

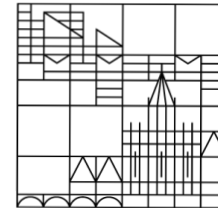
# (Multivariate) $k$ -SUM as barrier to succinct computation

ESA 2025

Geri Gokaj<sup>\*</sup>, Marvin Künnemann<sup>\*</sup>, Sabine Storandt<sup>+</sup>, Carina Truschel<sup>+</sup>



Universität  
Konstanz



Our setting:

Our setting:

**Problem  $P$ .**

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Our setting:

**Problem  $P$ .**

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Our setting:

Problem  $P$ .

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Problem succinct- $P$ .

Our setting:

Problem  $P$ .

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Problem succinct- $P$ .

Output:  $P(X = A + B, z)$ .

Our setting:

**Problem  $P$ .**

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

**Problem succinct- $P$ .**

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

Our setting:

**Problem  $P$ .**

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

**Problem succinct- $P$ .**

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

Why?

Our setting:

**Problem  $P$ .**

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

**Problem succinct- $P$ .**

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

**Why?** Problems occur naturally,  $X+Y$  sorting,  $X+Y$  Selection, Pareto Sum

Our setting:

Problem  $P$ .

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Problem succinct- $P$ .

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

**Why?** Problems occur naturally,  $X+Y$  sorting,  $X+Y$  Selection, Pareto Sum  
"Compression viewpoint". Given  $X = A + B$ , where  $|A| = |B| = n$ .

Our setting:

Problem  $P$ .

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Problem succinct- $P$ .

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

Why? Problems occur naturally,  $X+Y$  sorting,  $X+Y$  Selection, Pareto Sum

"Compression viewpoint". Given  $X = A + B$ , where  $|A| = |B| = n$ .

Possibly  $|X| = n^2 \implies$  more efficient to work with  $A, B$  if we can avoid decompressing

Our setting:

Problem  $P$ .

Input: Set  $X \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X, z)$ .

Problem succinct- $P$ .

Input: Set  $A, B \subset \mathbb{Z}^d$ , additional inputs  $z$ .

Output:  $P(X = A + B, z)$ .

$$A + B := \{a + b : a \in A, b \in B\}$$

Why? Problems occur naturally,  $X+Y$  sorting,  $X+Y$  Selection, Pareto Sum

"Compression viewpoint". Given  $X = A + B$ , where  $|A| = |B| = n$ .

Possibly  $|X| = n^2 \implies$  more efficient to work with  $A, B$  if we can avoid decompressing

An easy succinct Problem: Convex hull:  $\text{conv}(A + B) = \text{conv}(A) + \text{conv}(B)$ .

A difficult succinct problem: Bichromatic Closest Pair of points

A difficult succinct problem: Bichromatic Closest Pair of points

Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

$$\min_{x \in X, y \in Y} |x - y|_\infty = 0 \iff \min_{a_1 \in A_1, b_1 \in B_1, a_2 \in A_2, b_2 \in B_2} |(a_1 + b_1) - (a_2 + b_2)|_\infty = 0$$

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

$$\min_{x \in X, y \in Y} |x - y|_\infty = 0 \iff \min_{a_1 \in A_1, b_1 \in B_1, a_2 \in A_2, b_2 \in B_2} |(a_1 + b_1) - (a_2 + b_2)|_\infty = 0$$

$$\text{In 1-Dimension} \iff \exists a_1 \in A_1 \quad \exists b_1 \in B_1 \quad \exists a_2 \in -A_2 \quad \exists b_2 \in -B_2 : a_1 + b_1 + a_2 + b_2 = 0$$

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

$$\min_{x \in X, y \in Y} |x - y|_\infty = 0 \iff \min_{a_1 \in A_1, b_1 \in B_1, a_2 \in A_2, b_2 \in B_2} |(a_1 + b_1) - (a_2 + b_2)|_\infty = 0$$

$$\text{In 1-Dimension} \iff \exists a_1 \in A_1 \quad \exists b_1 \in B_1 \quad \exists a_2 \in -A_2 \quad \exists b_2 \in -B_2 : a_1 + b_1 + a_2 + b_2 = 0$$

4-SUM hypothesis:

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

$$\min_{x \in X, y \in Y} |x - y|_\infty = 0 \iff \min_{a_1 \in A_1, b_1 \in B_1, a_2 \in A_2, b_2 \in B_2} |(a_1 + b_1) - (a_2 + b_2)|_\infty = 0$$

$$\text{In 1-Dimension} \iff \exists a_1 \in A_1 \quad \exists b_1 \in B_1 \quad \exists a_2 \in -A_2 \quad \exists b_2 \in -B_2 : a_1 + b_1 + a_2 + b_2 = 0$$

4-SUM hypothesis:

Sets  $A, B, C, D$  of integers of size  $n$ , there is no  $O(n^{2-\epsilon})$  algorithm to decide 4-SUM i.e:

## A difficult succinct problem: Bichromatic Closest Pair of points

### Bi. Closest Pair of Points

Input: Sets  $X, Y \subset \mathbb{Z}^d$  of points

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

### Succinct Bi. Closest pair of points

Input: Sets  $A_1, B_1, A_2, B_2 \subseteq \mathbb{Z}^d$ , ( $X = A_1 + B_1, Y = A_2 + B_2$ )

Output:  $\min_{x \in X, y \in Y} |x - y|_\infty$

$$\min_{x \in X, y \in Y} |x - y|_\infty = 0 \iff \min_{a_1 \in A_1, b_1 \in B_1, a_2 \in A_2, b_2 \in B_2} |(a_1 + b_1) - (a_2 + b_2)|_\infty = 0$$

$$\text{In 1-Dimension} \iff \exists a_1 \in A_1 \quad \exists b_1 \in B_1 \quad \exists a_2 \in -A_2 \quad \exists b_2 \in -B_2 : a_1 + b_1 + a_2 + b_2 = 0$$

4-SUM hypothesis:

Sets  $A, B, C, D$  of integers of size  $n$ , there is no  $O(n^{2-\epsilon})$  algorithm to decide 4-SUM i.e:

$$\exists a \in A \exists b \in B \exists c \in C \exists d \in D : a + b + c + d = 0.$$

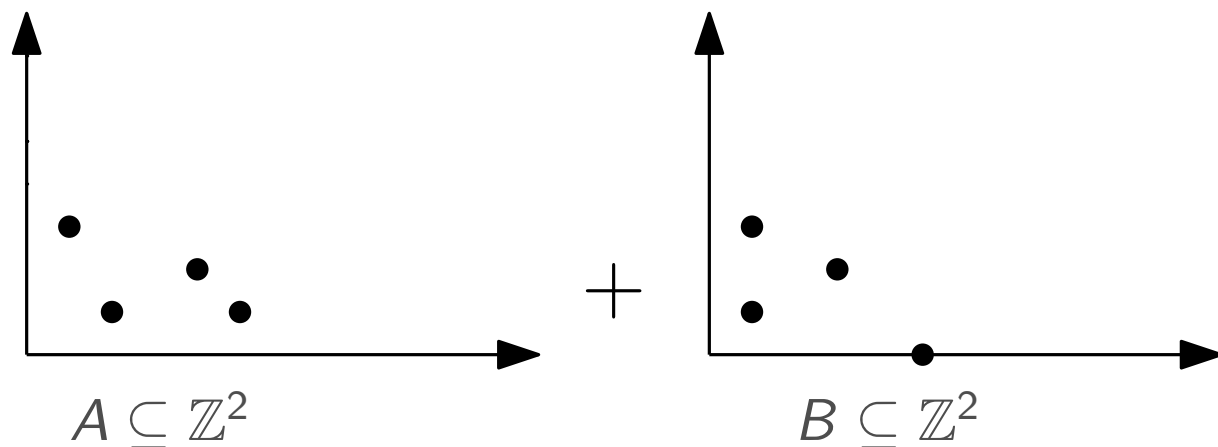
Another hard problem: 2-D Succinct-Skyline /Pareto Sum:

Another hard problem: 2-D Succinct-Skyline /Pareto Sum:

Def:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute set  $PS(A, B) := \{x \in X = A + B \mid \neg \exists x' \neq x : x' \geq x\}$ .

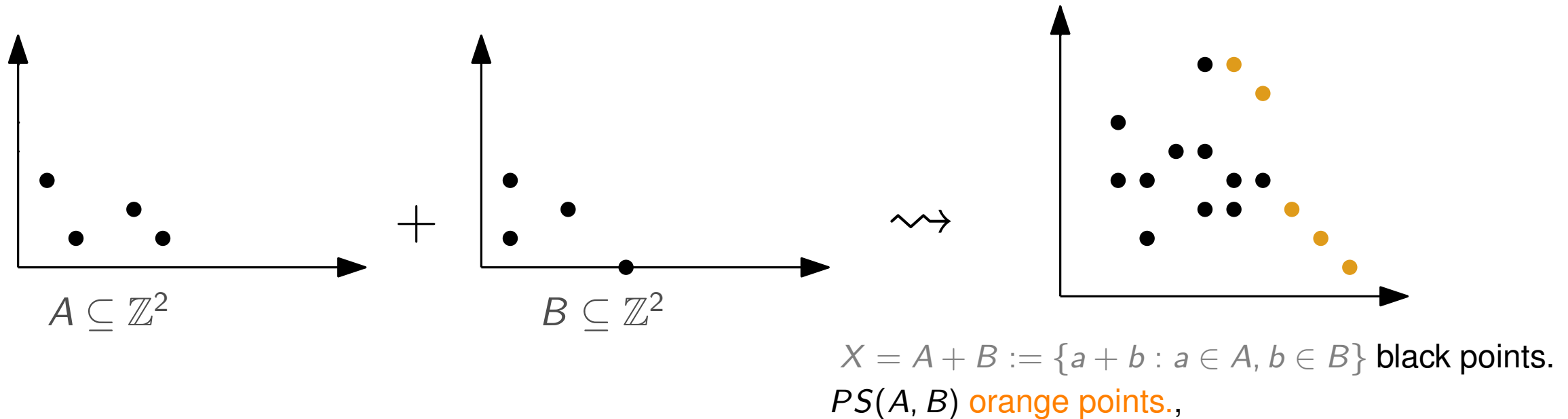
Another hard problem: 2-D Succinct-Skyline /Pareto Sum:

Def:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute set  $PS(A, B) := \{x \in X = A + B \mid \neg \exists x' \neq x : x' \geq x\}$ .



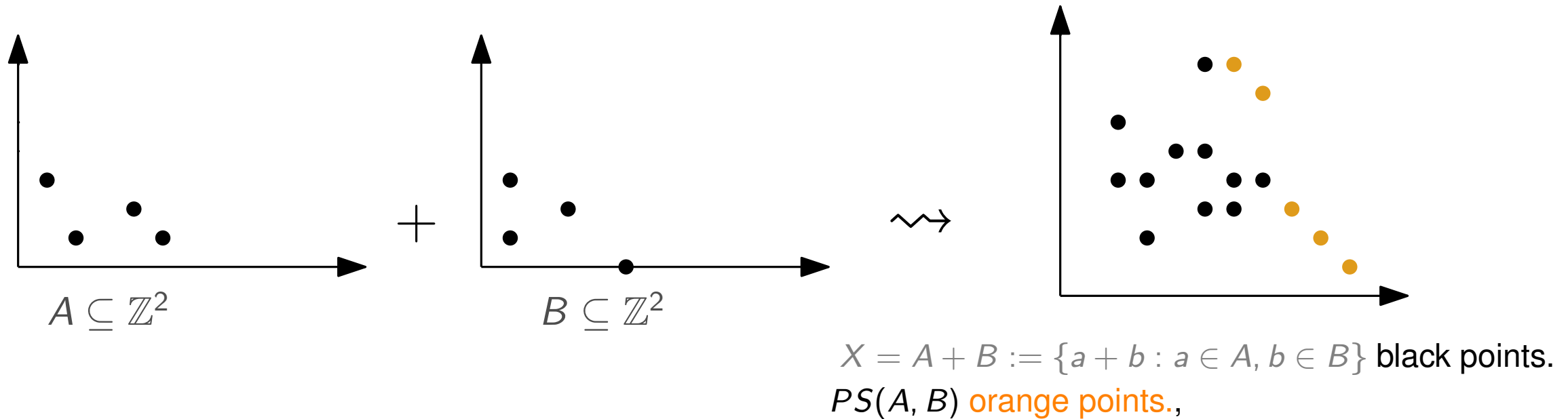
## Another hard problem: 2-D Succinct-Skyline /Pareto Sum:

Def:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute set  $PS(A, B) := \{x \in X = A + B \mid \neg \exists x' \neq x : x' \geq x\}$ .



## Another hard problem: 2-D Succinct-Skyline /Pareto Sum:

Def:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute set  $PS(A, B) := \{x \in X = A + B \mid \neg \exists x' \neq x : x' \geq x\}$ .



Lower bounds known, [GK25, FHSST24] (promise output size  $\Theta(n)$ ). Outp. sens UB in time  $\tilde{O}(nk)$ . [FHSST23]

Multivariate 3-SUM as a barrier.

Multivariate 3-SUM as a barrier.  
3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

$A := \{1, 3, 4, 5\}, B := \{7, 9, 10, 11\}, C := \{10, 15, 9\}$

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

$$A := \{1, 3, 4, 5\}, B := \{7, 9, 10, 11\}, C := \{10, 15, 9\} \\ 1, 1, 0$$

## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

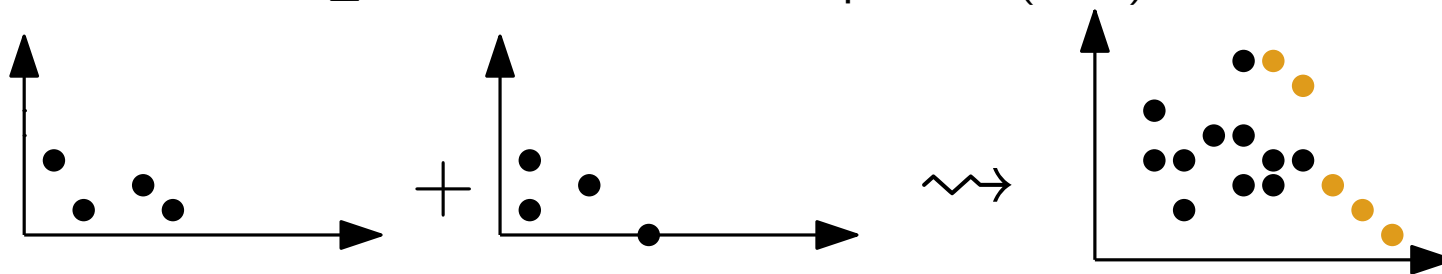
All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

$A := \{1, 3, 4, 5\}, B := \{7, 9, 10, 11\}, C := \{10, 15, 9\}$   
 $1, 1, 0$

Pareto Sum:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute  $PS(A, B)$  where  $k$  is the output size  $|PS(A, B)|$ .



## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

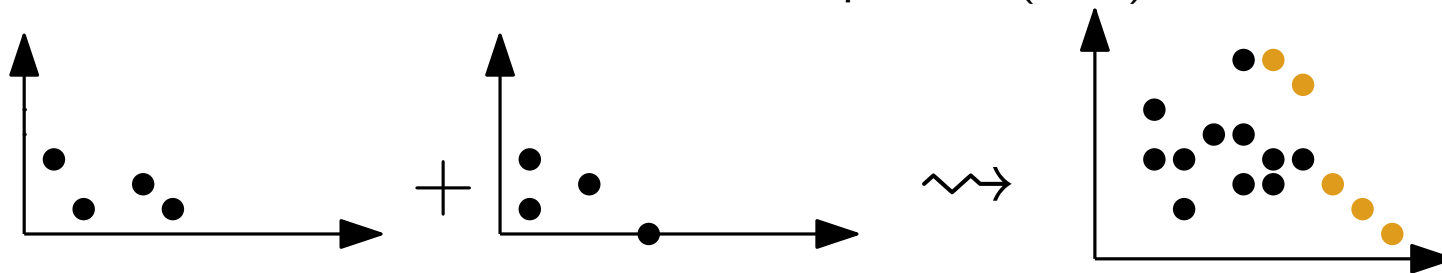
All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

$A := \{1, 3, 4, 5\}, B := \{7, 9, 10, 11\}, C := \{10, 15, 9\}$   
 $1, 1, 0$

Pareto Sum:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute  $PS(A, B)$  where  $k$  is the output size  $|PS(A, B)|$ .



## Multivariate 3-SUM as a barrier.

3-SUM: Given sets  $|A| = |B| = |C| = n$ .

Do there exist  $a \in A, b \in B, c \in C$  s.t  $a + b = c$ ?

i.e  $(A + B) \cap C \neq \emptyset$

3-SUM hypothesis: Let  $\epsilon > 0$ . No algorithm for 3-SUM in time  $O(n^{2-\epsilon})$ .

All-ints 3-SUM: Given sets  $A, B, C$  with  $|A| = |B| = |C| = n$ .

