On the Variants of Subset Sum: Projected and Unbounded

Pranjal Dutta (NUS), Mahesh Sreekumar Rajasree (IITD)

SYNASC 2023

- Introduction

- Introduction

- Randomized $\tilde{O}(n + t)$ algorithm for Unique Projection Subset Sum

- Introduction
- Randomized $\tilde{O}(n + t)$ algorithm for Unique Projection Subset Sum
- Reduction from Unbounded Subset Sum to Closest Vector Problem

- Introduction
- Randomized $\tilde{O}(n + t)$ algorithm for Unique Projection Subset Sum
- Reduction from Unbounded Subset Sum to Closest Vector Problem
- Other Results

- Introduction
- Randomized $\tilde{O}(n+t)$ algorithm for Unique Projection Subset Sum
- Reduction from Unbounded Subset Sum to Closest Vector Problem
- Other Results
- Conclusion

Introduction

Introduction

Introduction

Subset Sum Problem (SSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

Introduction

Subset Sum Problem (SSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Introduction

Subset Sum Problem (SSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

Introduction

Subset Sum Problem (SSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

$O(nt)$ time algorithm due to Bellman.

Introduction

Subset Sum Problem (SSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

$O(nt)$ time algorithm due to Bellman.

Randomized $\tilde{O}(n + t)$ time algorithm due to [Jin & Wu,Bringmann].

Introduction

Introduction

Unbounded Subset Problem (UBSSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $\beta_1, \ldots, \beta_n \in \mathbb{Z}_{\geq 0}$ such that

Introduction

Unbounded Subset Problem (UBSSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $\beta_1, \ldots, \beta_n \in \mathbb{Z}_{\geq 0}$ such that

$$\sum_{i=1}^{n} (\beta_i \cdot a_i) = t$$

Introduction

Unbounded Subset Problem (UBSSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $\beta_1, \ldots, \beta_n \in \mathbb{Z}_{\geq 0}$ such that

$$\sum_{i=1}^{n} (\beta_i \cdot a_i) = t$$

NP-complete problem. $O(nt)$ time algorithm due to Bellman.

Introduction

Unbounded Subset Problem (UBSSUM) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $\beta_1, \ldots, \beta_n \in \mathbb{Z}_{\geq 0}$ such that

$$\sum_{i=1}^{n} (\beta_i \cdot a_i) = t$$

NP-complete problem. $O(nt)$ time algorithm due to Bellman.

Randomized $\tilde{O}(n + t)$ time algorithm.

Introduction

Introduction

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

Introduction

Projection Subset Problem (PSSUM_p) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

Introduction

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$||\vec{a} - \vec{x}||_p$ is minimised

Introduction

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$$||\vec{a} - \vec{x}||_p \text{ is minimised}$$

NP-complete problem.

Introduction

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$$||\vec{a} - \vec{x}||_p \text{ is minimised}$$

NP-complete problem.

$O(nt)$ time algorithm?

$\tilde{O}(n + t)$ algorithm for SSUM

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

Claim:- $(a_1, \ldots, a_n, t) \in SSUM \iff coeff(f, x^t) \neq 0$

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

Claim:- $(a_1, \ldots, a_n, t) \in SSUM \iff coeff(f, x^t) \neq 0$

$$f(x) = 1 + x^{a_1} + \ldots + x^{a_n} + x^{a_1+a_2} + x^{a_1+a_3} + \ldots + x^{a_1+a_2+\ldots+a_n}$$

Randomized $\tilde{O}(n + t)$ algorithm for Unique Projection Subset Sum

Projection SSUM

Projection SSUM

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

Projection SSUM

Projection Subset Problem (PSSUM_p) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

Projection SSUM

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$||\vec{a} - \vec{x}||_p$ is minimised

Projection SSUM

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$||\vec{a} - \vec{x}||_p$ is minimised

UNIQUE PSSUM is such that any valid subset sum t', there exists a unique S, i.e.,

Projection SSUM

Projection Subset Problem (PSSUM$_p$) - Given $a_1, \ldots, a_n, t \in \mathbb{Z}_{\geq 0}$ and p, find $x_1, \ldots, x_n \in \mathbb{Z}$ such that there exists $S \subseteq [n]$

$$\sum_{i \in S} x_i = t$$

$||\vec{a} - \vec{x}||_p$ is minimised

UNIQUE PSSUM is such that any valid subset sum t', there exists a unique S, i.e.,

$$\sum_{i \in S} a_i = t'$$

Unique-PSSUM for $p = 1$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \setminus \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

$$\sum_{i \in T} y_i = t$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

$$\sum_{i \in T} y_i = t$$

$$|| \vec{a} - \vec{y} ||_1 < || \vec{a} - \vec{x} ||_1$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

$$\sum_{i \in T} y_i = t$$

$$||\vec{a} - \vec{y}||_1 < ||\vec{a} - \vec{x}||_1$$

then, we can show that

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

$$\sum_{i \in T} y_i = t$$

$$||\vec{a} - \vec{y}||_1 < ||\vec{a} - \vec{x}||_1$$

then, we can show that

$$|\sum_{i \in T} a_i - t| < |t - t'|$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \subseteq [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Suppose there is a better solution, i.e. , $y_1, \ldots, y_n$ such that

$$\sum_{i \in T} y_i = t$$

$$|| \vec{a} - \vec{y} ||_1 < || \vec{a} - \vec{x} ||_1$$

then, we can show that

$$|\sum_{i \in T} a_i - t| < |t - t'|$$

Claim: $t' \leq 2t$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $O(nt)$ deterministic time algorithm.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $O(nt)$ deterministic time algorithm.

Run Bellman's algorithm upto $2t$ to find the closest t'.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $O(nt)$ deterministic time algorithm.

Run Bellman's algorithm upto $2t$ to find the closest t'.

To find j, update the table with additional informations.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $O(nt)$ deterministic time algorithm.

Run Bellman's algorithm upto $2t$ to find the closest t'.

To find j, update the table with additional informations.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Unique-PSSUM for $p = 1$

Theorem: There is an $\tilde{O}(n + t)$ randomised time algorithm.

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $\tilde{O}(n+t)$ randomised time algorithm.

Use the $\tilde{O}(n+t)$ randomised algorithm to find t'. Recall, the algorithm checks whether $coeff(f, x^t) \neq 0$ where

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $\tilde{O}(n + t)$ randomised time algorithm.

Use the $\tilde{O}(n + t)$ randomised algorithm to find t'. Recall, the algorithm checks whether $coeff(f, x^t) \neq 0$ where

$$f(x) = \prod_{i=1}^{n} (1 + x^{a_i})$$

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $\tilde{O}(n + t)$ randomised time algorithm.

Use the $\tilde{O}(n + t)$ randomised algorithm to find t'. Recall, the algorithm checks whether $coeff(f, x^t) \neq 0$ where

$$f(x) = \prod_{i=1}^{n} (1 + x^{a_i})$$

To find j, use a different polynomial and binary search.

Unique-PSSUM for $p = 1$

Input: $a_1, \ldots, a_n, t$

Find t' such that

1. there exists $S \in [n]$ and $\sum_{i \in S} a_i = t'$.
2. $|t - t'|$ is minimised.

Let $j \in S$. We define

$$x_i := a_i, \forall i \in [n] \backslash \{j\}$$

$$x_j := a_j + t - t'$$

Theorem: There is an $\tilde{O}(n + t)$ randomised time algorithm.

