Journal of Mathematics and Statistics

Original Research Paper

Fast Approach to Factorize Odd Integers with Special Divisors

[1,2,3]Xingbo Wang and [1]Junjian Zhong

[1]*Department of Mechatronic Engineering, Foshan University, Foshan City, PRC, 528000, China*
[2]*State Key Laboratory of Information Security,*
Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[3]*Guangdong Engineering Center of Information Security for Intelligent Manufacturing System, China*

Article history
Received: 04-12-2019
Revised: 13-01-2020
Accepted: 25-01-2020

Corresponding Authors:
Xingbo Wang
Department of Mechatronic Engineering, Foshan University, Foshan City, PRC, 528000, China
Email: 153668@qq.com

Abstract: The paper proves that an odd composite integer N can be factorized in $O((\log_2 N)^4)$ bit operations if $N = pq$, the divisor q is of the form $2^\alpha u+1$ or $2^\alpha u-1$ with u being an odd integer and α being a positive integer and the other divisor p satisfies $1 < p \le 2^\alpha+1$ or $2^\alpha+1 < p \le 2^{\alpha+1}-1$. Theorems and corollaries are proved with detail mathematical reasoning. Algorithm to factorize the odd composite integers is designed and tested in Maple. The results in the paper demonstrate that fast factorization of odd integers is possible with the help of valuated binary tree.

Keywords: Cryptography, Integer Factorization, Binary Tree, Algorithm

Introduction

A Valuated Binary tree is a full perfect binary tree that has odd integers bigger than 1 put on it from top to bottom and left to right, as introduced in Wang's (2016a). With the help of the valuated binary tree, many new properties of the odd integers are discovered. For example, the properties of symmetric nodes and symmetric common divisors, the properties of subtree duplication and subtree transition and the properties of sum by level, root division and uniform sum were discovered in (Wang, 2016b; 2017a), the genetic properties of odd integers was disclosed in (Wang, 2017b) and the periodical divisibility traits along the leftmost path or the left side-path of the tree were demonstrated in (Wang and Guo, 2019). All these new properties enable us to know the integers in a different point of view, as stated and investigated in Wang's (2018). Integer factorization has been a hard problem in number theory and in cryptography over years, as overviewed in Yan's (2013), Sarnaik's *et al*. (2016) and Phulachand's (2016). Any new approach related with the integers shall of course be tried on the issue. Wang (2017b) proved that there should exist an algorithm of $O(\log_2 N)$ searching steps to factorize an odd integer N. But there has not been a convincible demonstration. Thereby, this paper, continues the studies on integer factorization and proves that there are odd integers that can be factorized in $O(\log_2 N)$ searching steps or in $O((\log_2 N)^4)$ bit operations.

Preliminaries

Definitions and Notations

A valuated binary tree T is such a binary tree that each of its nodes is assigned a value. An odd number N-rooted tree, denoted by T_N is a recursively constructed valuated binary tree whose root is the odd number N with $2N-1$ and $2N+1$ being the root's left and right sons, respectively. Each son is connected with its father via a path, but there is no path between the two sons. T_3 tree is the case $N = 3$. For convenience, symbol $N_{(k,j)}$ is by default the node at position j on level k of T_3, where $k = 1, 2,\cdots$ and $j = 0,1,\cdots, 2^k-1$. Symbol $N^N_{(k,j)}$ is to denote the node at position j on level k of T_N, where $k = 1,2,\cdots$ and $j = 0,1,\cdots,2^k-1$. Symbol $X\in l(T_N)$ means node X is in the left branch of T_N while symbol $X\in r(T_N)$ means X is in the right branch of T_N. Symbol $N^N_{(i,0)}$ is the leftmost node on level i of T_N; use symbol $N^N_{(i,-1)}$ to denote the odd number left to $N^N_{(i,0)}$, namely, $N^N_{(i,-1)} = N^N_{(i,0)} - 2$. Use symbol P^N_0 to indicate the leftmost path defined by $P^N_0 = \{N^N_{(0,0)}, N^N_{(1,0)},\cdots, N^N_{(i,0)},\cdots\}$ and symbol P^N_L to indicate the path defined by $P^N_L = \{N_{(1,-1)},\cdots, N_{(i,-1)},\cdots,\dots\}$, which is also called a left side-path, as depicted in Fig. 1. The leftmost path and the rightmost path together with their side-paths respectively are in all called border-path or simply border.