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

i.e compute  $(A + B) \cap C$

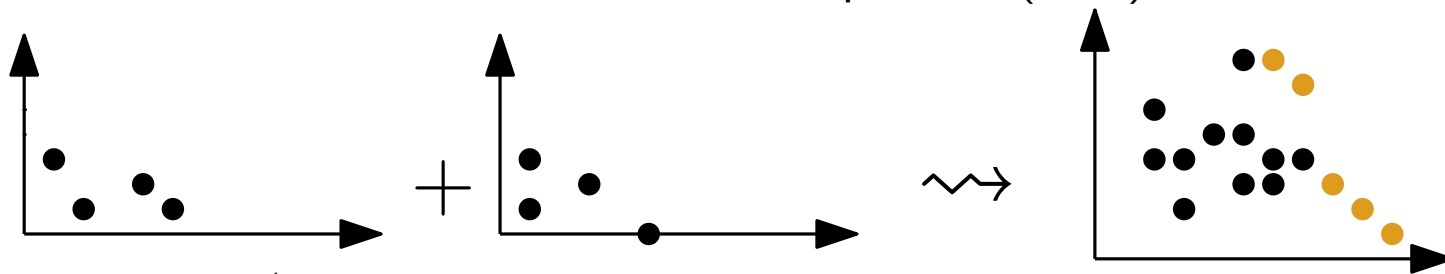
All-ints 3-SUM( $n, n, k$ ): Given sets  $A, B, C$  with  $|A| = |B| = n$  and  $|C| = k \leq n$ ,

Determine for each  $c \in C$  whether there exist  $a \in A, b \in B$  s.t  $c = a + b$ .

Solvable in time  $\tilde{O}(nk)$  (tight).

$A := \{1, 3, 4, 5\}, B := \{7, 9, 10, 11\}, C := \{10, 15, 9\}$   
 $1, 1, 0$

Pareto Sum:  $X = A + B \subseteq \mathbb{Z}^2$  succ. defined, compute  $PS(A, B)$  where  $k$  is the output size  $|PS(A, B)|$ .



**Thm:** Exists an  $O((nk)^{1-\epsilon})$  alg. for All-ints 3-SUM( $n, n, k$ ) iff exists an outp-sens. alg. for Pareto Sum in  $O((nk)^{1-\epsilon'})$ .

Upper bound Tool:

Upper bound Tool: All-ints FOP $\mathbb{Z}$  queries

Upper bound Tool: All-ints  $\text{FOP}_{\mathbb{Z}}$  queries

Let  $\varphi(a, b, c)$  be a linear arithmetic formula.

## Upper bound Tool: All-ints $\text{FOP}_{\mathbb{Z}}$ queries

Let  $\varphi(a, b, c)$  be a linear arithmetic formula.

All-ints $\varphi$ .
Sets $A \subseteq \mathbb{Z}^{d_1} B \subseteq \mathbb{Z}^{d_2} C \subseteq \mathbb{Z}^{d_3},  A  =  B  = n,  C  = k$
Determine for each $c \in C$ whether $\exists a \in A \exists b \in B : \varphi(a, b, c)$ .

## Upper bound Tool: All-ints $\text{FOP}_{\mathbb{Z}}$ queries

Let  $\varphi(a, b, c)$  be a linear arithmetic formula.

All-ints $\varphi$ .
Sets $A \subseteq \mathbb{Z}^{d_1} B \subseteq \mathbb{Z}^{d_2} C \subseteq \mathbb{Z}^{d_3},  A  =  B  = n,  C  = k$
Determine for each $c \in C$ whether $\exists a \in A \exists b \in B : \varphi(a, b, c)$ .

$\rightarrow \tilde{O}(1)$  calls All-ints 3-SUM( $n, n, k$ ). [GK25]

## Upper bound Tool: All-ints FOP $_{\mathbb{Z}}$ queries

Let  $\varphi(a, b, c)$  be a linear arithmetic formula.

All-ints $\varphi$ .
Sets $A \subseteq \mathbb{Z}^{d_1} B \subseteq \mathbb{Z}^{d_2} C \subseteq \mathbb{Z}^{d_3},  A  =  B  = n,  C  = k$
Determine for each $c \in C$ whether $\exists a \in A \exists b \in B : \varphi(a, b, c)$ .

$\rightarrow \tilde{O}(1)$  calls All-ints 3-SUM( $n, n, k$ ). [GK25]

Examples:  $\varphi_1(a, b, c) := a + b + c = 0,$

## Upper bound Tool: All-ints FOP $_{\mathbb{Z}}$ queries

Let  $\varphi(a, b, c)$  be a linear arithmetic formula.

All-ints $\varphi$ .
Sets $A \subseteq \mathbb{Z}^{d_1} B \subseteq \mathbb{Z}^{d_2} C \subseteq \mathbb{Z}^{d_3},  A  =  B  = n,  C  = k$
Determine for each $c \in C$ whether $\exists a \in A \exists b \in B : \varphi(a, b, c)$ .

$\rightarrow \tilde{O}(1)$  calls All-ints 3-SUM( $n, n, k$ ). [GK25]

Examples:  $\varphi_1(a, b, c) := a + b + c = 0,$

$\varphi_2(a, b, c) := c[0] \leq a + b \leq c[1]$

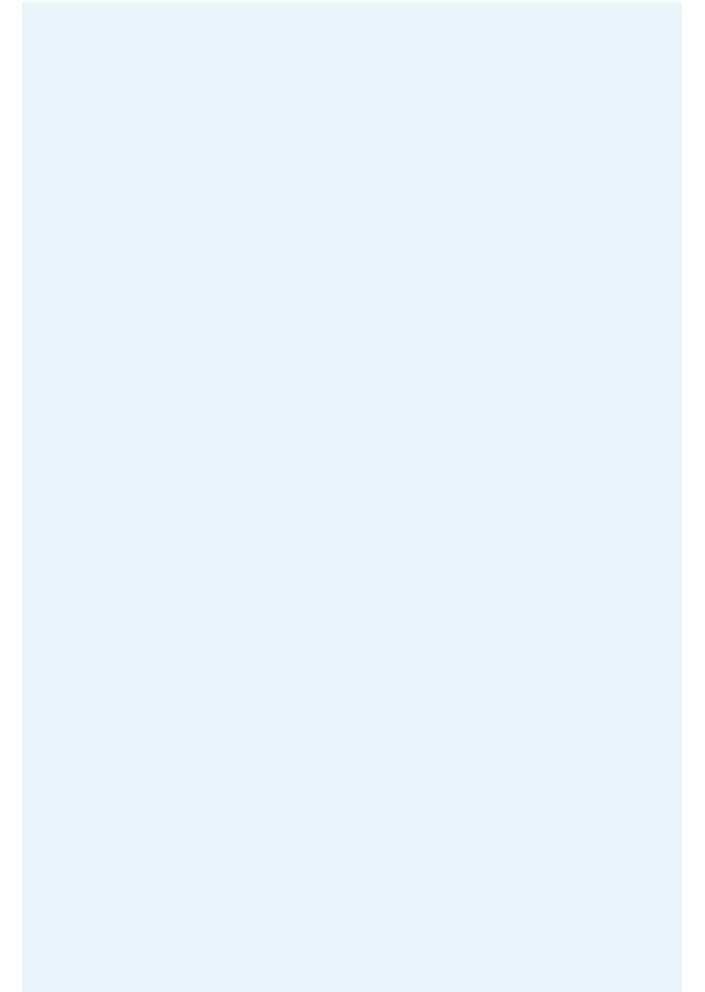
**Subroutine:**

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

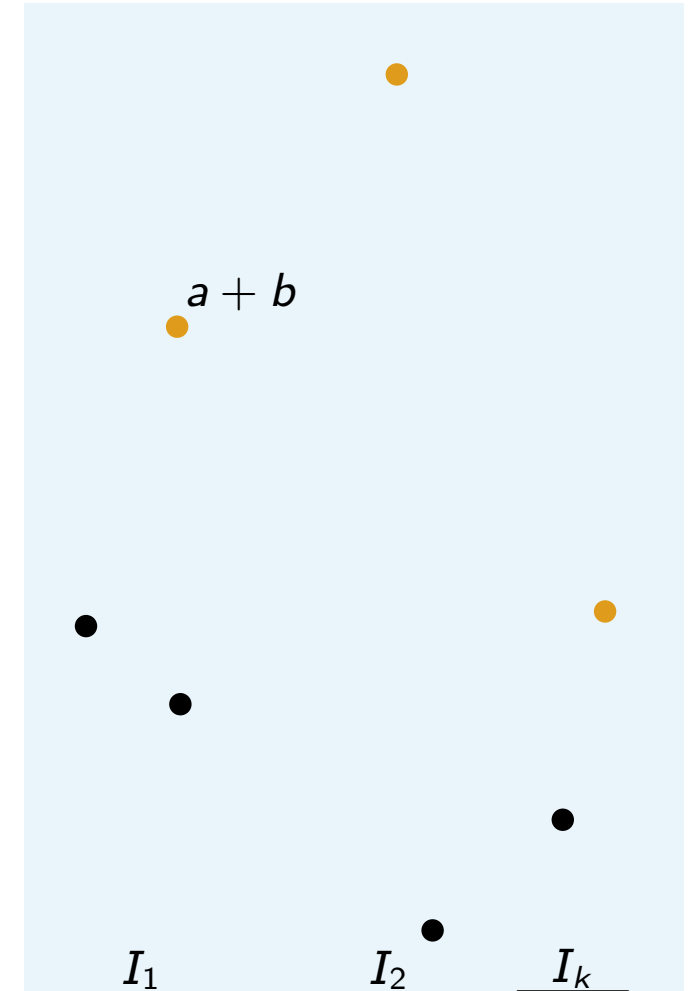
Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ .  
Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

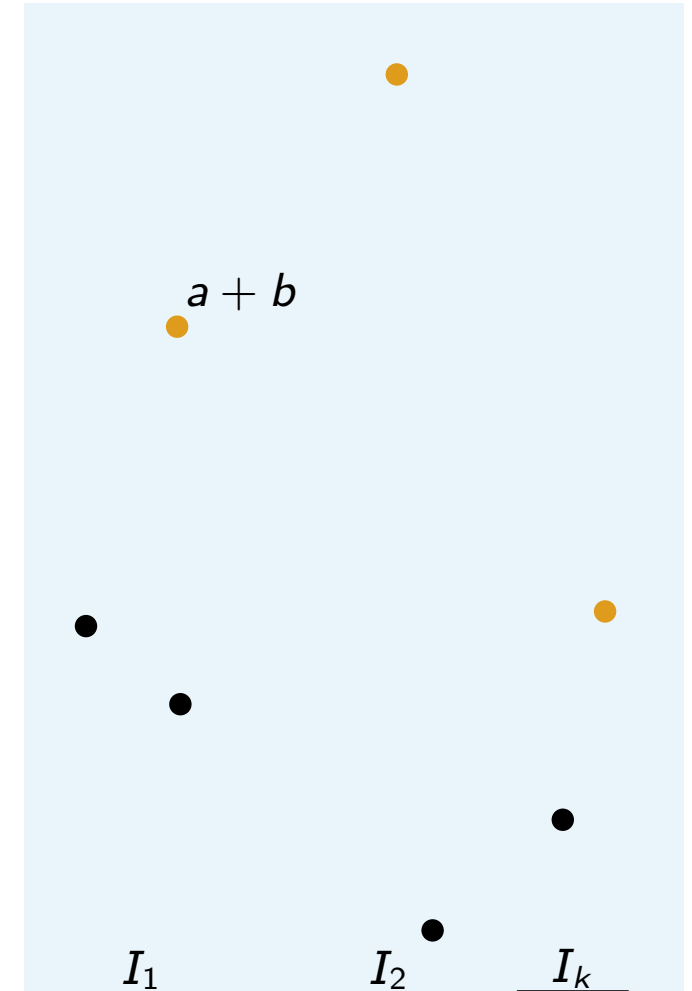
Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

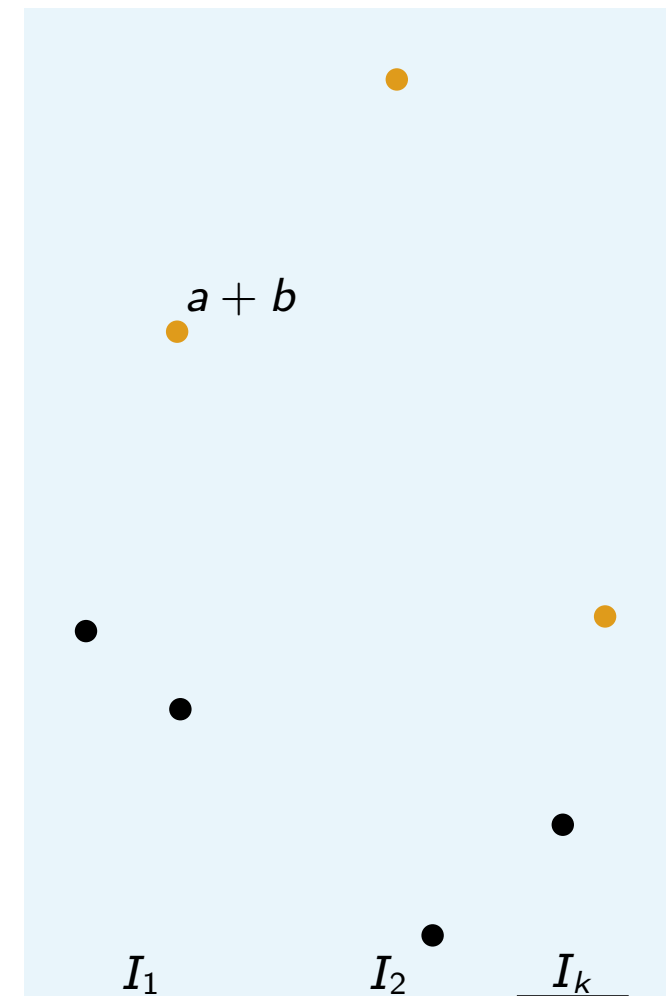


## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ .



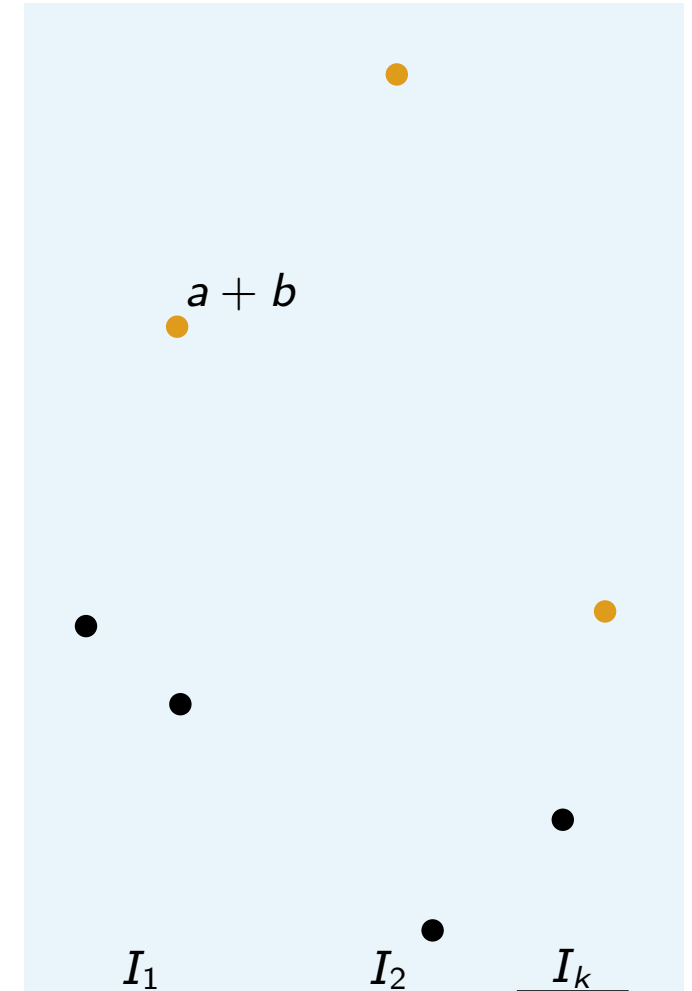
## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$



