Efficient Enumeration of Quadratic Lattices

Victor Chen, Rohan Garg, and Benny Wang

MIT PRIMES-USA

(Under the direction of Eran Assaf)

Fall-Term PRIMES Conference
October 18-19, 2025

Outline for Talk

Background

Definition

A rank n lattice in $\mathbb{Q}^n$ is the image of a non-degenerate linear transformation of $\mathbb{Z}^n$.

Background

Definition

A rank n lattice in $\mathbb{Q}^n$ is the image of a non-degenerate linear transformation of $\mathbb{Z}^n$.

Example

The basis vectors $(3/2, 0)$ and $(0, 3)$ generate the rank 2 lattice consisting of points $(3n/2, 3m)$ for $n, m \in \mathbb{Z}$.

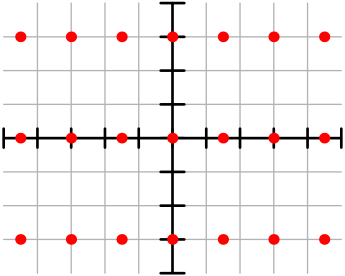
Background

Definition

A rank n lattice in $\mathbb{Q}^n$ is the image of a non-degenerate linear transformation of $\mathbb{Z}^n$.

Example

The basis vectors $(3/2, 0)$ and $(0, 3)$ generate the rank 2 lattice consisting of points $(3n/2, 3m)$ for $n, m \in \mathbb{Z}$.



Definition

A quadratic form on $\mathbb{Q}^n$ is a function $P : \mathbb{Q}^n \to \mathbb{Q}$ of the form

$$P(x_1, x_2, \dots x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

Definition

A quadratic form on $\mathbb{Q}^n$ is a function $P : \mathbb{Q}^n \to \mathbb{Q}$ of the form

$$P(x_1, x_2, \ldots x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

Given P and a choice of a basis for $\mathbb{Q}^n$, there exists a symmetric $n \times n$ matrix M such that

$$2P\left(\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}\right) = \begin{bmatrix} u_1 \cdots u_n \end{bmatrix} \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}.$$

Background

Definition

A quadratic form on $\mathbb{Q}^n$ is a function $P : \mathbb{Q}^n \to \mathbb{Q}$ of the form

$$P(x_1, x_2, \dots x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

Given P and a choice of a basis for $\mathbb{Q}^n$, there exists a symmetric $n \times n$ matrix M such that

$$2P\left(\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}\right) = \begin{bmatrix} u_1 \cdots u_n \end{bmatrix} \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}.$$

Definition

Given a quadratic form P and a lattice L, a Gram matrix of L is a matrix M of the form above, where the basis chosen is one that spans L. The determinant of L is the determinant of any one of its Gram matrices.

Background

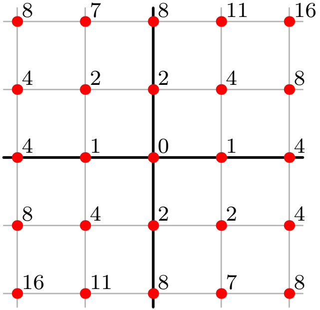
Lattice: $\mathbb{Z}^2$ Quadratic form: $P(x, y) = x^2 + xy + 2y^2$

Choice of basis: $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$

Gram matrix: $\begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}$

Determinant: 7

Example: $2P(1,2) = \begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix} = 22$



Background

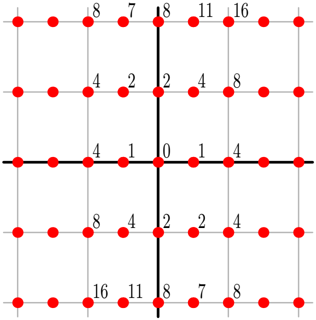
Lattice: $\frac{\mathbb{Z}}{2} \times \mathbb{Z}$ Quadratic form: $P(x, y) = 4x^2 + 2xy + 2y^2$

Choice of basis: $\begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Gram matrix: $\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$

Determinant: 7

Example: $2P(\frac{1}{2}, 2) = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 22$



Definition

Given a quadratic form P and a lattice L, we say L is integral if all the entries in the Gram matrix are integers. We say L is even-integral if all the entries in the Gram matrix are integers and the entries on the diagonal are even.

