UNIVERSITÉIT LËTZEBUERG

AKS Algorithm: Finding shortest nonzero vectors.

Hilder Vítor Lima Pereira

Introduction to lattices and their applications in Computer Science/Cryptography.
Doctoral Programme in Computer Science and Computer Engineering
University of Luxemburg.

12th June, 2019

Table of contents

Introduction

SVP:
Given a lattice $\mathcal{L}$, the shortest nonzero vector problem (SVP) is the problem of finding a point $\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}$ such that the Euclidean norm is minimized, that is, $\forall \mathbf{u} \in \mathcal{L} \setminus \{\mathbf{0}\}, \|\mathbf{v}\|_2 \leq \|\mathbf{u}\|_2$.

Introduction

Solving SVP:

Algorithms for approximate versions of SVP.

- LLL solves approx. SVP with exponential approximation factor in polynomial-time.
- BKZ solves approx. SVP with polynomial approximation factor roughly in exponential time.

What about AKS?

- It solves the exact version of SVP.
- Exponential time and memory.
- It is a randomized algorithm.
- It outputs the correct answer with overwhelming probability.

Introduction

Solving SVP:

Algorithms for approximate versions of SVP.

- LLL solves approx. SVP with exponential approximation factor in polynomial-time.
- BKZ solves approx. SVP with polynomial approximation factor roughly in exponential time.

What about AKS?

- It solves the exact version of SVP.
- Exponential time and memory.
- It is a randomized algorithm.
- It outputs the correct answer with overwhelming probability.

Overview

1. Sample $2^{O(n)}$ random lattice points inside a ball $\mathcal{B}(0, R)$.
2. Find "centers points" among them, i.e., points close to several other points.
3. Get new lattice points by computing the difference between the points and their centers. (Note that the new points lie in $\mathcal{B}(0, R')$ with $R' < R$).
4. Repeat it with those new points unless they are already shorter than some bound.
5. Output the shortest vector among the remaining ones.

1. How does one sample random lattice points within a ball?

We actually sample them indirectly:

- Sample a *real* vector $\mathbf{x} \in \mathcal{B}(0, R) \cap \mathbb{R}^n$.
- Compute $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ (the fundamental region) such that $\mathbf{y} - \mathbf{x} \in \mathcal{L}$.
- Define $\mathbf{z} = \mathbf{y} - \mathbf{x}$ as the random lattice point.

We compute $\mathbf{y}$ as $\mathbf{x} \mod \mathbf{B} := \mathbf{x} - \mathbf{B}\lfloor \mathbf{B}^{-1}\mathbf{x} \rfloor$, which is a "reduction modulo the basis $\mathbf{B}$".

Notice that $\mathbf{z}$ is the corner of the translated fundamental region that covers $\mathbf{x}$.

2. How does one find center points?

We use a procedure known as Sieve:

Algorithm 1: SIEVE

Input: A positive $R \in \mathbb{R}$ and $X := \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \mathcal{B}(0, R)$

Output: A set C of pairs (i, j) such that $\mathbf{x}_j$ is the center of $\mathbf{x}_i$.

```
1 C := ∅
2 for i = 1 until N do
3     if ∃(i', j) ∈ C such that ‖x_i − x_j‖_2 ≤ R/2 then
4         C = C ∪ {(i, j)} ;    ▷ x_j becomes the center of x_i
5     else
6         C = C ∪ {(i, i)} ;    ▷ x_i becomes its own center
```

Notice that the "centers" are defined by the second entry of the pairs (i, j) in C. For each $\mathbf{x}_i$, its center is $\mathbf{x}_j$.

2. How does one find center points?

Lemma

Let $R \in \mathbb{R}_{>0}$. For any set of points $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \mathcal{B}(0, R)$, let C be the set returned by SIEVE. *Then*

(i) *C defines at most 5^n centers and*

(ii) $\forall (i, j) \in C$, $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq R/2$

Moreover, (iii) SIEVE *runs in polynomial time in the input size.*

2. How does one find center points?

Proof.
Define balls of radius $\frac{R}{4}$ around each center. Notice that they are disjoint, because the distance between two centers is bigger than $\frac{R}{2}$. Furthermore, their union is contained in $\mathcal{B}(0, \frac{5R}{4})$.
Therefore, the number of balls (which equals the number of centers) is at most

$$\frac{\text{vol}(\mathcal{B}(0, \frac{5R}{4}))}{\text{vol}(\mathcal{B}(0, \frac{R}{4}))} = 5^n.$$

Propositions (ii) and (iii) are trivial. □

Remember that $\text{vol}(\mathcal{B}(0, R)) = \frac{\pi^{n/2} R^n}{\Gamma(n/2+1)}$.

