Available online at www.sciencedirect.com

SCIENCE DIRECT

Theoretical Computer Science 326 (2004) 419–429

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

On the ultimate complexity of factorials[☆]

Qi Cheng

School of Computer Science, University of Oklahoma, 200 Felgar Street, Norman, OK 73019, USA

Received 3 February 2004; received in revised form 9 June 2004; accepted 18 June 2004
Communicated by F. Cucker

Abstract

It has long been observed that certain factorization algorithms provide a way to write the product of many different integers succinctly. In this paper, we study the problem of representing the product of all integers from 1 to n (i.e. $n!$) by straight-line programs. Formally, we say that a sequence of integers a_n is ultimately $f(n)$-computable, if there exists a nonzero integer sequence m_n such that for any n, $a_n m_n$ can be computed by a straight-line program (using only additions, subtractions and multiplications) of length at most $f(n)$. Shub and Smale [12] showed that if $n!$ is ultimately hard to compute, then the algebraic version of $NP \neq P$ is true. Assuming a widely believed number theory conjecture concerning smooth numbers in a short interval, a subexponential upper bound ($\exp(c\sqrt{\log n \log\log n})$) for the ultimate complexity of $n!$ is proved in this paper, and a randomized subexponential algorithm constructing such a short straight-line program is presented as well.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Computational and structural complexity

1. Introduction

Computing the factorial function ($n!$) is an interesting problem in computational complexity. Because of the size of the number, computing $n!$ certainly takes exponential time.

E-mail address: qcheng@bachman.cs.ou.edu (Q. Cheng).

[☆] The preliminary version of this paper appeared in the Proceeding of the 20th International Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science 2607, Springer. This research is partially supported by NSF Career Award CCR-0237845.

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.06.020

One can instead study the modular factorial $n! \bmod m$. Given an integer $m \geqslant 2$, the smallest positive integer α such that $\gcd(\alpha! \bmod m, m) > 1$ is a prime factor of m. For every integer $n \geqslant \alpha$, $\gcd(n! \bmod m, m)$ is greater than 1, hence we can use binary search to find α if we know how to compute $n! \bmod m$ efficiently for any m and n. This shows that the integer factorization problem can be reduced to computing $n! \bmod m$. It is very interesting to compare modular exponentiation with modular factorial. In some sense, the reason that primality testing is easy while factoring is hard is because modular exponentiation is easy but modular factorial is hard. This statement may underestimate the complexity of modular factorial, as it is believed that computing $n! \bmod m$ is much harder than the integer factorization problem. We do not even know whether computing modular factorial is an NP-easy problem or not.

One approach we may take to compute $n! \bmod m$ is to find a short straight-line program for $n!$. This problem relates to the algebraic model of computation [1–3]. In the algebraic model, it makes sense to ask whether the factorial problem has a polynomial time algorithm, because in this context, to estimate the time complexity, we only count the number of ring operations used to compute $n!$ regardless of the size of the operands. Sometimes, algebraic complexity is also called non-scalar complexity. If the function $n!$ has polynomial time algebraic complexity, or equivalently, every integer $n!$ has a short straight-line program uniformly, then by doing modulo m in every step, we would obtain a polynomial time algorithm to compute $n! \bmod m$ on a Turing machine, which is thought to be unlikely. Throughout this paper, we assume that a straight-line program only contains ring operations. Shamir [11] showed that if division (computing remainder and quotient) is allowed, then $n!$ can be computed by a straight-line program of polynomial length.

The ultimate complexity of a number was first studied in [12] by Shub and Smale. They found a surprising relation between the ultimate complexity of $n!$ and the algebraic version of NP vs. P problem. We say that $n!$ is ultimately hard to compute, if there does not exist a non-zero integer sequence m_n, such that $n!m_n$ can be computed by straight-line programs of length $(\log n)^c$ for an absolute constant c. It was proved in [12]:

If $n!$ is ultimately hard to compute, then the algebraic version of $NP \neq P$ is true.

Note that in the Turing model, proving that the modular factorial problem is hard does not necessarily imply that $NP \neq P$. There is no corresponding concept of the ultimate complexity in the Turing model.