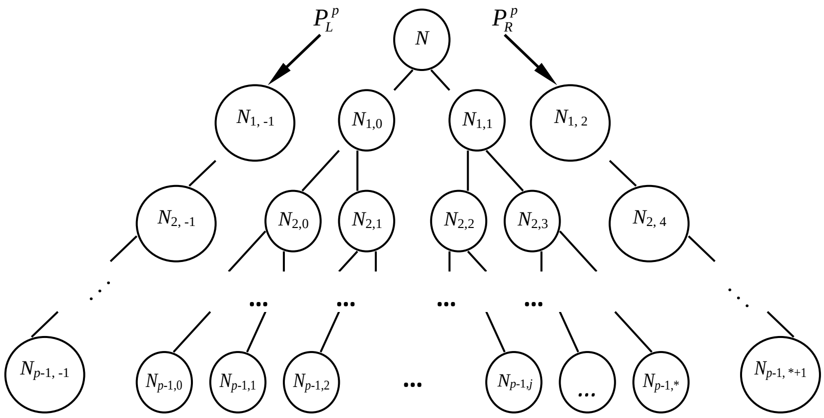

Fig. 1: T_p Tree and its side-paths

Symbol $A \Rightarrow B$ means result B is derived from condition A or A can derive B out. In this whole article, symbol $\lfloor x \rfloor$ denotes the floor function, an integer function of the real number x such that x-1 < $\lfloor x \rfloor \le x$ or equivalently $\lfloor x \rfloor \le x < \lfloor x \rfloor + 1$. Symbol $a|b$ means b can be divided by a; symbol (a, b) is to express the Greatest Common Divisor (GCD) of integers a and b. A tracing step or a searching step is the computation of a father based on a son or vice versa.

Lemmas

Lemma 1 (Node Calculation, see in (Wang, 2016a))

Node $N_{(k,j)}$ of T_3 is calculated by:

$$N_{(k,j)} = 2^{k+1} + 1 + 2j$$
$$k = 0,1,2,...; j = 0,1,...,2^k - 1$$

Node $N^X_{(k,j)}$ of T_X is computed by:

$$N^X_{(k,j)} = 2^k X - 2^k + 2j + 1$$
$$k = 0,1,2,...; j = 0,1,...,2^k - 1$$

Lemma 2 (Divisors on Borders, see in (Wang and Guo, 2019))

Let p be an odd integer and T_p be the p-rooted valuated binary tree and d be a positive integer with $1 \le d \le p$-1; if there exits a positive integer e such that $1 \le e \le 2^{d-1}$-1 and 2^d- $\underbrace{(2e-1)}_{odd} \equiv 0(\text{mod}p)$, then $p \mid N^p_{(d,e-1)}$; if there exits a positive integer f such that $0 \le p \le f \le 2^{d-1}$-2 and 2^d + $\underbrace{(2f-1)}_{odd} \equiv 0(\text{mod}p)$, then $p \mid N^p_{(d,p-f)}$. Particularly, if 2^d-1 $\equiv 0(\text{mod } p)$ then $N^p_{(d,0)} \equiv 0(\text{mod}p)$; if $2^d + 1 \equiv 0(\text{mod}p)$ then $N^p_{(d,p-1)} \equiv N^p_{(d,-1)} \equiv 0(\text{mod } p)$.

Lemma 3 (Floor Function, see in (Wang, 2019))

Properties of the floor functions with real numbers x and y and integers n:

(P1) $\lfloor x \rfloor + \lfloor y \rfloor \le \lfloor x + y \rfloor \le \lfloor x \rfloor + \lfloor y \rfloor + 1$
(P2) $\lfloor x \rfloor - \lfloor y \rfloor - 1 \le \lfloor x + y \rfloor \le \lfloor x \rfloor - \lfloor y \rfloor < \lfloor x \rfloor - \lfloor y \rfloor + 1$
(P13) $x \le y \Rightarrow \lfloor x \rfloor \le \lfloor y \rfloor$
(P32) $n \lfloor x \rfloor \le \lfloor nx \rfloor \le n(\lfloor x \rfloor + 1)$-1. Taking n = 2 yields $2\lfloor x \rfloor \le \lfloor 2x \rfloor \le 2\lfloor x \rfloor + 1$

Main Results and Proofs

Theorem 1

Let $p > 1$ be an odd integer and α be a positive integer; if $p < 2^\alpha + 1$ then it holds:

$$2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} \le 2^\alpha - 1$$
$$0 \le 2^{\alpha-1} - \frac{p+1}{2} \le 2^{\alpha-1} - 1$$

whereas if $p < 2^\alpha$-1 it holds:

$$2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} < 2^\alpha - 1$$
$$0 < 2^{\alpha-1} - \frac{p+1}{2} \le 2^{\alpha-1} - 1$$