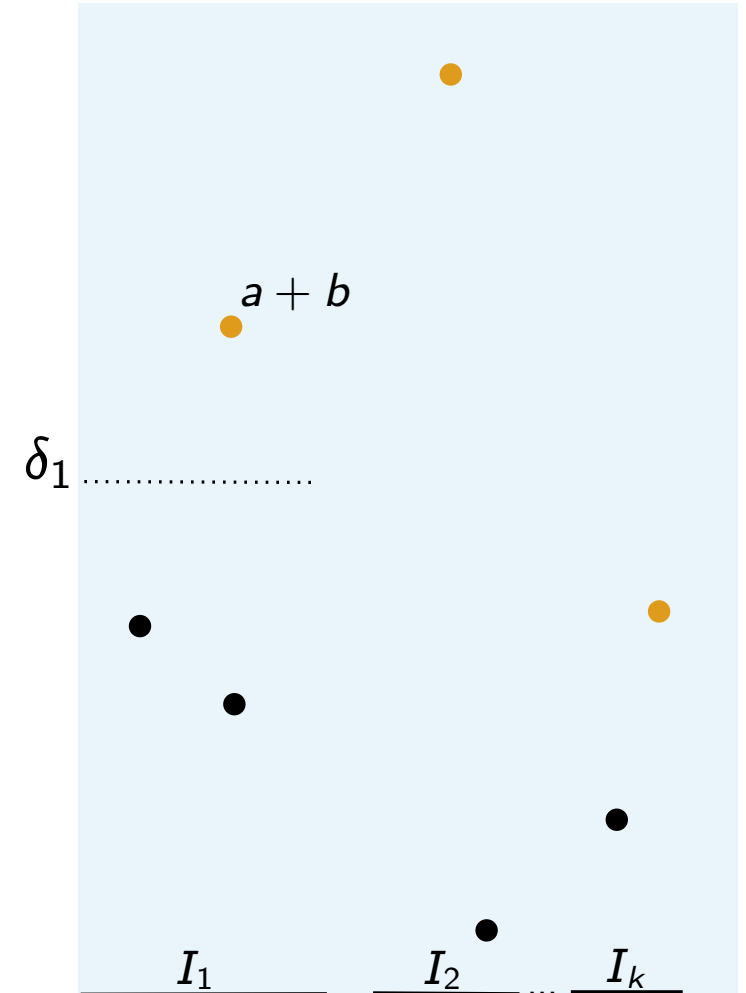
## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$



## Subroutine:

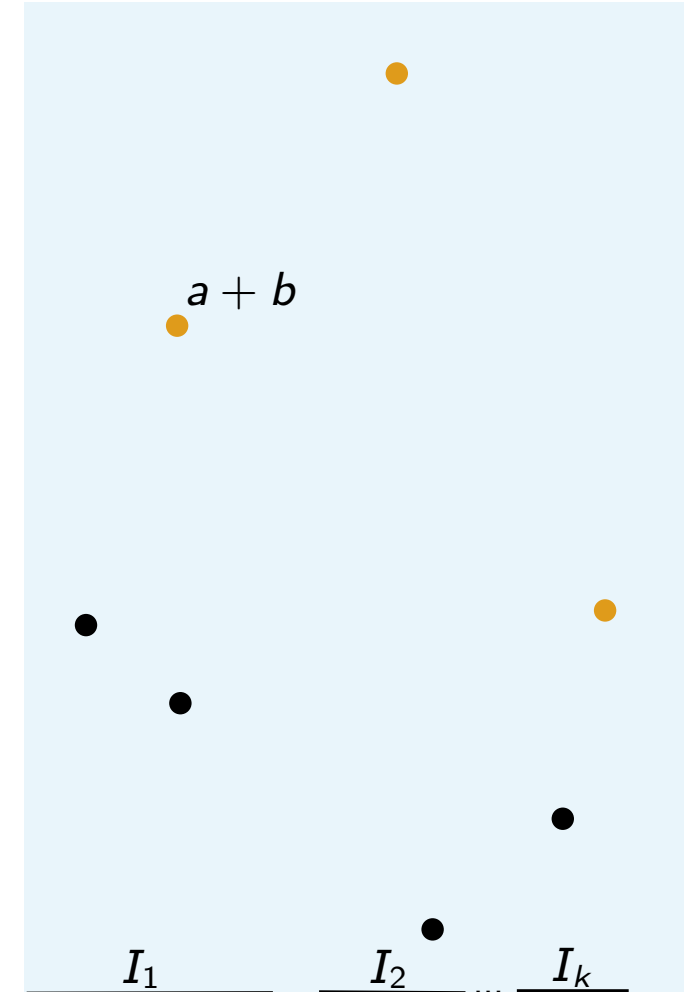
Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$

Parallel binary search all  $\delta_i$  in time  $O(\log(U))$ .



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

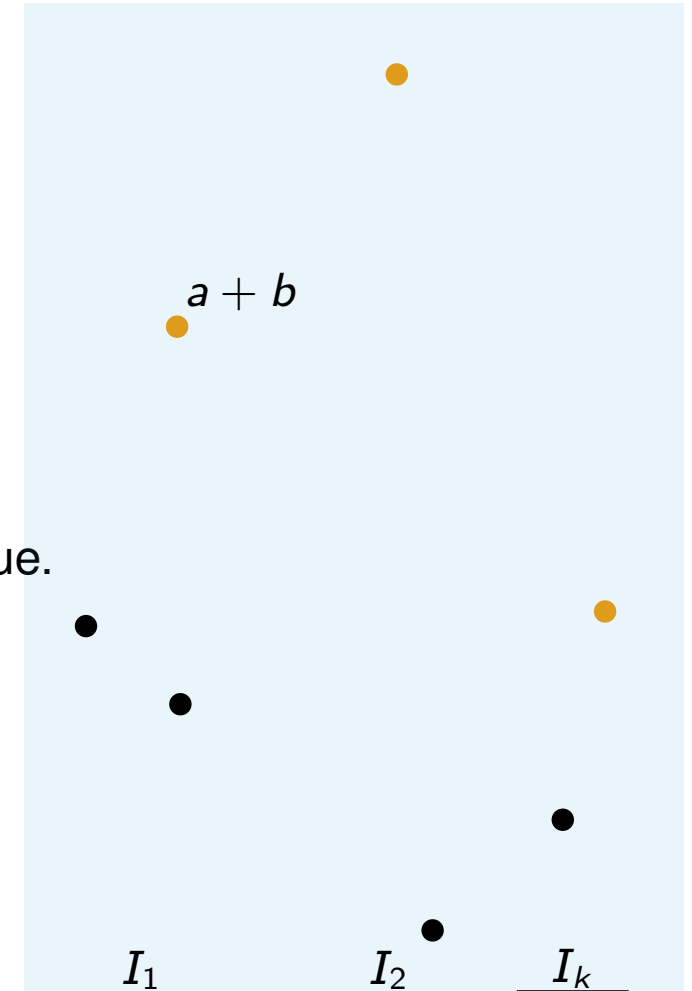
Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$

Parallel binary search all  $\delta_i$  in time  $O(\log(U))$ .

Let  $\delta^*$  be optimal  $y$ -value. Bsearch again on  $\gamma$  to find rightmost matching  $x$ -value.



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

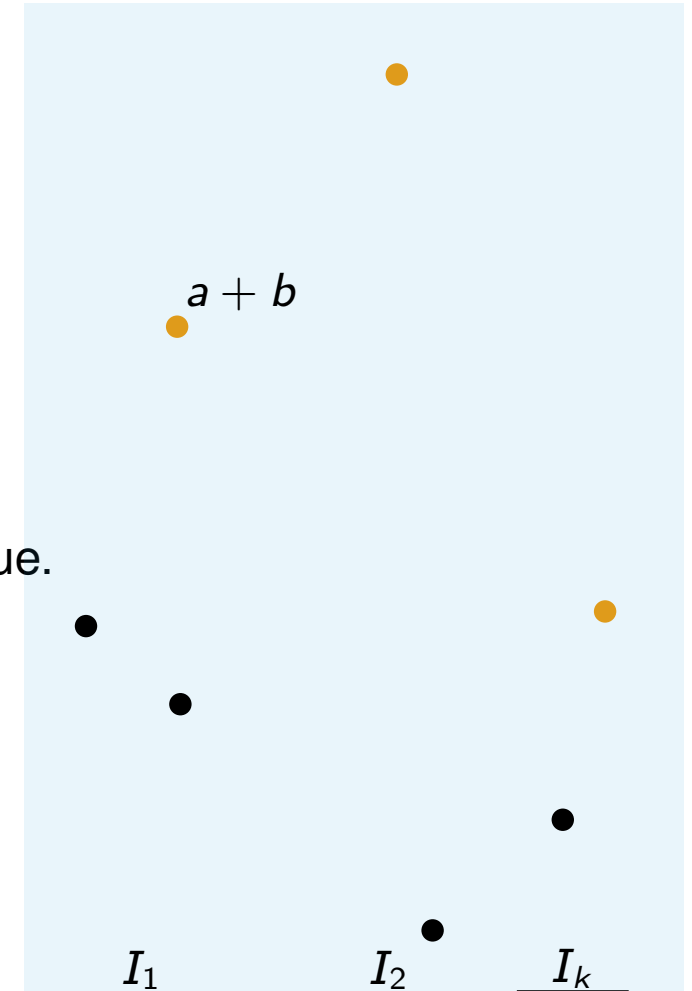
Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$

Parallel binary search all  $\delta_i$  in time  $O(\log(U))$ .

Let  $\delta^*$  be optimal  $y$ -value. Bsearch again on  $\gamma$  to find rightmost matching  $x$ -value.

Let  $C' = \{(s_i, e_i, \delta_i^*, \gamma_i) : i \in \{0, \dots, k\}\}$ .



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

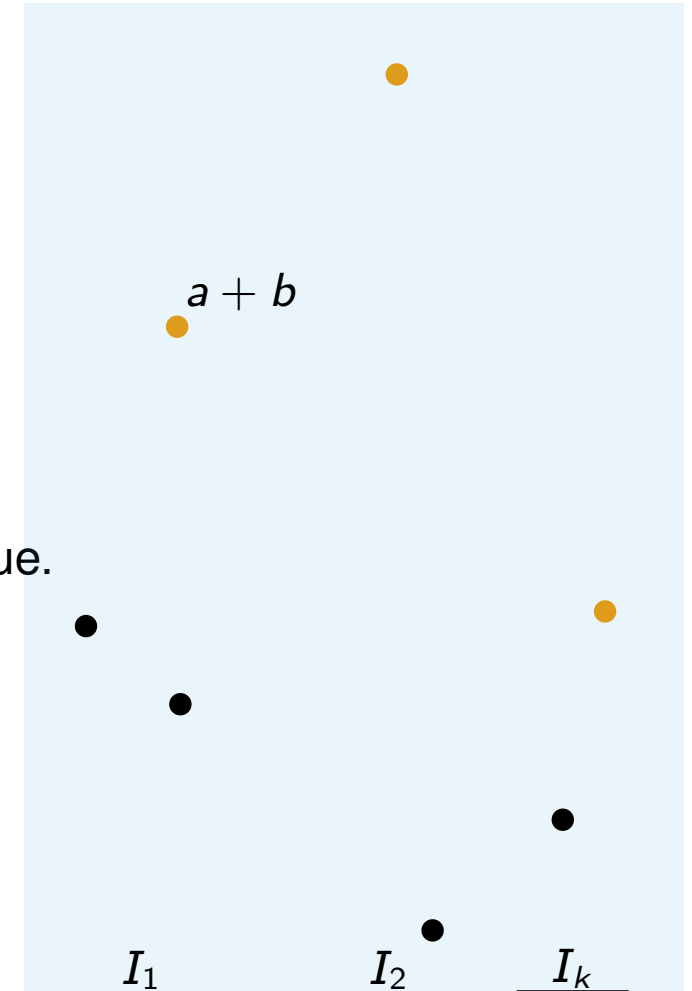
$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$

Parallel binary search all  $\delta_i$  in time  $O(\log(U))$ .

Let  $\delta^*$  be optimal  $y$ -value. Bsearch again on  $\gamma$  to find rightmost matching  $x$ -value.

Let  $C' = \{(s_i, e_i, \delta_i^*, \gamma_i) : i \in \{0, \dots, k\}\}$ .

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y = c.\delta^* \wedge a.x + b.x \geq c.\gamma$$



## Subroutine:

Let All-ints 3-SUM( $n, n, k$ ) in time  $T(n, k)$ .

Lemma: Given  $A, B \subseteq \mathbb{Z}^2$  and Intervals  $I_1 = [s_1, e_1], \dots, I_k = [s_k, e_k]$ . Determining for each interval  $I_j$  the highest point  $a+b$  with  $x$ -coord. inside  $I_j$  possible in  $\tilde{O}(T(n, k))$ .

Let  $C' = \{(s_i, e_i, \delta_i) : i \in \{1, \dots, k\}\}$ . All-ints FOP $_{\mathbb{Z}}$  query:

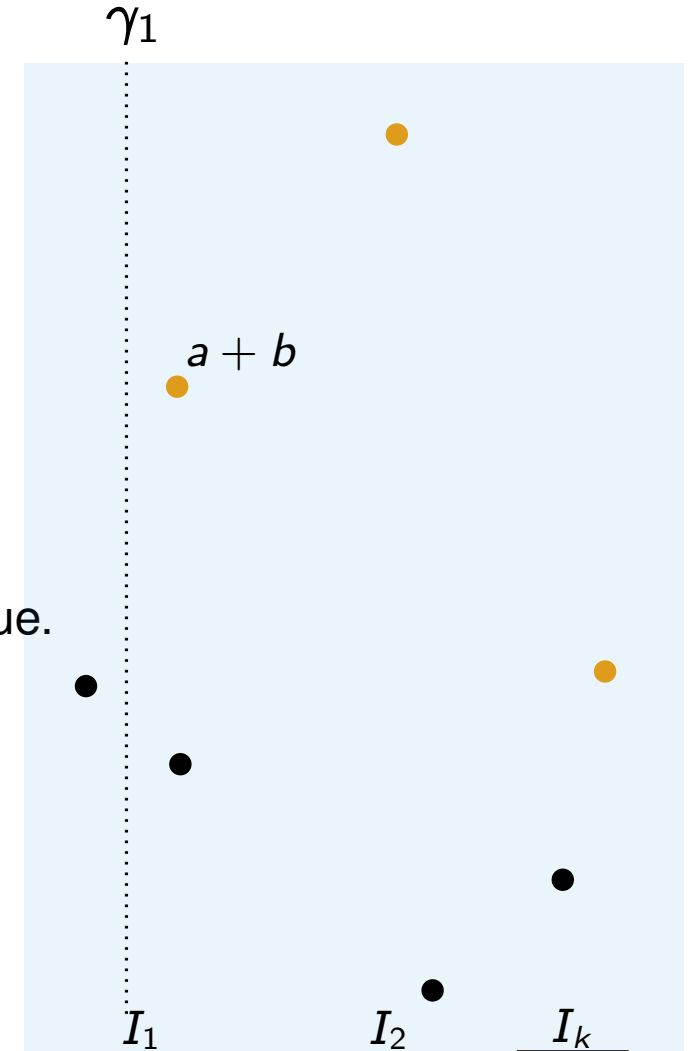
$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y \geq c.\delta$$

Parallel binary search all  $\delta_i$  in time  $O(\log(U))$ .

Let  $\delta^*$  be optimal  $y$ -value. Bsearch again on  $\gamma$  to find rightmost matching  $x$ -value.

Let  $C' = \{(s_i, e_i, \delta_i^*, \gamma_i) : i \in \{0, \dots, k\}\}$ .

$$\exists a \in A \exists b \in B : c.s \leq a.x + b.x \leq c.e \wedge a.y + b.y = c.\delta^* \wedge a.x + b.x \geq c.\gamma$$



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).