Use the $\tilde{O}(n + t)$ randomised algorithm to find t'. Recall, the algorithm checks whether $coeff(f, x^t) \neq 0$ where

$$f(x) = \prod_{i=1}^{n} (1 + x^{a_i})$$

To find j, use a different polynomial and binary search.

$$g(x) = \prod_{i=1}^{n/2} (1 + 2x^{a_i}) \prod_{i=n/2+1}^{n} (1 + x^{a_i})$$

Reduction from Unbounded Subset Sum to Closest Vector Problem

Lattice

Lattice

A *lattice* generated by a set of *linearly independent vectors* $B = \{b_1, \ldots, b_n\}$ is the set of all ***integer linear combinations*** of $\{b_1, \ldots, b_n\}$, i.e.,

Lattice

A *lattice* generated by a set of *linearly independent vectors* $B = \{b_1, \ldots, b_n\}$ is the set of all ***integer linear combinations*** of $\{b_1, \ldots, b_n\}$, i.e.,

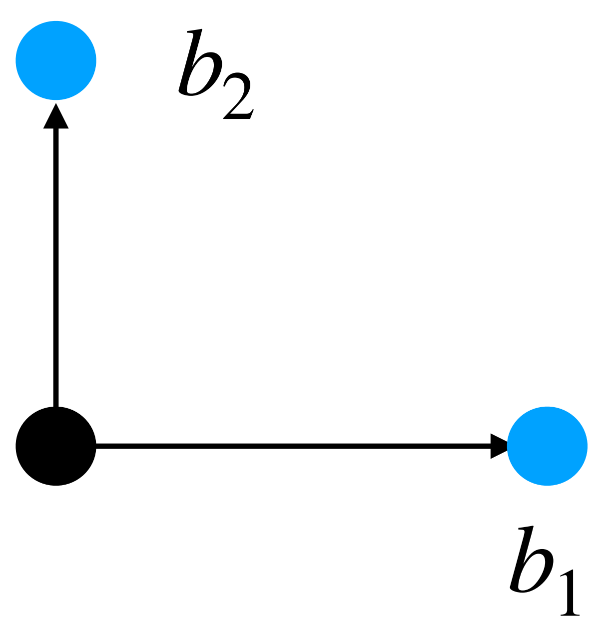
$$\mathscr{L}(b_1, \ldots, b_n) = \{ \sum_{i=1}^{n} z_i b_i \mid \forall (z_1, \ldots, z_n) \in \mathbb{Z}^n \}$$

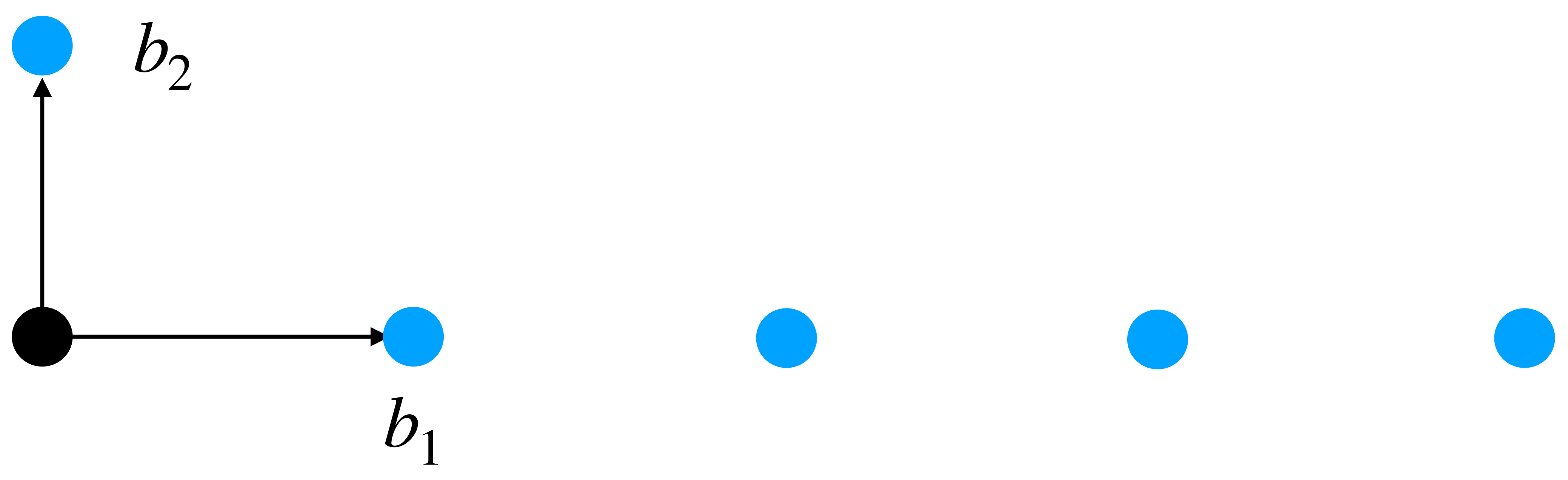
Lattice

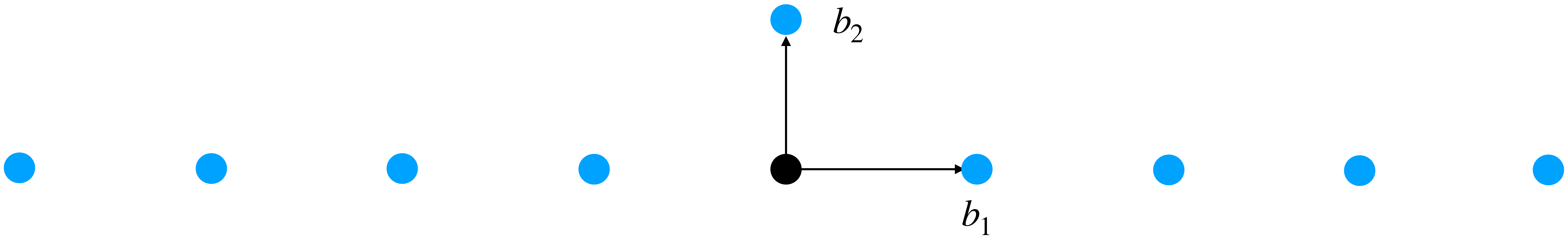
A *lattice* generated by a set of *linearly independent vectors* $B = \{b_1, \ldots, b_n\}$ is the set of all ***integer linear combinations*** of $\{b_1, \ldots, b_n\}$, i.e.,

$$\mathscr{L}(b_1, \ldots, b_n) = \{ \sum_{i=1}^{n} z_i b_i \mid \forall (z_1, \ldots, z_n) \in \mathbb{Z}^n \}$$