So far the best algorithm we know computes $n!$ in $O(\sqrt{n}\log^2 n)$ ring operations over $\mathbf{Z}$ [14,4]. No better upper bound has been reported for the ultimate complexity of $n!$. It has long been noticed that certain factorization algorithms provide a way to write the product of many different primes succinctly. For instance, Lenstra's elliptic curve factorization method [10], performs algebraic computation modulo the integer to be factored, but the operations remain the same for all inputs of a certain size. The algebraic computation essentially generates a number with a lot of prime factors, since it factors almost all integers of the size. However, these algorithms do not directly give us a straight-line program to compute a product of $n!$, because

(1) Divisions are essential in these algorithms. For instance, in the elliptic curve factorization method, splitting of an integer n happens precisely when the inverse of a integer modulo n does not exist.

(2) More importantly, all the fast factorization algorithms are random in nature. The time complexity of a randomized algorithm is an average measurement. Practically the algorithms should work as expected, but theoretically it is possible that for any integer n, there are bad choices of random bits such that the algorithm will take exponential time to stop. Hence for any choice of random bits of a certain length, there are integers which cannot be factored.

In this paper, we give a formal proof of a subexponential upper bound for the ultimate complexity of $n!$ under a widely believed number theory conjecture. Our result is constructive in the sense that we can construct the straight-line program from n in subexponential time. More precisely, our paper presents a Monte Carlo algorithm (certainly in the Turing model), given a natural number n as input, output a straight-line program which computes a non-zero multiple of $n!$ with probability better than a constant. The algorithm runs in subexponential time, hence the output straight-line program will have at most a subexponential length. Our result suggests that the algebraic complexity of certain product of $n!$ is not as high as the complexity of $n!$, even though the product looks more random than $n!$. Note that by a simple counting argument, we can prove that there exists a divisor of $n!$ which has exponential straight-line complexity. Our result also shows that the complexity of certain multiple of $n!$ is much closer to the integer factorization problem than the complexity of $n!$ itself.

It is interesting to note that we do not know whether there exists a subexponential straight line program for the polynomial $(x-1)\cdots(x-n)$, or any polynomial with a lot of distinct integral roots. If we apply the same technique in the paper to construct straight line program, we encounter obstacles from the Uniform Boundedness Theorem [6].

1.1. Main results

We call a number smooth if all of its prime factors are small. More precisely, a number is said to be y-smooth, if all of its prime factors are less than or equal to y. Let

$$\Psi(x, y) = |\{n \leqslant x : n \textit{ is } y\textit{-smooth}\}|.$$

Throughout this paper, log denotes the natural logarithm. Let $L_x[c]$ denote $e^{c\sqrt{\log x \log\log x}}$. The following proposition about $\Psi(x, L_x[a])$ was proved in [5].

Proposition 1. *For any constant a, $\Psi(x, L_x[a]) = xL_x[-1/(2a) + o(1)]$.*

It was conjectured that the smooth number in some short interval is as dense as in a large interval. In particular,

Conjecture 1. *For any constant $a > 0$,*

$$\Psi(p+1+2\sqrt{p}, L_p[a]) - \Psi(p+1-2\sqrt{p}, L_p[a]) = \sqrt{p}L_p[-1/(2a) + o(1)].$$

Though this conjecture has not been proved yet, it is widely believed to be true. See [9,10] for details. In fact, Lenstra's elliptic curve factorization algorithm relies on this conjecture to achieve the subexponential time complexity.

Theorem 1. *Assume that Conjecture 1 is true. Then there exist absolute constants c_1 and c_2 such that for any natural number n, a non-zero multiple of $n!$ can be computed by a straight-line program of length at most $L_n[c_1]$. Furthermore, the straight-line program can be constructed in time $L_n[c_2]$ by a probabilistic Turing machine.*

