Exercises

Anna Somoza

Exercise 1. Let $M \in \mathbb{Z}^{n\times n}$ be a unimodular matrix.

(i) Show that M is invertible, and that M^{-1} is unimodular.

(ii) Show that if $n = 2$, then M is equal to $\pm I_n$ or $\pm \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ times a combination of the matrices $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and their inverses.

(iii) Prove that two bases with matrices B and C generate the same lattice if and only if there exists a unimodular matrix $M \in \mathbb{Z}^{n\times n}$ such that $B = CM$.

Exercise 2. Let $\Lambda \subseteq \Lambda'$ be two full rank lattices. Prove that if $\det \Lambda = \det \Lambda'$, then $\Lambda = \Lambda'$. Prove also that if $\Lambda \neq \Lambda'$, then $\det \Lambda \geq 2 \det \Lambda'$.

Exercise 3. Let Λ be a lattice of dimension n. Show that the number of vectors $x \in \Lambda$ such that $||x|| = \lambda(\Lambda)$ is upper-bounded by 3^n. This number is called the *kissing number*. One can look at the volume of the open balls centered on these points and with radius $\lambda(\Lambda)/2$.

Exercise 4. The goal of this exercise is to prove that every lattice Λ of dimension n has at most $2^{O(n^3)}$ reduced bases.

(i) Let $\lambda = \lambda(\Lambda)$ be the miniaml distance of Λ, and let $(b_1, \ldots, b_n)$ be a reduced basis. Show that $||b_1|| \leq r$ with $r = 2^{O(n)}\lambda$.

(ii) Consider a ball of radius r and the balls of radius $\lambda/2$. Show that there are at most $2^{O(n^2)}$ points of the lattice of length smaller or equal to r. Conclude on the number of possibilities for b_1.

(iii) Consider now the projection $(b'_2, \ldots, b'_n)$ of the vectors $(b_2, \ldots, b_n)$ on the hyperplane orthogonal to b_1. Show that $(b'_2, \ldots, b'_n)$ is sitll a reduced basis (for the lattice generated by $(b'_2, \ldots, b'_n)$).

(iv) Show that b'_2 cannot come from more than 2 b_2 of a reduced basis of Λ with b_1 fixed.

(v) Deduce that the number of possible b_2 is at most $2^{O(n-1)^2}$.

(vi) Conclude by recurrence the claim of the exercise.

Exercise 5. Let Λ be a lattice of dimension n.

(i) Using Minkiwski's theorem with a parallelepiped, show that there exists $x \in \Lambda$ nonzero such that $||x||_\infty \leq (\det L)^{1/n}$.

(ii) Show that for this x, we have $||x||_2 \leq \sqrt{n}(\det L)^{1/n}$.

We will now obtain a weaker, but constructive, result. Let $b_i^* = b_i - \sum_{j<i} \mu_{ij} b_j^*$.

(iii) Show by induction that we can always take $\mu_{i,i-1} \leq 1/2$, replacing b_i by $b'_i = b_i - \lfloor \mu_{i,i-1} \rceil b_{i-1}$ if necessary.

(iv) Show that the condition $||b^*_{i-1}||_2 \leq ||b^*_i + \mu_{i,i-1} b^*_{i-1}||_2$ can be interpreted geometrically in terms of the projection of b_{i-1} and b_i on $\langle b_1, \dots b_{i-2}\rangle^{\perp}$.

(v) Deduce that the property above is true, exchanging b_{i-1} and b_i if necessary.

We would like to obtain both properties at the same time. Consider the following algorithm:

Algorithm 1: Reduction procedure

Make all $\mu_{i,i-1}$ smaller or equal to $1/2$ in absolute value.
while $\exists i_0, ||b^*_{i_0-1}||_2 > ||b^*_{i_0} + \mu_{i_0,i_0-1} b^*_{i_0-1}||_2$ **do**
  Swap b_{i_0} and b_{i_0-1}.
  Make all $\mu_{i,i-1}$ smaller or equal to $1/2$ in absolute value.

(vi) Show that the algorithm finishes because the norms $(||b^*_1||_2, \dots ||b_n||_2)$ decrease strictly on each iteration.

(vii) Show that for $i > 1$ we have $3/4||b^*_{i-1}||_2^2 \leq ||b^*_i||_2^2$ by the end of the algorithm.

(viii) Using the fact that $\det \Lambda = \prod ||b^2_i||_2$, show Hermite's inequality

$$||b_1||_2 \leq \left(\frac{4}{3}\right)^{(n-1)/4} (\det \Lambda)^{1/n}.$$

Exercise 6. Consider the lattice Λ generated by the columns of the following matrix:

$$B = \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We consider $b'_4 = 2b_4 - b_1 - b_2 - b_3$. Show that b_1, b_2, b_3, b'_4 are linearly independent, and that they attain the minimal length 2, but that they do not generate the lattice Λ.

Note: In dimension ≥ 5, there exist lattices for which no choice of vectors attaining the minimal length forms a basis of the lattice.

Exercise 7. In dimension 2, consider the following algorithm, where we use $q(u) = ||u||^2$.

Algorithm 2: Gauss' algorithm

---

input : An ordered basis (u, v) with $q(u) \leq q(v)$
output: A reduced basis of the lattice
repeat
  $x = \lfloor \langle u, v\rangle / q(u) \rceil$
  $r = v - xu$
  $v = u$
  $u = r$
until $q(u) \geq q(v)$;
return (v, u)

First we focus on the correctness of the algorithm.

(i) Show that the output (U, V) is a basis of the lattice

(ii) Show that $q(U) \leq q(V)$ and that for all $y \in \mathbb{Z}$ we have $q(V + yU) \geq (V)$.

(iii) Using $q(U + V) \geq q(V)$ and $q(U - V) \geq q(V)$, deduce that $|\langle U, V\rangle| \leq q(U)/2$.

(iv) Show that $q(U)$ is minimal by proving that if we have $q(x_1U + x_2V) < q(U)$ then $x_1 = x_2 = 0$.

(v) Show that $q(V)$ attains the second minimum, that is, it is not possible to have $q(x_1U + x_2V) < q(V)$ with $x_2 \neq 0$.

Now we focus on the execution time of the algorithm.

(vi) Show that if $x = 0$, then it ends loop.

(vii) Prove that $|x| = 1$ can only occur on the two first or the last iteration of the algorithm. Do so by contradiction, by showing that then r is not the minimal choice.

(viii) Assume $|x| > 1$. Prove that in that case we have $\langle u, v\rangle/q(u) \geq 3/2$.

(ix) Let $v^\perp$ be the projection of v on $\langle y\rangle^\perp$. Prove that $q(v) \geq q(v^\perp) + 9/4q(u)$.

(x) Prove that $q(r) \leq q(v^\perp) + 1/4q(u)$.

(xi) Deduce that $q(v) \geq q(r) + 2q(u)$, and that if we are not on the last iteration, then $q(v) \geq 3q(r)$.

(xii) Deduce that, except on the first two or the last iteration of the algorithm, $q(u)q(v)$ decreases by a factor of 3 on each iteration. Denote by λ_1 the minimum of the lattice, and u_0, v_0 the input vectors, and prove that the number of iterations is at most $2\log_3 q(v_0)/\lambda_1^2 + 2 = O(\log q(v_0))$.

(xiii) The cost of each step inside the loop is upper-bounded by the cost of the computation of x, that is an euclidean division. If we write $a = bq + r$, then the cost is $O(\log(a)^2)$. Deduce the total cost of the algorithm.