AKS for $\lambda_1 \in [2, 3)$

We are almost ready to see the algorithm AKS. Before defining it to the general case, let's assume we are working over lattices for which

$$\lambda_1 \in [2, 3).$$

We will see in the end how to remove this restriction.

Algorithm 2: AKS*, for $\lambda_1 \in [2, 3)$

Input: A basis $\mathbf{B}$ of an n-dimensional lattice whose $\lambda_1 \in [2, 3)$

Output: A shortest nonzero vector of $\mathcal{L}(\mathbf{B})$

$R := n \cdot \max \|\mathbf{b}_j\|_2 + 2$

$N := 2^{8n} \log R$

Sample $X := \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ unif. in $\mathcal{B}(0, 2) \cap \mathbb{R}^n$

$Y := \{\mathbf{y}_i := \mathbf{x}_i \mod \mathbf{B} : \mathbf{x}_i \in X\} \subset \mathcal{P}(\mathbf{B})$

while $R > 6$ **do**

  $C := \mathrm{SIEVE}(Y)$

  for *each center* $\mathbf{y}_j$ *defined by* C **do**

    $Y = Y \setminus \{\mathbf{y}_j\}$; $X = X \setminus \{\mathbf{x}_j\}$

  for *each* $\mathbf{y}_j$ *in* Y **do**

    Let $\mathbf{y}_c$ be the center of $\mathbf{y}_j$

    $\mathbf{y}_j = \mathbf{y}_j - (\mathbf{y}_c - \mathbf{x}_c)$

  $R = R/2 + 2$

Return the shortest $(\mathbf{y}_i - \mathbf{x}_i) - (\mathbf{y}_j - \mathbf{x}_j)$ (among Y and X)

AKS for $\lambda_1 \in [2, 3)$

Lemma

The number of iterations of the while loop in AKS^* *is at most* $2 \log R_0$*, where* R_0 *is the first value assigned to* R*.*

Proof.

Let R_k be the value of R on the beginning of the k-th iteration.

Then, we have $R_1 = R_0$, $R_2 = R_0/2 + 2$, $R_3 = R_0/2^2 + 1 + 2$, $R_4 = R_0/2^3 + 1/2 + 1 + 2$, etc.

In general, $R_k = R_0/2^{k-1} + 2 + \sum_{i=0}^{k-3} 1/2^i$.

For $k = \lceil \log R_0 \rceil + 1$ we have $R_k \leq 1 + 2 + 2 = 5 < 6$ and then the while loop is aborted.

Therefore, the number of iterations is at most

$$\lceil \log R_0 \rceil + 1 \leq 2 \log R_0.$$

□

AKS for $\lambda_1 \in [2, 3)$

Lemma

The number of iterations of the while loop in AKS^* *is at most* $2 \log R_0$*, where* R_0 *is the first value assigned to* R*.*

Proof.

Let R_k be the value of R on the beginning of the k-th iteration.
Then, we have $R_1 = R_0$, $R_2 = R_0/2 + 2$, $R_3 = R_0/2^2 + 1 + 2$, $R_4 = R_0/2^3 + 1/2 + 1 + 2$, etc.
In general, $R_k = R_0/2^{k-1} + 2 + \sum_{i=0}^{k-3} 1/2^i$.
For $k = \lceil \log R_0 \rceil + 1$ we have $R_k \leq 1 + 2 + 2 = 5 < 6$ and then the while loop is aborted.
Therefore, the number of iterations is at most

$$\lceil \log R_0 \rceil + 1 \leq 2 \log R_0.$$

□

AKS for $\lambda_1 \in [2, 3)$

Lemma

AKS^* *runs in time* $2^{O(n)}$ *times some polynomial in the input size.*

Proof.

Let $S = \log(R_0)$ be the input length.

The initialization step, before the while loop, already costs $2^{O(n)}\,\mathrm{poly}(S)$. The final step, after the loop, is clearly cheaper than this (since we have removed several points from Y and X).