The essential part of the proof of Theorem 1 is based on Lenstra's elliptic curve factorization method. Let $\mathcal{E}$ be an elliptic curve $y^2 = x^3 + ax + b$ with $a, b \in \mathbf{Z}$ and $P_s(x)$ be the univariate sth division polynomial of $\mathcal{E}$. Given n and x, $P_n(x)$ can be computed by a straight-line program of length $\mathrm{O}(\log n)$ using 1, x, a and b as constants. If x, a and b are integers less than n, then $P_n(x)$ can be calculated by $\mathrm{O}(\log n)$ arithmetic operations using 1 as the only constant. Let x be an integer which is not the abscissa of a torsion on $\mathcal{E}$, i.e. $P_i(x) \neq 0$ for any positive integer i. For any prime p, we have $p|P_s(x)$ if s is divisible by $|E(\mathbf{F}_p)|$, where E is the reduction of $\mathcal{E}$ at p and x mod p is the abscissa of a point on $E(\mathbf{F}_p)$ (x may or may not be an abscissa of a point on $\mathcal{E}(\mathbf{Q})$).

If the reduction of $\mathcal{E}$ at a random prime p takes a random number between $p-2\sqrt{p}+1$ and $p+2\sqrt{p}+1$ as the order over $\mathbf{F}_q$, then with probability greater than 1 over a subexponential function on $\log p$, the reduction curve has a smooth order over $\mathbf{F}_p$. Furthermore, given an elliptic curve $E/\mathbf{F}_p$, a random integer x mod p becomes an abscissa of a point on $E(\mathbf{F}_p)$ with a constant probability (about $\frac{1}{2}$). Hence if S is a large smooth number and x is an arbitrary integer, $P_S(x)$ contains a lot of distinct prime factors. In order to get a multiple of $n!$, we only need to collect subexponentially many elliptic curves and evaluate their Sth division polynomials at polynomially many integers. We will show that randomly chosen elliptic curves and integers suffice. The effects of the global torsions will be carefully controlled.

This paper is organized as follows. In Section 2, we define the straight-line program and the ultimate complexity, and prove a lemma about bipartite graphs. In Section 3, we review some facts about elliptic curves. In Section 4, we formally prove the main theorem. We conclude this paper by a discussion section.

2. Preliminaries

A straight-line program of an integer is a sequence of ring operations, which outputs the integer in the last operation. Formally,

Definition 1. A straight-line program of an integer m is a sequence of instructions

$$z \leftarrow x \alpha y$$

where $\alpha \in \{+, -, *\}$, x, y are two previously appeared symbols or 1 and z is a new symbol, such that after we execute the instructions sequentially, the last symbol will represent the value of m. The length of the program is the number of instructions. The length of the shortest straight-line program of m is called the straight-line complexity of m.

An integer n has a straight-line complexity at most $2 \log n$. In some cases, a straight-line program is a very compact description of an integer. It can represent a huge number in small length. For example, the number n^m can be computed using the repeated squaring technique and hence has a straight-line complexity at most $2\log n + 2\log m$.

Definition 2. Let u be a real number. An integer a is ultimately u-computable, if there exists a nonzero integer sequence m such that am can be computed by a straight-line program of length at most u. The smallest u is called the ultimate complexity of a. Let f be a function in $\mathbf{R} \to \mathbf{R}$. A sequence of integers a_n is ultimately f-computable, if for any n, there exists a nonzero integer m_n such that $a_n m_n$ is ultimately $f(n)$-computable.

In this paper, we study the ultimate complexity of $n!$. First we show that this problem can be reduced to studying the ultimate complexity of the product of primes up to n.

Lemma 1. *Let p_n be the nth prime number. If the sequence $\alpha_n = p_1 p_2, \ldots, p_m$, where p_m is the largest prime less than or equal to n, can be ultimately computed by a straight-line program of length $f(n)$, then $n!$ can be ultimately computed by a straight-line program of length $f(n) + 2\log n$.*

Proof. This follows from a simple fact that $n! | (\alpha_n)^n$. Note that the exponent n is the minimum possible. $\square$

Now we prove a lemma about bipartite graphs. Given a bipartite graph $G = (X \cup Y, E)$ ($E \subseteq X \times Y$), we say that a subset $A \subseteq X$ dominates a subset $B \subseteq Y$, if every vertex in B is adjacent to at least one vertex in A.