Proof

See the following deductions:

$$p < 2^\alpha + 1 \Rightarrow \frac{p-1}{2} < 2^{\alpha-1}$$
$$\Rightarrow 2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} < 2^\alpha$$
$$\Rightarrow 2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} \le 2^\alpha - 1$$

$$p < 2^{\alpha} + 1 \Rightarrow \frac{p+1}{2} < 2^{\alpha-1} + 1$$
$$\Rightarrow -1 < 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha-1} - 2$$
$$\Rightarrow 0 \le 2^{\alpha-1} + \frac{p-1}{2} \le 2^{\alpha} - 1$$

$$p < 2^{\alpha} - 1 \Rightarrow \frac{p-1}{2} < 2^{\alpha-1} - 1$$
$$\Rightarrow 2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} < 2^{\alpha} - 1$$
$$\Rightarrow 2^{\alpha-1} < 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha} - 1$$

$$p < 2^{\alpha} - 1 \Rightarrow \frac{p+1}{2} < 2^{\alpha-1}$$
$$\Rightarrow 0 < 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha-1} - 2$$
$$\Rightarrow 0 < 2^{\alpha-1} - \frac{p+1}{2} \le 2^{\alpha-1} - 1$$

Theorem 2

Let $p > 1$ be an odd integer and α be a positive integer; if $2^{\alpha}+1 < p \le 2^{\alpha+1}-1$ then it holds:

$$2^{\alpha-1} \le 2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha} - 1$$

and:

$$0 < 2^{\alpha-1} - 2^{\alpha} + \frac{p-1}{2} \le 2^{\alpha-1} - 1$$

Proof

See the following deductions:

$$\begin{aligned}
&2^{\alpha} + 1 < p \le 2^{\alpha-1} - 1 \\
&\Rightarrow 2^{\alpha} + 2 < p + 1 \le 2^{\alpha+1} \\
&\Rightarrow 2^{\alpha-1} + 1 < \frac{p+1}{2} \le 2^{\alpha} \\
&\Rightarrow -2^{\alpha} \le -\frac{p+1}{2} < -2^{\alpha-1} - 1 \\
&\Rightarrow \underbrace{2^{\alpha} + 2^{\alpha-1}}_{added\ \ items} - 2^{\alpha} \le \underbrace{2^{\alpha} + 2^{\alpha-1}}_{added\ \ items} - \frac{p+1}{2} \\
&< \underbrace{2^{\alpha} + 2^{\alpha-1}}_{added\ \ items} - 2^{\alpha-1} - 1 \\
&\Rightarrow 2^{\alpha-1} \le 2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha} - 1
\end{aligned}$$

$$\begin{aligned}
&2^{\alpha} + 1 < p \le 2^{\alpha+1} - 1 \\
&\Rightarrow 2^{\alpha} < p - 1 \le 2^{\alpha+1} - 2 \\
&\Rightarrow 2^{\alpha-1} < \frac{p-1}{2} \le 2^{\alpha} - 1 \\
&\Rightarrow \underbrace{2^{\alpha-1} - 2^{\alpha}}_{added\ \ items} + 2^{\alpha-1} < \underbrace{2^{\alpha-1} - 2^{\alpha}}_{added\ \ items} + \frac{p-1}{2} \\
&\le \underbrace{2^{\alpha-1} + 2^{\alpha}}_{added\ \ items} + 2^{\alpha} - 1 \\
&\Rightarrow 0 < 2^{\alpha-1} - 2^{\alpha} + \frac{p-1}{2} < 2^{\alpha-1} - 1
\end{aligned}$$

Theorem 3

Let $N = pq$ with $1 < p \le q$ being odd integers; then $\lfloor \log_2 N \rfloor \ge \max(2\lfloor \log_2 p \rfloor, \lfloor \log_2 q \rfloor)$.

Proof

Without loss of generality, assume $1 < p \le \sqrt{N} \le q$. Then By Lemma 3 (P13) and (P32):

$$\log_2 N > \log_2 q \Rightarrow \lfloor \log_2 N \rfloor \ge \lfloor \log_2 q \rfloor$$

and:

$$\log_2 N \ge 2\log_2 p \Rightarrow \lfloor \log_2 N \rfloor \ge \lfloor 2\log_2 p \rfloor \ge 2\lfloor \log_2 p \rfloor$$

Hence it holds:

$$\lfloor \log_2 N \rfloor \ge \max\left(2\lfloor \log_2 p \rfloor, \lfloor \log_2 q \rfloor\right)$$

Corollary 1

Suppose p and q are odd integers with $1 < p < q$; then $N = pq$ can be factorized in $\lfloor \log_2 N \rfloor + 1$ searching steps if one of p and q is in the form $2^{\alpha}+1$ or $2^{\alpha}-1$ with α being a positive integer.