The procedure SIEVE runs in polynomial time in the number of points it receives, that is, $O(\mathrm{poly}(2^{8n} \log R)) = 2^{O(n)}\,\mathrm{poly}(S)$.

By the last lemma, SIEVE is executed at most $2S$ times, therefore, the cost of the loop is also $2^{O(n)}\,\mathrm{poly}(S)$. □

AKS for $\lambda_1 \in [2, 3)$

Lemma
AKS^* *runs in time* $2^{O(n)}$ *times some polynomial in the input size.*

Proof.
Let $S = \log(R_0)$ be the input length.

The initialization step, before the while loop, already costs $2^{O(n)}\,\mathrm{poly}(S)$. The final step, after the loop, is clearly cheaper than this (since we have removed several points from Y and X).

The procedure SIEVE runs in polynomial time in the number of points it receives, that is, $O(\mathrm{poly}(2^{8n} \log R)) = 2^{O(n)}\,\mathrm{poly}(S)$.

By the last lemma, SIEVE is executed at most $2S$ times, therefore, the cost of the loop is also $2^{O(n)}\,\mathrm{poly}(S)$. □

AKS for $\lambda_1 \in [2, 3)$

Lemma

Let $Z := \{(\mathbf{x}_i, \mathbf{y}_i) : \mathbf{x}_i \in X \wedge \mathbf{y}_i := \mathbf{x}_i \mod \mathbf{B}\}$. *At the end of* AKS^*, *the set* Z *has an exponential number of pairs and each pair gives us a lattice vector with norm bounded by* 8.

Remark

Several pairs $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$ define the same lattice point...

AKS for $\lambda_1 \in [2, 3)$

Proof.

By the definition of $\mathbf{y}_i$, at the beginning of the algorithm, we have $\mathbf{y}_i \in \mathcal{P}(\mathbf{B})$, thus, $\|\mathbf{y}_i\|_2 \leq \sum \|\mathbf{b}_j\|_2 \leq R_0$. And at each iteration k, $\mathbf{y}_i$ is updated to $\mathbf{y}_i - (\mathbf{y}_c - \mathbf{x}_c)$, therefore, its norm becomes

$$\|\mathbf{y}_i - (\mathbf{y}_c - \mathbf{x}_c)\|_2 \leq \|\mathbf{y}_i - \mathbf{y}_c\|_2 + \|\mathbf{x}_c\|_2 \leq R_k/2 + 2.$$

Thus, at the end of the last iteration, we have $\|\mathbf{y}_i\|_2 \leq 6$.
Therefore, we have $\mathbf{y}_i - \mathbf{x}_i \in \mathcal{L}$ and

$$\|\mathbf{y}_i - \mathbf{x}_i\|_2 \leq 6 + 2 = 8.$$

Now notice that at the each iteration, at most 5^n points are removed from X and Y, therefore

$$|Z| \geq N - 5^n \cdot 2 \log R_0 = (2^{8n} - 2 \cdot 5^n) \log R_0 \geq 2^{7n} \log R_0.$$

AKS for $\lambda_1 \in [2, 3)$

Lets breath a bit...

- We have proved that AKS^* finds an exponentially large set of pairs which define (possibly repeated) very short lattice points.
- Remember that we are supposing $\lambda_1 \in [2, 3)$ and all those lattice points have norm smaller than 8. Therefore, they are already a very good approximation to a shortest nonzero vector.
- Intuitively, it is very likely that a shortest nonzero vector is indeed among them.

AKS for $\lambda_1 \in [2, 3)$

How can we prove that AKS^* really finds a shortest nonzero vector with high probability?

Intuition:

- Notice that if we sample the points $\mathbf{x}_i$ differently, but keeping the same distribution, the algorithm's output must be the same.
- For analysis purposes, sample $\mathbf{x}_i$ such that many of them are equal to a fixed $\mathbf{w}$ and many have the form $\mathbf{w} \pm \mathbf{v}$, where $\mathbf{v}$ is a shortest nonzero vector.
- Then, at the end of the algorithm, with high probability, we will have $\mathbf{x}_i$ and $\mathbf{x}_j$ whose difference equals $\pm\mathbf{v}$.

AKS for $\lambda_1 \in [2, 3)$

A lemma that we will need later...

