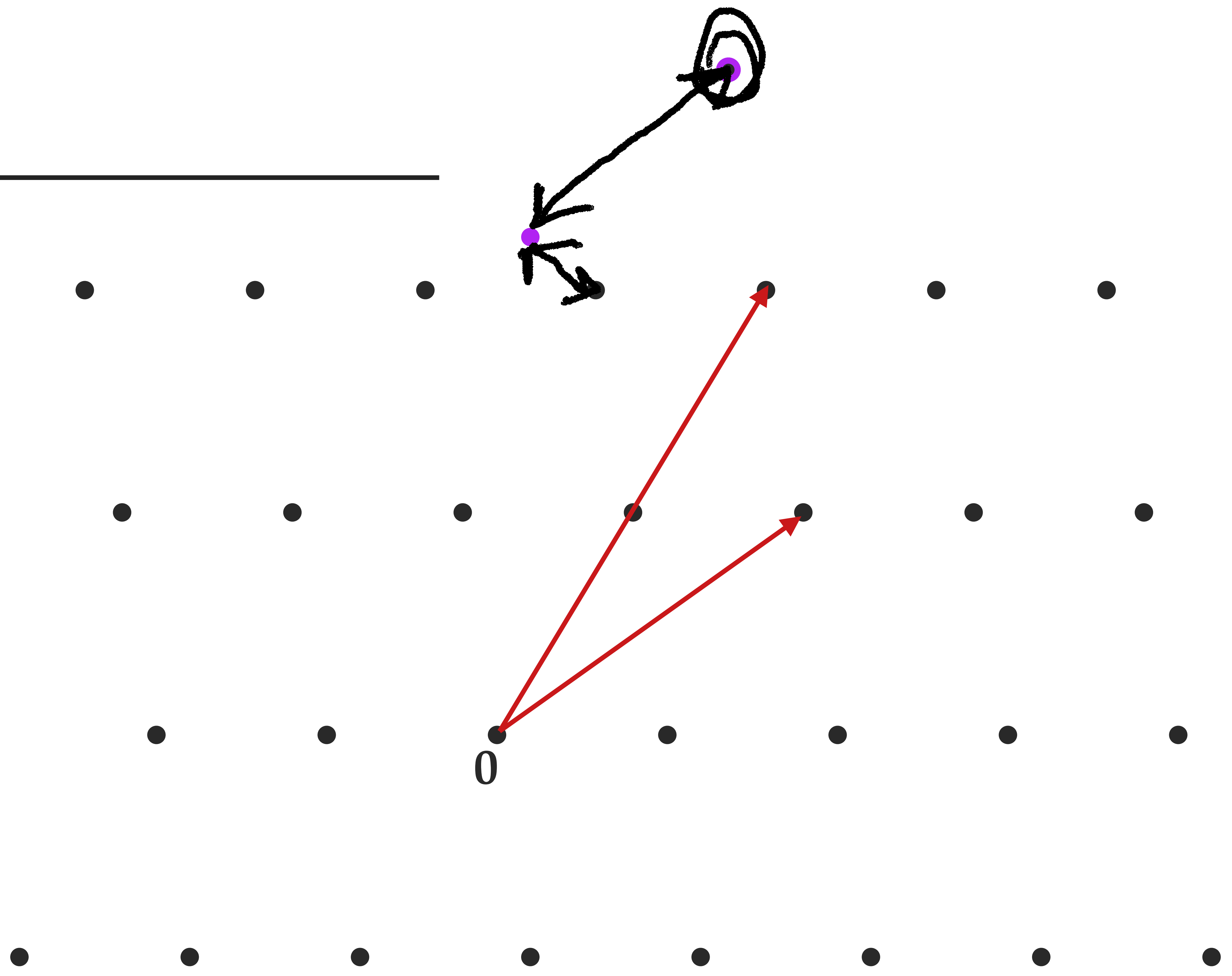
$$(\lambda_1, \lambda_2)\begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix} = (1,11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}'_1 + 1.8\,\mathbf{b}'_2$$

# Good basis VS bad basis




Hard problem




CVP input $\mathbf{t} = (1,11)$

**Good basis**

$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1,11)$$

$$\mathbf{t} \;= -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

$\big\downarrow$ rounding

$$\mathbf{c} \;= -1\mathbf{b}_1 + 2\mathbf{b}_2 \quad ✓$$

**Bad basis**

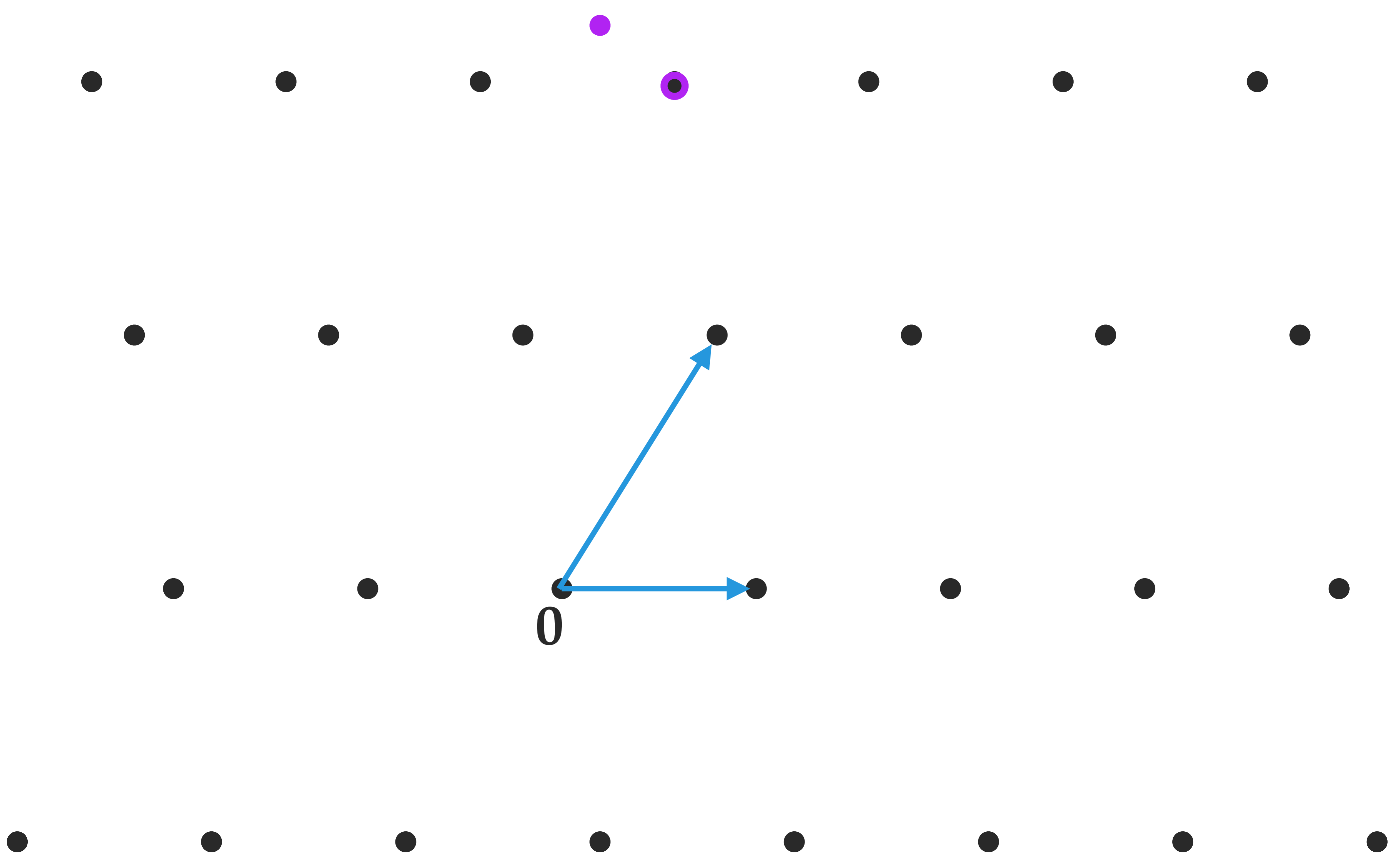
$$(\lambda_1, \lambda_2)\begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix} = (1,11)$$