Proof

According to the given conditions, there are 4 cases, $q = 2^{\alpha}+1$, $q = 2^{\alpha}-1$, $p = 2^{\alpha}+1$ and $p = 2^{\alpha}-1$, to be considered.

Consider the first case $q = 2^{\alpha}+1$; then $N = 2^{\alpha}p + p$. Rewrite this by:

$$N = 2^{\alpha}p - 2^{\alpha} + 2^{\alpha} + p = 2^{\alpha}p - 2^{\alpha} + 2\left(2^{\alpha-1} + \frac{p-1}{2}\right) + 1$$

Referring to Lemma 1 and Theorem 1, it yields:

$$N = N^{p}_{\left(\alpha, 2^{\alpha-1} + \frac{p-1}{2}\right)}$$

This implies that N is a node in the right branch of T_p. Consequently, there are at most α steps by tracing upwards and finding out the GCD between N and its ancestors in T_p. Since $q = 2^\alpha+1$, it yields:

$$\alpha = \lfloor \log_2 (q-1) \rfloor \le \log_2 q < \lfloor \log_2 q \rfloor + 1 \tag{1}$$

For the case $q = 2^\alpha-1$, it holds $N = 2^\alpha p - p = 2^\alpha p - 2^\alpha + 2\left(2^{\alpha-1} - \frac{p+1}{2}\right) + 1$. Again referring to Theorem 1, it leads to $N = N^p_{\left(\alpha, 2^{\alpha-1}-\frac{p+1}{2}\right)}$. This case says N is a node in the left branch of T_p.

For the case $p = 2^\alpha+1$ or $p = 2^\alpha-1$, by Lemma 2, it knows $N^p_{(\alpha,-1)} \equiv 0 (\mathrm{mod}\, p)$ or $N^p_{(\alpha,0)} \equiv 0 (\mathrm{mod}\, p)$ respectively. Since $\alpha \le \lfloor \log_2 p \rfloor + 1$, by genetic property it knows p can be found in at most $2\lfloor \log_2 p \rfloor + 1$ steps by tracing downwards and finding the GCD between N and nodes along the leftmost path or left side-path of T_N.

Example 1

Let $N = 527$; then N's ancestors are 263,131, 65,33 and 17, as depiected with Fig. 2. It can see that 17 is the divisor of $527 = 17\times31$ and $31 = 2^5-1$.

Example 2

Let $N = 561$, then N's ancestors are 281,141,71,35 and 17, as depiected with Fig. 3. It can see that 17 is the divisor of $561 = 17\times33$ and $33 = 2^5+1$.

Proposition 1

Suppose p and q are odd integers with $1 < p < q$; then $N = pq$ can be factorized in $\lfloor \log_2 N \rfloor + 1$ searching steps if q is in either form of $2^\alpha-1$ and $2^\alpha+1$ with α being a positive integer.