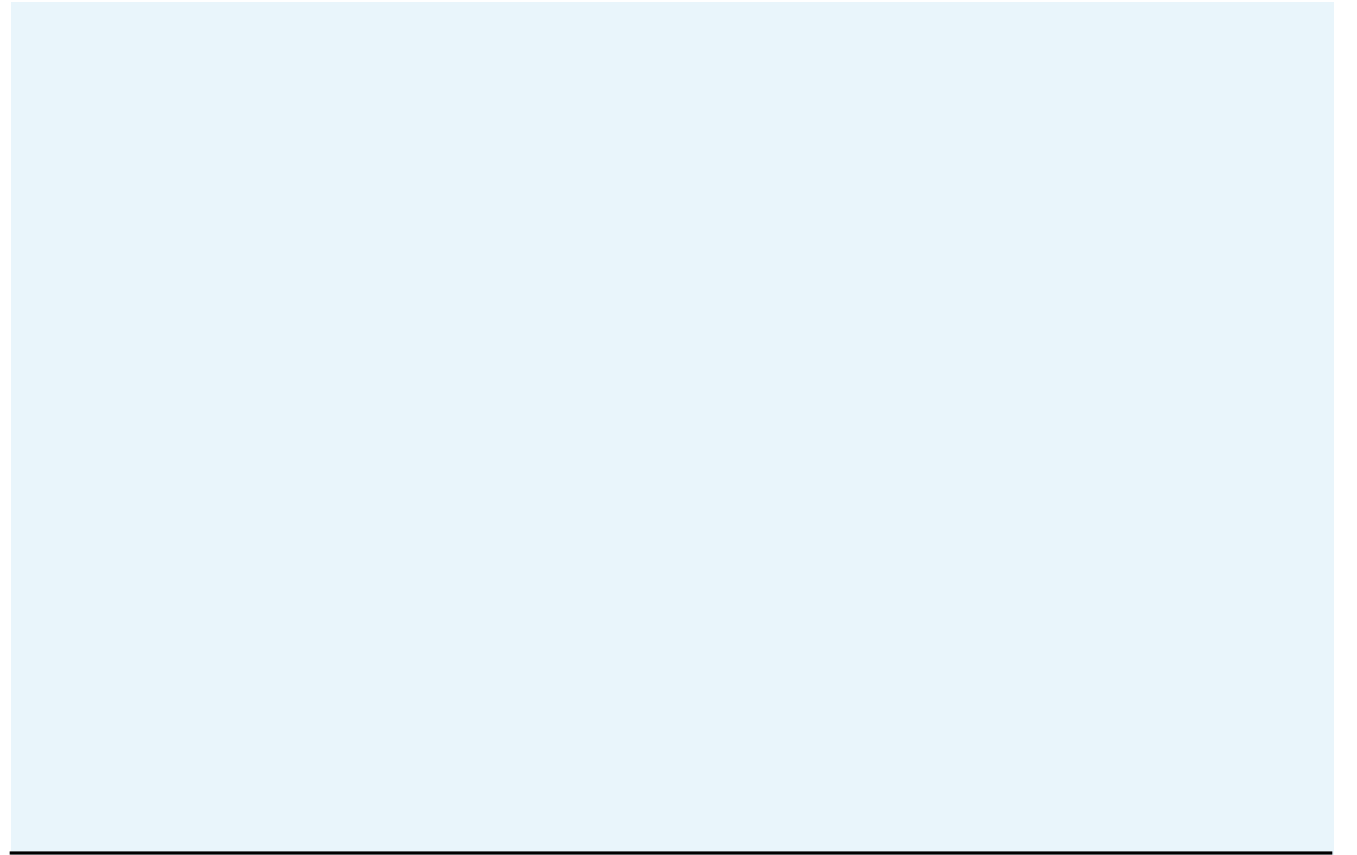
2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

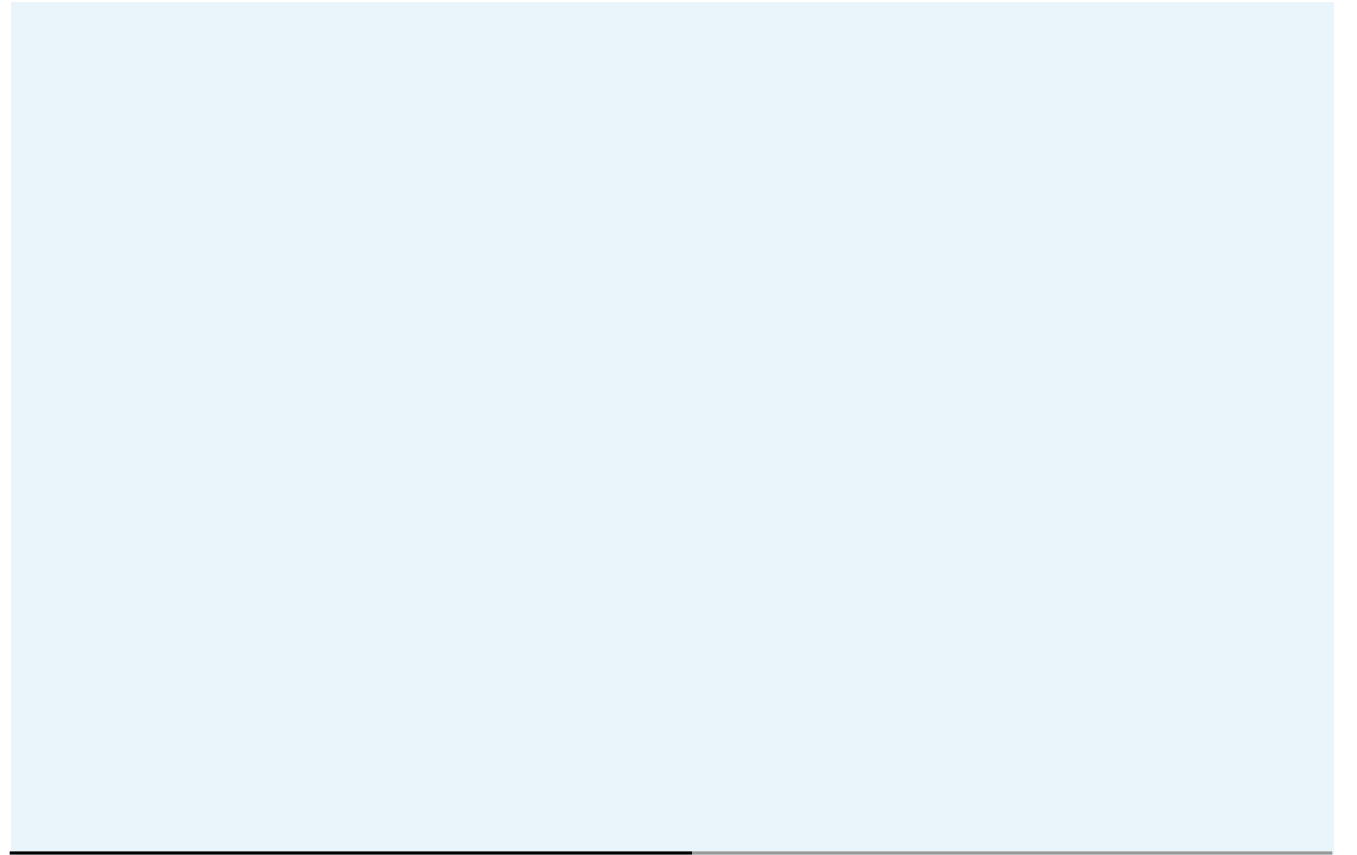
Start with interval  $I = [-M, M]$ ,



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

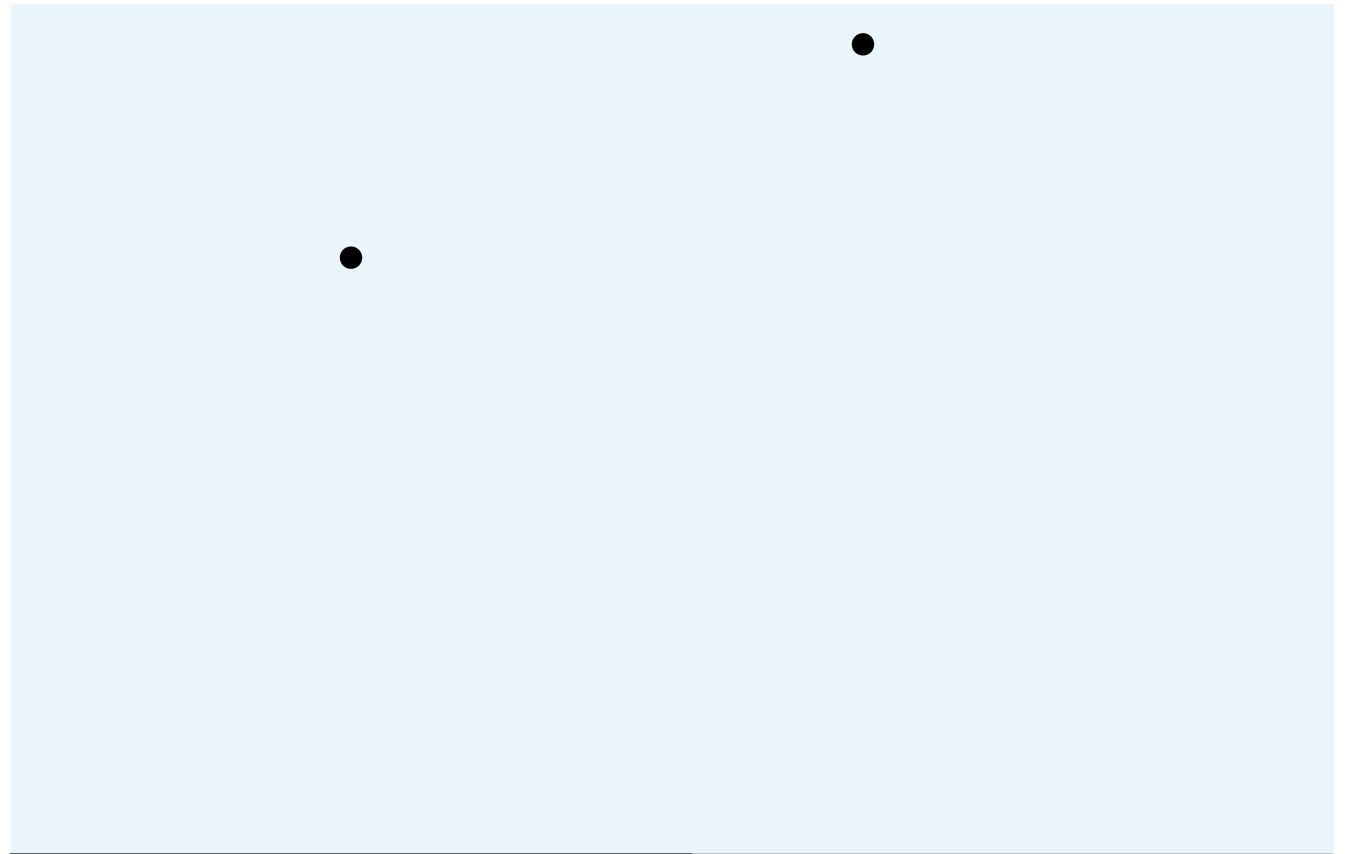
Halve the intervals and compute highest rightmost-points.