$$\mathbf{t} \;= -1.4\,\mathbf{b}'_1 + 1.8\,\mathbf{b}'_2$$

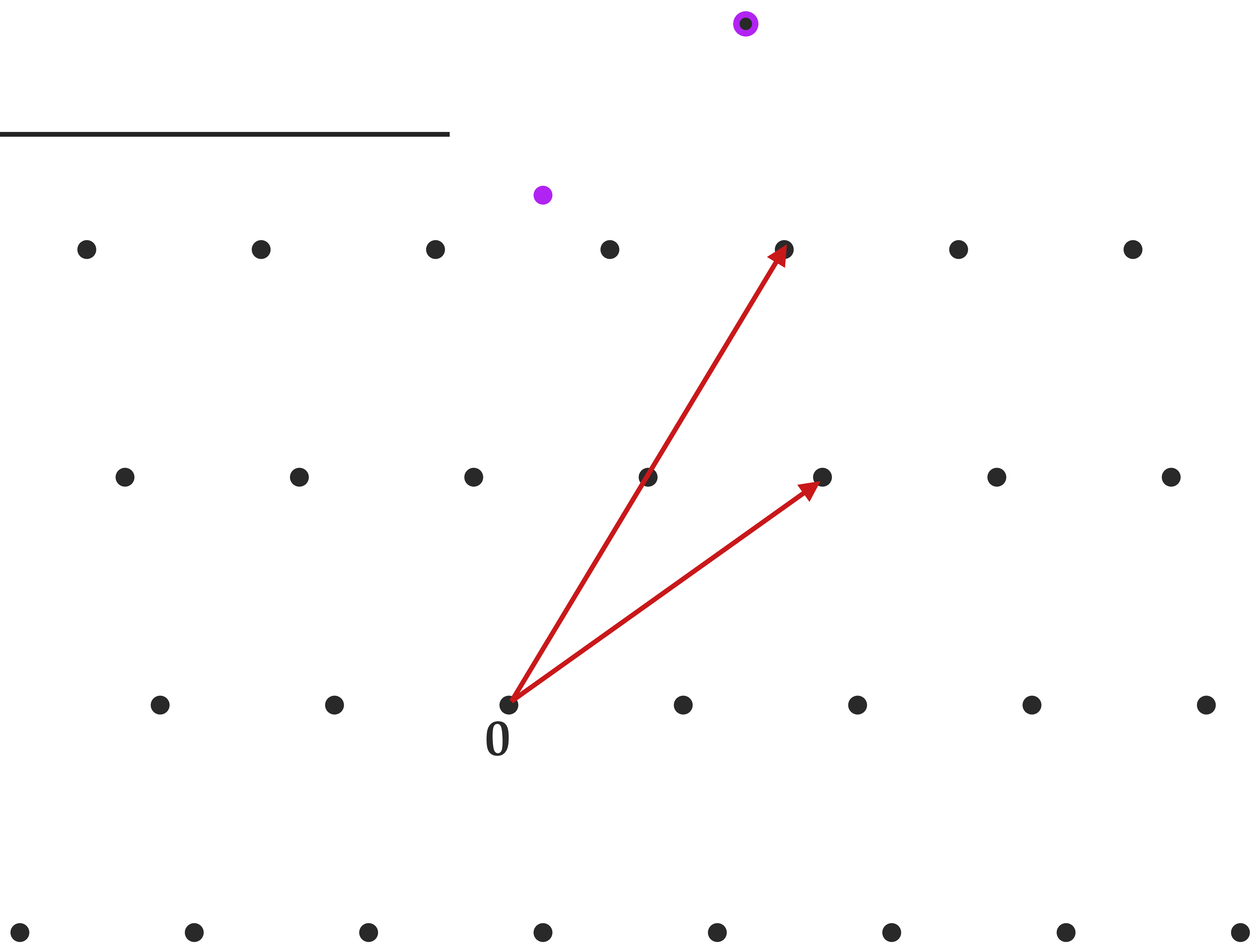
$\big\downarrow$ rounding

$$\mathbf{c} \;= -1\mathbf{b}'_1 + 2\mathbf{b}'_2$$

# Good basis VS bad basis




Hard problem




Good basis

CVP input $\mathbf{t} = (1{,}11)$

Bad basis

$$(\lambda_1, \lambda_2)\begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix} = (1{,}11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1 + 2.2\,\mathbf{b}_2$$

$$\Big\downarrow \text{rounding}$$

$$\mathbf{c} = -1\mathbf{b}_1 + 2\mathbf{b}_2 \quad ✓$$

$$(\lambda_1, \lambda_2)\begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix} = (1{,}11)$$

$$\mathbf{t} = -1.4\,\mathbf{b}_1' + 1.8\,\mathbf{b}_2'$$

$$\Big\downarrow \text{rounding}$$

$$\mathbf{c} = -1\mathbf{b}_1' + 2\mathbf{b}_2' \quad ✗$$

# Lagrange-Gauss lattice reduction

In dimension 2, takes as input an arbitrary basis $\mathbf{b}'_1, \mathbf{b}'_2$ of a lattice $L$ and outputs a 'best' basis $\mathbf{b}_1, \mathbf{b}_2$.

# Lagrange-Gauss lattice reduction

In dimension 2, takes as input an arbitrary basis $\mathbf{b}_1', \mathbf{b}_2'$ of a lattice $L$ and outputs a 'best' basis $\mathbf{b}_1, \mathbf{b}_2$.

**Do**

- **Swap:** If $||\mathbf{b}_1|| > ||\mathbf{b}_2||$, then swap $\mathbf{b}_1$ and $\mathbf{b}_2$.
- **Reduce:** While $||\mathbf{b}_1 \pm \mathbf{b}_2|| < ||\mathbf{b}_2||$, replace $\mathbf{b}_2 \leftarrow \mathbf{b}_2 \pm \mathbf{b}_1$.

**Until** no progress is made from an iteration in the loop

# Lagrange-Gauss lattice reduction

In dimension 2, takes as input an arbitrary basis $\mathbf{b}_1', \mathbf{b}_2'$ of a lattice $L$ and outputs a 'best' basis $\mathbf{b}_1, \mathbf{b}_2$.

**Do**

- **Swap:** If $||\mathbf{b}_1|| > ||\mathbf{b}_2||$, then swap $\mathbf{b}_1$ and $\mathbf{b}_2$.
- **Reduce:** While $||\mathbf{b}_1 \pm \mathbf{b}_2|| < ||\mathbf{b}_2||$, replace $\mathbf{b}_2 \leftarrow \mathbf{b}_2 \pm \mathbf{b}_1$.

**Until** no progress is made from an iteration in the loop

Assignment exercise: iterate algorithm for $\mathbf{b}_1 = \begin{pmatrix} 144 \\ 0 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} 89 \\ 1 \end{pmatrix}$.

# Basis reduction algorithms

- Lagrange-Gauss reduction (in two dimensions)
- LLL
- BKZ
- Enumeration
- Sieving

# Cryptographic constructions

# Keygen

Solving the hard problems (SVP & CVP) with a good basis is easy and solving them with a bad basis is hard.

# Keygen

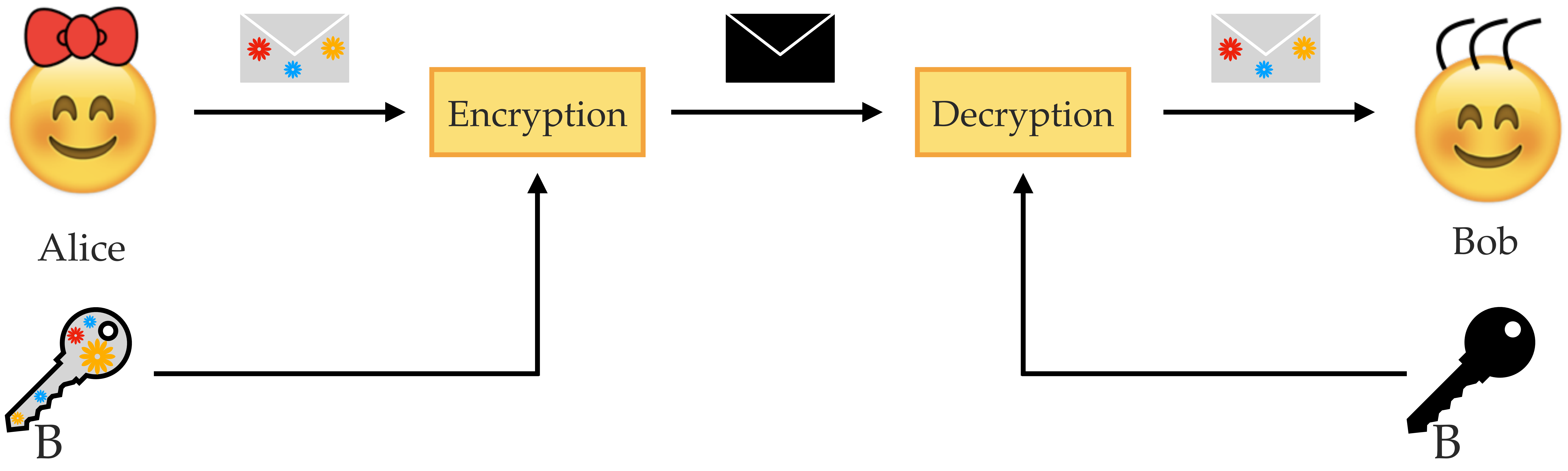
Solving the hard problems (SVP & CVP) with a good basis is easy and solving them with a bad basis is hard.