Background

Definition

Given a quadratic form P and a lattice L, we say L is integral if all the entries in the Gram matrix are integers. We say L is even-integral if all the entries in the Gram matrix are integers and the entries on the diagonal are even.

Equivalently, L is even-integral if $P(v)$ is an integer for all $v \in \mathbb{L}$.

Definition

Given a quadratic form P and a lattice L, we say L is integral if all the entries in the Gram matrix are integers. We say L is even-integral if all the entries in the Gram matrix are integers and the entries on the diagonal are even.

Equivalently, L is even-integral if $P(v)$ is an integer for all $v \in \mathbb{L}$.

Example

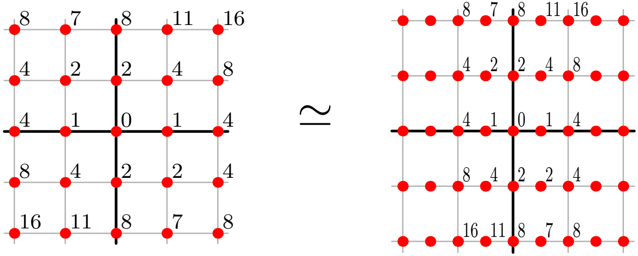
A lattice with gram matrix

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -4 \\ 0 & -4 & 3 \end{pmatrix}$$

is integral, but not even, since 3 is on the diagonal.

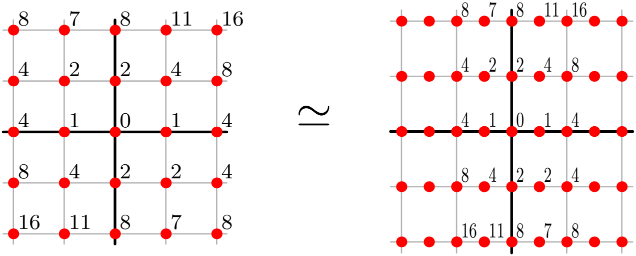
Definition

Let L_1 be a lattice with quadratic form P_1 and let L_2 be a lattice with quadratic form P_2. Lattices L_1 and L_2 are isometric if there exists a linear bijection $f : L_1 \to L_2$ such that $P_1(v_1) = P_2(f(v_1))$ for all $v_1 \in L_1$.



Definition

Let L_1 be a lattice with quadratic form P_1 and let L_2 be a lattice with quadratic form P_2. Lattices L_1 and L_2 are isometric if there exists a linear bijection $f : L_1 \to L_2$ such that $P_1(v_1) = P_2(f(v_1))$ for all $v_1 \in L_1$.

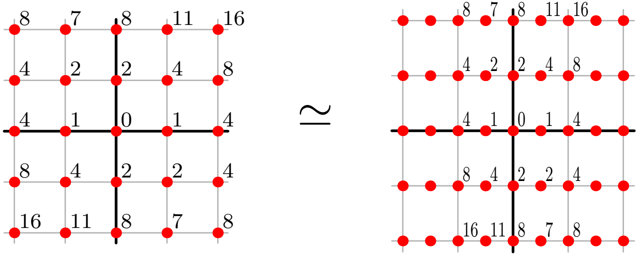


Equivalently, L_1 and L_2 are equivalent if every Gram matrix of L_1 is a Gram matrix of L_2 and vice versa.

Background

Definition

Let L_1 be a lattice with quadratic form P_1 and let L_2 be a lattice with quadratic form P_2. Lattices L_1 and L_2 are isometric if there exists a linear bijection $f : L_1 \to L_2$ such that $P_1(v_1) = P_2(f(v_1))$ for all $v_1 \in L_1$.