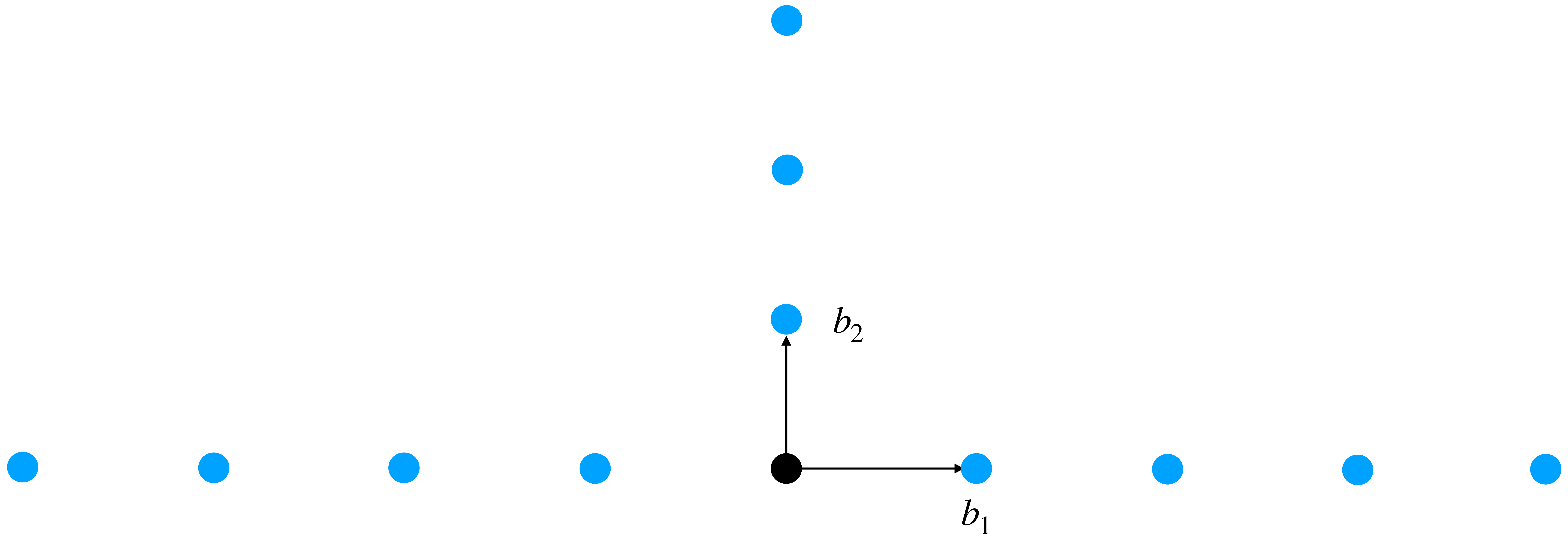
B is called a *basis* of $\mathscr{L}$.