Lemma 2. *For a simple undirected bipartite graph $G = (X \cup Y, E)$, let $m = |X|$ and $n = |Y|$. If every vertex in X has degree greater than $d = \lceil n/r \rceil$ where $2 < r < n/(2\log m)$, then there exists a subset $S \subseteq Y$, with cardinality $g = \lceil 2r \log m \rceil$, which dominates X. Moreover, if we randomly choose a subset of Y with cardinality g, it dominates X with probability greater than $1 - (1/m)$.*

Proof. From $X \times Y$, we construct a new bipartite graph $X \times \mathcal{Y}$ as follows. $\mathcal{Y}$ is the set of all the subsets of Y with g elements. For any $u \in X$ and $v \in \mathcal{Y}$, u and v are joined by an edge iff in $X \times Y$, u is adjacent to at least one vertex in $v \subseteq Y$.

For every $u \in X$, its degree in $X \times \mathcal{Y}$ is greater than $\binom{n}{g} - \binom{n-d}{g}$. The total number of edges in $X \times \mathcal{Y}$ is thus greater than $m(\binom{n}{g} - \binom{n-d}{g})$. The average degree of elements in $\mathcal{Y}$ is greater than

$$\frac{m\left(\binom{n}{g} - \binom{n-d}{g}\right)}{\binom{n}{g}} = m\left(1 - \binom{n-d}{g} \Big/ \binom{n}{g}\right).$$

We have

$$\binom{n-d}{g} \Big/ \binom{n}{g} = \frac{(n-d)!/(n-d-g)!}{n!/(n-g)!}$$

$$
\begin{aligned}
&= \frac{(n-d)(n-d-1)\cdots(n-d-g+1)}{n(n-1)\cdots(n-g+1)} \\
&< \left(1-\frac{d}{n}\right)^{g} < \left(1-\frac{1}{r}\right)^{2r\log m} \\
&< \frac{1}{m^2}.
\end{aligned}
$$

Suppose that $x|\mathcal{Y}|$ vertices in $\mathcal{Y}$ have degree less than m. The average degree of vertices in $\mathcal{Y}$ is less than $m(1-x)+(m-1)x = m-x$. Hence $m-x > m(1-(1/m^2))$. This implies that $x < 1/m$. $\square$

This lemma will be used in several places in the paper. First we present a simple consequence of the lemma.

Corollary 1. *Let p be a prime. If we randomly pick $n = \lceil 6\log p\rceil$ integers $a_1, a_2, \ldots, a_n$ between 2 and p inclusive, then with probability at least $1 - 2\log p/p$, for every prime q, $2 < q \leqslant p$, at least one of integers in $\{a_1, a_2, \ldots, a_n\}$ is a quadratic nonresidue modulo q.*

Proof. For every prime q, $2 < q \leqslant p$, at most $p/3$ of the integers between 2 and p inclusive have prime factor q. For the rest of integers, half of them are quadratic nonresidues modulo q. Hence at least $p/3$ of the integers in the same range are quadratic nonresidues modulo q. By replacing r with 3 in Lemma 2 we obtain the corollary, as there are $(1+\varepsilon)(p/\log p)$ primes less than p. $\square$

3. Elliptic curves

An elliptic curve is a smooth cubic curve. Let k be a field. If the characteristic of k is not 2 or 3, we may assume that the elliptic curve is given by an equation of the form

$$y^2 = x^3 + ax + b, \quad a, b \in k.$$

The discriminant of this curve is defined as $-16(4a^3 + 27b^2)$, whose essential part is the discriminant of the polynomial $x^3 + ax + b$. It should be non-zero as the curve is smooth. For detailed information about elliptic curves, we refer to Silverman's book [13].

The set of points on an elliptic curve consists of the solution set of the definition equation plus a point at infinity. These points form an abelian group with the infinity point as the identity. We call a point *a torsion* if it has a finite order in the group. The abscissa of the torsions of order $n > 3$ are the solutions of $P_n^{\mathcal{E}}(x)$, the nth division polynomial of $\mathcal{E}$. Sometimes we omit the superscription $\mathcal{E}$ if no confusion is possible. These polynomials can be computed recursively as follows:

$$
\begin{aligned}
P_1 &= 1, \\
P_2 &= 1, \\
P_3 &= 3x^4 + 6ax^2 + 12bx - a^2,
\end{aligned}
$$