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

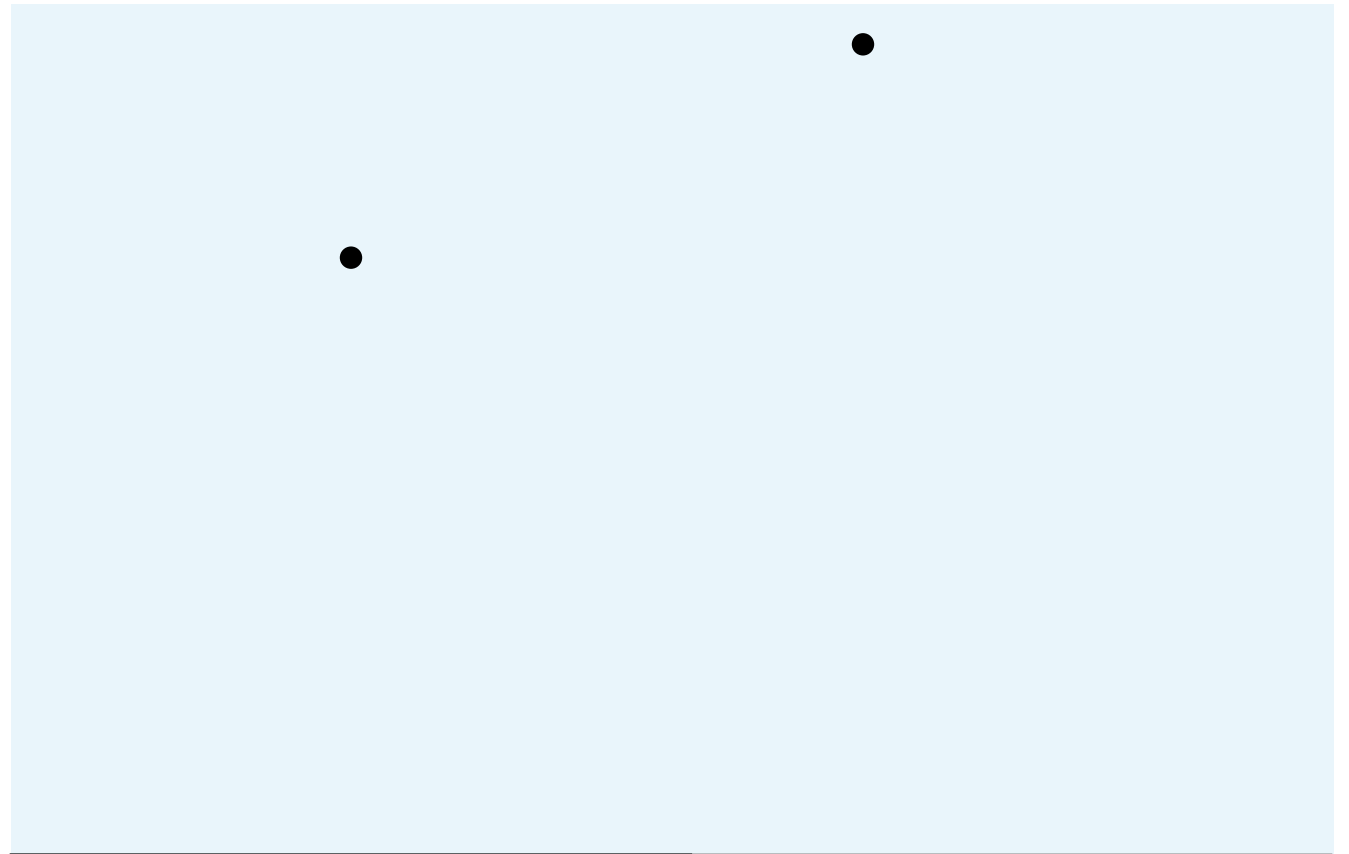


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

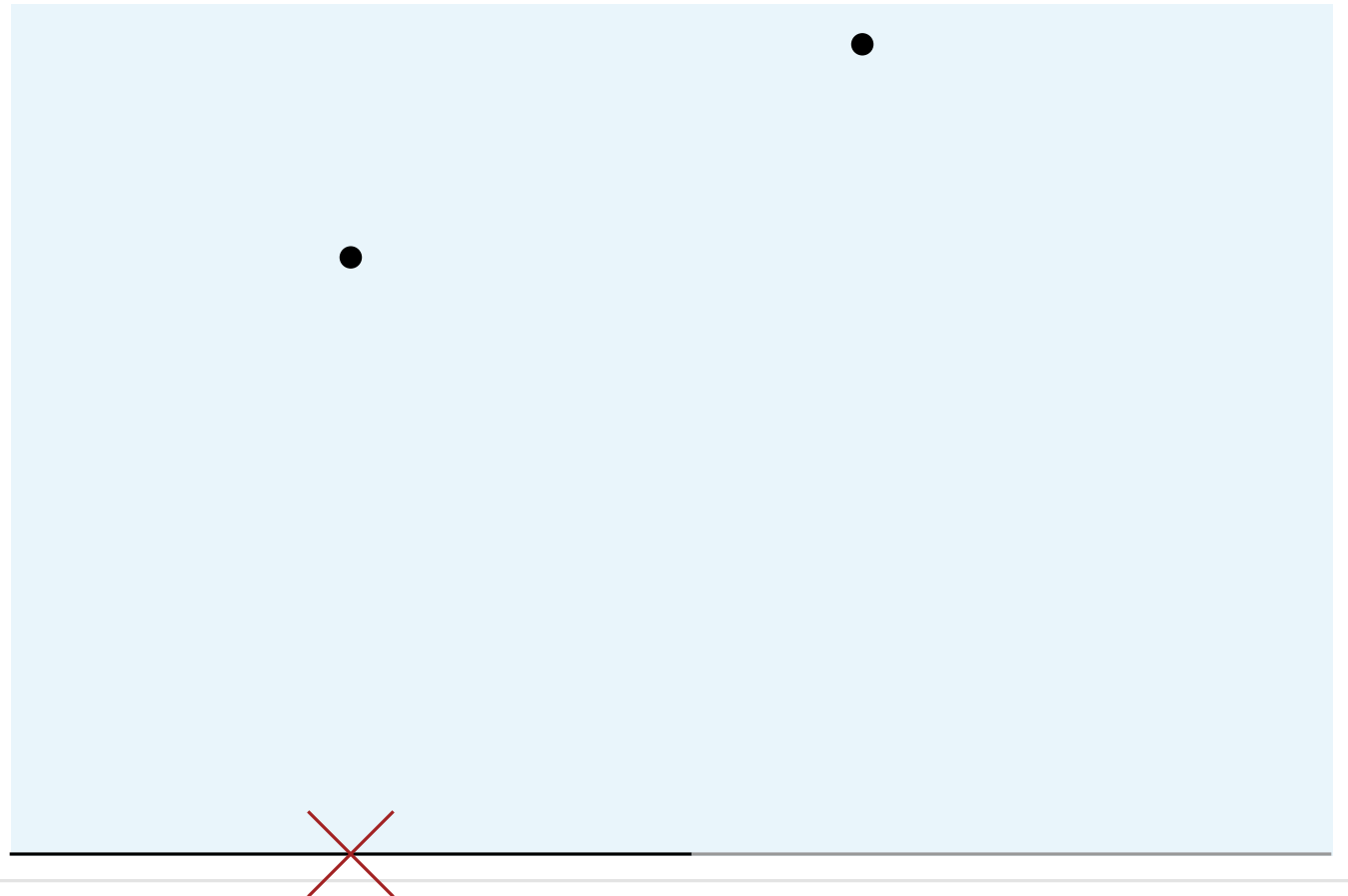


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

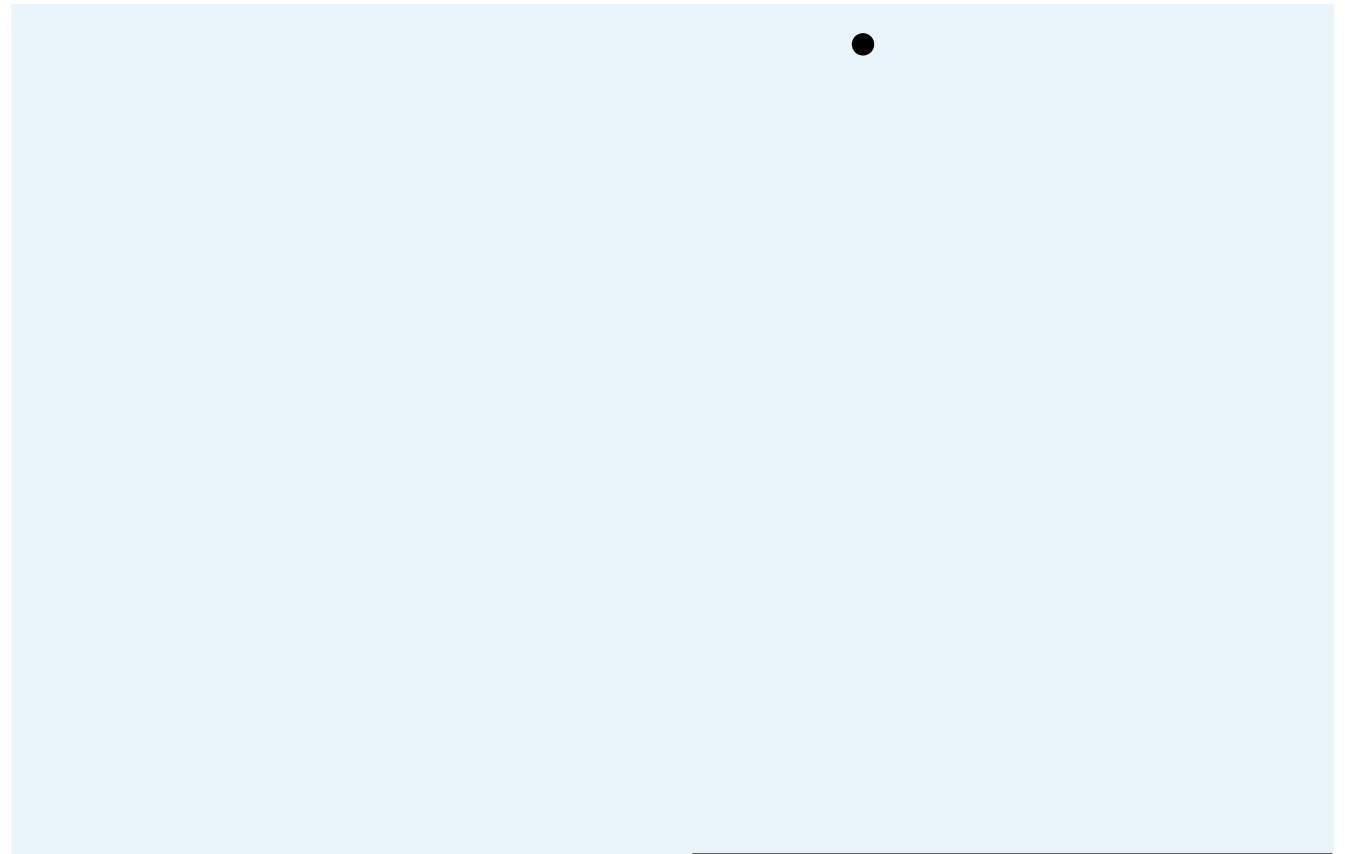


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

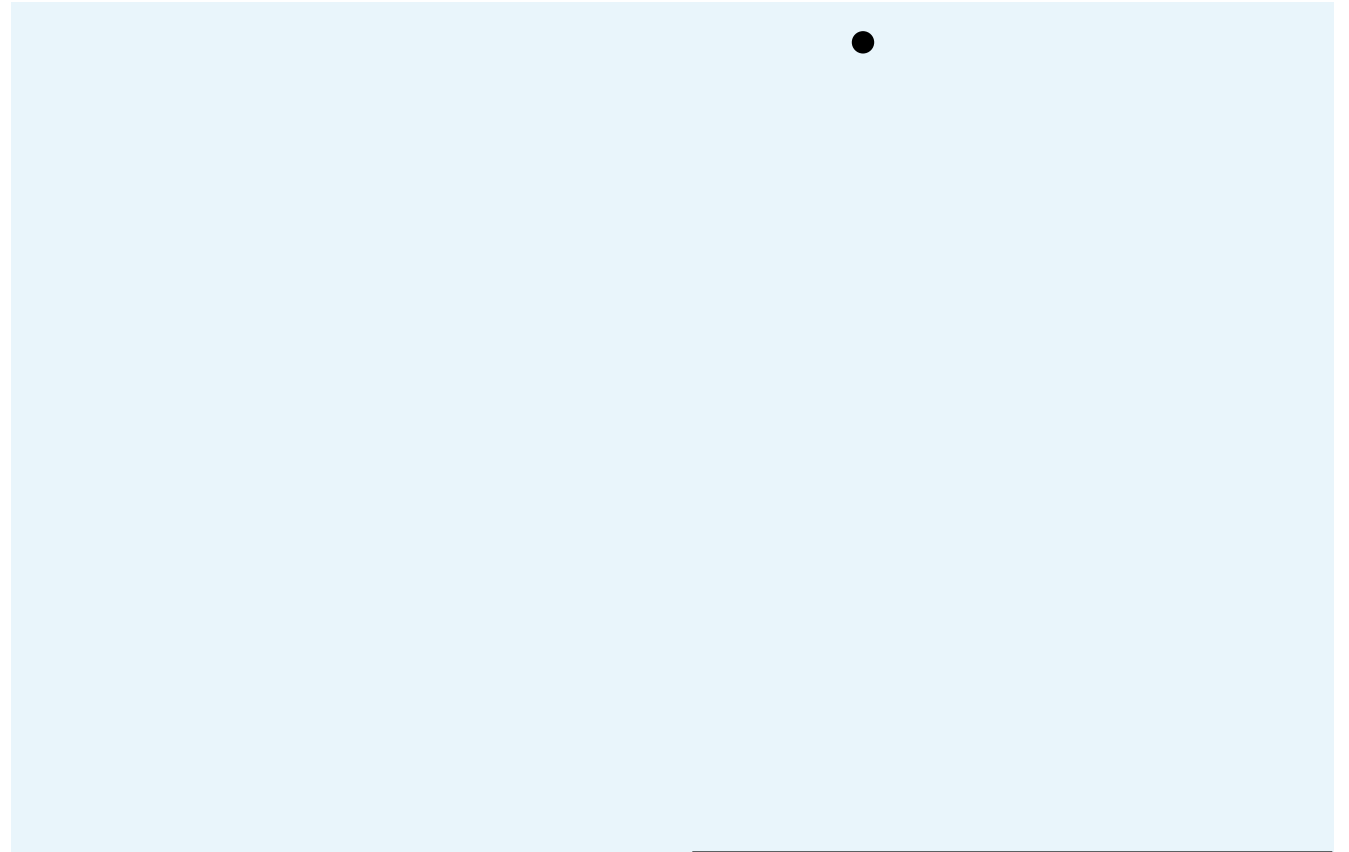


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

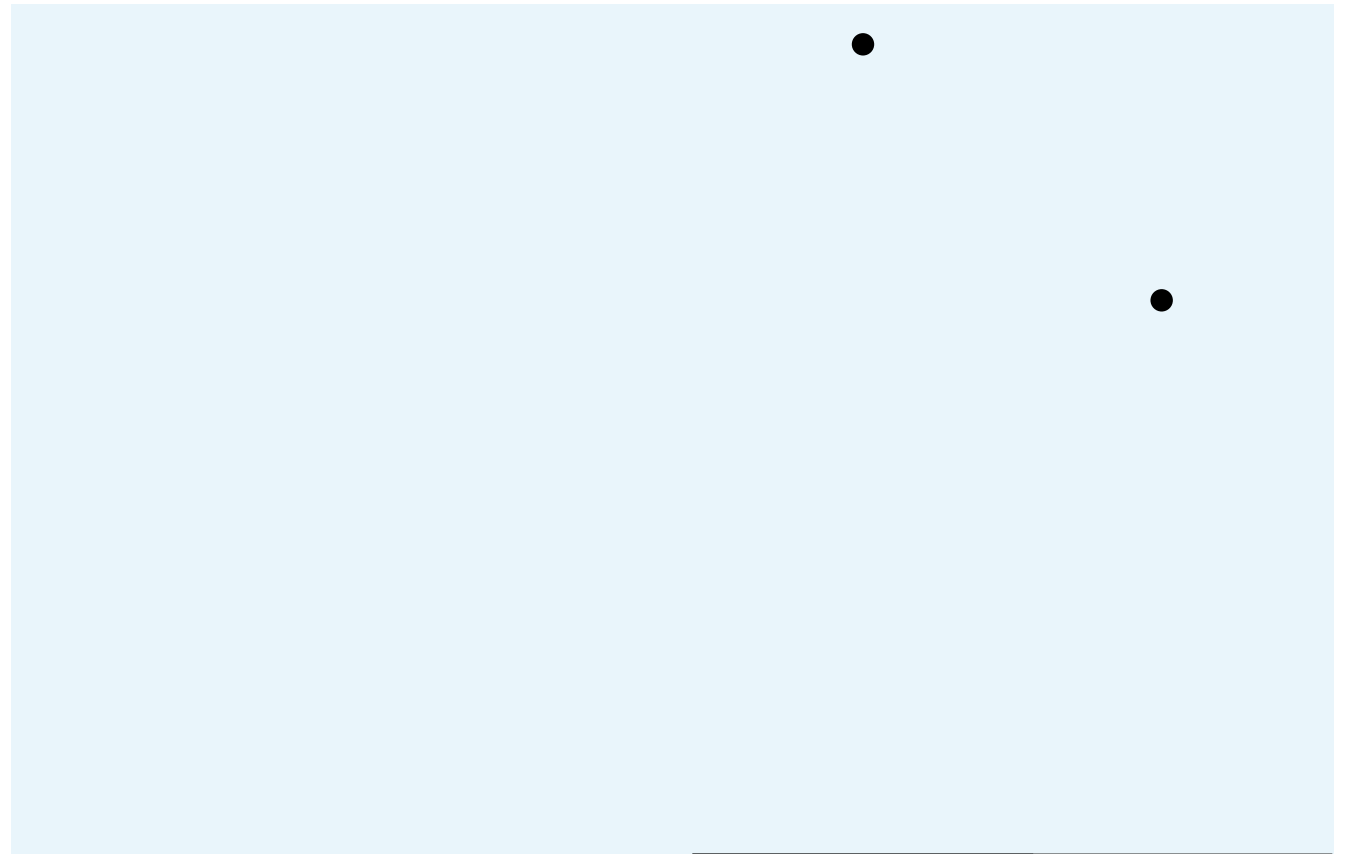


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

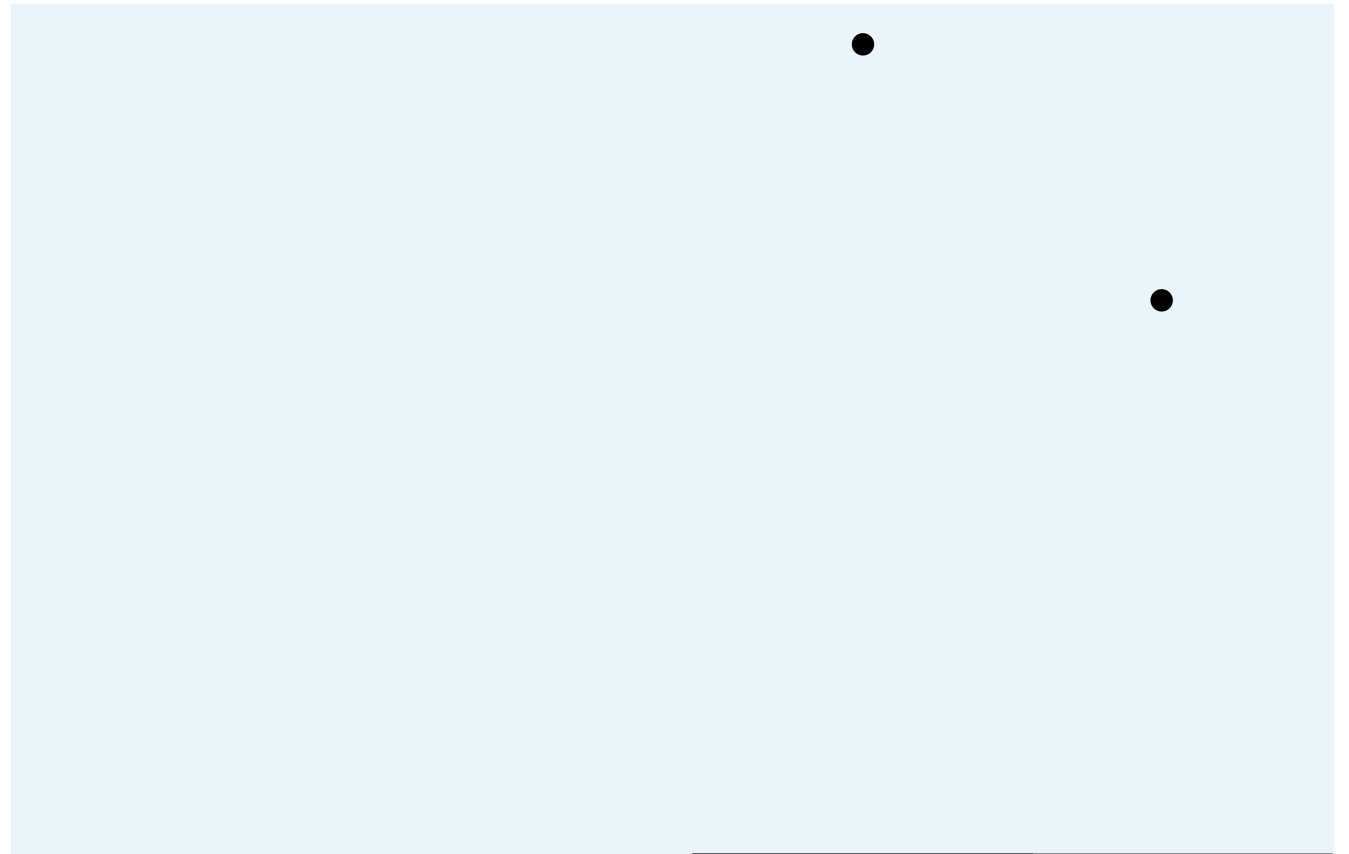


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

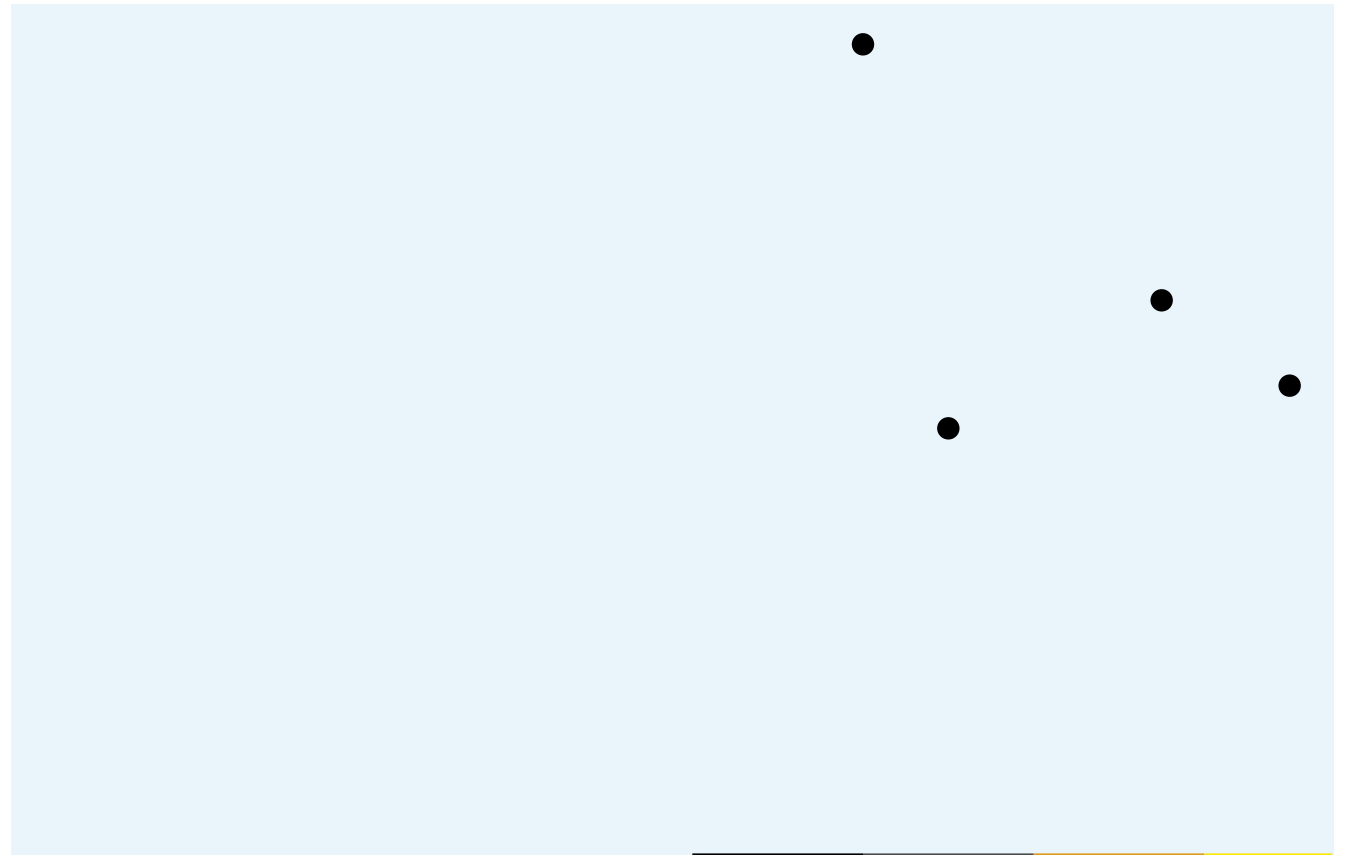