b_2
b_1

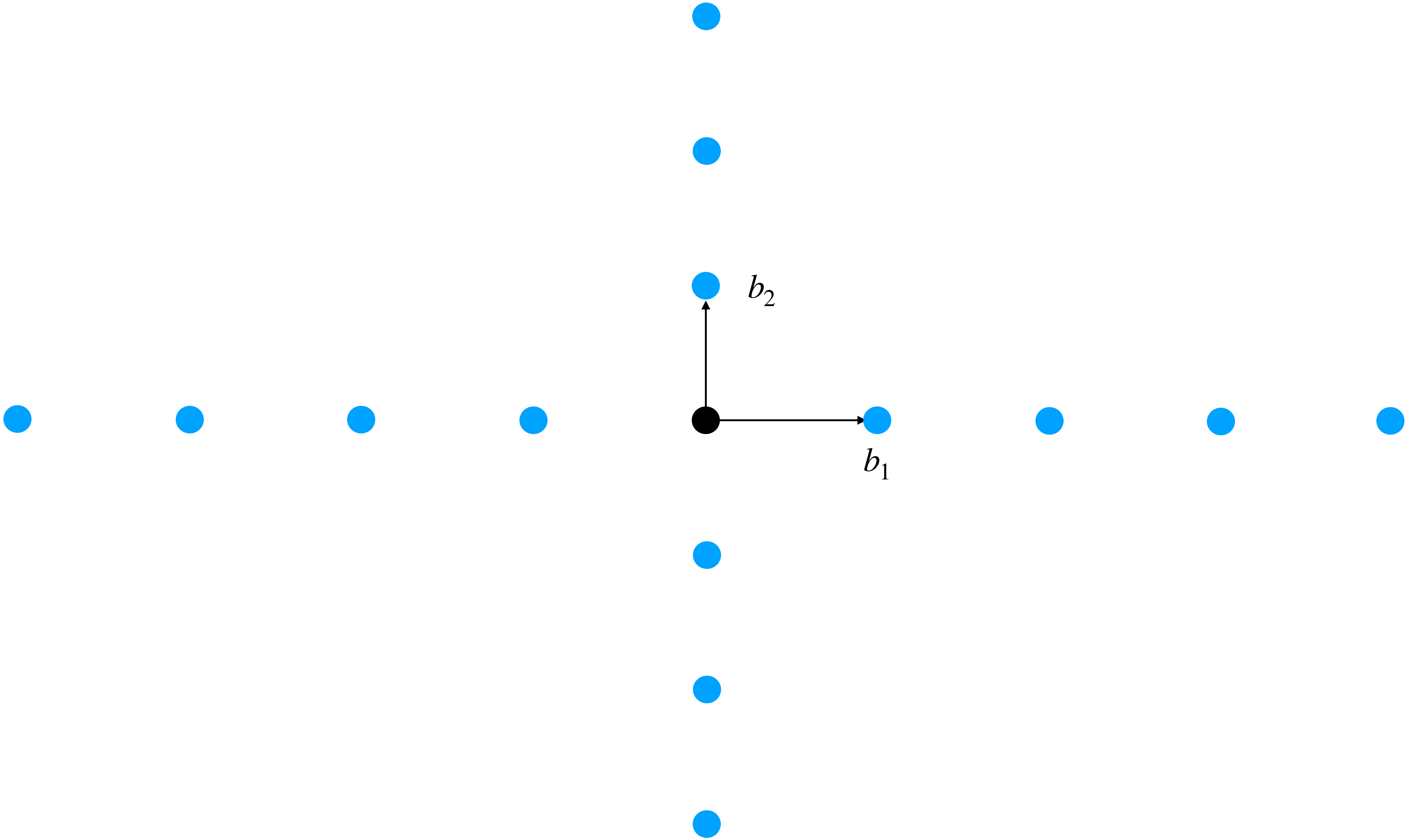

b_2
b_1

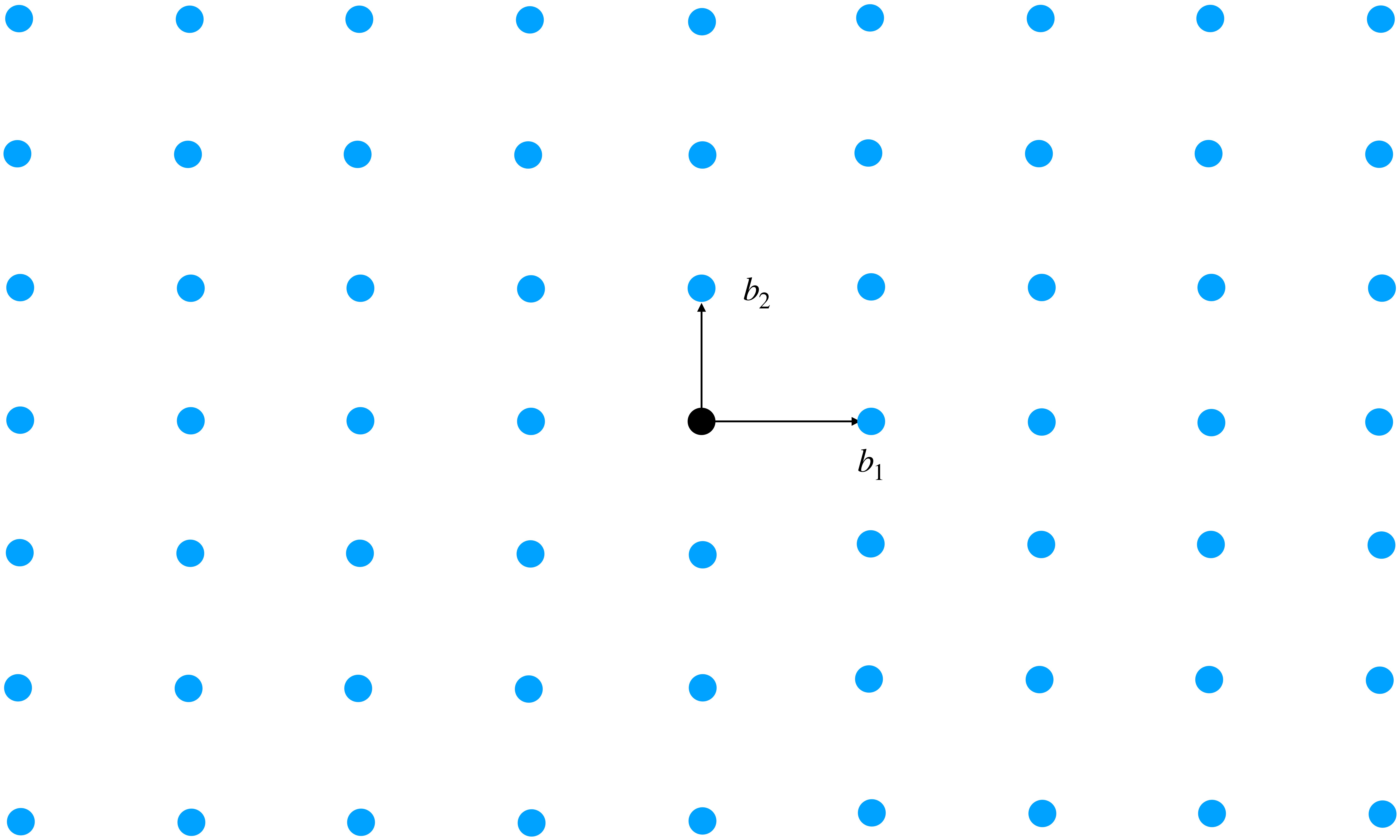


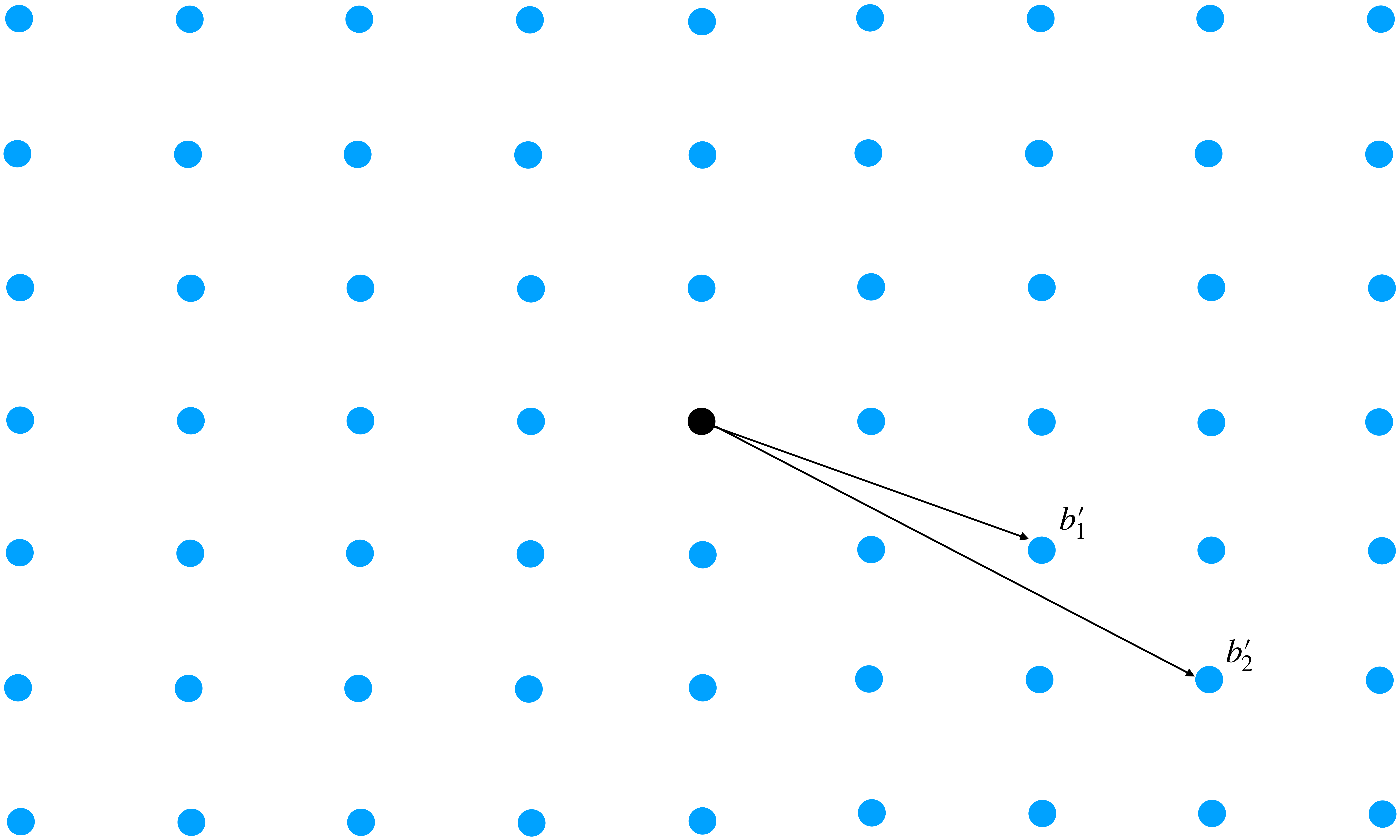
b_2
b_1