$$\begin{aligned}
P_4 &= 2(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3),\\
P_{4n+1} &= 16(x^3 + ax + b)P_{2n+2}P_{2n}^3 - P_{2n-1}P_{2n+1}^3,\\
P_{4n+2} &= P_{2n+1}(P_{2n+3}P_{2n}^2 - P_{2n-1}P_{2n+2}^2),\\
P_{4n+3} &= P_{2n+3}P_{2n+1}^3 - 16(x^3 + ax + b)P_{2n}P_{2n+2}^3,\\
P_{4n+4} &= P_{2n+2}(P_{2n+4}P_{2n+1}^2 - P_{2n}P_{2n+3}^2).
\end{aligned}$$

We have

Proposition 2. *For any positive integers n and x, the integer $P_n^{\mathcal{E}}(x)$ can be computed by a straight-line program of length* $\mathrm{O}(\log n + \log(|x| + 1) + \log(|a| + 1) + \log(|b| + 1))$, *where $\mathcal{E}$ is the elliptic curve $y^2 = x^3 + ax + b$ with $a, b \in \mathbf{Z}$.*

See [6] for the proof of (a stronger version of) the proposition. It is based on the ideas of repeated doubling and dynamical programming.

Proposition 3. *Let $\mathcal{E} : y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathbf{Z}$. Assume that p does not divide the discriminant. If x is an integer and*
(1) *x mod p is the abscissa of a point on $E(\mathbf{F}_p)$,*
(2) *the point $(x, \sqrt{x^3 + ax + b})$ is not a torsion on $\mathcal{E}$,*
then $P_l(x) \neq 0$ and $p|P_l(x)$, where l is any non-zero multiple of $|E(\mathbf{F}_p)|$.

Proof. For any l, $P_l(x) \neq 0$ since x is not a torsion. We have $p|P_l(x)$, since the point with x mod p as its abscissa has order dividing $|E(\mathbf{F}_p)|$ and l. $\square$

Let $\mathcal{E}: y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathbf{Z}$. The torsion points on $\mathcal{E}$ with integral abscissa (thus y-coordinates are integers or quadratic algebraic numbers) have order at most 18, as shown in the celebrated Uniform Boundedness Theorem in the quadratic number fields [7,8]. Hence such integers must be the roots of some $P_n(x)$ where $n \leqslant 18$, or of $x^3 + ax + b$. The maximal possible roots of those equations are bounded by the sum of the degrees of the equations, which is an absolute constant. Let B denote this constant. One can take $B = 1035$. Define

$$R_{\mathcal{E}}(p) = \{x | x \in \mathbf{Z}, 1 \leqslant x \leqslant p, (x, \sqrt{x^3 + ax + b}) \text{ is not a torsion on } \mathcal{E}\}.$$

Then $|R_{\mathcal{E}}(p)| \geqslant p - B$. Given an integer, we can decide whether the integer is in $R_{\mathcal{E}}(p)$ in polynomial time. From Lemma 2, we conclude

Corollary 2. *Let p be a prime and $\mathcal{E} : y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathbf{Z}$ with $1 \leqslant a \leqslant p - 1$ and $1 \leqslant b \leqslant p - 1$. If $n = \lceil 6 \log p \rceil$ integers $x_1, x_2, \ldots, x_n$ are randomly chosen from $R_{\mathcal{E}}(p)$, then with probability greater than $1 - 2\log p/p$, for any prime q satisfying $7B < q \leqslant p$ and $q \nmid 4a^3 + 27b^2$, one of x_i mod q is the abscissa of a point on the reduction of $\mathcal{E}$ at q.*

Proof. We construct a bipartite graph $P \times Y$ as follows. The set P consists of all the prime numbers from $7B$ to p which are not the prime factors of $4a^3 + 27b^2$. Let $Y = R_{\mathcal{E}}(p)$. For any $q \in P$ and $x \in Y$, draw an edge between q and x iff $x^3 + ax + b$ is a quadratic residue modulo q.