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

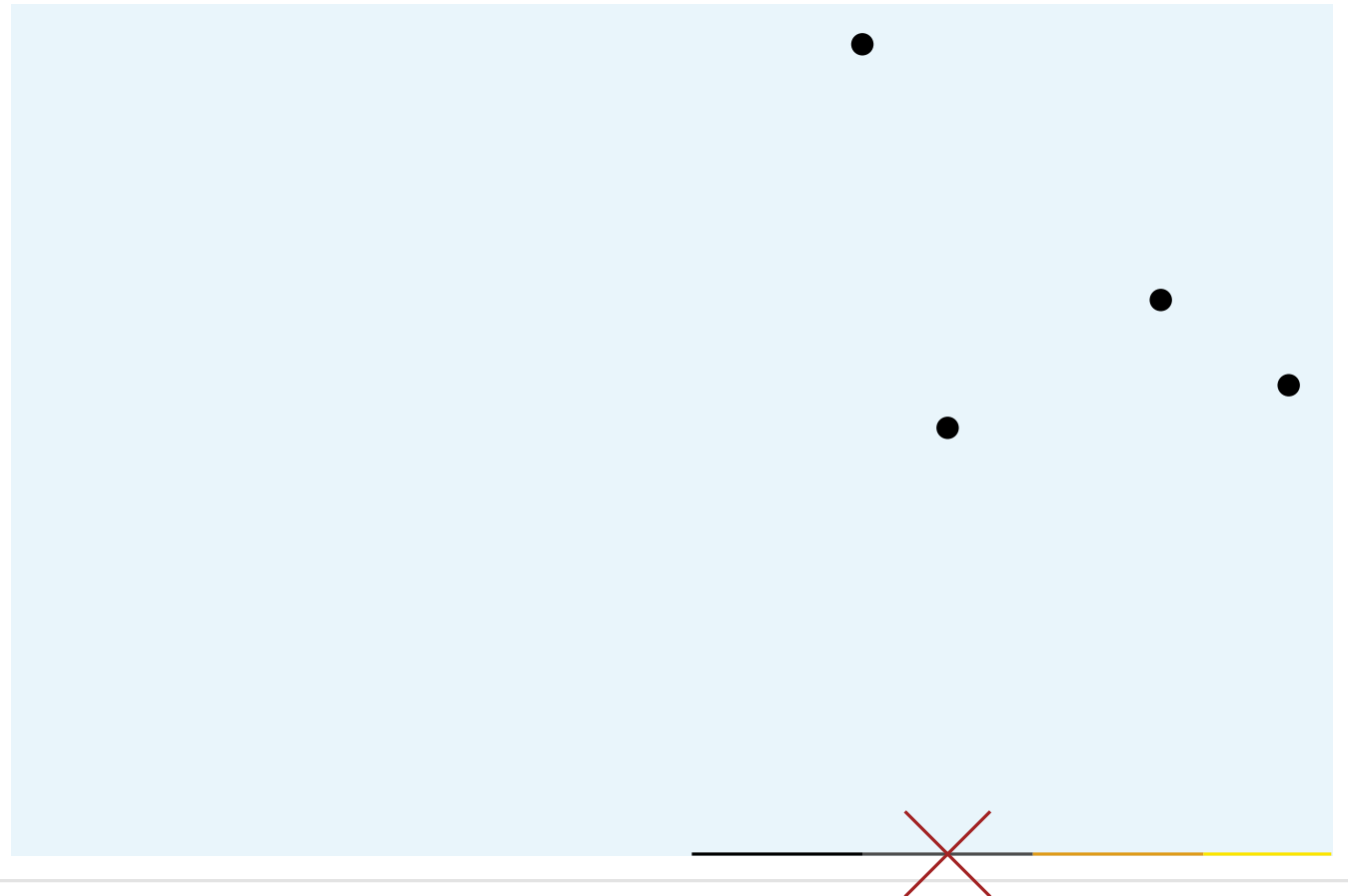


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

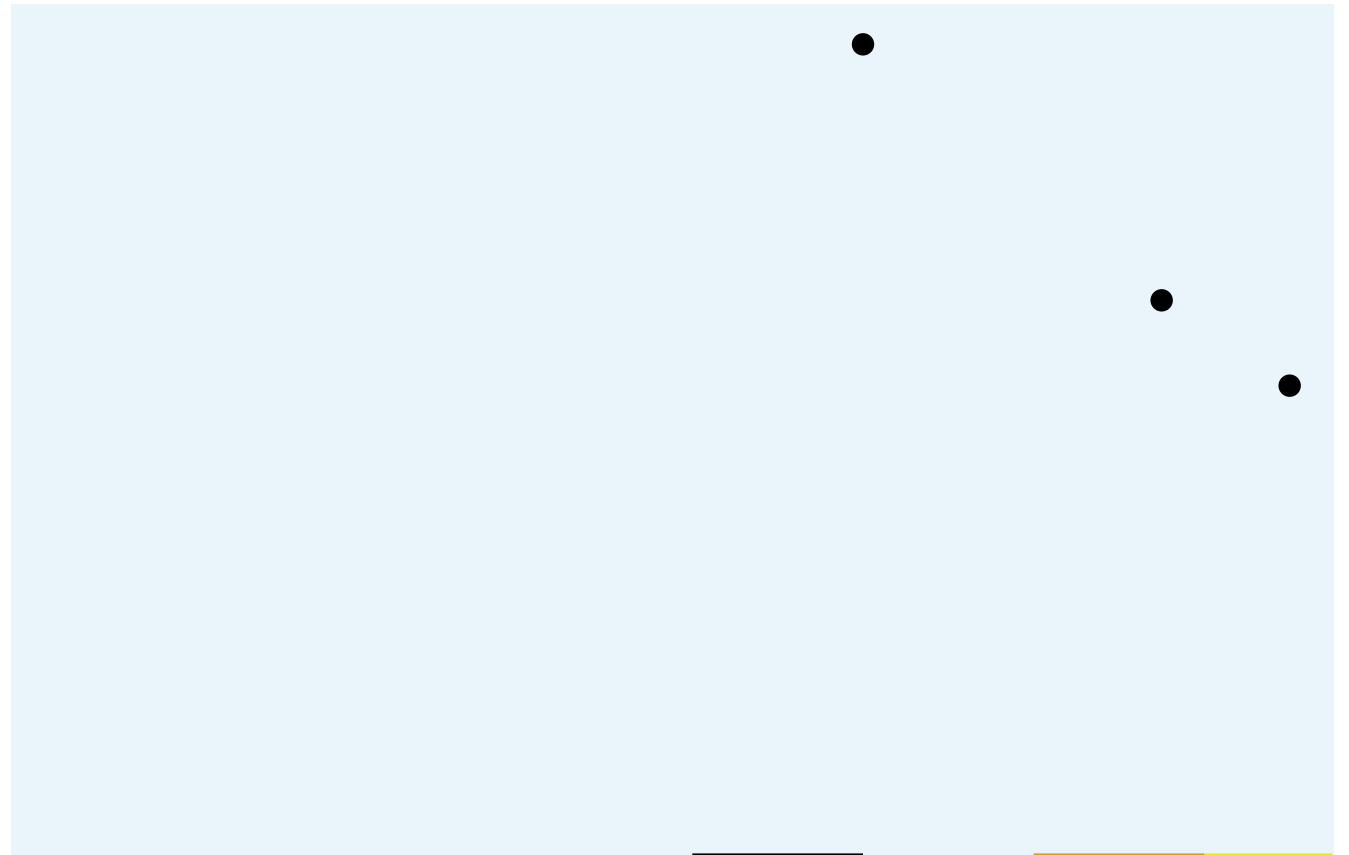


2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.



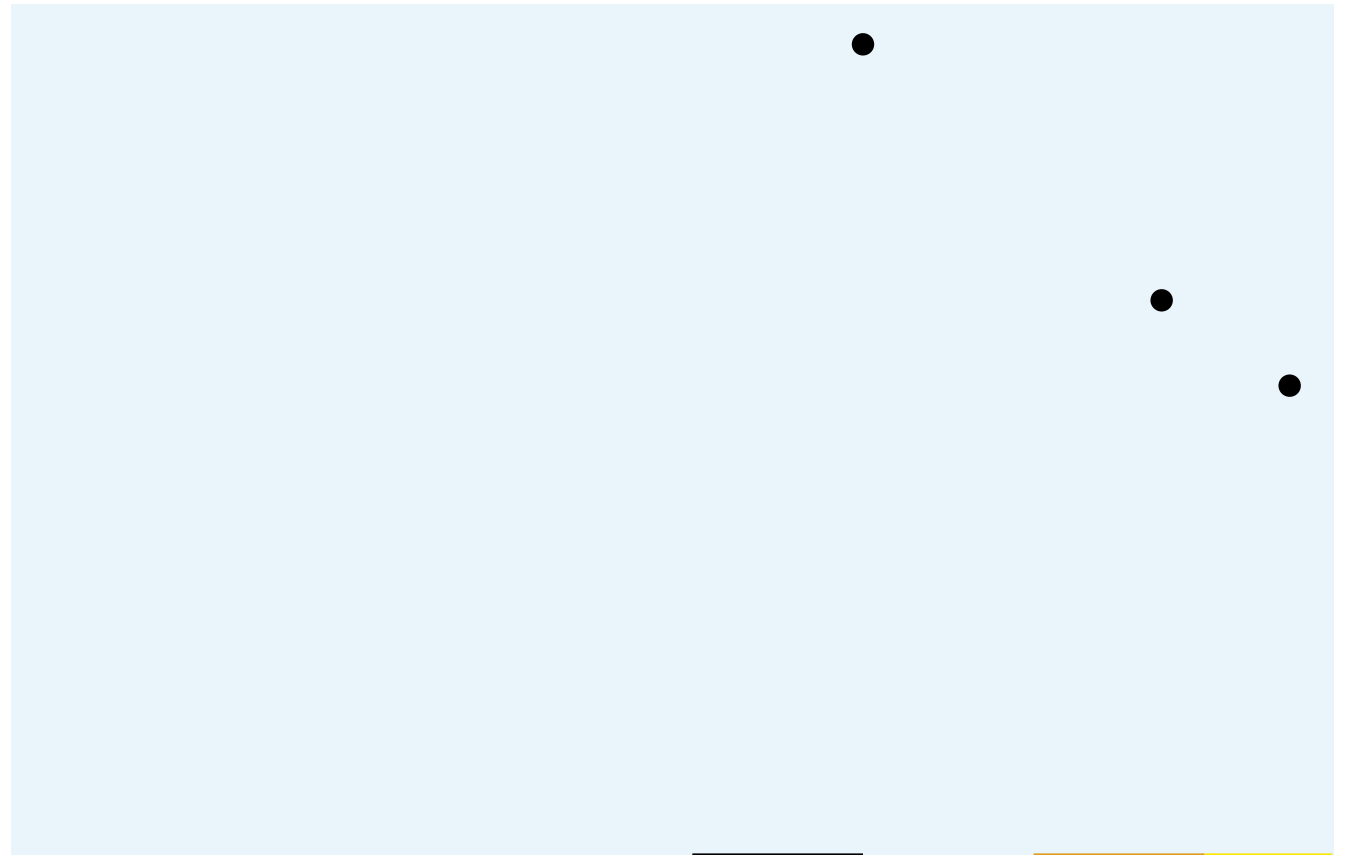
2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

After logarithmically many steps intervals of size 1.



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

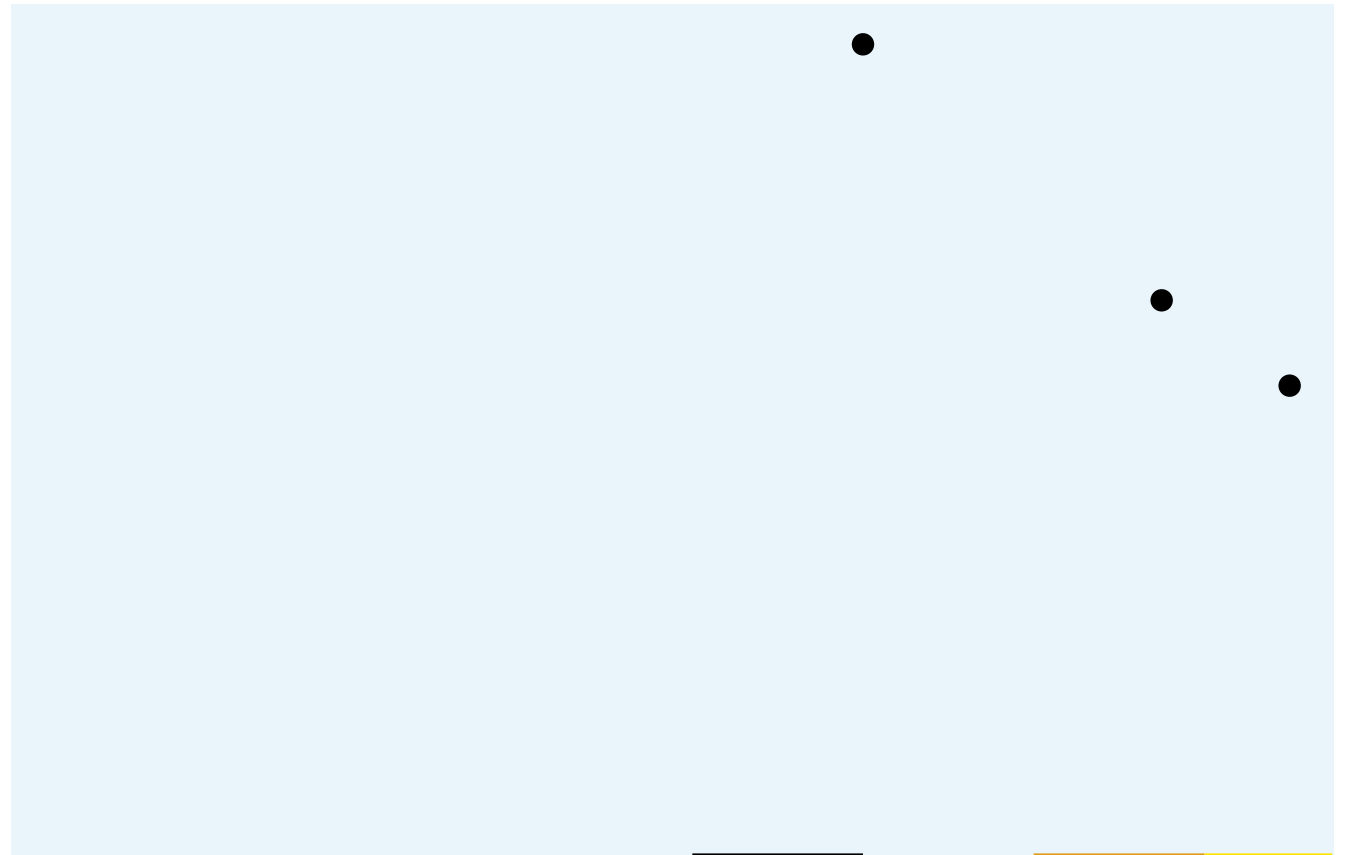
Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

After logarithmically many steps intervals of size 1.

$O(\log(U))$  many All-ints  $\text{FOP}_{\mathbb{Z}}$  queries.