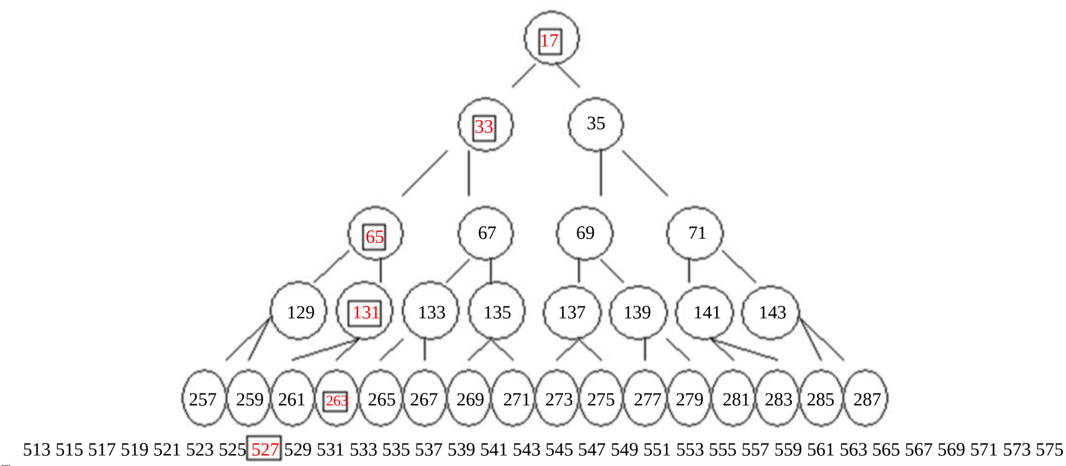

Fig. 2: The ancestors of $N = 527$ in T_{17}

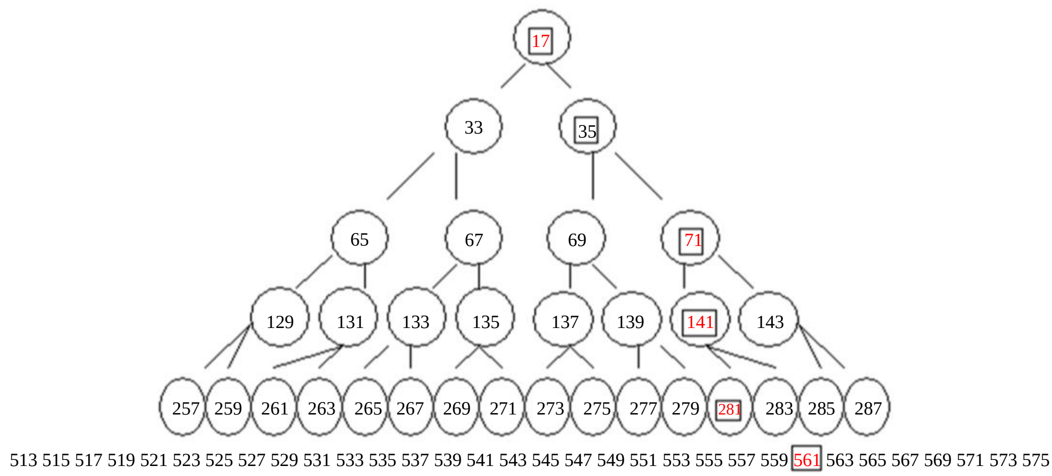

Fig. 3: The ancestors of $N = 561$ in T_{17}

DOI: 10.3844/jmssp.2020.24.34

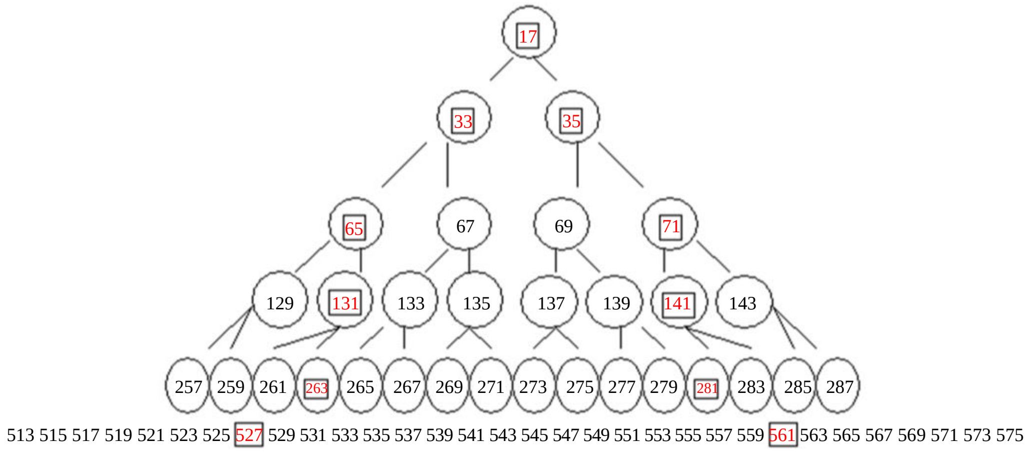

Fig. 4: Symmetric divisors with q = 31 and q = 33

Example 3

Figure 4, symmetric divisors distributed in a tree are again exhibited with q in the form 2^{α}-1 or 2^{α} +1.

Example 4

Let N = 731; then the left side-path of T_{731} is 1459, 2919, 5839 and 11679, as depicted in Fig. 5. It can see GCD(11679,731) = 17. Likewise, the right side path is 1465, 2929, 5857 and 11713, among which it fits GCD(11713,731) = 17.

Corollary 2

Let p and q be odd integers with $1 < p < q$ and suppose $q = 2^{\alpha}u + 1$ with $u \geq 1$ being an old integer, α being a positive integer and $1 < p < 2^{\alpha} + 1$; then $N = pq$ can be factorized in $\lfloor \log_2 p \rfloor + 1$ searching steps.

Proof

The condition $q = 2^{\alpha}u + 1$ leads to:

$$N = \left(2^{\alpha}u + 1\right)p = 2^{\alpha}up - 2^{\alpha} + 2^{\alpha} + p$$
$$= 2^{\alpha}up - 2^{\alpha} + 2\left(2^{\alpha-1} + \frac{p-1}{2}\right) + 1$$

Since $1 < p < 2^{\alpha} + 1$, it knows by 1, $2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} \leq 2^{\alpha}-1$. Thereby:

$$N = N^{up}_{\left(\alpha, 2^{\alpha-1} + \frac{p-1}{2}\right)}$$

This says that N is a node in the right branch of T_{up}. Thus there are at most α searching steps to trace upwards and find out the GCD between N and its ancestors in T_{up}.