For a prime q, there are at least $q - 2\sqrt{q} + 1$ many points on the reduction of $\mathcal{E}$ at q. Hence there are at least $q - 2\sqrt{q}/2$ many x between 1 and q inclusive such that $x^3 + ax + b$ are quadratic residues modulo q. Among them, $(q - 2\sqrt{q}/2) - B$ many are not torsion points. To count such elements between 1 and p inclusive, we need to multiple the number by $\lfloor p/q \rfloor$. Thus the degree of q in P is greater than $((q - 2\sqrt{q}/2) - B) \times \lfloor p/q \rfloor > q/3 \times \lfloor p/q \rfloor = p/3$ for $q > 7B$. The theorem now follows from Lemma 2. $\square$

The j-invariant of the curve $y^2 = x^3 + ax + b$ is defined as $j = 1728(4a^3/(4a^3 + 27b^2))$. Two elliptic curves with a same j-invariant are isomorphic over the algebraic closed field. For elliptic curves defined over a prime finite field $\mathbf{F}_p$ where $p > 3$, two curves with a same j-invariant may not be isomorphic. If $j \neq 0$ or 1728, there are exactly two isomorphic classes which have the same j-invariant, one can be represented by $y^2 = x^3 + kx + k$ and the other by $y^2 = x^3 + c^2kx + c^3k$, where $k = 27j/4(1728 - j)$ and c is a quadratic nonresidue modulo p. There are different number of points over the two classes of curves. There are at most 6 isomorphic classes with $j = 0$, and at most 4 isomorphic classes with $j = 1728$.

We are interested in counting the number of isomorphic classes of elliptic curves with the number of points coming from a given set. In [10], the following proposition was proved.

Proposition 4. *There exist two constants c_1, c_2 such that if A is a set of integers between $p + 1 - \sqrt{p}$ and $p + 1 + \sqrt{p}$, the number of non-isomorphic classes of elliptic curves defined over $\mathbf{F}_p$ whose number of points over $\mathbf{F}_p$ are in A is*

$$c_1\sqrt{p}(|A| - 2)/\log p \leqslant N \leqslant c_2\sqrt{p}|A| \log p(\log\log p)^2.$$

4. Proof of the main theorem

Our goal is to construct a straight-line program of some multiple of $\alpha_p = 2 \times 3 \times 5 \times \cdots \times p$ in $L_p[c_1]$ time for some constant c_1. Firstly, we compute a number $S = 2^{e_1} \times 3^{e_2} \times \cdots \times p_s^{e_s}$, where p_s is the maximal prime less than or equal to $L_p[1]$ and for every $1 \leqslant i \leqslant s$, $p_i^{e_i}$ is the least p_i-power greater than $p + 1 + 2\sqrt{p}$. Obviously we can compute S in time $L_p[2 + o(1)]$.

Secondly, we randomly choose $l = \lceil 6 \log p \rceil$ integers $c_1, c_2, \ldots, c_l$ between 2 and p inclusive. We call the step successful if for every prime $2 < q \leqslant p$, at least one of the integers is a quadratic nonresidue mod q. The step succeeds with probability greater than $1 - (2\log p/p)$ according to Lemma 1.

Denote by $\mathbf{D}$ the set of elliptic curve $\{y^2 = x^3 + ax + a | 1 \leqslant a \leqslant p\} \cup \{y^2 = x^3 + ac_i^2x + ac_i^3 | 1 \leqslant i \leqslant l, 1 \leqslant a \leqslant p\}$. Construct a bipartite graph $X \times \mathbf{D}$ as follows. X consists of all the primes between $7B + 1$ and p inclusive. For any prime $q \in X$ and any elliptic curve $\mathcal{E} \in \mathbf{D}$, connect q and $\mathcal{E}$ by an edge iff the reduction curve E of $\mathcal{E}$ at q is non-singular, and the order of $E(\mathbf{F}_q)$ is $L_p[1]$-smooth.