b_2
b_1


b_2
b_1


b_2
b_1



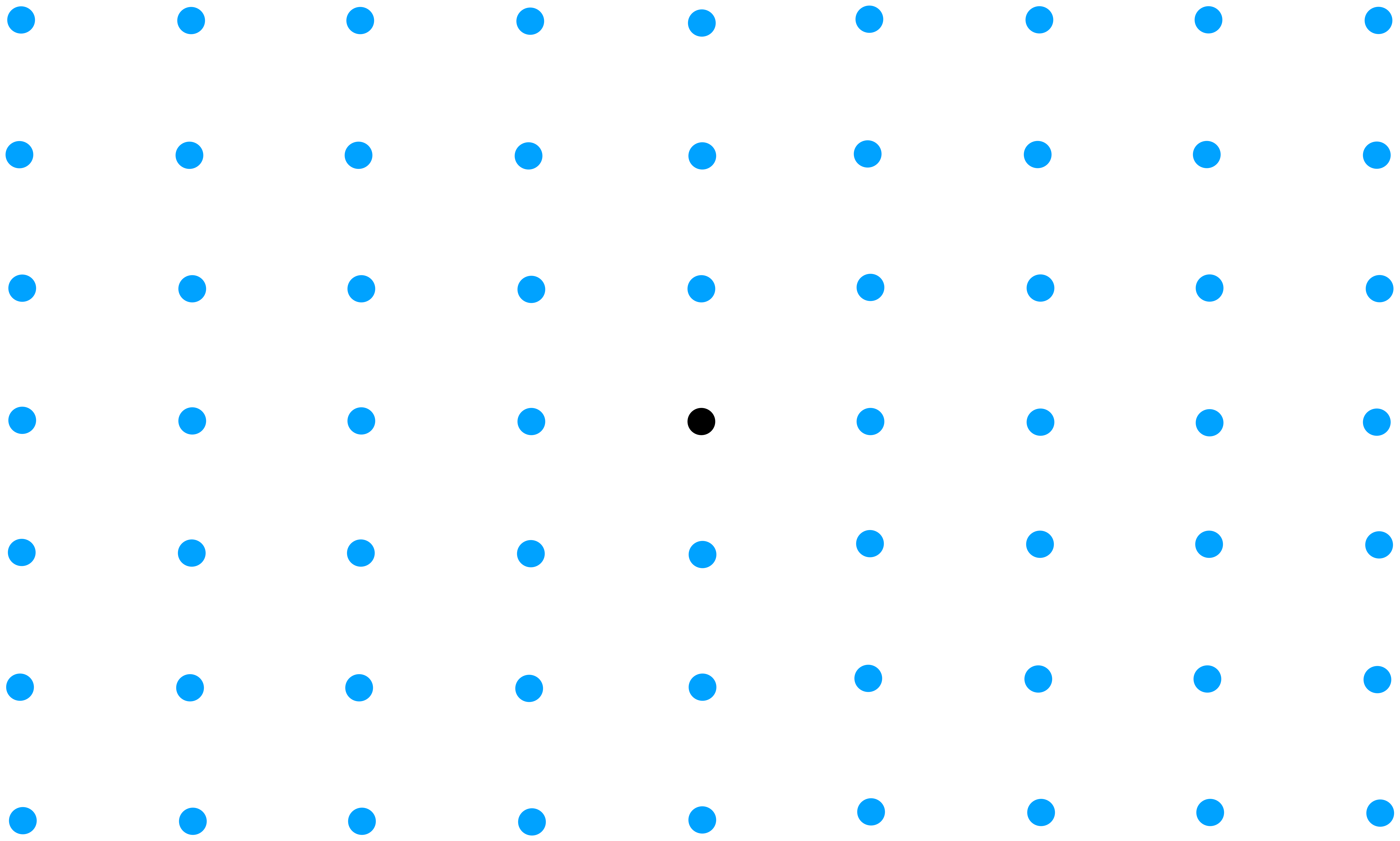
b_1'
b_2'

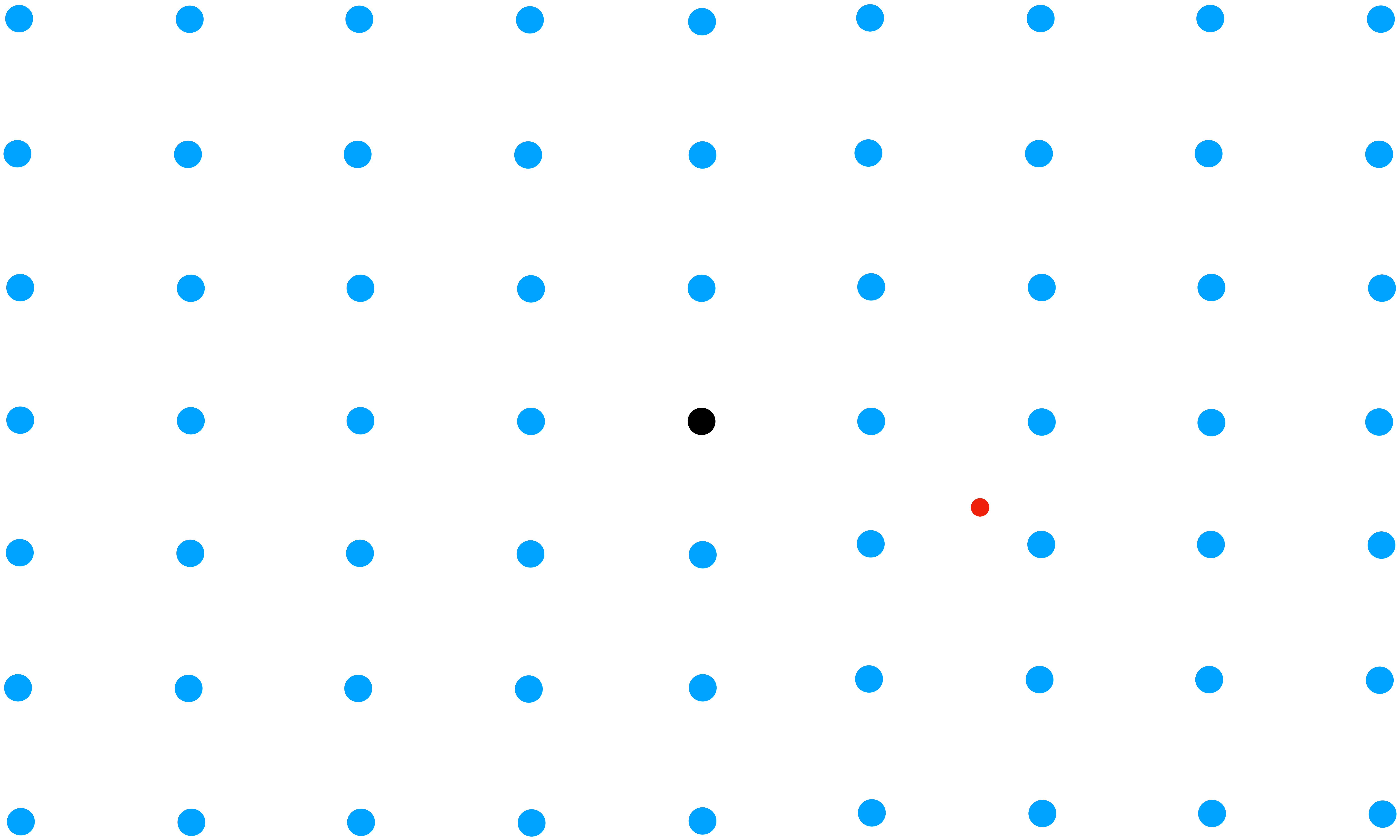
Closest Vector Problem (CVP)