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

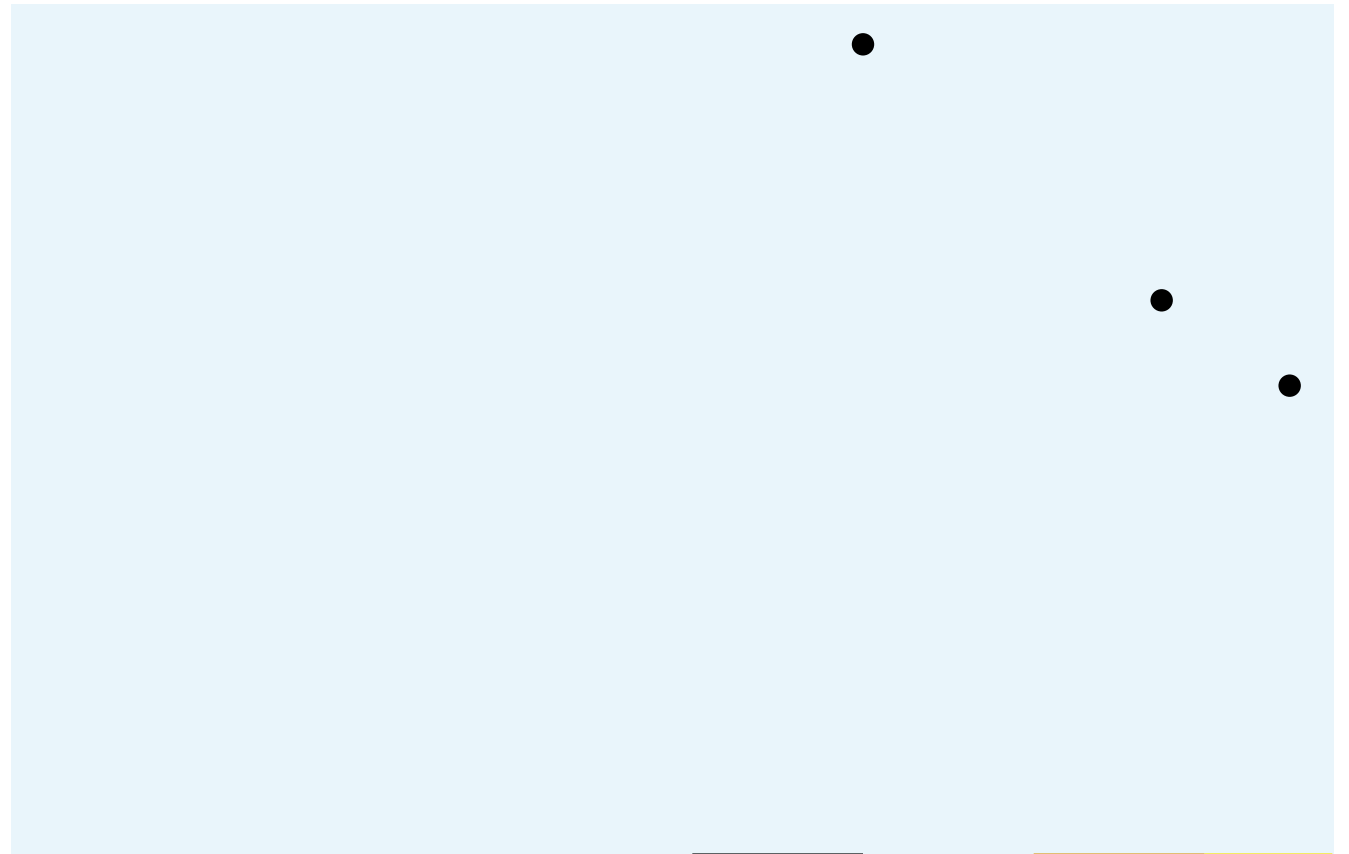
Halve the intervals and compute highest rightmost-points.

Disregard intervals which have a higher point to the right.

After logarithmically many steps intervals of size 1.

$O(\log(U))$  many All-ints  $\text{FOP}_{\mathbb{Z}}$  queries.

Every interval has atleast one Pareto Sum point.



2-D Pareto Sum  $\rightarrow$  All-ints 3-SUM( $n, n, k$ ).

Start with interval  $I = [-M, M]$ ,

Halve the intervals and compute highest rightmost-points.

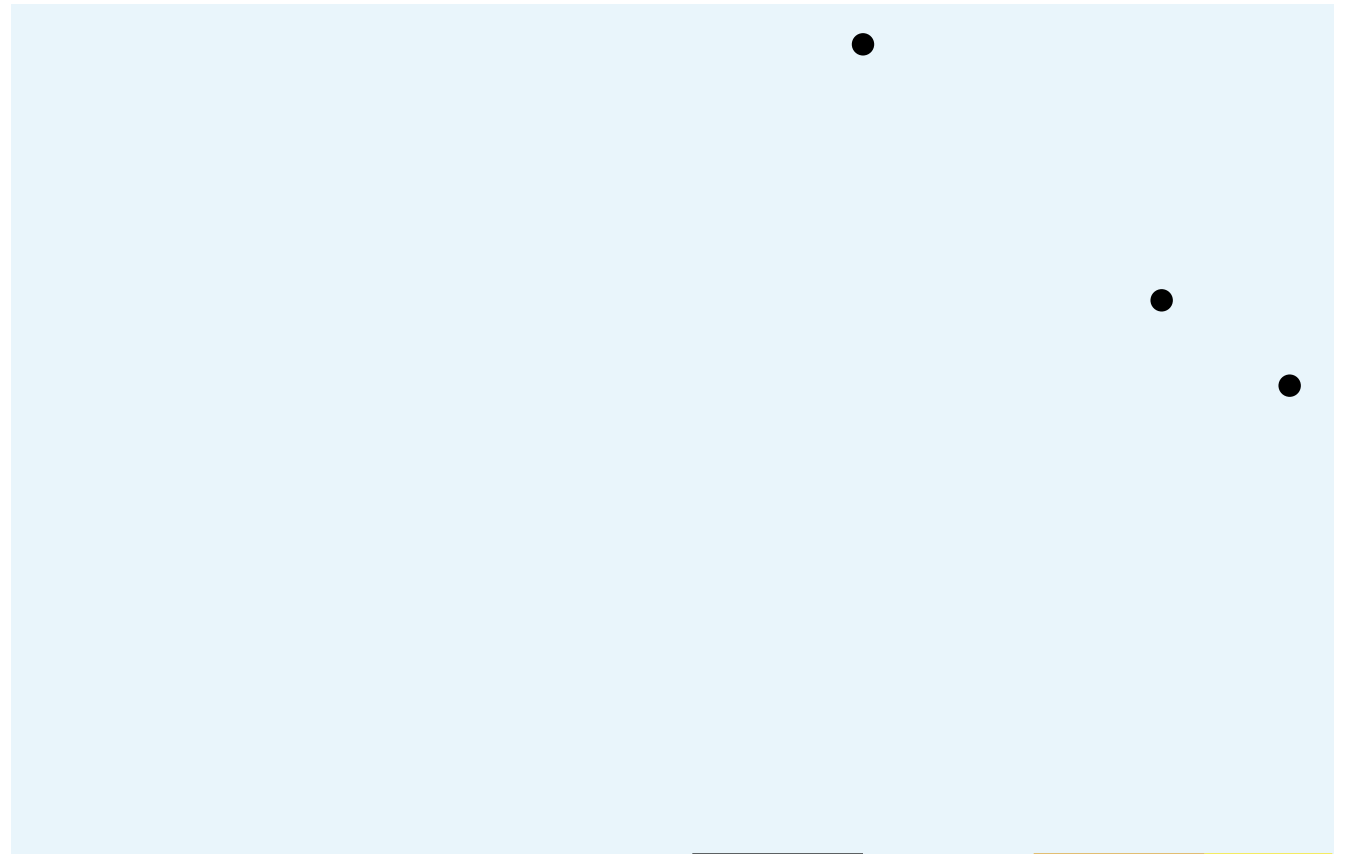
Disregard intervals which have a higher point to the right.

After logarithmically many steps intervals of size 1.

$O(\log(U))$  many All-ints FOP $_{\mathbb{Z}}$  queries.

Every interval has atleast one Pareto Sum point.

No more than  $2k$  active segments.



All-ints 3-SUM( $n, n, k$ )  $\rightarrow$  2-D Pareto Sum

Lower bound tools:

Tool 1:

All-ints 3-SUM( $n, n, k$ )  $\rightarrow$  2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

All-ints 3-SUM( $n, n, k$ )  $\rightarrow$  2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k - 1], B[0, \dots, k - 1]$  of size  $n/k$

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k - 1]$ ,  $B[0, \dots, k - 1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k - 1]$ .

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

Determine for each  $i$ :

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

Determine for each  $i$ :  $\exists j \in \{0, \dots, k-1\}, \exists a \in A[j], \exists b \in B[i-j] : C[i] = a + b$ .

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

Determine for each  $i$ :  $\exists j \in \{0, \dots, k-1\}, \exists a \in A[j], \exists b \in B[i-j] : C[i] = a + b$ .

Lemma: All-intsMixedConv 3-SUM( $n, n, k$ )  $\equiv$  All-ints 3-SUM( $n, n, k$ ).

Tool 2:

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ) :

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

Determine for each  $i$ :  $\exists j \in \{0, \dots, k-1\}, \exists a \in A[j], \exists b \in B[i-j] : C[i] = a + b$ .

Lemma: All-intsMixedConv 3-SUM( $n, n, k$ )  $\equiv$  All-ints 3-SUM( $n, n, k$ ).

Tool 2:

**Def:**  $\mathcal{Q}$ -Pareto Sum: Let  $X = A + B \subseteq \mathbb{Z}^2$ , a quadrant  $\mathcal{Q} = [q_1, \infty) \times [q_2, \infty)$ .

## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ):

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

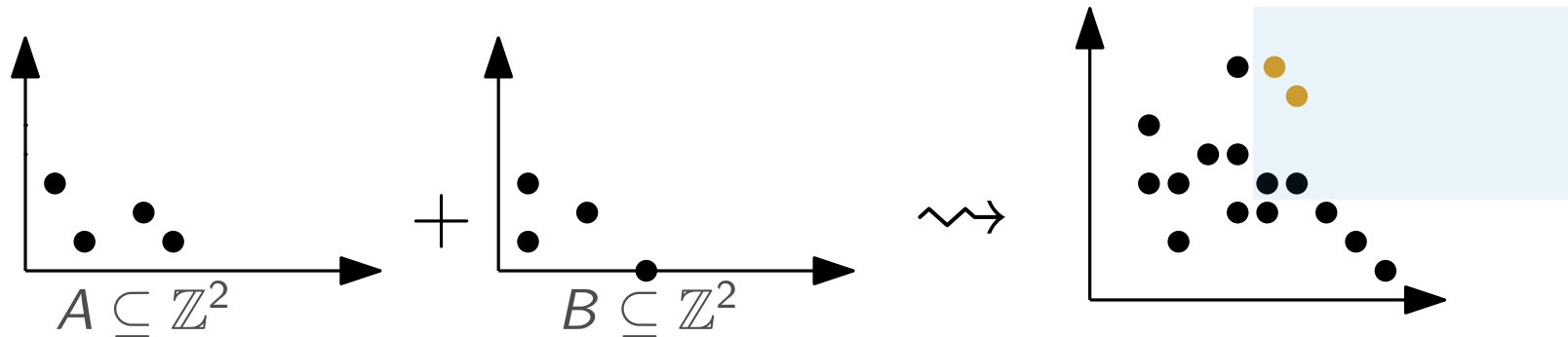
Determine for each  $i$ :  $\exists j \in \{0, \dots, k-1\}, \exists a \in A[j], \exists b \in B[i-j] : C[i] = a + b$ .

Lemma: All-intsMixedConv 3-SUM( $n, n, k$ )  $\equiv$  All-ints 3-SUM( $n, n, k$ ).

Tool 2:

**Def:**  $\mathcal{Q}$ -Pareto Sum: Let  $X = A + B \subseteq \mathbb{Z}^2$ , a quadrant  $\mathcal{Q} = [q_1, \infty) \times [q_2, \infty)$ .

Compute  $PS_{\mathcal{Q}}(A, B) = \{x \in X = A + B \mid x \in \mathcal{Q} \wedge \neg \exists x' \neq x : x' \geq x.\}$



## All-ints 3-SUM( $n, n, k$ ) $\rightarrow$ 2-D Pareto Sum

Lower bound tools:

Tool 1:

**Def:** All-intsMixedConv 3-SUM( $n, n, k$ ):

Given arrays of sets of numbers  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array of numbers  $C[0, \dots, k-1]$ .

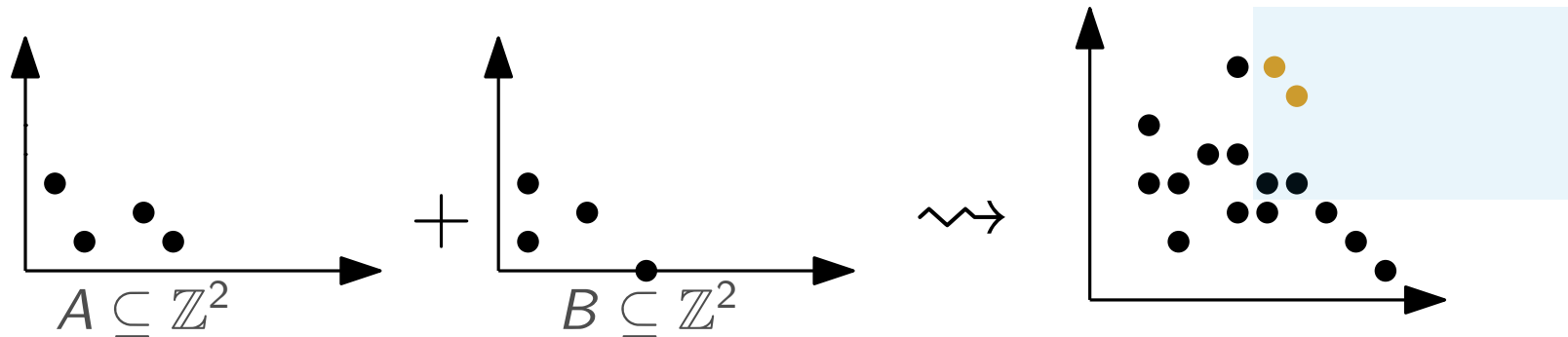
Determine for each  $i$ :  $\exists j \in \{0, \dots, k-1\}, \exists a \in A[j], \exists b \in B[i-j] : C[i] = a + b$ .

Lemma: All-intsMixedConv 3-SUM( $n, n, k$ )  $\equiv$  All-ints 3-SUM( $n, n, k$ ).

Tool 2:

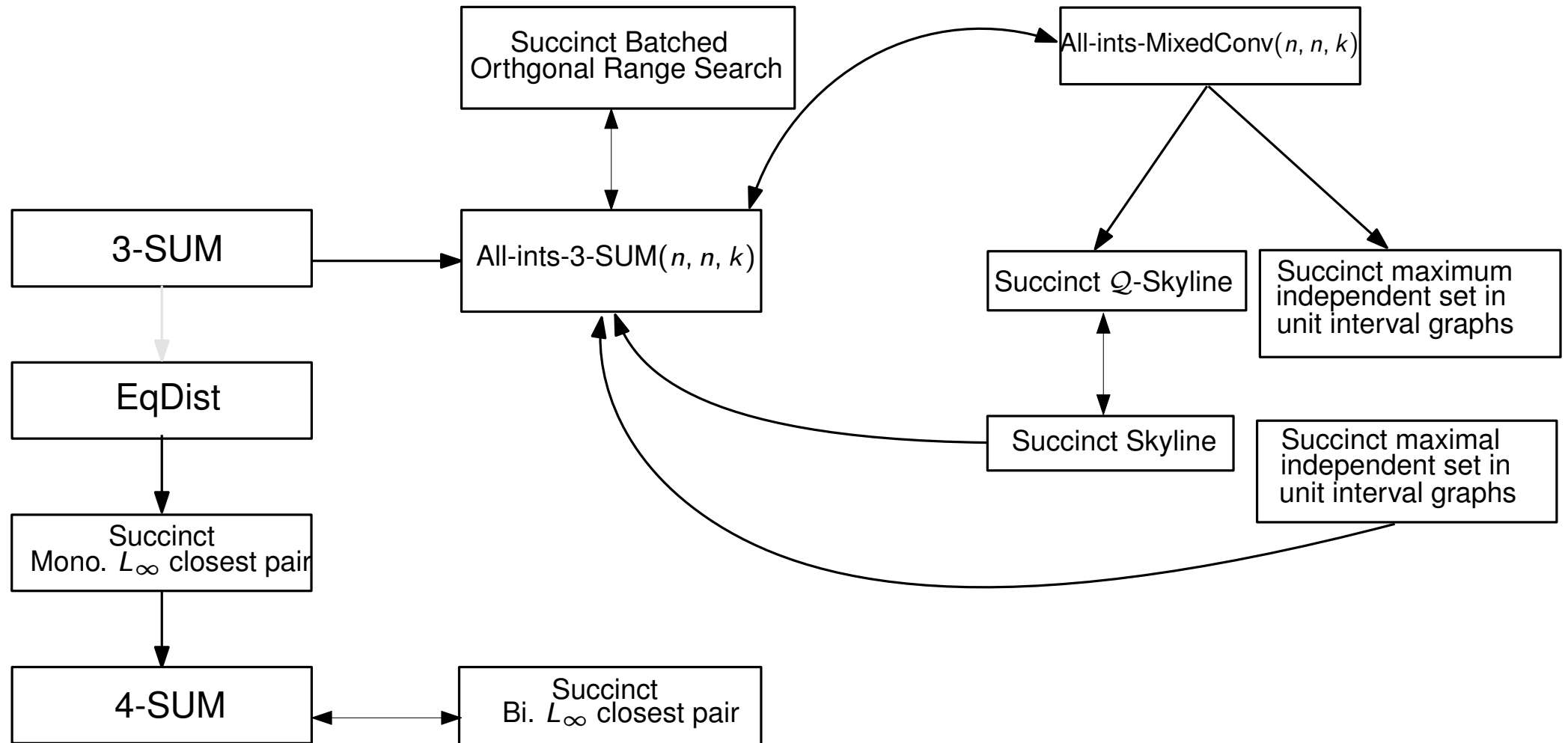
**Def:**  $\mathcal{Q}$ -Pareto Sum: Let  $X = A + B \subseteq \mathbb{Z}^2$ , a quadrant  $\mathcal{Q} = [q_1, \infty) \times [q_2, \infty)$ .