Secret key: a good basis

Public key: a bad basis

Gives rise to assumptions like NTRU, SIS, LWE

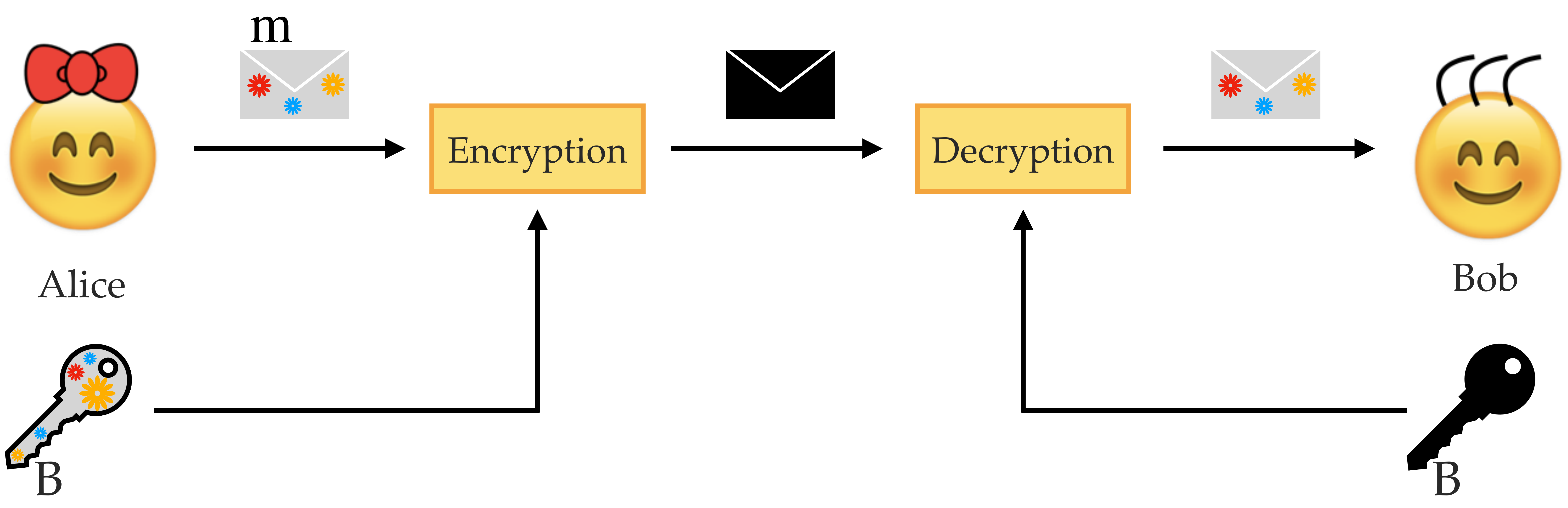
# The GGH encryption scheme




$R = \begin{pmatrix} \mathbf{r}_1 \\ \cdots \\ \mathbf{r}_n \end{pmatrix}$

$B = \begin{pmatrix} \mathbf{b}_1 \\ \cdots \\ \mathbf{b}_n \end{pmatrix}$

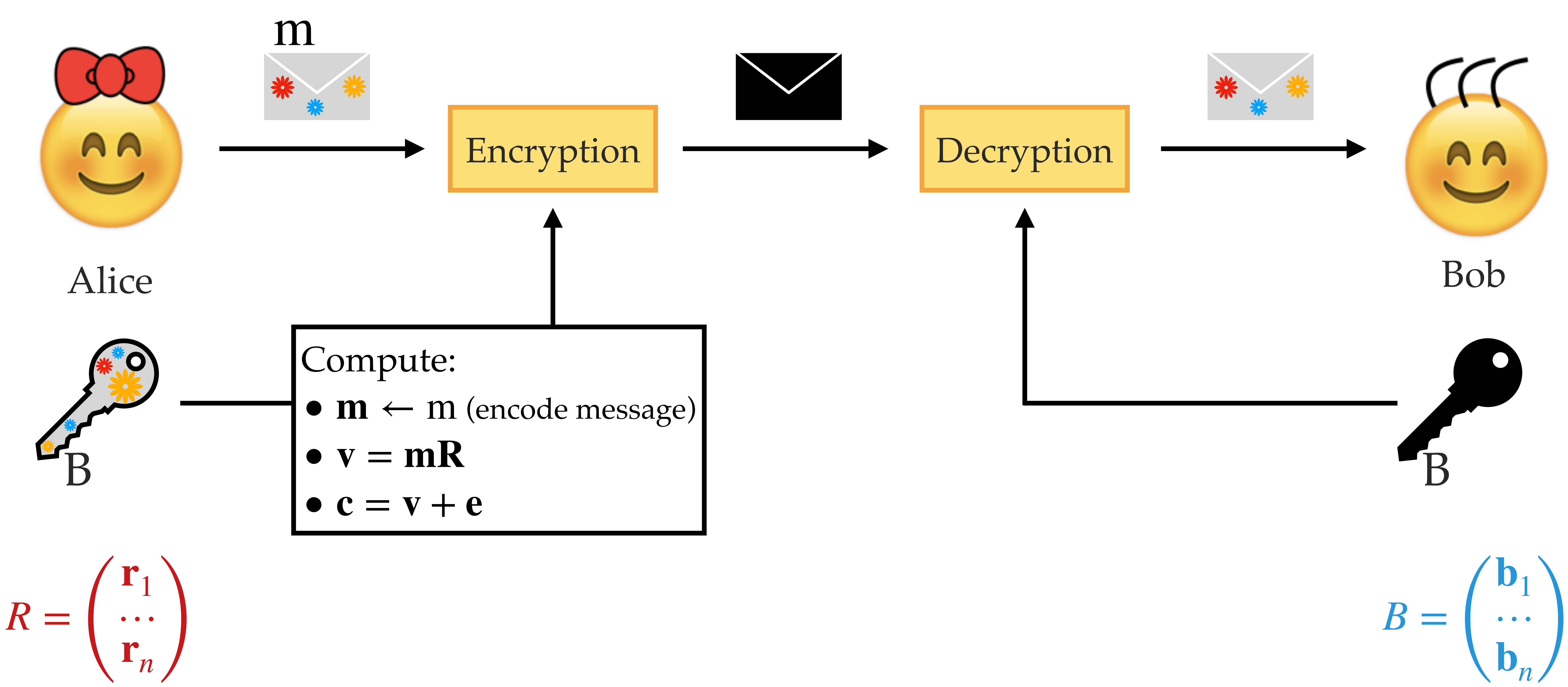
# The GGH encryption scheme



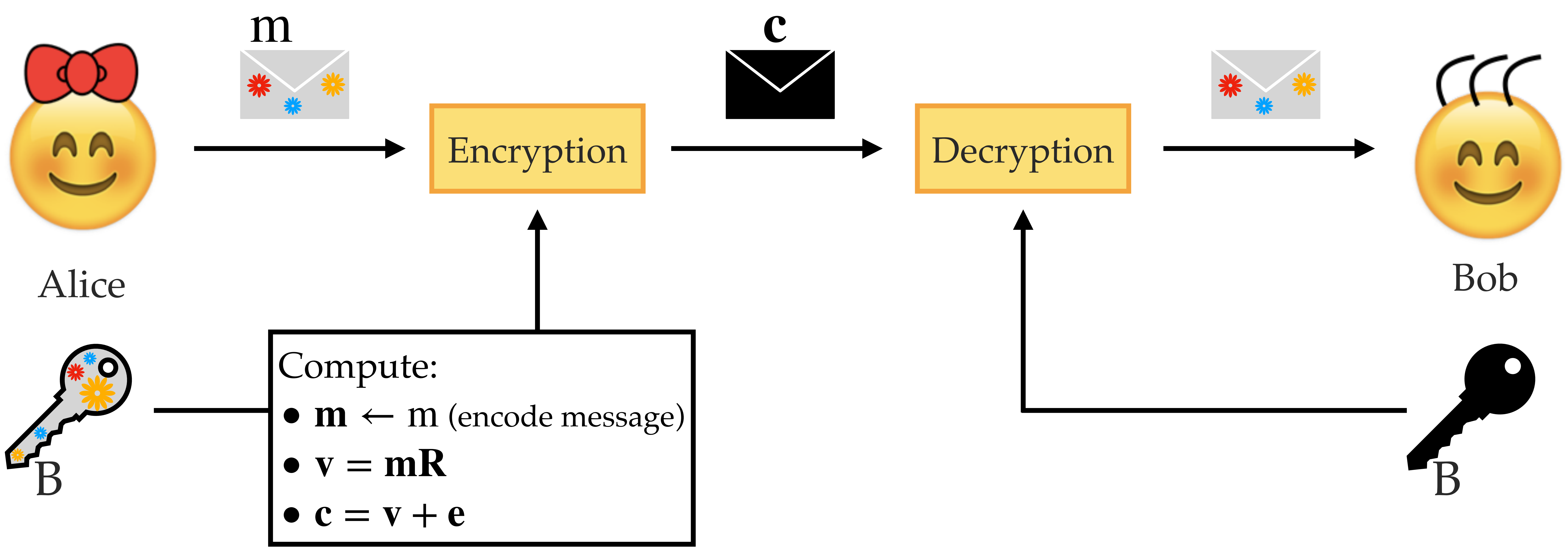

$$R = \begin{pmatrix} \mathbf{r}_1 \\ \cdots \\ \mathbf{r}_n \end{pmatrix}$$

$$B = \begin{pmatrix} \mathbf{b}_1 \\ \cdots \\ \mathbf{b}_n \end{pmatrix}$$