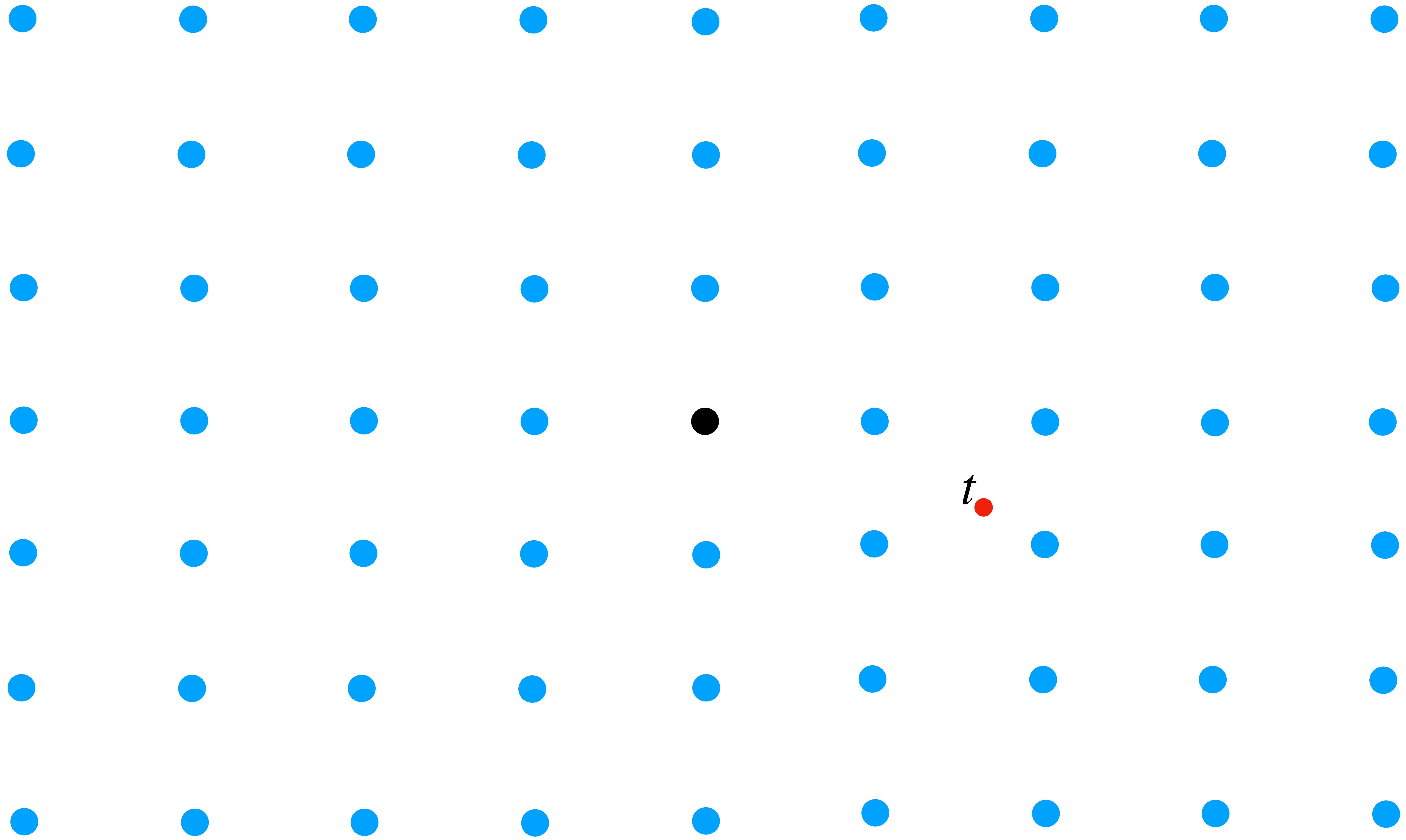
Closest Vector Problem (CVP)

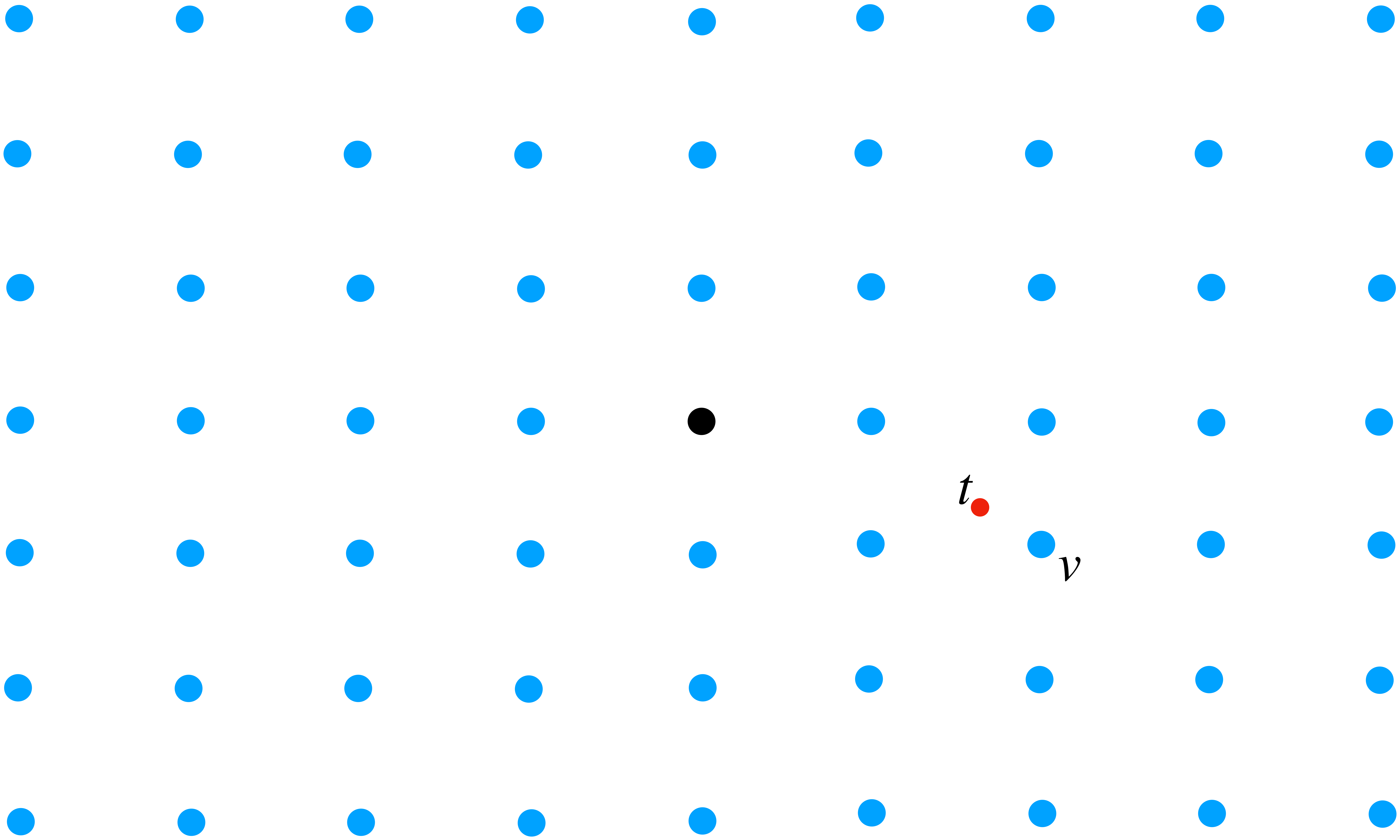
Given a basis $B = \{b_1, \ldots, b_n\}$ and a target $t \in \mathbb{R}^n$, find a vector $v \in \mathscr{L}(B)$ such that v is closest to t, i.e.,

