Factoring univariates over $\mathbb{Q}$

- Suppose $f(x) \in \mathbb{Q}[x]$ is a polynomial to be factored.

  By multiplying it with a positive integer we could clear away the denominators.

- So, wlog $f(x) \in \mathbb{Z}[x]$. Let n be its degree and the coefficients a_i be of ℓ-bits.

- How do we factor, or test the irreducibility of, the <u>integral</u> polynomial $f(x)$?

- Starting idea is to factor it modulo a prime p, do Hensel lifting and "solve a linear system" (much like bivariate factoring).

- Let us first see the algorithm & then the analysis.

  It was discovered by (Lenstra, Lenstra, Lovász) in 1982, starting a new field.

Input: $f = \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x]$, $|a_i| < 2^{\ell-1}$ $(0 \le i \le n)$.
Output: A nontrivial integral factor (if one exists).
L^3-algorithm:

1) Preprocess: Assume that f is square-free. Find the smallest prime p s.t. $\begin{cases} p \nmid a_n, \\ f(x) \bmod p \text{ is sq.-free.} \end{cases}$

[If f is sq.-full then $\gcd(f, f')$ factors f.
$f(x) \bmod p$ is sq.-full iff $p \mid \mathrm{res}(f, f')$.
Now, $|a_n \cdot \mathrm{res}(f, f')| < 2^{\ell} \cdot (2^{\ell})^{n+1} \cdot (2^{\ell + \lg n})^{n} \cdot (2n+1)!$
$\Rightarrow$ # primes p dividing $a_n \cdot \mathrm{res}(f, f')$ are at most $2\ell(n+1) + 3n\lg n < 3n \cdot (\ell + \lg n)$ many.
$\Rightarrow$ A prime $p = \tilde{O}(\ell n)$ will work.]

2) Factor mod p: Using Berlekamp's algorithm compute a factorization $f(x) \equiv g_0 \cdot h_0 \pmod{p}$
where $g_0(x) \bmod p$ is monic, irreducible & coprime to h_0.

3) Hensel lift: Compute $f \equiv g_k \cdot h_k \pmod{p^{2^k}}$,

for $k = \lceil \lg 2n^3\ell \rceil$.

4) Linear system: Find $\tilde{g}, \ell_k$ s.t. $\tilde{g} \equiv g_k \cdot \ell_k \pmod{p^{2^k}}$ with $\deg \tilde{g} < n$ & the coefficients of $\tilde{g}$ are at most $2^{n\cdot(\ell+\lg n)}$ in magnitude.

5) Output $\gcd(f, \tilde{g})$.

Analysing the steps

Step 2: Since $p = \tilde{O}(\ell n)$, this step finds g_0 in deterministic poly$(n\ell)$ time. $\square$

Step 3: Clearly, in poly$(n\ell)$ time. $\square$

Step 4: For this we need to estimate the size of the factors of f.

Lemma 1: (Mignotte's bound) Any root $\alpha \in \mathbb{C}$ of a polynomial

$f(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x]$ satisfies $|\alpha| \le n \cdot \max_i |a_i|$.

Proof:

- If $|\alpha| < 1$ then done. (consider rev(f) to get a lower bound on α.)
- Else $|f(\alpha)| = |a_n \alpha^n + \sum_{i=0}^{n-1} a_i \alpha^i|$

$$\geqslant |\alpha|^n - \sum_{i=0}^{n-1} |a_i \alpha^i|$$

$$\geqslant |\alpha|^n - n \cdot (\max_i |a_i|) \cdot |\alpha|^{n-1}$$

$$\Rightarrow |\alpha| \leqslant n \cdot \max_i |a_i| . \qquad \square$$

Lemma 2: Any factor g of f has coefficients of magnitude at most $2^{(\ell + \lg n - 1)n}$.

Proof:
- Let $g(x) = \prod_{i=1}^{m} (x - \alpha_i)$, $\alpha_i \in \mathbb{C}$.

- The coeff. of x^{m-j} is $\sum_{S \in \binom{[m]}{j}} \prod_{i \in S} (-\alpha_i)$.