# The GGH encryption scheme



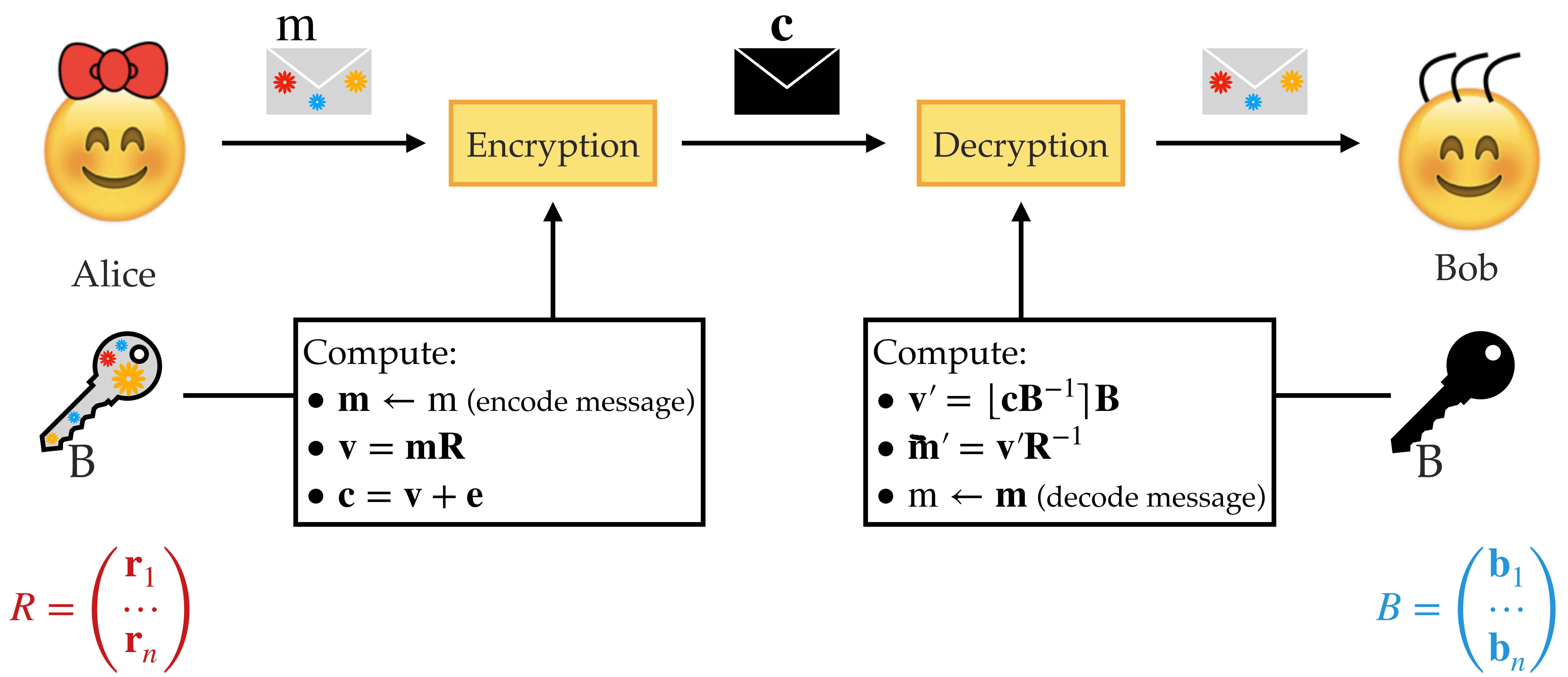
# The GGH encryption scheme



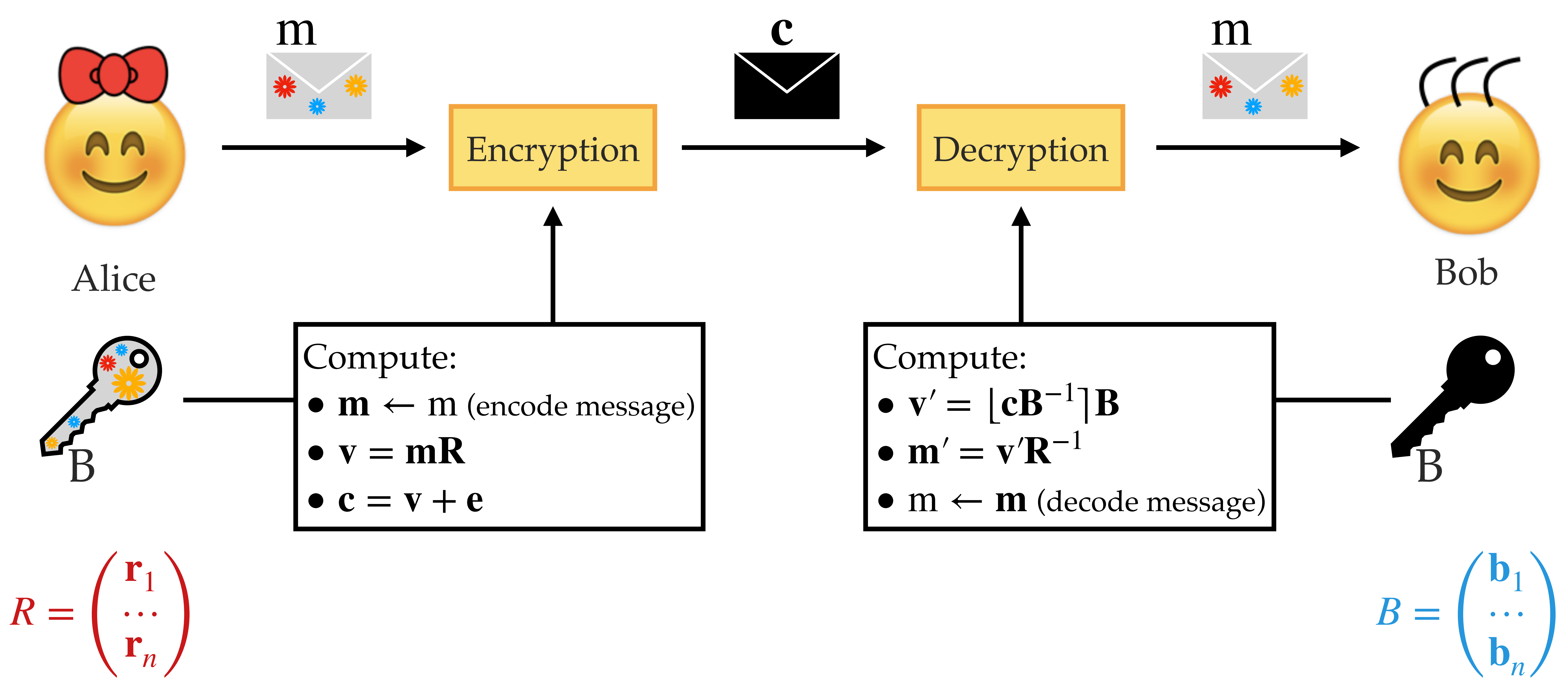

$$R = \begin{pmatrix} \mathbf{r}_1 \\ \cdots \\ \mathbf{r}_n \end{pmatrix}$$

$$B = \begin{pmatrix} \mathbf{b}_1 \\ \cdots \\ \mathbf{b}_n \end{pmatrix}$$

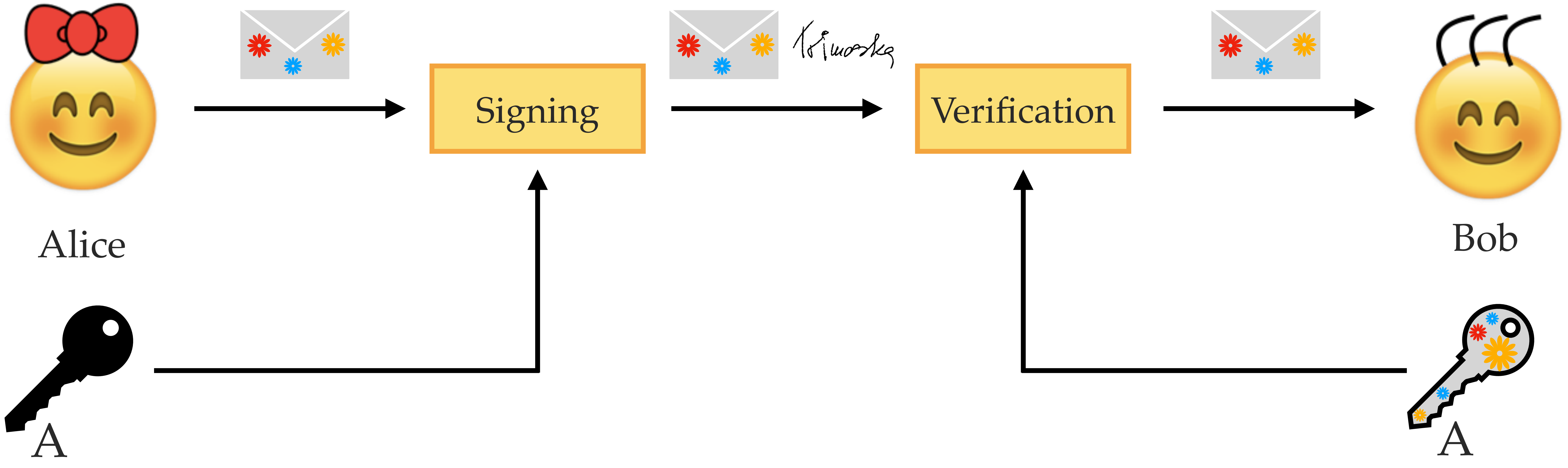
# The GGH encryption scheme



# The GGH encryption scheme

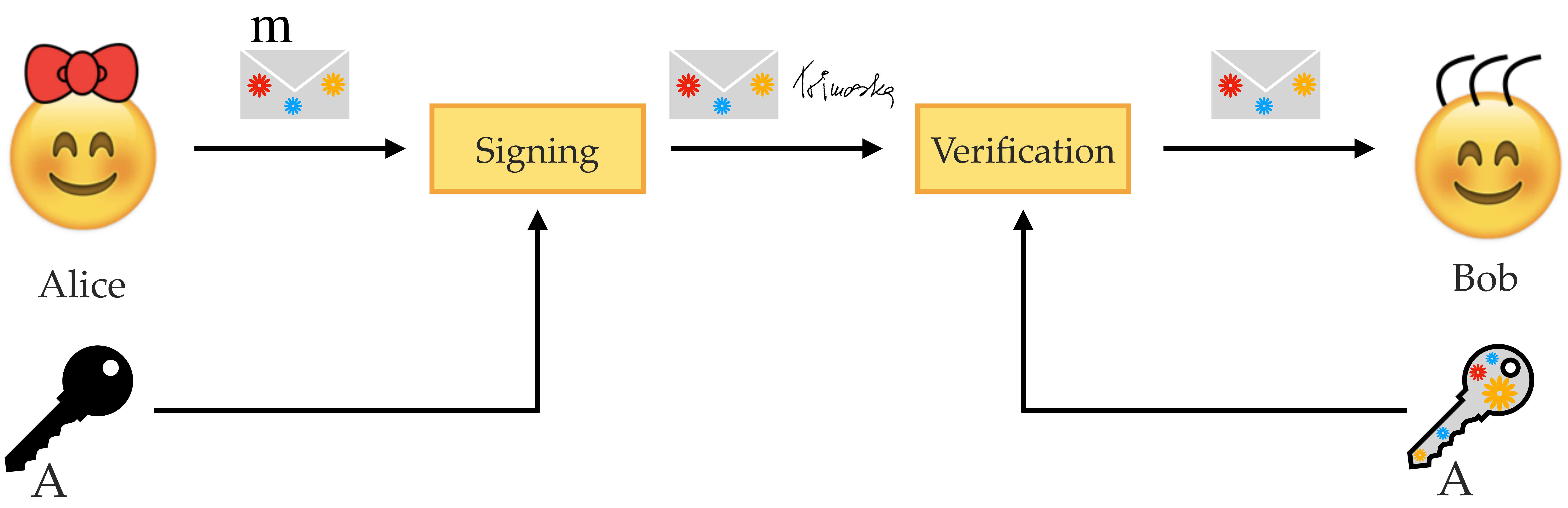


# The GGH signature scheme




$$B = \begin{pmatrix} \mathbf{b}_1 \\ \cdots \\ \mathbf{b}_n \end{pmatrix} \qquad R = \begin{pmatrix} \mathbf{r}_1 \\ \cdots \\ \mathbf{r}_n \end{pmatrix}$$