Compute  $PS_{\mathcal{Q}}(A, B) = \{x \in X = A + B \mid x \in \mathcal{Q} \wedge \neg \exists x' \neq x : x' \geq x\}$



Lemma:  $\mathcal{Q}$ -Pareto Sum  $\equiv$  Pareto Sum

# All results



Open Problems: Fast approximation of  $X$  as  $A + B$ .

Open Problems: Fast approximation of  $X$  as  $A + B$ .

Equivalence All-ints 3-SUM( $n, n, k$ ) and 3-SUM( $n, n, k$ ) ?

Open Problems: Fast approximation of  $X$  as  $A + B$ .

Equivalence All-ints 3-SUM( $n, n, k$ ) and 3-SUM( $n, n, k$ ) ?

# Thanks!

All-ints 3-SUM  $(n, n, k) \rightarrow$  2D-Pareto Sum

All-ints 3-SUM  $(n, n, k) \rightarrow$  2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k - 1]$ ,  $B[0, \dots, k - 1]$  of size  $n/k$  and array  $C[0, \dots, k - 1]$ .

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k - 1]$ ,  $B[0, \dots, k - 1]$  of size  $n/k$  and array  $C[0, \dots, k - 1]$ .

(Determine for each index  $i$ :  $\exists j \in \{0, k - 1\} : \exists a \in A[j] \quad \exists b \in B[i - j] : a + b = C[i]$ )

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i$ :  $\exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i$ :  $\exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i$ :  $\exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i: \exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

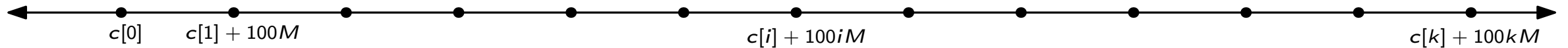
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in B[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1], B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i: \exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

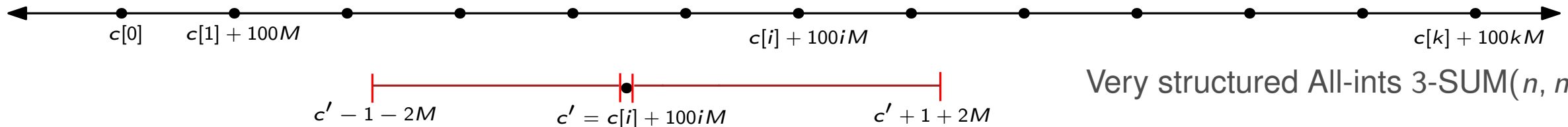
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1], B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i: \exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

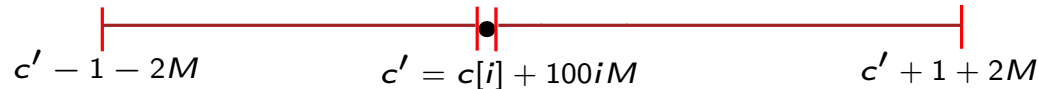
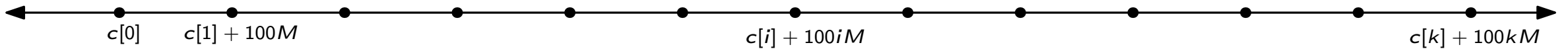
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

At cluster around  $100iM$ : no 3-SUM triple  $\iff$  all  $a' + b'$  land in red area.

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1], B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i: \exists j \in \{0, k-1\} : \exists a \in A[j] \exists b \in B[i-j] : a + b = C[i]$ )

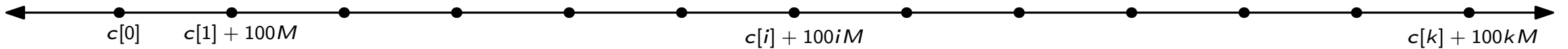
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in b[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

At cluster around  $100iM$ : no 3-SUM triple  $\iff$  all  $a' + b'$  land in red area.

$$\text{Alternatively, at this cluster: } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix} \text{ or } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix}$$

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1]$ ,  $B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i$ :  $\exists j \in \{0, k-1\} : \exists a \in A[j] \quad \exists b \in B[i-j] : a + b = C[i]$ )

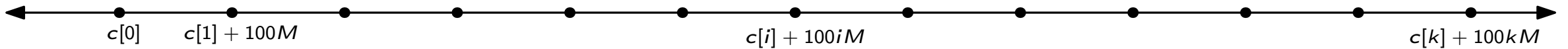
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in B[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

At cluster around  $100iM$ : no 3-SUM triple  $\iff$  all  $a' + b'$  land in red area.

$$\text{Alternatively, at this cluster: } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix} \text{ or } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix}$$

i.e a dominance condition is fulfilled.

## All-ints 3-SUM $(n, n, k) \rightarrow$ 2D-Pareto Sum

Start with instance of All-intsMixedConv 3-SUM $(n, n, k)$ .

Array of sets  $A[0, \dots, k-1], B[0, \dots, k-1]$  of size  $n/k$  and array  $C[0, \dots, k-1]$ .

(Determine for each index  $i: \exists j \in \{0, k-1\} : \exists a \in A[j] \exists b \in B[i-j] : a + b = C[i]$ )

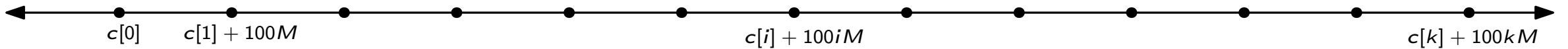
$$A' = \{a + 100iM : i \in \{0, \dots, k-1\}, a \in A[i]\}$$

$$C' = \{C[i] + 100iM : i \in \{0, \dots, k-1\}\}$$

$$B' = \{b + 100iM : i \in \{0, \dots, k-1\}, b \in B[i]\}$$

$$|A'| = |B'| = n, |C'| = k$$

Due large  $M$  Index  $i$  has solution  $\iff \exists a' \in A' b' \in B' : a' + b' = a + 100jM + b + 100(i-j)M = c[i] + 100iM = c'$



Very structured All-ints 3-SUM $(n, n, k)$ .

At cluster around  $100iM$ : no 3-SUM triple  $\iff$  all  $a' + b'$  land in red area.

$$\text{Alternatively, at this cluster: } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix} \text{ or } \begin{pmatrix} a' \\ -a' \end{pmatrix} + \begin{pmatrix} b' \\ -b' \end{pmatrix} \leq \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix}$$

i.e a dominance condition is fulfilled.

$$\text{Create the above sets: } A^* = \left\{ \begin{pmatrix} a' \\ -a' \end{pmatrix} : a' \in A' \right\}, B^* = \left\{ \begin{pmatrix} b' \\ -b' \end{pmatrix} : b' \in B' \right\}, C^* = \bigcup_{c' \in C'} \left\{ \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix}, \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix} \right\}$$

Verifying Dominance by Computing Pareto Sum.

Verifying Dominance by Computing Pareto Sum.

Define  $H := A^* \cup \left( C^* + \left\{ \begin{pmatrix} 1000kM \\ -1000kM \end{pmatrix} \right\} \right)$ ,  $G := B^* \cup \left\{ \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix} \right\}$ , abbreviate  $w = \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix}$

## Verifying Dominance by Computing Pareto Sum.

Define  $H := A^* \cup \left( C^* + \left\{ \begin{pmatrix} 1000kM \\ -1000kM \end{pmatrix} \right\} \right)$ ,  $G := B^* \cup \left\{ \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix} \right\}$ , abbreviate  $w = \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix}$

$$A^* + \{w\}$$

$$H + G \rightsquigarrow$$

$$(A^* + B^*), C^*$$

$$C^* + B^*\{+w\}$$

## Verifying Dominance by Computing Pareto Sum.

Define  $H := A^* \cup \left( C^* + \left\{ \begin{pmatrix} 1000kM \\ -1000kM \end{pmatrix} \right\} \right)$ ,  $G := B^* \cup \left\{ \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix} \right\}$ , abbreviate  $w = \begin{pmatrix} -1000kM \\ +1000kM \end{pmatrix}$

$$A^* + \{w\}$$

$$H + G \rightsquigarrow$$

$$(A^* + B^*), C^*$$

$$C^* + B^*\{+w\}$$

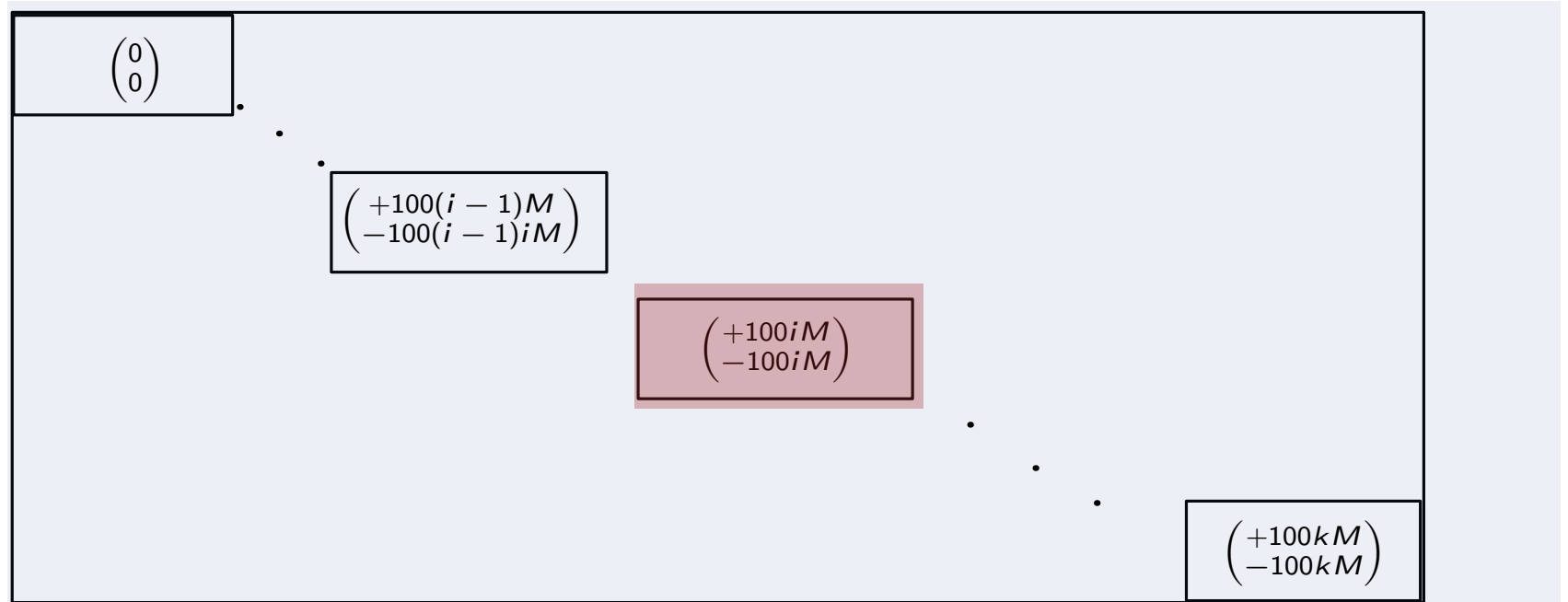
Crucial step !!!

## Verifying Dominance by Computing Pareto Sum.

$$A' = \{a + 100iM : i \in \{0, \dots, k\} a \in A[i]\} B' = \{b + 100iM : i \in \{0, \dots, k\} b \in b[i]\} C' = \{c[i] + 100iM : i \in \{0, \dots, k\}\}$$

$$A^* = \left\{ \begin{pmatrix} a' \\ -a' \end{pmatrix} : a' \in A' \right\}, B^* = \left\{ \begin{pmatrix} b' \\ -b' \end{pmatrix} : b' \in B' \right\}, C^* = \bigcup_{c' \in C'} \left\{ \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix}, \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix} \right\}$$

$$(A^* + B^*), C^* \rightsquigarrow$$



## Verifying Dominance by Computing Pareto Sum.

$$A' = \{a + 100iM : i \in \{0, \dots, k\} a \in A[i]\} B' = \{b + 100iM : i \in \{0, \dots, k\} b \in b[i]\} C' = \{c[i] + 100iM : i \in \{0, \dots, k\}\}$$

$$A^* = \left\{ \begin{pmatrix} a' \\ -a' \end{pmatrix} : a' \in A' \right\}, B^* = \left\{ \begin{pmatrix} b' \\ -b' \end{pmatrix} : b' \in B' \right\}, C^* = \bigcup_{c' \in C'} \left\{ \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix}, \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix} \right\}$$

$$\text{Cluster} \begin{pmatrix} 100iM \\ -(100iM) \end{pmatrix} \rightsquigarrow$$

$$\bullet \begin{pmatrix} c[i] + 100iM - 1 \\ -(c[i] + 98M - 1) \end{pmatrix}$$

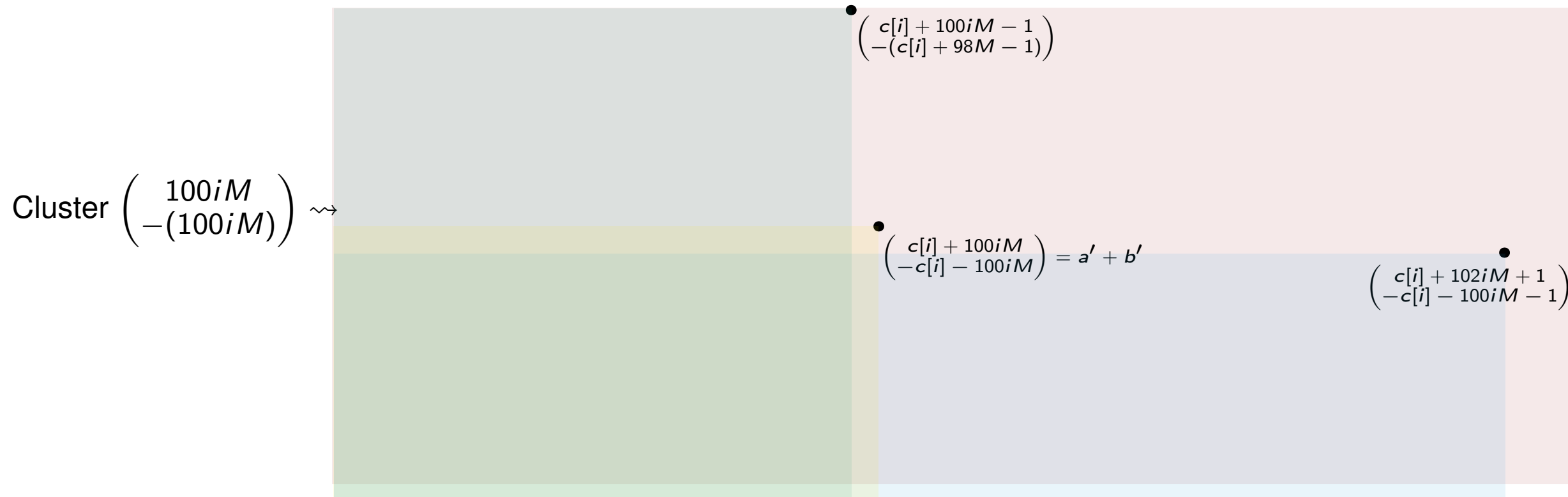
$$\bullet \begin{pmatrix} c[i] + 100iM \\ -c[i] - 100iM \end{pmatrix} = a' + b'$$

$$\bullet \begin{pmatrix} c[i] + 102iM + 1 \\ -c[i] - 100iM - 1 \end{pmatrix}$$

## Verifying Dominance by Computing Pareto Sum.

$$A' = \{a + 100iM : i \in \{0, \dots, k\} a \in A[i]\} B' = \{b + 100iM : i \in \{0, \dots, k\} b \in b[i]\} C' = \{c[i] + 100iM : i \in \{0, \dots, k\}\}$$

$$A^* = \left\{ \begin{pmatrix} a' \\ -a' \end{pmatrix} : a' \in A' \right\}, B^* = \left\{ \begin{pmatrix} b' \\ -b' \end{pmatrix} : b' \in B' \right\}, C^* = \bigcup_{c' \in C'} \left\{ \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix}, \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix} \right\}$$



## Verifying Dominance by Computing Pareto Sum.

$$A' = \{a + 100iM : i \in \{0, \dots, k\} a \in A[i]\} B' = \{b + 100iM : i \in \{0, \dots, k\} b \in b[i]\} C' = \{c[i] + 100iM : i \in \{0, \dots, k\}\}$$

$$A^* = \left\{ \begin{pmatrix} a' \\ -a' \end{pmatrix} : a' \in A' \right\}, B^* = \left\{ \begin{pmatrix} b' \\ -b' \end{pmatrix} : b' \in B' \right\}, C^* = \bigcup_{c' \in C'} \left\{ \begin{pmatrix} c' - 1 \\ -(c' - 1 - 2M) \end{pmatrix}, \begin{pmatrix} c' + 1 + 2M \\ -(c' + 1) \end{pmatrix} \right\}$$