Equivalently, L_1 and L_2 are equivalent if every Gram matrix of L_1 is a Gram matrix of L_2 and vice versa.

Theorem (Minkowski)

For a fixed rank and determinant, there are finitely many isometry classes.

Background

Definition

Let L_1 and L_2 be integral lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 are locally isometric at p if

$$L_1 \otimes \mathbb{Z}_p \simeq L_2 \otimes \mathbb{Z}_p$$

as quadratic $\mathbb{Z}_p$-lattices.

Background

Definition

Let L_1 and L_2 be integral lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 are locally isometric at p if

$$L_1 \otimes \mathbb{Z}_p \simeq L_2 \otimes \mathbb{Z}_p$$

as quadratic $\mathbb{Z}_p$-lattices.

Let L_1 and L_2 have gram matrices G_1 and G_2, respectively. An equivalent definition is that for each positive integer k, there exists a matrix $U \in \mathsf{GL}_n(\mathbb{Z}/p^k\mathbb{Z})$ such that

$$U^\intercal G_1 U \equiv G_2 \pmod{p^k}.$$

Background

Definition

Let L_1 and L_2 be integral lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 are locally isometric at p if

$$L_1 \otimes \mathbb{Z}_p \simeq L_2 \otimes \mathbb{Z}_p$$

as quadratic $\mathbb{Z}_p$-lattices.

Let L_1 and L_2 have gram matrices G_1 and G_2, respectively. An equivalent definition is that for each positive integer k, there exists a matrix $U \in \mathsf{GL}_n(\mathbb{Z}/p^k\mathbb{Z})$ such that

$$U^\intercal G_1 U \equiv G_2 \pmod{p^k}.$$

Definition

Let L_1 and L_2 be lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 have the same signature if $L_1 \otimes \mathbb{R} \simeq L_2 \otimes \mathbb{R}$.

Definition

Let L_1 and L_2 be integral lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 are locally isometric at p if

$$L_1 \otimes \mathbb{Z}_p \simeq L_2 \otimes \mathbb{Z}_p$$

as quadratic $\mathbb{Z}_p$-lattices.

Let L_1 and L_2 have gram matrices G_1 and G_2, respectively. An equivalent definition is that for each positive integer k, there exists a matrix $U \in \mathsf{GL}_n(\mathbb{Z}/p^k\mathbb{Z})$ such that

$$U^\intercal G_1 U \equiv G_2 \pmod{p^k}.$$

Definition

Let L_1 and L_2 be lattices with quadratic forms P_1 and P_2, respectively. Lattices L_1 and L_2 have the same signature if $L_1 \otimes \mathbb{R} \simeq L_2 \otimes \mathbb{R}$.

Equivalently, there exists an invertible $n \times n$ matrix U of reals such that $U^\intercal G_1 U = G_2$.

Definition

The genus of an integral lattice L is the set of all integral lattices L' such that:

- L and L' are locally isometric at p for all primes p,
- L and L' have the same signature.

Definition

The genus of an integral lattice L is the set of all integral lattices L' such that:

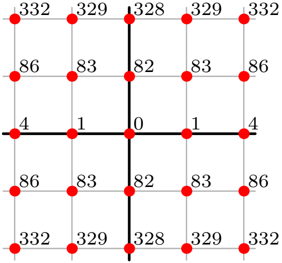
- L and L' are locally isometric at p for all primes p,
- L and L' have the same signature.

Idea: The genus groups together lattices that "look the same" locally everywhere, though they may differ in actuality.

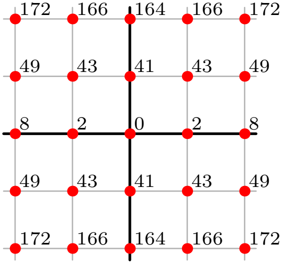
Background

$$L_1 = L_2 = \mathbb{Z}^2, \qquad P_1(x, y) = x^2 + 82y^2, \qquad P_2(x, y) = 2x^2 + 41y^2.$$

Claim: Lattices L_1 and L_2 are in the same genus.