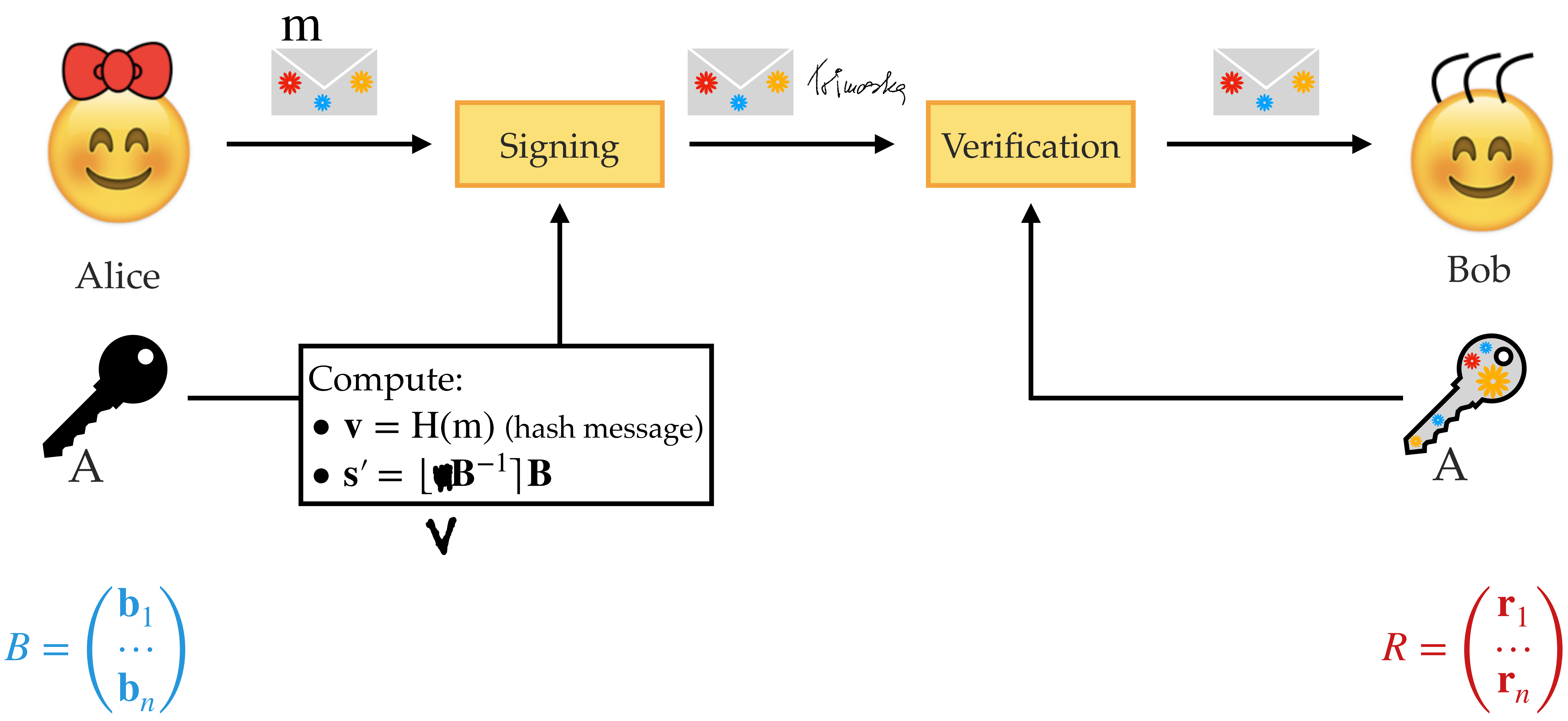
# The GGH signature scheme



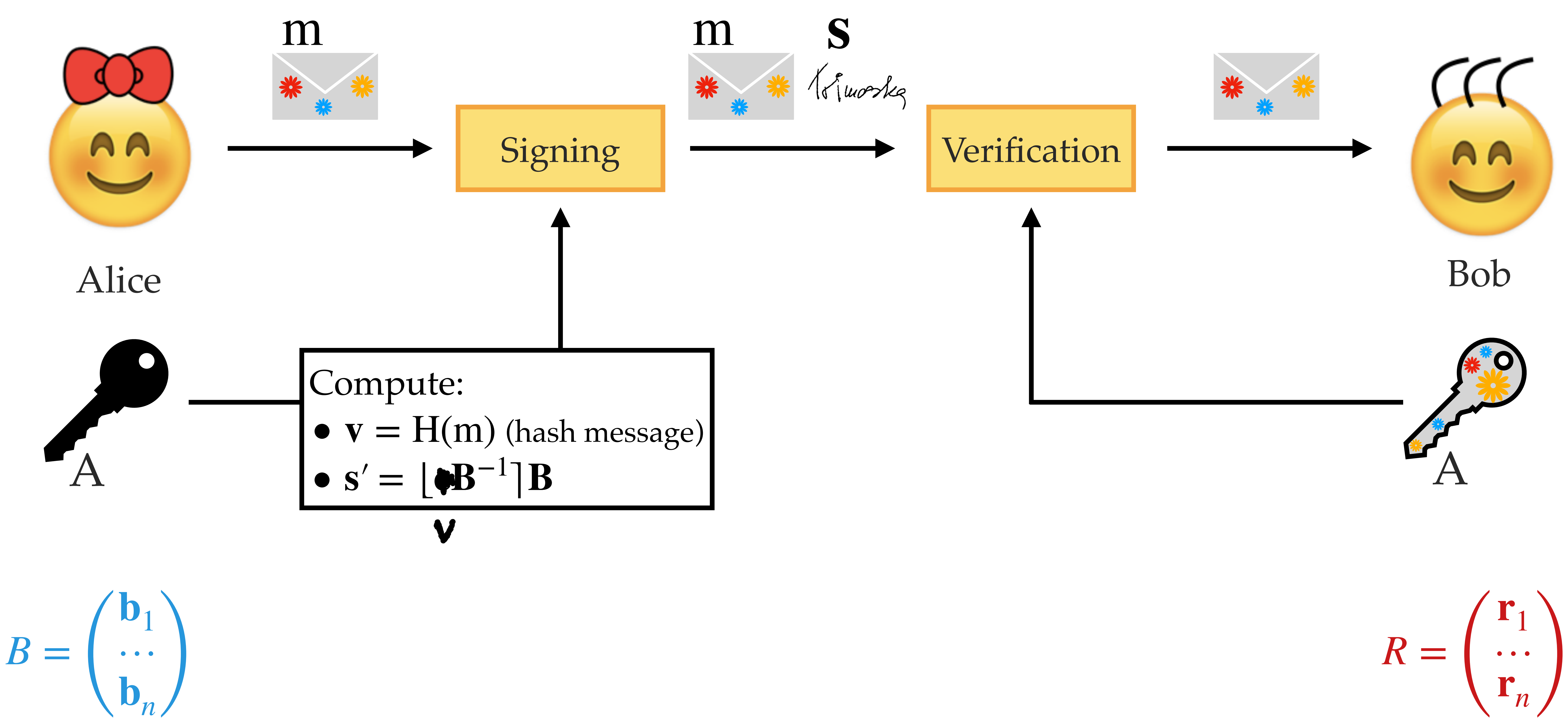

$$B = \begin{pmatrix} \mathbf{b}_1 \\ \cdots \\ \mathbf{b}_n \end{pmatrix}$$

$$R = \begin{pmatrix} \mathbf{r}_1 \\ \cdots \\ \mathbf{r}_n \end{pmatrix}$$