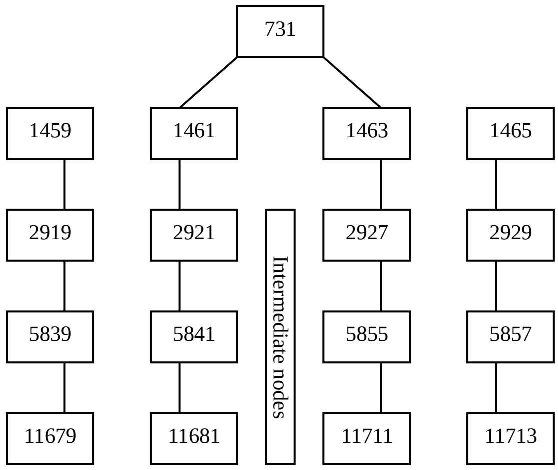

Fig. 5: Side-paths and border-path of T_{731}

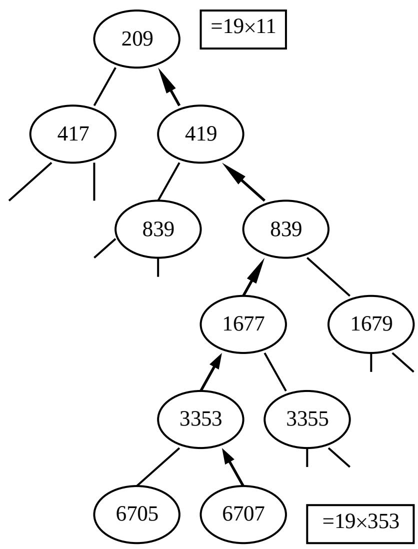

Fig. 6: Tracing ancestors of 6707

Example 5

Let N = 6707; then N's ancestors are 3353, 1677, 839, 419, 209, among which GCD(6707,209) = 19, which results in 6707 = 19×353 = 19×(2^5×11+1). Figure 6 shows the tracing path from 6707 to 209. Seen from the figure, N = 6707 is sure in the right branch of T_{209}.

Corollary 3

Let p and q be odd integers with $1 < p < q$ and suppose $q = 2^\alpha u$-1 with $u \geq 1$ being an old integer, α being a positive integer and $1 < p < 2^\alpha + 1$; then $N = pq$ can be factorized in $\lfloor \log_2 p \rfloor + 1$ searching steps.

Proof

By Theorem 1, the condition $1 < p < 2^\alpha + 1$ leads to $0 \leq 2^{\alpha-1} - \frac{p+1}{2} \leq 2^{\alpha-1} - 1$. Considering:

$$N = \left(2^\alpha u - 1\right) p = 2^\alpha up - 2^\alpha + 2^\alpha - p$$
$$= 2^\alpha up - 2^\alpha + 2\left(2^{\alpha-1} - \frac{p+1}{2}\right) + 1$$

it knows:

$$N = N^{up}_{\left(\alpha, 2^{\alpha-1} - \frac{p+1}{2}\right)}$$

This says that N is a node in the left branch of T_{up}. Thus there are at most α searching steps to trace upwards and find out the GCD between N and its ancestors in T_{up}.

Example 6

Let N = 45601; then N's ancestors are 22801, 11401, 5701, 2851, 1425, 713, among which GCD(45601,713) = 31, which results in 45601 = 31×1471 = 31×(2^6×23-1). Figure 7 shows the tracing path from 45601 to 713. Seen from the figure, N = 45601 is sure in the left branch of T_{713}.

Corollary 4

Let p and q be odd integers with $1 < p < q$ and suppose $q = 2^\alpha u + 1$ with $u \geq 1$ being an old integer, α being a positive integer and $1 < p < 2^\alpha$-1; then $N = pq$ can be factorized in $\lfloor \log_2 p \rfloor + 1$ searching steps.

Proof

By Theorem 1, the condition $1 < p < 2^\alpha - 1$ leads to $2^{\alpha-1} < 2^{\alpha-1} + \frac{p+1}{2} < 2^\alpha - 1$. Since:

$$N = \left(2^\alpha u + 1\right) p = 2^\alpha up - 2^\alpha + 2^\alpha + p$$
$$= 2^\alpha up - 2^\alpha + 2\left(2^{\alpha-1} + \frac{p-1}{2}\right) + 1$$
$$= N^{up}_{\left(\alpha, 2^{\alpha-1} + \frac{p-1}{2}\right)}$$

it knows that N is a node in the right branch of T_{up}. Thus there are at most α searching steps to trace upwards and find out the GCD between N and its ancestors in T_{up}.