Lemma

Let $\mathcal{L}$ be a lattice such that $\lambda_1 \in [2, 3)$. Then there are at most 9^n lattice points inside $\mathcal{B}(0, 8)$.

Proof.

Let m be the number of points in $\mathcal{L} \cap \mathcal{B}(0, 8)$.
Because $\lambda_1 \geq 2$, we can consider m disjoint balls of radius 1 centered in each lattice point inside $\mathcal{B}(0, 8)$.
Then, the union of all these balls is contained in $\mathcal{B}(0, 9)$.
Thus, we have $m \cdot \text{vol}(\mathcal{B}(0, 1)) \leq \text{vol}(\mathcal{B}(0, 9))$. Therefore,

$$m \leq \frac{\text{vol}(\mathcal{B}(0, 9))}{\text{vol}(\mathcal{B}(0, 1))} = 9^n.$$

□

AKS for $\lambda_1 \in [2, 3)$

A lemma that we will need later...

Lemma

Let $\mathcal{L}$ be a lattice such that $\lambda_1 \in [2, 3)$. Then there are at most 9^n lattice points inside $\mathcal{B}(0, 8)$.

Proof.

Let m be the number of points in $\mathcal{L} \cap \mathcal{B}(0, 8)$.
Because $\lambda_1 \geq 2$, we can consider m disjoint balls of radius 1 centered in each lattice point inside $\mathcal{B}(0, 8)$.
Then, the union of all these balls is contained in $\mathcal{B}(0, 9)$.
Thus, we have $m \cdot \mathrm{vol}(\mathcal{B}(0, 1)) \leq \mathrm{vol}(\mathcal{B}(0, 9))$. Therefore,

$$m \leq \frac{\mathrm{vol}(\mathcal{B}(0, 9))}{\mathrm{vol}(\mathcal{B}(0, 1))} = 9^n.$$

□

AKS for $\lambda_1 \in [2, 3)$

Theorem

If $\lambda_1 \in [2, 3)$, then AKS^* returns a shortest nonzero vector with probability exponentially close to 1, i.e., bigger than $1 - 2^{-n}$.

AKS for $\lambda_1 \in [2, 3)$

Sketch of the proof

Let $\mathbf{v}$ be a shortest nonzero vector, thus $\|\mathbf{v}\|_2 \in [2, 3)$.
Define $C_1 := \mathcal{B}(0, 2) \cap \mathcal{B}(-\mathbf{v}, 2)$ and $C_2 := \mathcal{B}(0, 2) \cap \mathcal{B}(\mathbf{v}, 2)$.

(a) Example for $\|\mathbf{v}\|_2 = 2$.

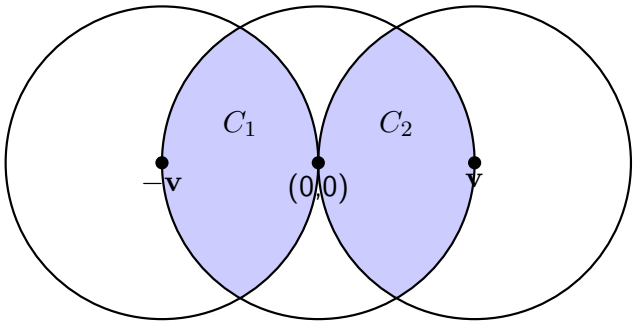
(b) Example for $\|\mathbf{v}\|_2 = 3$.

UNIVERSITÉIT LËTZEBUERG

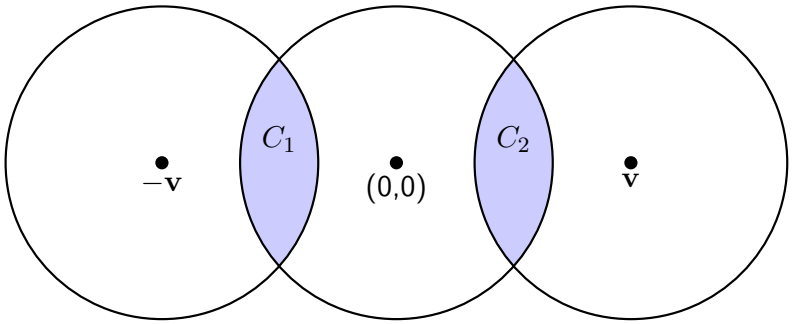
AKS for $\lambda_1 \in [2, 3)$

Sketch of the proof

Let $\mathbf{v}$ be a shortest nonzero vector, thus $\|\mathbf{v}\|_2 \in [2, 3)$.
Define $C_1 := \mathcal{B}(0, 2) \cap \mathcal{B}(-\mathbf{v}, 2)$ and $C_2 := \mathcal{B}(0, 2) \cap \mathcal{B}(\mathbf{v}, 2)$.



