Computational problems on lattices: [problems parameterized by lattice dimension $n$] (can solve exactly using Gauss' algorithm)

- Shortest vector problem (SVP): Given a basis $B$ of an $n$-dimensional lattice $L = L(B)$, find $v \in L$ such that $\|v\| = \lambda_n(L)$

- Approximate SVP ($\text{SVP}_\gamma$): Given a basis $B$ of an $n$-dimensional lattice $L = L(B)$, find $v \in L$ such that $\|v\| \leq \gamma \cdot \lambda_n(L)$

- Decisional approximate SVP ($\text{GapSVP}_\gamma$): Given a basis $B$ of an $n$-dimensional lattice $L = L(B)$, decide if $\lambda_n(L) \leq 1$ or if $\lambda_n(L) > \gamma$

(excitement problem; one of these cases is guaranteed)

- Approximate shortest independent vectors ($\text{SIVP}_\gamma$): Given basis of full-rank $n$-dimensional lattice $L = L(B)$, output a set of linearly independent vectors $b_1, \ldots, b_n$ where $\|b_i\| \leq \gamma \cdot \lambda_n(L)$ for all $i \in [n]$.

Main problems we use for cryptography are short integer solutions ($\text{SIS}$) and learning with errors ($\text{LWE}$)

$\Rightarrow$ These reduce to $\text{GapSVP}_\gamma$ and $\text{SIVP}_\gamma$

$\Rightarrow$ Currently open: basing crypto on search-SVP ($\text{SVP}$ or $\text{SVP}_\gamma$)

Complexity of $\text{GapSVP}_\gamma$ depends on approximation factor $\gamma$:

<table>
<thead>
<tr>
<th>Complexity</th>
<th>$1$</th>
<th>$O(1)$</th>
<th>$2^{(\log n)^2}$</th>
<th>$\sqrt{n \log n}$</th>
<th>$\gamma n$</th>
<th>$O(n)$</th>
<th>$2^{n \log n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approx.</td>
<td>NP-hard*</td>
<td>quasi-NP-hard*</td>
<td>NP $\cap$ coNP</td>
<td>NP $\cap$ coNP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

under randomized reductions $^{\text{example language in NP $\cap$ coNP is graph isomorphism}}$

(approximation factor sufficient for cryptography (e.g., OWF/PKE exist))

[map NO instances to NO instances w.p. $1$
 and YES instances to YES instances w.p. $2/3$]

Unlikely to allow basing crypto on NP hardness since for approximation factors bigger than $\sqrt{n}$, $\text{GapSVP}_\gamma \not\in \text{NP} \cap \text{coNP}$

Open questions: Derandomizing reductions for some gap? (NP-hardness result known for low norm up to nearly polynomial factors)

Poly-time reductions for super-constant approximation factor?

Algorithms for SVP:
- Lenstra - Lenstra - Lovász (LLL) algorithm (lattice reduction)
- Polynomial time algorithm for $\gamma = 2^{\frac{n \log n}{(\log n)^2}}$ approximation
- Known algorithms for $\text{poly}(n)$ approx run in time $2^{O(n)}$ (may need similar space as well)
- Can trade-off time for approximation factor: solve GapSVP$_\gamma$ in time $2^{O(\gamma n \log n)}$ with quantum algorithms
For cryptographic constructions, it is often more convenient to use average-case problems (which admit reductions from GapSVP).

Specifically, we rely on the short integer solutions (SIS) or the learning with errors (LWE) problems, which are average-case problems.

Both the SIS and the LWE problems can be based on the hardness of the GapSVP problem (e.g., an adversary that solves SIS or LWE can be used to solve GapSVP in the worst-case).

**Short Integer Solutions (SIS):** The SIS problem is defined with respect to lattice parameters \( n, m, q \) and a norm bound \( q \). The SIS\(_{n,m,q} \) problem says that for \( A \in \mathbb{Z}_q^{n \times m} \), no efficient adversary can find a non-zero vector \( x \in \mathbb{Z}^m \) where
\[
Ax = 0 \pmod{q}
\]
In lattice-based cryptography, the lattice dimension \( n \) will be the primary security parameter.

Notes: The norm bound \( q \) should satisfy \( q \leq m \). Otherwise, a trivial solution is to set \( x = (q, 0, \ldots, 0) \).

- We need to choose \( m, n, q \) to be large enough so that a solution does exist.
- When \( m = \Omega(n \log q) \) and \( q > \sqrt{m} \), a solution always exists. In particular, when \( m \geq \lceil n \log q \rceil \), there always exists \( x \in \{-1, 0, 1\}^n \) such that \( Ax = 0 \):
  - There are \( 2^n \geq 2^{\Omega(n)} \) vectors \( y \in \{-1, 1\}^n \) and \( x \in \{-1, 0, 1\}^n \).
  - Since \( y \in \mathbb{Z}_q \), there are at most \( q^n \) possible outputs of \( A \).
  - Thus, if we choose \( x = y, y_1 \in \{-1, 0, 1\}^n \), then \( Ax = A(y-y_1) = A_1 - A_2 = 0 \in \mathbb{Z}_q^m \).

SIS can be viewed as an average-case SVP on a lattice defined by \( A \in \mathbb{Z}_q^{n \times m} \):
\[
L(A) = \{ x \in \mathbb{Z}^m : Ax = 0 \pmod{q} \}
\]
\[
\uparrow
called a "q-ary" lattice
\]
\[
\text{in coding-theoretic terms, the matrix } A \text{ is a "parity-check" matrix}
\]
\[
\text{since } \frac{q}{n} \mathbb{Z}^n \leq L(A)
\]

SIS problem is essentially finding short vectors in the lattice \( L(A) \) where \( A \in \mathbb{Z}_q^{n \times m} \).

**Theorem:** For any \( m = \text{poly}(n) \), any \( \beta > 0 \), and sufficiently large \( q \gg \beta \cdot \text{poly}(n) \), there is a probabilistic polynomial time (PPT) reduction from solving GapSVP\(_y \) or \( \text{SIVP}_y \) in the worst case to solving SIS\(_{n,m,q} \) with non-negligible probability, where \( y = \beta \cdot \text{poly}(n) \).

**Implication:** Algorithm for SIS\(_{n,m,q} \) \( \Rightarrow \) algorithm for GapSVP\(_y \) or SIVP\(_y \) in the worst case.

**GapSVP\(_y \) or SIVP\(_y \) hard in the worst case \( \Rightarrow \) SIS\(_{n,m,q} \) hard on average \( \checkmark \) for cryptographic applications

**Tightness:** 
- Micciancio - Regev [MRO]: \( Y = \beta \cdot \mathcal{O}(n) \) - can set \( \beta \) so that \( Y = \delta(n) \) with \( \delta = \mathcal{O}(n \sqrt{m}) = \Theta(1) \)
- Gentry - Peikert - Vaikuntanathan [GPV]: improved bound on \( \beta \) to be \( \beta = \mathcal{O}(n) \) with \( Y = \beta \cdot \delta(n) \) approximation factor
- Micciancio - Peikert [MPR]: improve bound on \( \beta \) to \( \beta = n^2 \) for any \( \epsilon > 0 \) (nearly optimal since \( \beta \ll n \))

**Difficulty:** In these worst-case to average-case reductions, it is needing to take an arbitrary problem instance and embedding it in a random instance.

[We will not cover it today, but may discuss in future lecture.]