Example 7

Let N = 42711; then N's ancestors are 21355, 10677, 5339, 2669, 1335, 667, among which GCD(42711,667) = 23, which results in 42711 = 23×1857 = 23×(2^6×29+1). Figure 8 shows the tracing path from 42711 to 667. Seen from the figure, N = 42711 is sure in the right branch of T_{667}.

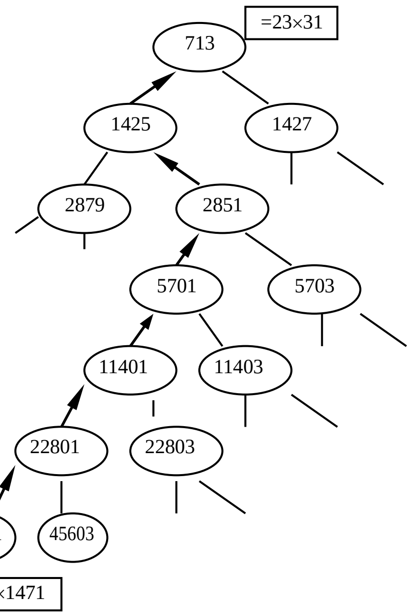

Fig. 7: Tracing ancestors of 45601

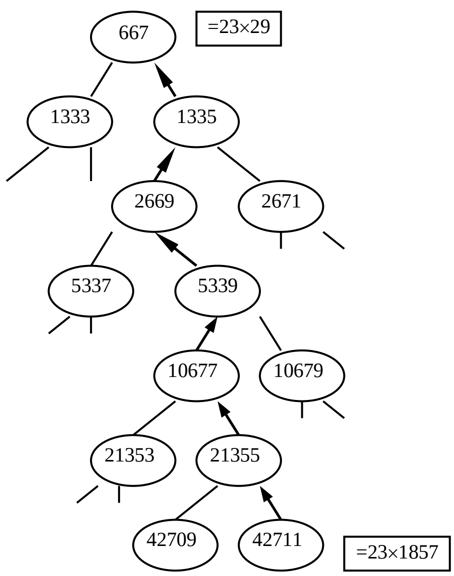

Fig. 8: Tracing ancestors of 42711

Corollary 5

Let p and q be odd integers with $1 < p < q$ and suppose $q = 2^{\alpha}u$-1 with $u \geq 1$ being an old integer, α being an positive integer and $1 < p < 2^{\alpha}$ -1; then $N = pq$ can be factorized in $\lfloor \log_2 p \rfloor$ +1 searching steps.

Proof

By Theorem 1, the condition $1 < p < 2^{\alpha}$ -1 leads to $0 < 2^{\alpha-1} - \frac{p+1}{2} \leq 2^{\alpha-1}$ -1. Since:

$$\begin{aligned} N &= \left(2^{\alpha}u-1\right)p = 2^{\alpha}up - 2^{\alpha} + 2^{\alpha} - p \\ &= 2^{\alpha}up - 2^{\alpha} + 2\left(2^{\alpha-1} - \frac{p+1}{2}\right) + 1 \\ &= N^{up}_{\left(\alpha, 2^{\alpha-1} - \frac{p+1}{2}\right)} \end{aligned}$$

it knows that N is a node in the left branch of T_{up}. Thus there are at most α searching steps to trace upwards and find out the GCD between N and its ancestors in T_{up}.

Example 8

Let N = 383031; then N's ancestors are 191515, 95757, 47879, 23939, 11969, 5985 and 2993, among which GCD(383031, 2993) = 73, which results in Figure 9 shows the tracing path from 383031 to 2993. Seen from the figure, N = 383031 is sure in the left branch of T_{2993}.