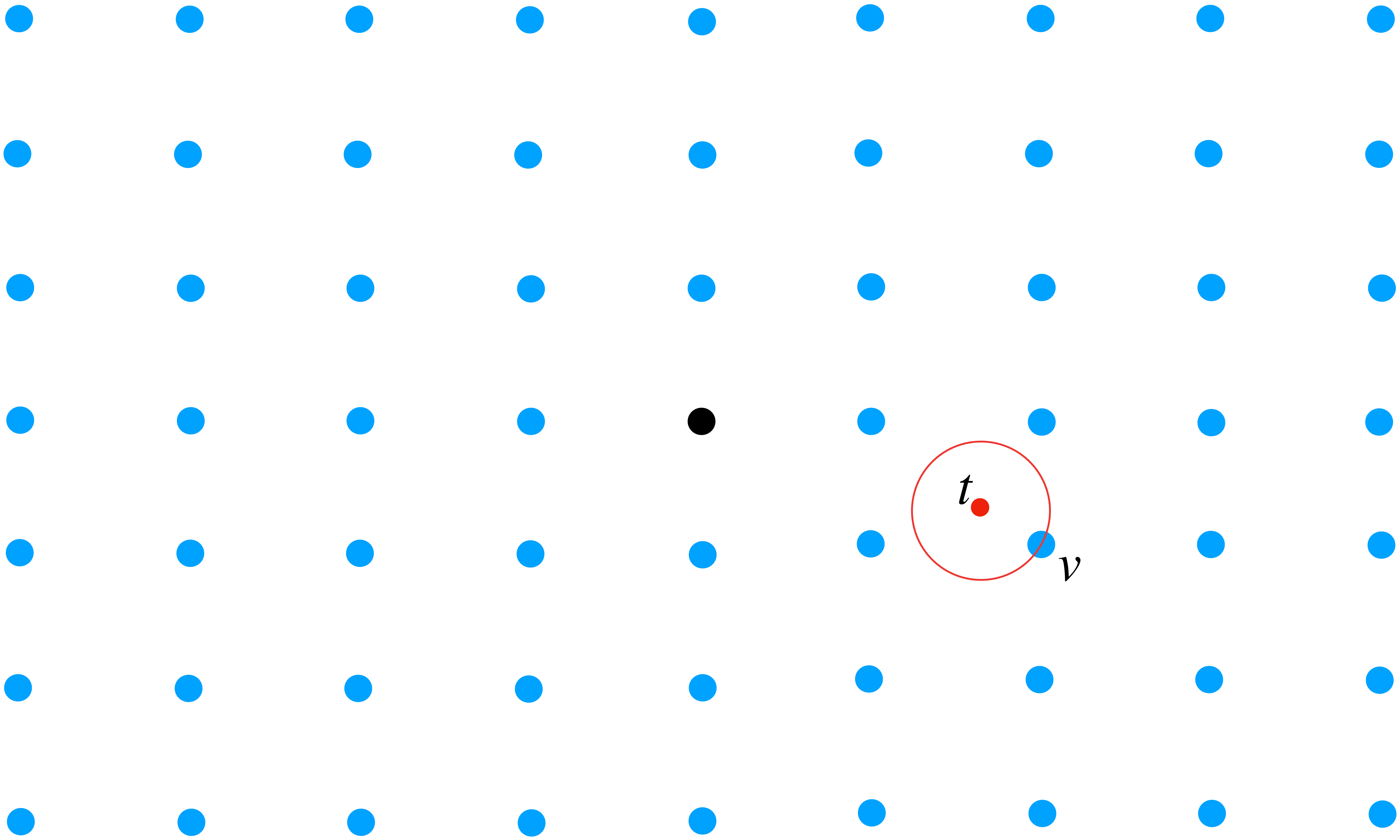
$$||v - t|| \leq ||u - t||, \forall u \in \mathscr{L}(B)$$