- Its magnitude $\leqslant \sum_{S} \prod_{i \in S} |\alpha_i|$

$$< \binom{m}{j} \cdot (n 2^{\ell-1})^j < (1 + n 2^{\ell-1})^{n-1} < 2^{(\ell + \lg n - 1)n} . \qquad \square$$

– Thus, $\exists$ suitable $\tilde{g}$ if f is reducible. $\square$

<u>Step 5:</u> • If a $\tilde{g}$ exists in Step 4, and $(f,\tilde{g})=1$, then $\exists u,v \in \mathbb{Z}[x]$, $\quad uf + v\tilde{g} = res(f,\tilde{g}) \neq 0$.

$\Rightarrow u\cdot g_k\cdot h_k + v\cdot g_k\cdot l_k \equiv res(f,\tilde{g}) \pmod{p^{2^k}}$.

$\Rightarrow g_k\cdot(uh_k + vl_k) \equiv$ " " . ---- (i)

• Note that $|res(f,\tilde{g})| < (2n+1)!\cdot(2^{\ell})^{n+1}\cdot\left(2^{(\ell+\lg n)\cdot n}\right)^{n}$
$< 2^{2n^3\ell} < p^{2^k}$.

$\Rightarrow$ In eqn. (i), the RHS is a nonzero <u>constant</u> while the LHS is a <u>nonconstant</u> integral polynomial. $\lightning$

$\Rightarrow$ This contradiction implies that Step 5 factors f, whenever $\tilde{g}$ exists. $\quad\Box$

<u>How do we compute $\tilde{g}$ (with "small" coeffs.)?</u>

– Let g_k be of deg $n' < n$. The unknown polynomials are: $\tilde{g} = \sum_{i=0}^{n-1} c_i x^i$ & $l_k = \sum_{i=0}^{n-1-n'} \alpha_i x^i$ s.t. $\tilde{g} \equiv g_k\cdot l_k \pmod{p^{2^k}}$.

- This can be rephrased as an integral system:

$$\sum_{i=0}^{n-1} c_i x^i = \sum_{i=0}^{n-1-n'} \alpha_i \cdot (x^i g_k) + \sum_{i=0}^{n-1} \beta_i \cdot (p^{2^k} x^i), \quad \text{---- (ii)}$$

where the unknown c's, α's & β's are in $\mathbb{Z}$.

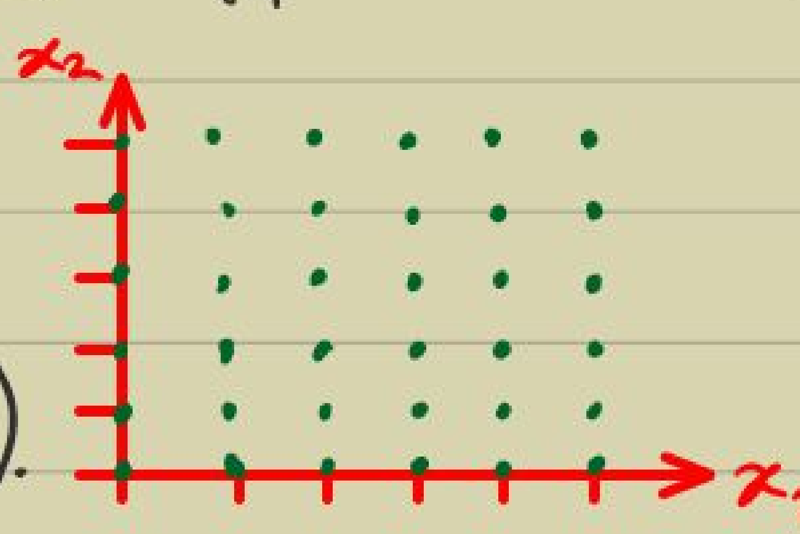
- We want a solution to (ii) s.t. $\|\bar{c}\| := \left(\sum_{i=0}^{n-1} c_i^2\right)^{1/2}$ is "small" $(< 2^{(\ell+\lg n)\cdot n})$.
[assuming the existence of length $< 2^{(\ell+\lg n-1)n}$.]

- So the related fundamental problem to be solved is:

Given $b_1, \ldots, b_m \in \mathbb{Z}^n$, find $\gamma_1, \ldots, \gamma_m \in \mathbb{Z}$ s.t. $\| \sum \gamma_i b_i \|$ is "small".