(a) Example for $\|\mathbf{v}\|_2 = 2$.



(b) Example for $\|\mathbf{v}\|_2 = 3$.

AKS for $\lambda_1 \in [2, 3)$

Sketch of the proof

Define the function $\tau : X \to X$ that flips vectors from C_1 to C_2 and vice-versa:

$$\tau(\mathbf{x}_i) = \begin{cases} \mathbf{x}_i + \mathbf{v}, & \text{if } \mathbf{x}_i \in C_1 \\ \mathbf{x}_i - \mathbf{v}, & \text{if } \mathbf{x}_i \in C_2 \\ \mathbf{x}_i, & \text{otherwise} \end{cases}$$

Notice that τ is a bijection, therefore, X and $\tau(X)$ follow the same distribution.
Moreover, $\mathbf{x}_i = \mathbf{x}_i \pm \mathbf{v} \mod \mathbf{B}$, therefore

$$\mathbf{y}_i = \mathbf{x}_i \mod \mathbf{B} \Leftrightarrow \mathbf{y}_i = \tau(\mathbf{x}_i) \mod \mathbf{B}.$$

Therefore, AKS^* has the same output given X or $\tau(X)$.

AKS for $\lambda_1 \in [2, 3)$

Sketch of the proof

Hence, apply τ to all $\mathbf{x}_i$.
As proved earlier, we have more than 2^{7n} vectors $\mathbf{x}_i$ at the end of the algorithm. For each of them, we have a lattice point $\mathbf{z}_i := \mathbf{y}_i - \mathbf{x}_i$ and $\mathbf{z}_i \in \mathcal{B}(0, 8)$. But there are at most 9^n lattice points inside $\mathcal{B}(0, 8)$.
Therefore, there exists $\mathbf{w} \in \mathcal{L}$ yielded by at least $2^{7n}/9^n \geq 2^{3.8n}$ pairs of $\mathbf{x}_i$ and $\mathbf{y}_i$.
Then, with high probability*, for such $\mathbf{w}$, at least one $\mathbf{x}_i$ is in $C_1 \cup C_2$ and at least one $\mathbf{x}_j$ doesn't belong to $C_1 \cup C_2$.
But then, $\mathbf{x}_i$ is flipped by τ and $\mathbf{x}_j$ isn't, and therefore, AKS^* returns

$$(\mathbf{y}_i - \mathbf{x}_i \pm \mathbf{v}) - (\mathbf{y}_j - \mathbf{x}_j) = \mathbf{w} \pm \mathbf{v} - \mathbf{w} = \pm\mathbf{v}.$$

Sketch of the proof

* It is not hard to see that for such $\mathbf{w}$, at least one $\mathbf{x}_i$ is in $C_1 \cup C_2$ with probability bigger than $1 - 2^{-n}$.
Notice that $\mathcal{B}(\mathbf{v}/2, 0.5)$ fits in C_1 and also in C_2. Hence, $\text{vol}(C_1) \geq \text{vol}(\mathcal{B}(\mathbf{v}/2, 0.5))$. Therefore,

$$\frac{\text{vol}(C_1)}{\text{vol}(\mathcal{B}(0,2))} \geq \frac{\text{vol}(\mathcal{B}(\mathbf{v}/2, 0.5))}{\text{vol}(\mathcal{B}(0,2))} = \frac{0.5^n}{2^n} = 2^{-2n}.$$

Thus, we have $\forall \mathbf{x}_i \leftarrow \mathcal{U}\left(\mathcal{B}(0,2)\right), Pr[\mathbf{x}_i \in C_1 \cup C_2] \geq 2 \cdot 2^{-2n}$.
Then, considering the (more than) $2^{3.8n}$ vectors $\mathbf{x}_i$ associated to $\mathbf{w}$ and using Chebyshev's inequality, we have

$$Pr[\exists i : \mathbf{x}_i \in C_1 \cup C_2] \geq 1 - \frac{2^{2n}}{2^{3.8n}} = 1 - \frac{1}{2^{1.8n}} \geq 1 - 2^{-n}.$$

AKS for $\lambda_1 \in [2, 3)$

Sketch of the proof.