Lattice L_1 with quadratic form P_1



Lattice L_2 with quadratic form P_2

L_1 and L_2 aren't isometric because L_1 represents 1 at $(1, 0)$ while L_2 doesn't represent 1.

Background

$$L_1 = L_2 = \mathbb{Z}^2, \qquad P_1(x,y) = x^2 + 82y^2, \qquad P_2(x,y) = 2x^2 + 41y^2.$$

Theorem

Lattices L_1 and L_2 are in the same genus iff they share a common determinant d and are locally isometric at p for $p \mid 2d$.

- $L_1 \otimes \mathbb{Z}_2 \simeq L_2 \otimes \mathbb{Z}_2$ because we can take the bijection that sends (x, y) to $(y\sqrt{41}, x/\sqrt{41})$; we have

$$P_1(x,y) = P_2\big(y\sqrt{41}, x/\sqrt{41}\big).$$

- Similarly, $L_1 \otimes \mathbb{Z}_{41} \simeq L_2 \otimes \mathbb{Z}_{41}$ because we can take the bijection that sends (x, y) to $(x/\sqrt{2}, y\sqrt{2})$; we have

$$P_1(x,y) = P_2\big(x/\sqrt{2}, y\sqrt{2}\big).$$

- Lastly, $L_1 \otimes \mathbb{R} \simeq L_2 \otimes \mathbb{R}$ because we send (x, y) to $(x/\sqrt{2}, y\sqrt{2})$.

Definition

A genus symbol is a compact way to encode the local invariants of a quadratic lattice at all primes p. It records the isometry class of $L \otimes \mathbb{Z}_p$ for each p.

Background

Definition

A genus symbol is a compact way to encode the local invariants of a quadratic lattice at all primes p. It records the isometry class of $L \otimes \mathbb{Z}_p$ for each p.

- For most p, the local structure is simple, so only finitely many primes contribute nontrivial data.
- The genus symbol thus describes the entire genus, without listing all lattices in it.

Background

Definition

A genus symbol is a compact way to encode the local invariants of a quadratic lattice at all primes p. It records the isometry class of $L \otimes \mathbb{Z}_p$ for each p.

- For most p, the local structure is simple, so only finitely many primes contribute nontrivial data.
- The genus symbol thus describes the entire genus, without listing all lattices in it.

Example

The genus symbol of $L = \mathbb{Z}^2$ and $P(x, y) = x^2 + 82y^2$ tells us that the quadratic form is positive definite and the determinant is 82. It also gives us additional info on $p = 2$ and $p = 41$. For example, at $p = 41$ the local genus symbol is

$$1^1 41^1$$

Problem

Problem: Given a rank n and determinant d, find every genus of that rank and determinant and find a representative for each genus (a lattice in the genus).

Counting Genus Symbols

To find the complexity of finding a representative for every genus of a given rank and determinant, we need to know how many genus symbols have that rank and determinant.

To find the complexity of finding a representative for every genus of a given rank and determinant, we need to know how many genus symbols have that rank and determinant.

Theorem (A.-C.-G.-W., 2025)

Let $n, d \in \mathbb{Z}$ such that $n > 0$. Let $p \mid d$ be an odd prime with $\nu_p(d) < n$. Then $L_p(n, d)$, which is the number of valid p-adic genus symbols of rank n and determinant d is

$$L_p(n, d) \sim \frac{1}{8\nu_p(d)} e^{\pi\sqrt{\nu_p(d)}}.$$

To find the complexity of finding a representative for every genus of a given rank and determinant, we need to know how many genus symbols have that rank and determinant.