Lemma 3. *The degree of every element in X is greater than $pL_p[-\frac{1}{2}+\mathrm{o}(1)]$ under Conjecture 1.*

Proof. For any prime $7B < q \leqslant p$, consider the subset of $\mathbf{D}$:

$$\mathbf{D}_q = \{y^2 = x^3 + ax + a \mid 1 \leqslant a \leqslant q\} \\ \cup \{y^2 = x^3 + ac_i^2 x + ac_i^3 \mid 1 \leqslant i \leqslant l,\ 1 \leqslant a < q\}.$$

The j-invariants of $y^2 = x^3 + ax + a$ and $y^2 = x^3 + ac_i^2 x + ac_i^3$ are $1728(4a/(4a+27))$. If one of integers in $\{c_1, c_2, \ldots, c_n\}$ is a quadratic nonresidue modulo q, then there exist representations of all the isomorphic classes of elliptic curves over $\mathbf{F}_q$ in $\mathbf{D}_q$, except for the curves with j-invariants 0 or 1728. There are at least $\sqrt{q}/L_q[\frac{1}{2}+\mathrm{o}(1)]$ many $L_q[1]$-smooth integers between $q - 2\sqrt{q} + 1$ and $q + 2\sqrt{q} + 1$ according to Conjecture 1. Hence there are at least $\sqrt{q}\,(\sqrt{q}/(L_q[1/2+\mathrm{o}(1)])) = q/(L_q[\frac{1}{2}+\mathrm{o}(1)])$ curves in $\mathbf{D}_q$ which have $L_q[1]$-smooth orders over $\mathbf{F}_q$ according to Proposition 4. In the set $\mathbf{D}$, we need to multiply this number by $\lfloor p/q \rfloor$, i.e. there are at least $q/L_q[\frac{1}{2}+\mathrm{o}(1)]\lfloor p/q \rfloor > p/L_p[\frac{1}{2}+\mathrm{o}(1)]$ curves in $\mathbf{D}$ have $L_p[1]$-smooth order over $\mathbf{F}_q$. Hence the degree of q in $X \times \mathbf{D}$ is greater than $p/L_p[\frac{1}{2}+\mathrm{o}(1)]$. □

Now we proceed to the third step. We randomly choose $w = \lceil L_p[1] \rceil$ curves $\mathcal{E}_1, \ldots, \mathcal{E}_w$ from D. We call the step successful if for any prime $7B \leqslant q \leqslant p$, q does not divide the discriminant of at least one of the curves in $\{\mathcal{E}_1, \ldots, \mathcal{E}_w\}$ and the reduction of this curve at q has a $L_p[1]$-smooth order over $\mathbf{F}_q$. In the other words, in graph $X \times \mathbf{D}$, $\{\mathcal{E}_1, \ldots, \mathcal{E}_w\} \subseteq \mathbf{D}$ dominates X. Since $L_p[1] > 2\log pL_p[\frac{1}{2}+\mathrm{o}(1)]$, the step succeeds with probability at least $1 - (2\log p/p)$ according to Lemmas 2 and 3.

In the fourth step, for each $1 \leqslant i \leqslant w$, we pick $h = \lceil 6\log p \rceil$ random integers $x_{i,1}, x_{i,2}, \ldots, x_{i,h}$ in $R_{\mathcal{E}_i}(p)$. We call the ith sub-step successful, if for any prime $7B < q \leqslant p$, at least one integer in $\{x_{i,1}, x_{i,2}, \ldots, x_{i,h}\}$ modulo q is the abscissa of a $\mathbf{F}_q$-point in the reduction curve of $\mathcal{E}_i$ at q. The successful probability for each sub-step is greater than $1-(2\log p/p)$ according to Corollary 2. Hence the successfully probability for this step is greater than $(1-(2\log p/p))^w$.

Lemma 4. *All these four steps are successful with probability*

$$\left(1 - \frac{2\log p}{p}\right)^{L_p[1/2+\mathrm{o}(1)]} > \frac{1}{3}.$$

If all the four steps are successful, then we can get a multiple of α_p by evaluating the Sth division polynomials of $\mathcal{E}_1, \ldots, \mathcal{E}_w$ on $x_{1,1}, x_{1,2}, \ldots, x_{1,h}; \cdots; x_{w,1}, \ldots, x_{w,h}$, respectively and multiplying the results together. Now we are ready to write the straight-line program for a multiple of $2 \times 3 \times 5 \times \cdots \times p$.