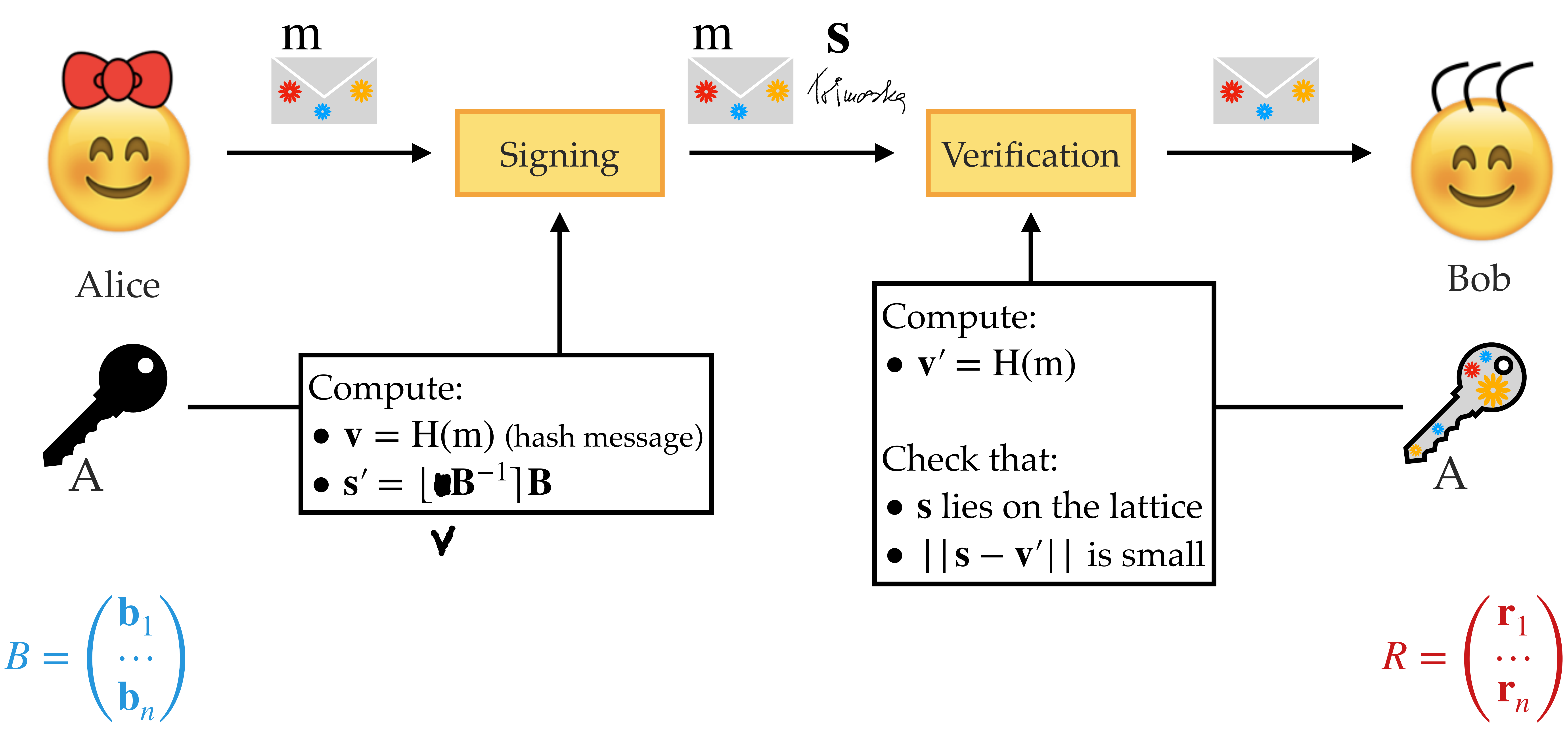
# The GGH signature scheme



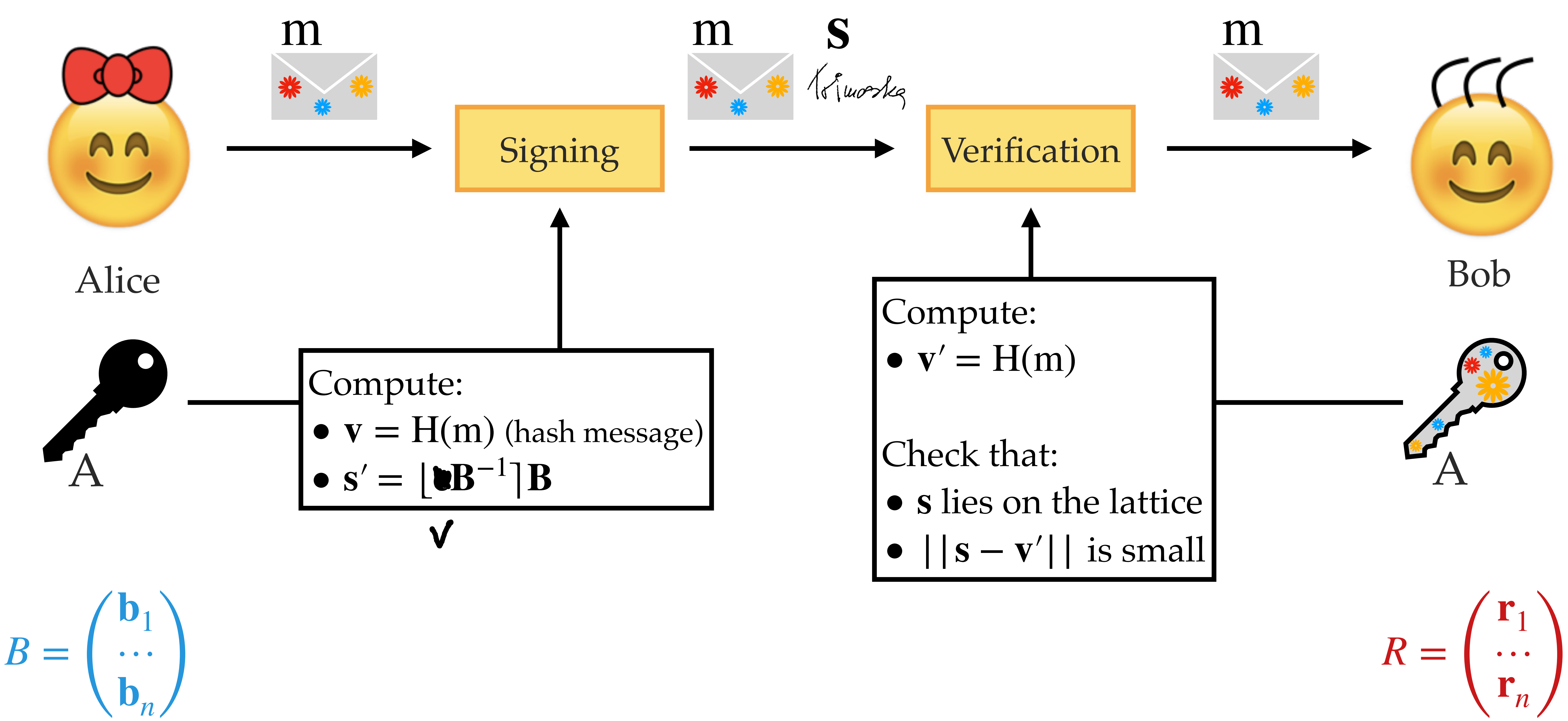
# The GGH signature scheme



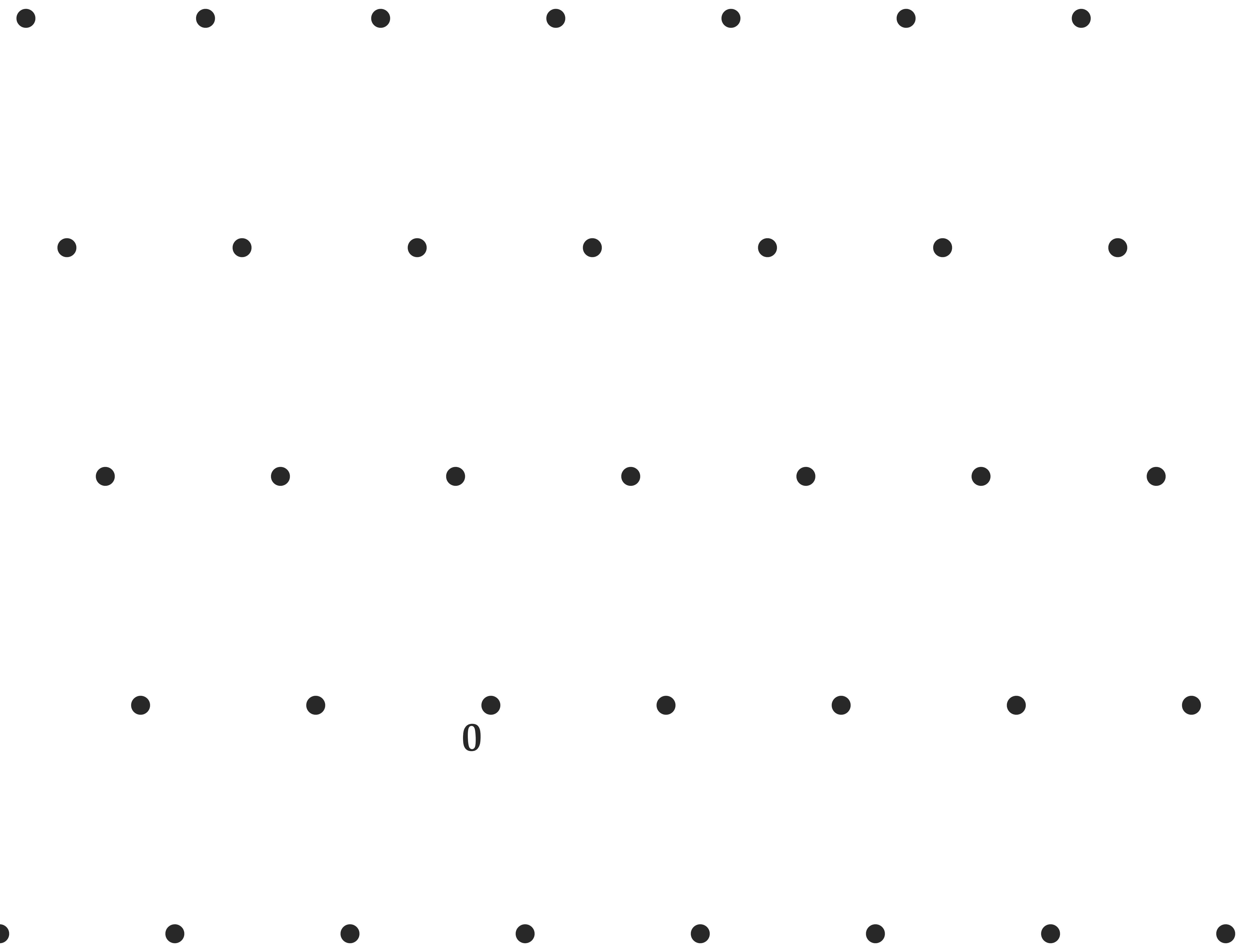