Theorem (A.-C.-G.-W., 2025)

Let $n, d \in \mathbb{Z}$ such that $n > 0$. Let $p \mid d$ be an odd prime with $\nu_p(d) < n$. Then $L_p(n, d)$, which is the number of valid p-adic genus symbols of rank n and determinant d is

$$L_p(n, d) \sim \frac{1}{8\nu_p(d)} e^{\pi\sqrt{\nu_p(d)}}.$$

However, for $p = 2$, it is significantly harder to count the number of genus symbols due to the inability to diagonalize the form. Therefore, a precise formula for the number of genus symbols for $p = 2$ is difficult.

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'

Computing a Representative

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'
3. For each prime p, find a local representative $\mathcal{L}_p$ of γ at $\mathbb{Z}_p$

Computing a Representative

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'
3. For each prime p, find a local representative $\mathcal{L}_p$ of γ at $\mathbb{Z}_p$
4. At each p, locally modify L' so $L'_p = \mathcal{L}_p$

Computing a Representative

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'
3. For each prime p, find a local representative $\mathcal{L}_p$ of γ at $\mathbb{Z}_p$
4. At each p, locally modify L' so $L'_p = \mathcal{L}_p$
5. Return the resulting lattice

Computing a Representative

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'
3. For each prime p, find a local representative $\mathcal{L}_p$ of γ at $\mathbb{Z}_p$
4. At each p, locally modify L' so $L'_p = \mathcal{L}_p$
5. Return the resulting lattice

Note: This algorithm is restricted by the time complexity of finding a maximal lattice

The Brandhorst algorithm is designed to compute a lattice representative of a genus. It is the currently implemented algorithm in SageMath.

Overview of the algorithm:

1. Find a rational representative L of the genus γ
2. Take the maximal overlattice of L to be L'
3. For each prime p, find a local representative $\mathcal{L}_p$ of γ at $\mathbb{Z}_p$
4. At each p, locally modify L' so $L'_p = \mathcal{L}_p$
5. Return the resulting lattice

Note: This algorithm is restricted by the time complexity of finding a maximal lattice

Theorem (A.-C.-G.-W., 2025)

The Brandhorst Algorithm has the same time complexity as finding a maximal lattice.

Issue: The current algorithm to find maximal lattices is computationally inefficient, so the bulk of the computation time is spent doing this.

Computing a Representative

Issue: The current algorithm to find maximal lattices is computationally inefficient, so the bulk of the computation time is spent doing this.

Brandhorst Maximal Overlattice Algorithm:
At each prime p, take the overlattice of L that is maximal over $\mathbb{Z}_p$.

Computing a Representative

Issue: The current algorithm to find maximal lattices is computationally inefficient, so the bulk of the computation time is spent doing this.

Brandhorst Maximal Overlattice Algorithm:
At each prime p, take the overlattice of L that is maximal over $\mathbb{Z}_p$.

Theorem (A.-C.-G.-W., 2025)

Let ψ be the exponent of the time complexity of matrix multiplication. Then, the Brandhorst Maximal Overlattice Algorithm runs in $O(2^{4n} + n^{1+\psi} \log d)$ time.

Computing a Representative

Issue: The current algorithm to find maximal lattices is computationally inefficient, so the bulk of the computation time is spent doing this.

Brandhorst Maximal Overlattice Algorithm:
At each prime p, take the overlattice of L that is maximal over $\mathbb{Z}_p$.

Theorem (A.-C.-G.-W., 2025)

Let ψ be the exponent of the time complexity of matrix multiplication. Then, the Brandhorst Maximal Overlattice Algorithm runs in $O(2^{4n} + n^{1+\psi} \log d)$ time.

Corollary (A.-C.-G.-W., 2025)

The Brandhorst Algorithm for computing representatives runs in $O(2^{4n} + n^{1+\psi} \log d)$ time.

Original Maximal Overlattice Algorithm

The Hanke algorithm is an algorithm designed specifically for computing the maximal overlattice of a lattice and can be used along with the previous algorithm to find the representative.