<u>Defn</u>: The $\mathbb{Z}$-linear-combinations of $\{b_i\}$ form a <u>lattice</u> <u>$\mathcal{L}(b_1, \ldots, b_m)$</u> $:= \left\{ \sum_{i=1}^{m} \gamma_i b_i \mid \gamma_i \in \mathbb{Z} \right\}$.

- Eg. $\mathcal{L}\left(\binom{1}{0}, \binom{1}{1}\right)$ is:



▷ $\mathcal{L}\left(\binom{1}{0}, \binom{1}{1}\right) = \mathcal{L}\left(\binom{1}{0}, \binom{0}{1}\right)$.

[Ajtai'98] - Computing a <u>shortest</u> vector in $\mathcal{L}(b_1, \ldots, b_m)$ is an NP-hard problem (<u>SVP</u>).

But, we need merely a 2^n-approximation. (better approx. are believed to be hard)

- First, we do a preprocessing step:

<u>Lemma 1</u>: We could assume, wlog, that $\{b_1, \ldots, b_m\} =: B$ are linearly independent. (requires integral b_i's.)

<u>Proof</u>:

- Consider the matrix $B := \begin{pmatrix} b_{11} & b_{21} & \cdots & b_{m1} \\ b_{12} & b_{22} & \cdots & b_{m2} \\ \vdots & \vdots & & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{mn} \end{pmatrix}$
- Let $\sum_{i=1}^{m} a_i b_{i1} = g := \gcd(b_{11}, b_{21}, \ldots, b_{m1})$.
- Apply the extended-Euclid-algo. transformations on the columns. (to corner g)
- Say, the new columns are $b_1', b_2', \ldots, b_m'$.
- Next, transform the cols. $2 \leq j \leq m$, $b_j' \leftarrow b_j' - \frac{b_{j1}'}{g} \cdot b_1'$.
- This gives us a $B' = \left(\begin{array}{c|ccc} g & 0 & \cdots & 0 \\ \hline * & & & \\ \vdots & & * & \\ * & & & \end{array} \right)_{n \times m}$.

• The transformation is $B' = B \cdot \mathcal{U}$, where

$$\mathcal{U} := \quad \mathcal{E} \quad \cdot \begin{pmatrix} 1 & -b_{21}/g & \cdots & -b_{m1}/g \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

& $\mathcal{E}$ is the product of matrices following the Euclid's algorithm on the numbers $\{b_{11}, b_{21}, \ldots, b_{m1}\}$.

• Note that each step in the Euclid's algo. is unimodular, i.e. $|\mathcal{E}| = \pm 1$

$\Rightarrow \quad |\mathcal{U}| = \pm 1$.

$\Rightarrow \quad \mathcal{L}(B') = \mathcal{L}(B)$. [$\mathcal{E}^{-1}$ is integral.]

• On repeatedly applying this Gauss-Euclid trick, we get a matrix

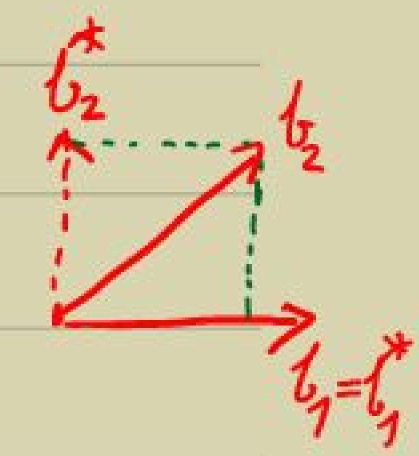
$$\tilde{B} := \left(\begin{array}{c|c} A_{m' \times m'} & \\ & O_{n \times (m-m')} \\ C_{(n-m') \times m'} & \end{array} \right)$$

where A is lower-triangular and $\mathcal{L}(\tilde{B}) = \mathcal{L}(B)$.