# The GGH signature scheme



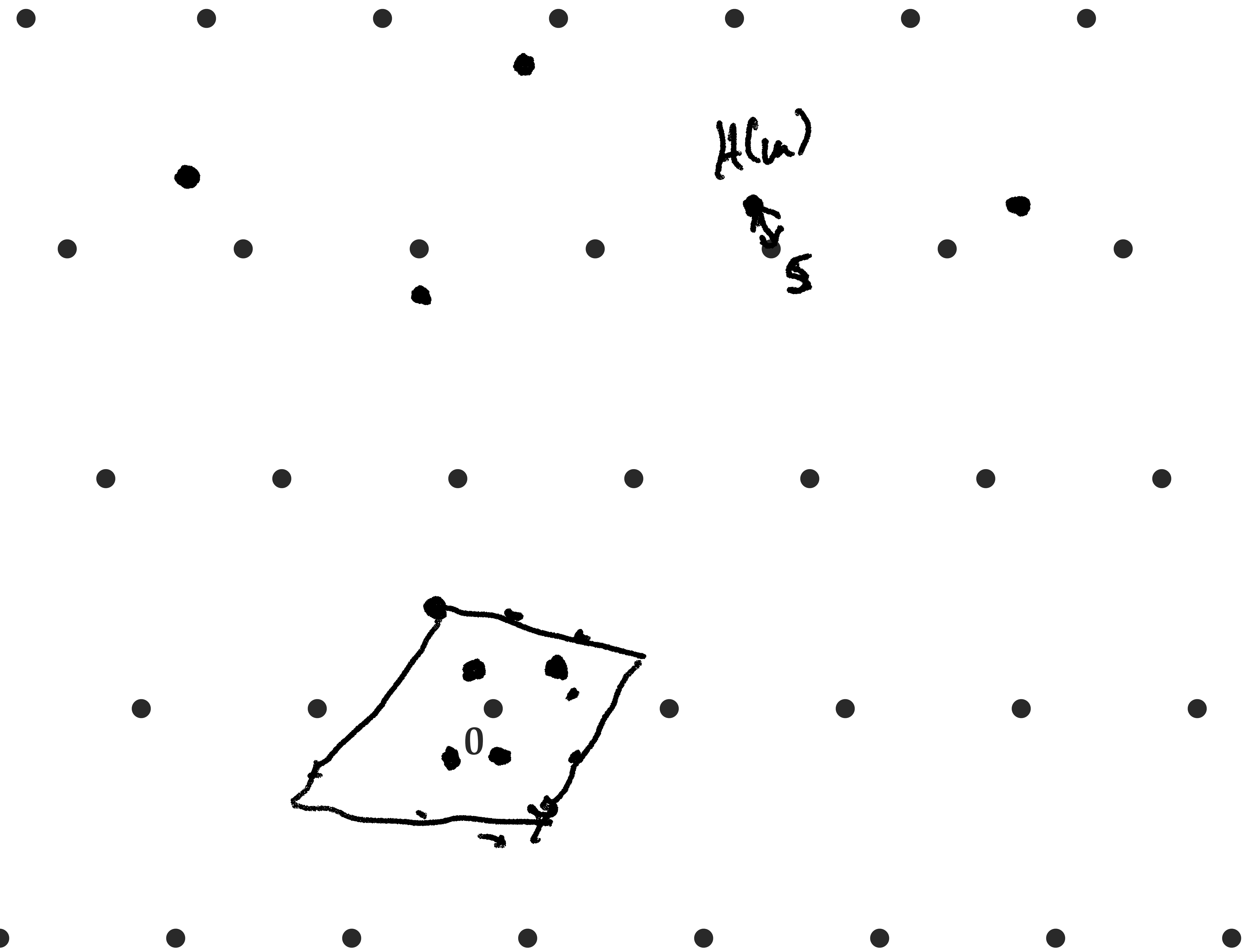
# The GGH signature scheme



# Learning a parallelepiped:



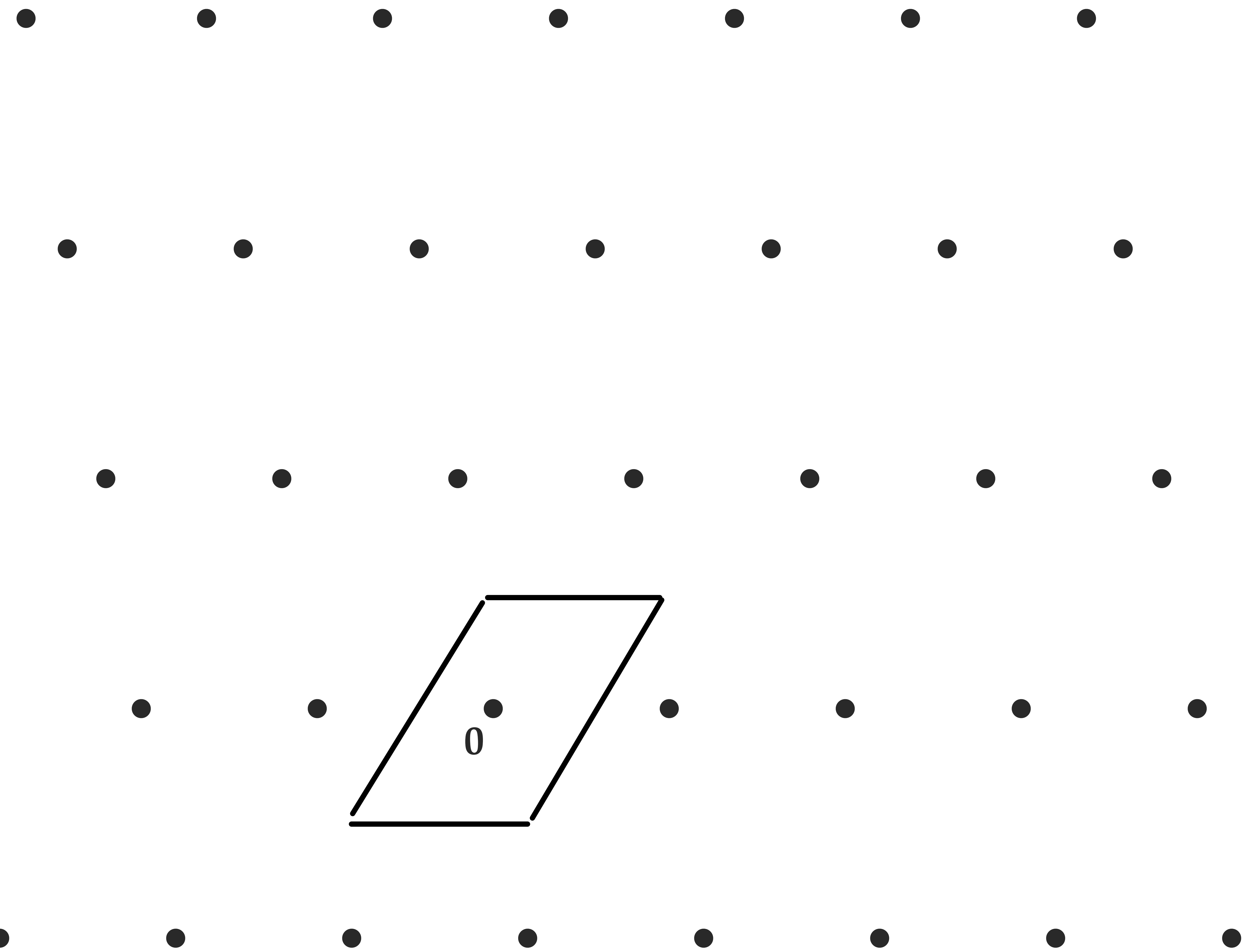
# Learning a parallelepiped:




**Repeat**

- Ask for a signature **s** on m.
- Plot H(m) − **s**.