Original Maximal Overlattice Algorithm

The Hanke algorithm is an algorithm designed specifically for computing the maximal overlattice of a lattice and can be used along with the previous algorithm to find the representative.

Theorem (A.-C.-G.-W., 2025)

The Hanke algorithm runs in $O(2^n n^3)$ time.

Original Maximal Overlattice Algorithm

The Hanke algorithm is an algorithm designed specifically for computing the maximal overlattice of a lattice and can be used along with the previous algorithm to find the representative.

Theorem (A.-C.-G.-W., 2025)

The Hanke algorithm runs in $O(2^n n^3)$ time.

This algorithm still runs in exponential time, but the cost of this exponential step is faster than the exponential step in the previous algorithm.

However, we are able to optimize the algorithm. The only exponential step is finding an even 2-neighbor of our current lattice. A 2-neighbor of a lattice L is a lattice L' such that $L/(L \cap L') \cong L'/(L \cap L') \cong \mathbb{F}_2$.

However, we are able to optimize the algorithm. The only exponential step is finding an even 2-neighbor of our current lattice. A 2-neighbor of a lattice L is a lattice L' such that $L/(L \cap L') \cong L'/(L \cap L') \cong \mathbb{F}_2$.

- We were able to prove that this is equivalent to finding a primitive vector v in a certain sublattice such that $Q(v) \equiv 0 \pmod 8$.

However, we are able to optimize the algorithm. The only exponential step is finding an even 2-neighbor of our current lattice. A 2-neighbor of a lattice L is a lattice L' such that $L/(L \cap L') \cong L'/(L \cap L') \cong \mathbb{F}_2$.

- We were able to prove that this is equivalent to finding a primitive vector v in a certain sublattice such that $Q(v) \equiv 0 \pmod 8$.
- It is known in literature that such a vector always exists for quadratic forms of at least 5 variables. Therefore, we can set any variable after the 5th to zero, and brute force the remaining up to 8^5 possibilities.

However, we are able to optimize the algorithm. The only exponential step is finding an even 2-neighbor of our current lattice. A 2-neighbor of a lattice L is a lattice L' such that $L/(L \cap L') \cong L'/(L \cap L') \cong \mathbb{F}_2$.

- We were able to prove that this is equivalent to finding a primitive vector v in a certain sublattice such that $Q(v) \equiv 0 \pmod 8$.
- It is known in literature that such a vector always exists for quadratic forms of at least 5 variables. Therefore, we can set any variable after the 5th to zero, and brute force the remaining up to 8^5 possibilities.
- This allows the algorithm to be optimized to $O(n^3 \log d)$, which is the fastest known algorithm to compute maximal overlattices.

- Using the Hanke algorithm, we can achieve a complexity of $O(n^3 \log d)$; faster than another existing $O(n^8 \log d)$ algorithm for computing representatives (due to Dubey and Holenstein).

Conclusion and Future Steps

- Using the Hanke algorithm, we can achieve a complexity of $O(n^3 \log d)$; faster than another existing $O(n^8 \log d)$ algorithm for computing representatives (due to Dubey and Holenstein).
- We hope to implement and test these algorithms to classify more lattices on the LMFDB, which is a database of number theoretic objects.

Conclusion and Future Steps

- Using the Hanke algorithm, we can achieve a complexity of $O(n^3 \log d)$; faster than another existing $O(n^8 \log d)$ algorithm for computing representatives (due to Dubey and Holenstein).
- We hope to implement and test these algorithms to classify more lattices on the LMFDB, which is a database of number theoretic objects.
- We can use the local version of the optimized Hanke algorithm in the Sage algorithm to speed it up.

Acknowledgements

We would like to give our heartfelt gratitude to:

- Our amazing mentor Eran Assaf for his support and insights during the entire research process
- The PRIMES-USA research program for giving us this amazing opportunity to learn and conduct research
- Our friends and family for their support throughout this process.