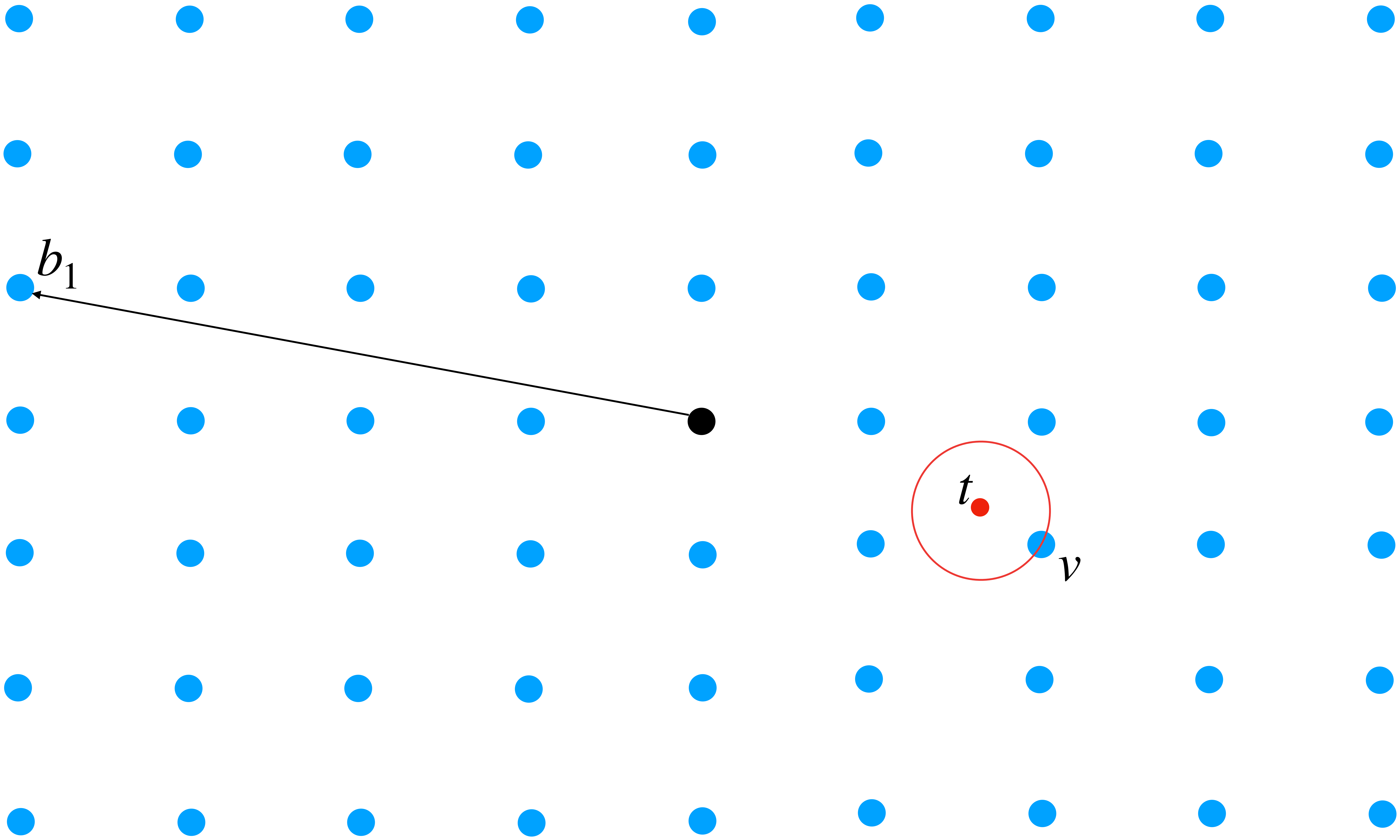


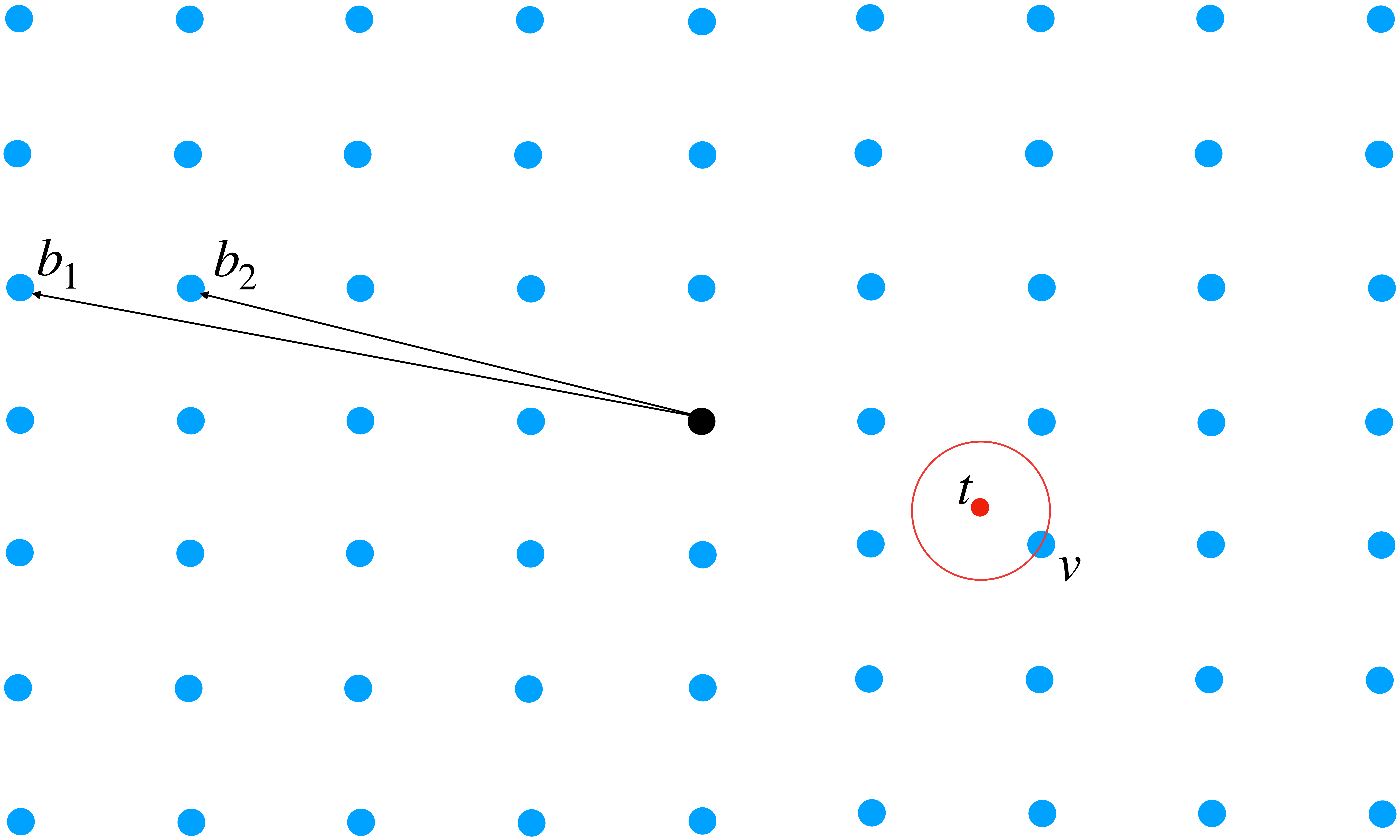



t


t
v


t
v


b_1
t
v


b_1
b_2
t
v

UBSSUM to CVP_∞

UBSSUM to CVP_∞

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

UBSSUM to CVP_∞

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} b \\ b \\ \vdots \\ b \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} b - x_1 \\ b - x_2 \\ \vdots \\ b - x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

UBSSUM to CVP_∞

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} b \\ b \\ \vdots \\ b \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} b - x_1 \\ b - x_2 \\ \vdots \\ b - x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

Assume, CVP is **YES**, then this implies that $\exists x$ such that $||t - Bx||_\infty \leq b$.

UBSSUM to CVP$_\infty$

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} b \\ b \\ \vdots \\ b \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} b - x_1 \\ b - x_2 \\ \vdots \\ b - x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

Assume, CVP is **YES**, then this implies that $\exists x$ such that $||t - Bx||_\infty \leq b$.

Since, λ is very large, this implies $(t - Bx)_{n+1} = 0 \implies \vec{a} \cdot x = b$. Also, $x_i \geq 0$.

UBSSUM to CVP_1

UBSSUM to CVP_1

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

UBSSUM to CVP_1

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} -a_1 & 0 & \cdots & 0 \\ 0 & -a_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & -a_n \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} a_1 x_1 \\ a_2 x_2 \\ \vdots \\ a_n x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

UBSSUM to CVP_1

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} -a_1 & 0 & \cdots & 0 \\ 0 & -a_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & -a_n \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} a_1 x_1 \\ a_2 x_2 \\ \vdots \\ a_n x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

Assume, CVP is **YES**, then this implies that $\exists x$ such that $||t - Bx||_1 \leq b$.

UBSSUM to CVP_1

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} -a_1 & 0 & \cdots & 0 \\ 0 & -a_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & -a_n \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} a_1 x_1 \\ a_2 x_2 \\ \vdots \\ a_n x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

Assume, CVP is **YES**, then this implies that $\exists x$ such that $||t - Bx||_1 \leq b$.

Since, λ is very large, this implies $(t - Bx)_{n+1} = 0 \implies \vec{a} \cdot x = b$. Also,

UBSSUM to CVP$_1$

Input: Unbounded subset sum instance $a_1, \ldots, a_n, t$

$$B = \begin{bmatrix} -a_1 & 0 & \cdots & 0 \\ 0 & -a_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & -a_n \\ \lambda a_1 & \lambda a_2 & \cdots & \lambda a_n \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lambda b \end{bmatrix}, d = b, t - Bx = \begin{bmatrix} a_1 x_1 \\ a_2 x_2 \\ \vdots \\ a_n x_n \\ \lambda(b - \sum a_i x_i) \end{bmatrix}$$

Assume, CVP is **YES**, then this implies that $\exists x$ such that $||t - Bx||_1 \leq b$.

Since, λ is very large, this implies $(t - Bx)_{n+1} = 0 \implies \vec{a} \cdot x = b$. Also,

$$b \geq \sum_{i=1}^{n} |a_i x_i| = \sum_{i=1}^{n} a_i |x_i| \geq \sum_{i=1}^{n} a_i x_i = b$$

Other Results

Other Results

Other Results

- There is an $\tilde{O}(k(n+t))$ time deterministic algorithm for Hamming-k-SUBSSUM (ask to return all $\sum_{i\in[n]} \beta_i$ where the number of solutions is atmost k)

Other Results

- There is an $\tilde{O}(k(n+t))$ time deterministic algorithm for Hamming-k-SUBSSUM (ask to return all $\sum_{i\in[n]} \beta_i$ where the number of solutions is atmost k)
- There is a $poly(knt)$-time and $O(log(knt))$-space deterministic algorithm which solves k-SUBSSUM (ask to return all $(\beta_1, \ldots, \beta_n)$ where the number of solutions is atmost k)

Conclusion

Conclusion

- We saw an $O(nt)$ and $\tilde{O}(n + t)$ time algorithm for unique-PSSUM$_1$.

Conclusion

- We saw an $O(nt)$ and $\tilde{O}(n+t)$ time algorithm for unique-PSSUM$_1$.
- We saw reductions from UBSSUM to CVP.

Conclusion

- We saw an $O(nt)$ and $\tilde{O}(n+t)$ time algorithm for unique-PSSUM_1.
- We saw reductions from UBSSUM to CVP.
- Can we find an $\tilde{O}(n+t)$ time algorithm for PSSUM_1.

Conclusion

- We saw an $O(nt)$ and $\tilde{O}(n+t)$ time algorithm for unique-PSSUM_1.
- We saw reductions from UBSSUM to CVP.
- Can we find an $\tilde{O}(n+t)$ time algorithm for PSSUM_1.
- Extend it to other values of p?

Thank You!