(1) Start by computing the product of all the primes less than $7B$. Let the result be T_1.
(2) Add instructions to compute

$$P_S^{\mathcal{E}_1}(x_{1,1}), \ldots, P_S^{\mathcal{E}_1}(x_{1,h}); \cdots; P_S^{\mathcal{E}_w}(x_{w,1}), \ldots, P_S^{\mathcal{E}_w}(x_{w,h}).$$

(3) Add instructions to compute

$$T_2 \leftarrow \prod_{1 \leqslant i \leqslant w, 1 \leqslant k \leqslant h} P_S^{\mathcal{E}_i}(x_{i,k}).$$

(4) Add $T \leftarrow T_1 \times T_2$ into the straight-line program.
Based on the analysis in this paper, it can be verified that the above straight-line program computes a product of α_p and it has subexponential length.

5. Discussion

The relation between ultimate complexity and integer factorization can be further explored.

Firstly, can we derive a factorization algorithm from a straight-line program for a multiple of n!? The only problem here is that the multiple of $n!$, i.e. $n!m_n$, may contain primes greater that n. We must try to restrict the integer m_n such that it only has primes less than n. It seems hard to do so with the algorithm in this paper.

Secondly, is the lower bound of the ultimate complexity of $n!$ also subexponential? Since this problem is closely related to the integer factorization problem, which is believed not to have a polynomial time algorithm, we suspect that the answer to this question is positive.

The existence of a short straight-line program for a large number does not imply that we can construct the short straight-line program in reasonable time. Given two integers m, n and a prime p, if m mod p is the generator of $\mathbf{F}_p^*$ and $p \nmid n$, then there exists a short straight-line program for a power of m which is congruent to n modulo p. But we do not know how to construct such a straight-line program from m, n and p, as the problem is equivalent to computing the discrete logarithm problem over $\mathbf{F}_p$. We believe that it might be possible that for some n, $n!$ or a multiple of $n!$ have very short straight-line programs, however constructing the program would be very hard.

References

[1] L. Blum, F. Cucker, M. Shub, S. Smale, Complexity and Real Computation, Springer, Berlin, 1997.

[2] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machine, Bull. Amer. Math. Soc. 21 (1) (1989).

[3] P. Burgisser, The complexity of factors of multivariate polynomials, in: Proc. 42th IEEE Symp. on Foundations of Computer Science, 2001, pp. 1–46.

[4] P. Burgisser, M. Clausen, M.A. Shokrollahi, Algebraic Complexity Theory, Grundlehren der Mathematischen, Vol. 315, Springer, Berlin, 1997.

[5] E.R. Canfield, P. Erdos, C. Pomerance, On a problem of Oppenheim concerning "Factorisatio Numerorum", J. Number Theory (1983) 1–28.

[6] Q. Cheng, Some remarks on the L-conjecture, in: Proc. 13th Annu. Internat. Symp. on Algorithms and Computation (ISAAC), Lecture Notes in Computer Science, Vol. 2518,Springer, Berlin, 2002.

[7] S. Kamienny, Torsion points on elliptic curves and q-coefficients of modular forms, Invent. Math. 109 (1992) 221–229.

[8] M. Kenku, F. Momose, Torsion points on elliptic curves defined over quadratic fields, Nagoya Math. J. 109 (1988) 125–149.

[9] A. Lenstra, H.W. Lenstra Jr., Handbook of Theoretical Computer Science A, Algorithms in Number Theory, Elsevier and MIT Press, Amsterdam, 1990, pp. 673–715.
[10] H.W. Lenstra, Factoring integers with elliptic curves, Ann. Math. 126 (1987) 649–673.
[11] A. Shamir, Factoring numbers in $O(\log n)$ arithmetic steps, Inform. Process. Lett. 1 (1979) 28–31.
[12] M. Shub, S. Smale, On the intractability of Hilbert's nullstellensatz and an algebraic version of "$\mathrm{P} = \mathrm{NP}$?", Duke Math. J. 81 (1995) 47–54.
[13] J.H. Silverman, The Arithmetic of Elliptic Curves, Springer, Berlin, 1986.
[14] V. Strassen, Einige resultate uber berechnungskomplexitat, Jber. Deutsch. Math.-Verein 78 (1) (1976/77) 1–8.