$\Rightarrow$ The first m' columns of $\tilde{B}$ form a basis of size $m' \leq \min(n, m)$ spanning our lattice. $\square$

- So, we work with l.i. $b_1, \ldots, b_m \in \mathbb{Z}^n$.
- In the vector space $\mathrm{span}_{\mathbb{R}}(b_1, \ldots, b_m) =: V(B)$ there is an <u>orthogonal</u> basis:

- <u>Idea</u>: • Orthogonalize $\{b_1, b_2\}$ to
$$\left\{ b_1^* = b_1 \,,\; b_2^* := b_2 - \frac{\langle b_2, b_1^* \rangle}{\|b_1^*\|^2} \cdot b_1^* \right\}.$$



▷ It is easily seen that the shorter of b_1^*, b_2^* is the shortest vector in $\mathcal{L}(b_1^*, b_2^*)$.
[$\because \|\alpha_1 b_1^* + \alpha_2 b_2^*\|^2 = \|\alpha_1 b_1^*\|^2 + \|\alpha_2 b_2^*\|^2$.]

<u>Gram-Schmidt Orthogonalization</u>:

1) Let $b_1^* := b_1$.
2) For all $2 \leq i \leq m$, do
$$b_i^* := b_i - \sum_{j=1}^{i-1} \underbrace{\frac{\langle b_i, b_j^* \rangle}{\|b_j^*\|^2}}_{\mu_{ij}} \cdot b_j^* .$$

<u>Lemma 2</u>: Each nonzero vector $b \in \mathcal{L}(b_1, \ldots, b_m)$ satisfies $\|b\| \geq \min_i \|b_i^*\|$.

<u>Proof</u>:

• Let $b = \lambda_1 b_1 + \ldots + \lambda_m b_m$ for λ's in $\mathbb{Z}$ s.t. $\lambda_m \neq 0$.

- $\Rightarrow b = \lambda_1 b_1^* + \lambda_2 (b_2^* + \mu_{21} b_1^*) + \ldots + \lambda_m (b_m^* + \mu_{m,m-1} b_{m-1}^* + \ldots$
$+ \mu_{m1} b_1^*)$.

$\Rightarrow \|b\|^2 = (\ldots)^2 \cdot \|b_1^*\|^2 + (\ldots)^2 \cdot \|b_2^*\|^2 + \ldots + \lambda_m^2 \cdot \|b_m^*\|^2$

$\Rightarrow \|b\| \geqslant |\lambda_m| \cdot \|b_m^*\| \geqslant \|b_m^*\|$. □

- Using $\mathbb{Z}$-coefficients it may not be possible to orthogonalize $\mathcal{L}(B)$. So, L^3 tries to make the "angles" around 60°! [$\cos 60° = 1/2$]
[Gauss & Lagrange could solve $m=2$ case using this.]

Defn: • L^3 will find a reduced basis of $\mathcal{L}(b_1, \ldots, b_m)$.
These are lattice elements $c_1, \ldots, c_m$ s.t.

(i) $\forall i, \|c_i^*\|^2 \leq \frac{4}{3} \cdot \|c_{i+1}^* + \mu_{i+1,i} c_i^*\|^2$

(ii) $\forall i \geqslant j, |\mu_{ij}| \leq 1/2$

where $\mu_{ij} := \dfrac{\langle c_i, c_j^* \rangle}{\|c_j^*\|^2}$

▷ $\Rightarrow \|c_i^*\|^2 \leq \frac{4}{3} \|c_{i+1}^*\|^2 + \frac{1}{3} \cdot \|c_i^*\|^2$

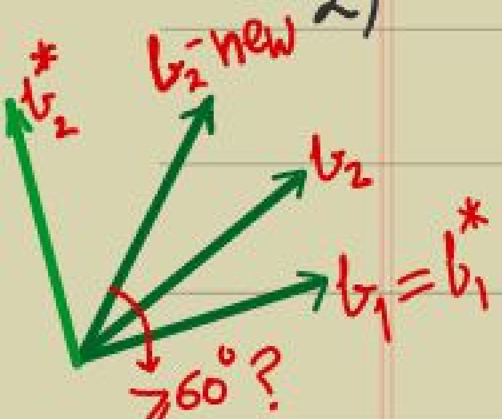
$\Rightarrow \|c_i^*\| \leq \sqrt{2} \cdot \|c_{i+1}^*\| \Rightarrow \|c_1^*\| \leq \min_i \{2^{\frac{i-1}{2}} \cdot \|c_i^*\|\}$.