* Using a similar argument we can also see that for such $\mathbf{w}$, at least one $\mathbf{x}_i$ is outside $C_1 \cup C_2$ with probability bigger than $1 - 2^{-n}$. $\square$

Removing the restriction $\lambda_1 \in [2, 3)$

"Okay, nice. But our lattices don't have such a small λ_1!"

- Impatient audience.

Removing the restriction $\lambda_1 \in [2, 3)$

Using LLL, we can find an estimate e for λ_1 such that

$$\lambda_1 \leq e \leq 2^n \lambda_1.$$

Manipulating that inequality, we get:

$$1 \leq \frac{e}{\lambda_1} \leq 2^n \Leftrightarrow \frac{1}{2^n} \leq \frac{\lambda_1}{e} \leq 1 \Leftrightarrow \frac{e}{2^n} \leq \lambda_1 \leq e.$$

Therefore, we know that the length of a shortest nonzero vector of $\mathcal{L}$ is in the interval $\left[\frac{e}{2^n}, e\right]$.

Removing the restriction $\lambda_1 \in [2, 3)$

- Consider the lattice $\mathcal{L}' := \frac{2^{n+1}}{e}\mathcal{L}$.
- Then $2 \leq \lambda_1(\mathcal{L}') \leq 2^{n+1}$.
- If $\mathbf{v}$ is a shortest nonzero vector of $\mathcal{L}'$, then $\frac{e}{2^{n+1}}\mathbf{v}$ is a shortest nonzero vector of $\mathcal{L}$.
- Therefore, it is sufficient to solve the SVP on $\mathcal{L}'$.

Removing the restriction $\lambda_1 \in [2, 3)$

How to solve SVP on $\mathcal{L}'$ knowing that $\lambda_1(\mathcal{L}') \in [2, 2^{n+1}]$?

- Write

$$[2, 2^{n+1}] \subset [2, 3) \cup [2x, 3x) \cup [2x^2, 3x^2) \cup ... \cup [2x^\ell, 3x^\ell).$$

- By choosing $x = 3/2$, we have $3x^k = 2x^{k+1}$.
- We need an ℓ such that $3x^\ell > 2^{n+1} \Leftrightarrow 3(3/2)^\ell > 2^{n+1} \Leftrightarrow 3^{\ell+1} > 2^{\ell+n+1}$, an it is sufficient to take $\ell = 2n$.
- Then, for some $k \in \{0, ..., \ell\}$, $\lambda_1(\mathcal{L}') \in [2x^k, 3x^k)$.
- Therefore, for such k, $\lambda_1(x^{-k}\mathcal{L}') = x^{-k}\lambda_1(\mathcal{L}') \in [2, 3)$.

The AKS algorithm (for any λ_1)

Algorithm 2: AKS

Input: A basis $\mathbf{B}$ of an n-dimensional lattice $\mathcal{L}$

Output: A shortest nonzero vector of $\mathcal{L}(\mathbf{B})$

Run LLL to get an estimate e for λ_1

Define $\mathcal{L}' := \frac{2^{n+1}}{e}\mathcal{L}$; ▷ `Just multiply` $\mathbf{B}$ `by` $\frac{2^{n+1}}{e}$

for $k = 0$ *until* $2n$ **do**

  Define $\mathcal{L}_k := x^{-k}\mathcal{L}'$

  $\mathbf{v}_k = \mathrm{AKS}^*(\mathcal{L}_k)$

Let $\mathbf{v}$ be a shortest nonzero vector among all $\mathbf{v}_k$

Let $\mathbf{u} = x^k\mathbf{v}$ be a shortest nonzero vector of $\mathcal{L}'$

Return $\frac{e}{2^{n+1}}\mathbf{u}$

References


M. Ajtai, R. Kumar, D. Sivakumar (2001)
A sieve algorithm for the shortest vector problem.
Procedings of the thirty-third annual ACM symposium on Theory of Computing. Pages 601 - 610.


Oded Regev (2004)
$2^{O(n)}$-time algorithm for SVP.
Lecture notes: Lattices in Computer Science. Tel Aviv University.