# Learning a parallelepiped:




**Repeat**

- Ask for a signature **s** on m.
- Plot H(m) – **s**.

From the shape of the parallelepiped, we can recover a short basis.

# FALCON

→ Chosen for standardisation by NIST (alongside CRYSTALS-Dilithium and SPHINCS+).

The hash-and-sign method

+

Solving approxCVP randomly
(sampling $\mathbf{s} \in L$ close to $\mathbf{t}$ but not closest)

+

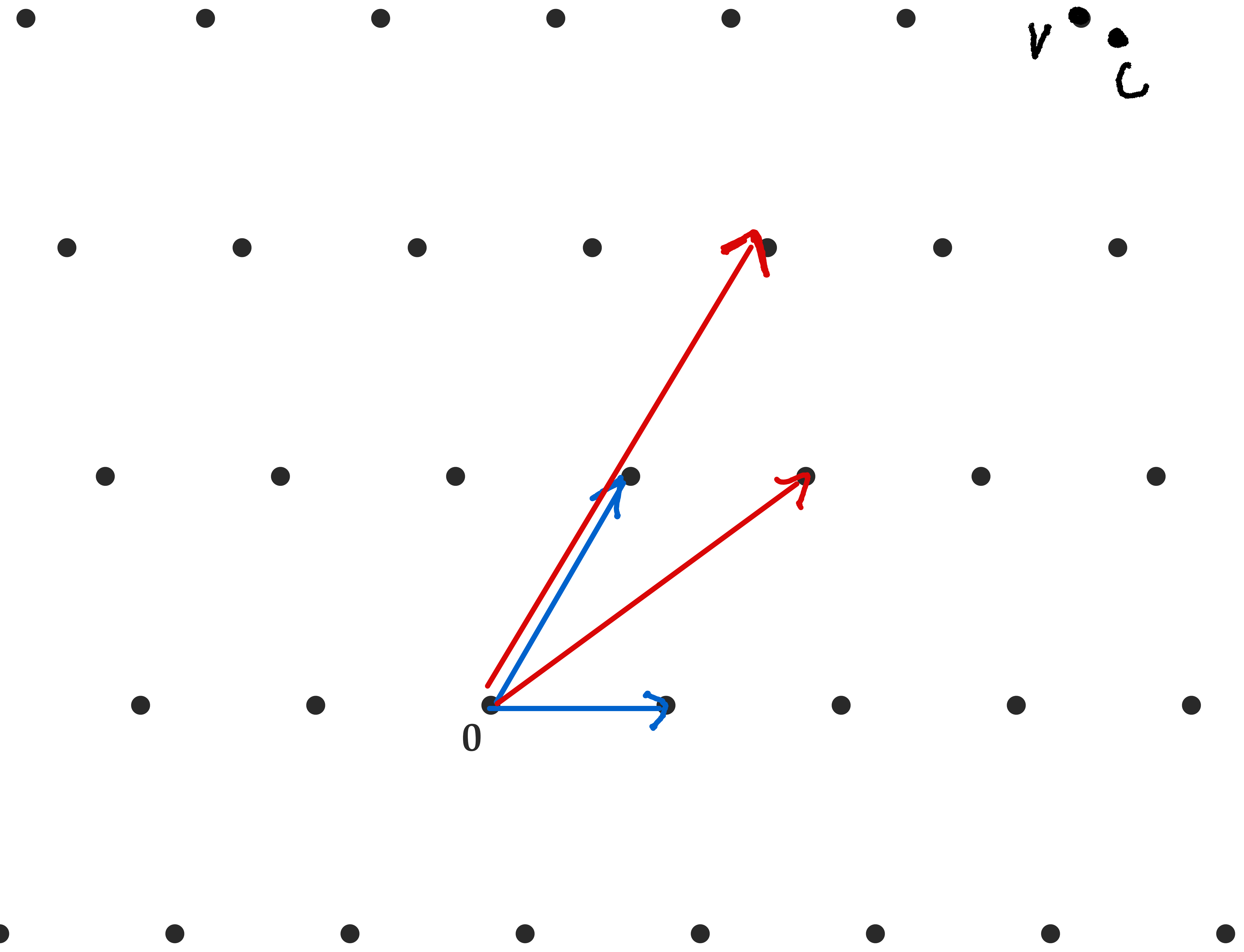
NTRU lattices

Assignment ex 1.

$\longrightarrow$ You should obtain

$$b_1 = (8, -8) \quad , \quad b_2 = (13, 5)$$

## ~~Learning a parallelepiped:~~ Toy example GGH encryption (assignment ex. 2)




$B = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$ $\quad R = \begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix}$

$m = (1,1)$

Encrypt:

$v = mR = (1,1)\begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix} = (13,15)$

Let $e = (0.1, -0.3)$

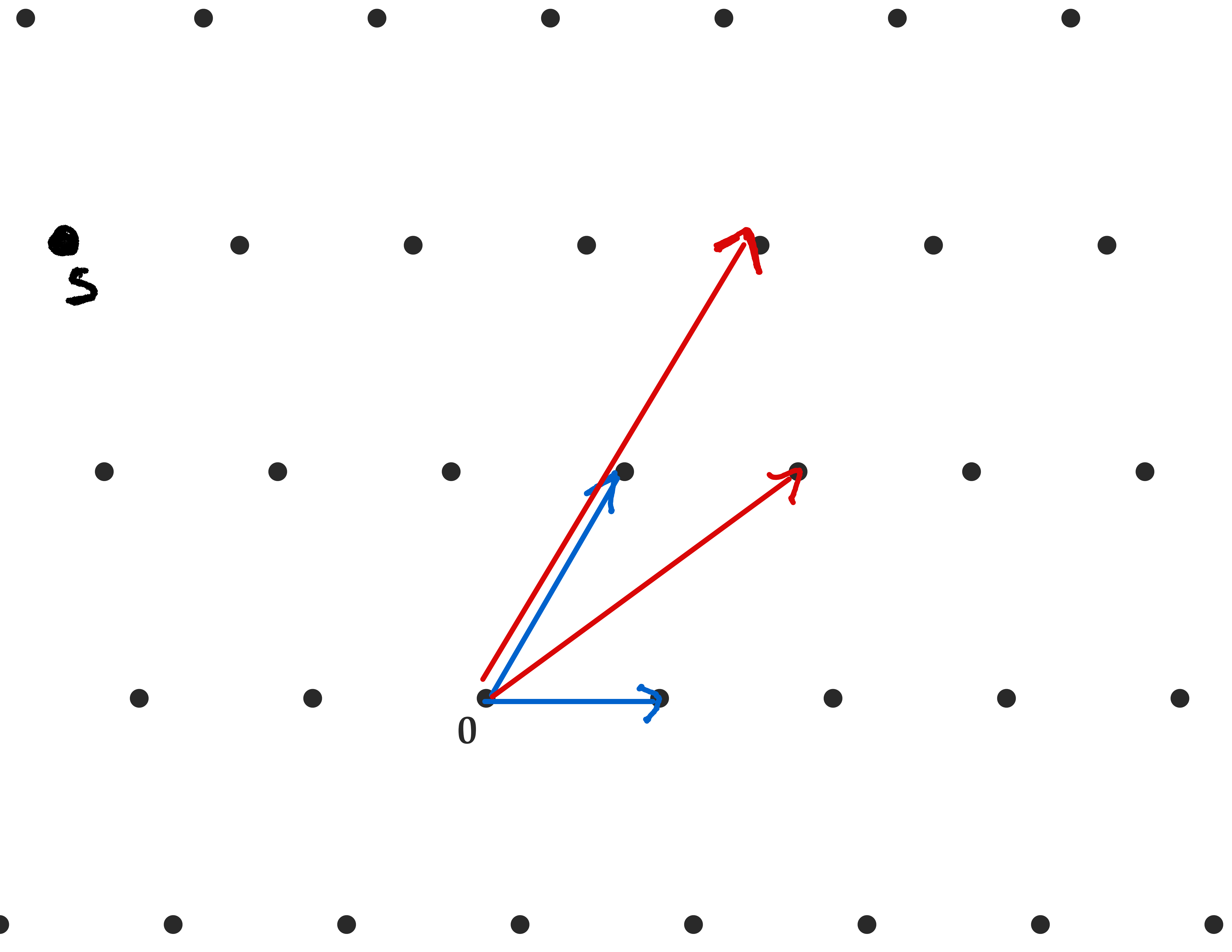
$c = v + e = (13.1, 14.7)$

Decrypt:

$v' = \lfloor c B^{-1} \rceil B = (1,3)B = (13,15)$

$m' = v' R^{-1} = (13,15)R^{-1} = (1,1)$

# ~~Learning a parallelepiped:~~ Toy example GGH signature (assignment ex. 2)




$B = \begin{pmatrix} 4 & 0 \\ 3 & 5 \end{pmatrix}$ $\quad R = \begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix}$

Sign: $v = H(m) = (-9.5, 11)$

$s = \lfloor v B^{-1} \rceil B = \lfloor (-4.02, 2.2) \rceil B = (-4, 2) B =$
$= (-10, 10)$

Verify:

$v' = (-7.8, 11)$

1) $s$ lies on the lattice:

$(a_1, a_2) R = s$

$(a_1, a_2) \begin{pmatrix} 7 & 5 \\ 6 & 10 \end{pmatrix} = (-10, 10)$

$a_1 = -4, \; a_2 = 3$ ✓

2) $\| s - v \| = 1.11 < \lambda_1(L) = 2$ ✓