$\Rightarrow \|c_1^*\| \leq 2^{\frac{m-1}{2}} \cdot \lambda(\mathcal{L})$ & $\|\overset{c_1}{\overset{=}{c_1^*}}\| \geqslant \lambda(\mathcal{L})$.

where $\lambda(\mathcal{L})$ is the shortest length in $\mathcal{L}(B)$.

L^3-reduced basis algorithm

1) Compute the GS-orthogonalization of $b_1, \ldots, b_m$.
2) For $i = 2$ to m
   For $j = i-1$ to 1
   $$b_i \leftarrow b_i - \left\lfloor \frac{\langle b_i, b_j^* \rangle}{\| b_j^* \|^2} \right\rceil \cdot b_j \, .$$
   rounding to nearest integer



3) If $\exists i$, $\| b_i^* \|^2 > \frac{4}{3} \cdot \| b_{i+1}^* + \mu_{i+1,i} \, b_i^* \|^2$
   then swap $\{b_i, b_{i+1}\}$ & GOTO (1).
   Attempt to reduce b_i^*
4) Output $\{b_1, \ldots, b_m\}$.

Analysis

Step 2: • Note that $b_2 \leftarrow b_2 - \left\lfloor \frac{\langle b_2, b_1 \rangle}{\| b_1 \|^2} \right\rceil \cdot b_1$

$$\Rightarrow \frac{\langle b_2, b_1 \rangle}{\| b_1 \|^2} \leftarrow \frac{\langle b_2, b_1 \rangle}{\| b_1 \|^2} - \left\lfloor \frac{\langle b_2, b_1 \rangle}{\| b_1 \|^2} \right\rceil \cdot \frac{\langle b_1, b_1 \rangle}{\| b_1 \|^2}$$

$\Rightarrow |\mu_{2,1}|$ is being reduced to $\leq 1/2$.

• The same holds true for $|\mu_{i,i-1}|$, $i \in [m]$.

• Also, the transformation is unimodular, so the lattice remains unchanged.

Step 3: • To show that this step will not happen many times, we need a potential function:

$$D(b_1,\dots,b_m) := \prod_{i=1}^{m} \|b_i^*\|^{2(m-i)}.$$

• Step 2 has no effect on this.

While each Step 3 swap reduces D by a factor of $\frac{\|b_{i+1}^*\|^2}{\|b_i^*\|^2} < \left(\frac{3}{4} - \mu_{i+1,i}^2\right) < \frac{3}{4}$.

Lemma 3: $|D(b_1,\dots,b_m)|$ is a positive integer of value under $2^{\tilde{O}(n^5 \ell)}$.

Proof:

• Write D as $\prod_{j=1}^{m-1} D_j$, where $D_j := \prod_{i=1}^{j} \|b_i^*\|^2$.

• We now relate D_j with vol$(b_1,\dots,b_j)$:

• D_j is the det. of $(b_1^*,\dots,b_j^*)^T \cdot (b_1^*,\dots,b_j^*)$ which is the same as $((b_1,\dots,b_j)\cdot C)^T \cdot ((b_1,\dots,b_j)\cdot C)$, for a unimodular transformation C.

$\Rightarrow D_j = |(b_1,\dots,b_j)^T \cdot (b_1,\dots,b_j)| \in \mathbb{Z}_{>0}$.

• The bound follows from the size of b_j's:

$$D_j = 2^{\tilde{O}(n^3 \cdot \ell) \cdot j}$$

$$\Rightarrow D = 2^{\tilde{O}(n^5 \cdot \ell)} . \qquad \square$$

▷ Thus, step 3 can repeat at most $\tilde{O}(n^5 \cdot \ell)$ times in the L^3-algorithm.

▷ A crude time estimate of poly. fact. algorithm is then $\tilde{O}(n^{11} \cdot \ell^2)$. $= n^5\ell \cdot n^3 \cdot \tilde{O}(n^3 \ell)$

▷ Assuming $L :=$ max <u>bit-size</u> in b_i's, we get a crude estimate for the L^3-algo. (to approximate a shortest vector) of : $\tilde{O}(L \cdot m \cdot m^2)^2 = \tilde{O}(m^6 \cdot L^2)$.

for pre-processing (m) — for the potential-fn. (m^2)

<u>Application to simultaneous Diophantine approx.</u>

– L^3-algorithm, & the idea of <u>reduced basis</u>, is used in many places.

- Eg. computational problems in algebraic number theory, faster arithmetic in number fields, Knapsack problem, testing conjectures (Merten's conjecture, ABC-conjecture, ...).

- The main reason is the following property of L^3: [relation to the volume]

<u>Theorem</u>: If $b_1, \ldots, b_n$ is a reduced basis for a lattice $L \trianglelefteq \mathbb{Z}^n$ & $b_1^*, \ldots, b_n^*$ is its GSO, then:

(i) $\|b_j\| \leq 2^{\frac{i-1}{2}} \cdot \|b_i^*\|$, $\forall\ 1 \leq j \leq i \leq n$.

(ii) $d(L) \leq \prod_{i=1}^{n} \|b_i\| \leq 2^{n(n-1)/4} \cdot d(L)$.

(iii) $\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot d(L)^{1/n}$.

[$d(L) := |(b_1, \ldots, b_n)|$ is the <u>determinant</u> of L.]

<u>Proof</u>:

(i) We have $\|b_j\|^2 = \|b_j^*\|^2 + \sum_{k=1}^{j-1} \mu_{jk}^2 \cdot \|b_k^*\|^2$.

$\leq \|b_j^*\|^2 + \sum_{1 \leq k \leq j-1} \frac{1}{4} \cdot 2^{j-k} \cdot \|b_j^*\|^2$

$= \|b_j^*\|^2 \cdot \left(1 + \frac{2^j - 2}{4}\right) \leq 2^{j-1} \cdot \|b_j^*\|^2$.

(ii) By unimodularity of GSO, $d(L) = |(b_1^*, \ldots, b_n^*)|$.

$\Rightarrow \quad d(L) = \prod_{1 \leq i \leq n} \|b_i^*\|$

- Since $\|b_i^*\| \leq \|b_i\|$, we get $d(L) \leq \prod_{i=1}^{n} \|b_i\|$.

- From (i) we have, $\prod_{j=1}^{n} \|b_j\| \leq \prod_{j=1}^{n} 2^{\frac{j-1}{2}} \cdot \|b_j^*\|$
$= 2^{n(n-1)/4} \cdot d(L)$.

(iii) Putting $j=1$ in (i) we have,

$$\prod_{1 \leq i \leq n} \|b_1\|^2 \leq \prod_{1 \leq i \leq n} 2^{i-1} \cdot \|b_i^*\|^2$$

$$\Rightarrow \quad \|b_1\|^{2n} \leq 2^{n(n-1)/2} \cdot d(L)^2$$

$$\Rightarrow \quad \|b_1\| \leq 2^{\frac{n-1}{4}} \cdot d(L)^{1/n}.$$

$\square$

– Suppose we are given rationals $\alpha_1, \ldots, \alpha_n, \varepsilon$ & we want to find integers $p_1, \ldots, p_n, q$ s.t. $\forall i$, $|p_i - q \cdot \alpha_i| \leq \varepsilon$ & q is "small".

– L^3 provides a poly-time algorithm!

- <u>Idea:</u> Consider the lattice $\mathcal{L}$ generated by the columns of

$$B = \begin{pmatrix} 1 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & \cdots & 0 & -\alpha_2 \\ \vdots & & & \vdots & \\ 0 & 0 & \cdots & 1 & -\alpha_n \\ 0 & 0 & \cdots & 0 & 2^{-n(n+1)/4} \cdot \varepsilon^{n+1} \end{pmatrix}$$

- It has elements like

$$\left(p_1 - q\alpha_1,\ p_2 - q\alpha_2, \ldots, p_n - q\alpha_n,\ q \cdot 2^{\frac{-n(n+1)}{4}} \cdot \varepsilon^{n+1}\right),$$

for integers $p_1, \ldots, p_n, q$. ----- (a)

- By the previous theorem, L^3-algo. gives a vector b_1 in poly-time s.t.

$$\|b_1\| \leq 2^{n/4} \cdot d(\mathcal{L})^{1/n+1} = \varepsilon.$$

$\Rightarrow$ the p's & q corresponding to b_1 in eqn. (a) are not <u>too large</u>.

$\triangleright$ In particular, $q \leq 2^{n(n+1)/4} \cdot \varepsilon^{-n}$.