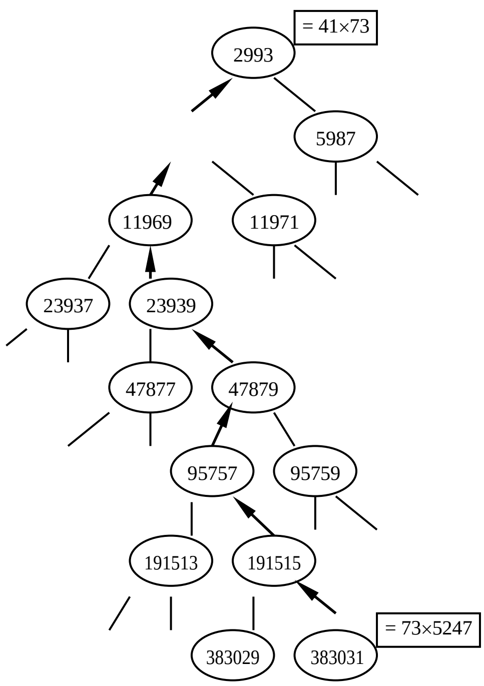

Fig. 9: Tracing ancestors of 383031

Theorem 4

Let $N = pq$ be an odd integer with p and q being odd integers and $1 < p < q$; suppose $q = 2^{\alpha}u \pm 1$ with $u \geq 1$ being an old integer, and $1 < p \leq 2^{\alpha} \pm 1$; then N can be factorized in $\lfloor \log_2 N \rfloor$ +1 steps or in $O((\log_2 N)^4)$ bit operations.

Proof

Let $J_1 = 2^{\alpha-1} - \frac{p+1}{2}$ and $J_2 = 2^{\alpha-1} + \frac{p-1}{2}$; summarizing Corollaries 1 to 5 yields Table 1.

Seen from the table and referring to the Corollaries 1 to 5, it knows the theorem holds considering it needs $O((\log_2 N)^4)$ bit operations in computation of the GCD at each step.

Corollary 6

Let $N = pq$ be an odd integer with p and q being odd integers and $1 < p < q$; suppose $q = 2^{\alpha}u$-1 with $u \geq 1$ being an old integer and α being an positive integer; if $2^{\alpha} + 1 < p \leq 2^{\alpha+1}$ -1 then N can be factorized in $\lfloor \log_2 N \rfloor$ +1 searching steps.

Proof

Direction calculation yields:

$$\begin{aligned} N &= \left(2^{\alpha}u-1\right)p = 2^{\alpha}up - p \\ &= 2^{\alpha}\left(up-2\right) - 2^{\alpha} + 2\left(2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2}\right) + 1 \\ &= 2^{\alpha}\left(\left(\underline{up-2}\right)-1\right) + 2\left(2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2}\right) + 1 \end{aligned}$$

Let $n = up$-2; by Theorem 2, $2^{\alpha-1} \leq 2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2} < 2^{\alpha}$ -1; consequently:

$$N = N^{n}_{\left(\alpha, 2^{\alpha} + 2^{\alpha-1} - \frac{p+1}{2}\right)}$$

That is to say, tracing upwards from N by α steps will reach n, the node left to up; then:

$$p = GCD\left(n+2, N\right)$$

The relations described in Corollary 6 among n, N and up are illustrated in Figure 10.

Corollary 7

Let $N = pq$ be an odd integer with p and q being odd integers and $1 < p < q$; suppose $q = 2^{\alpha}u$ +1 with $u \geq 1$ being an old integer and α being an positive integer; if $2^{\alpha} + 1 < p \leq 2^{\alpha+1}$ -1 then N can be factorized in $\lfloor \log_2 N \rfloor$ +1 searching steps.

Table 1: Summarized cases from Corollaries 1 to 5

q	p	J	N	
$q = 2^{\alpha}u\text{-}1$	$1 < p \le 2^{\alpha}\text{-}1$	$0 < 2^{\alpha-1}\text{-}\ \frac{p+1}{2} \le 2^{\alpha-1}\text{-}1$	$N = N^{up}_{\left(\alpha, 2^{\alpha-1}-\frac{p+1}{2}\right)}$	$N \in l(T_{up})$
	$1 < p \le 2^{\alpha}+1$	$0 \le 2^{\alpha-1}\text{-}\ \frac{p+1}{2} \le 2^{\alpha-1}\text{-}1$		
$q = 2^{\alpha}u+1$	$1 < p \le 2^{\alpha}\text{-}1$	$2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} < 2^{\alpha-1}\text{-}1$	$N = N^{up}_{\left(\alpha, 2^{\alpha-1}+\frac{p-1}{2}\right)}$	$N \in r(T_{up})$
	$1 < p \le 2^{\alpha}+1$	$2^{\alpha-1} < 2^{\alpha-1} + \frac{p-1}{2} \le 2^{\alpha-1}\text{-}1$		

Proof

Direction calculation yields:

$$\begin{aligned} N &= \left(2^{\alpha}u+1\right)p = 2^{\alpha}up + p \\ &= 2^{\alpha}up + 2^{\alpha+1} - 2^{\alpha} - 2^{\alpha+1} + 2^{\alpha} + p \\ &= 2^{\alpha}up + 2^{\alpha+1} - 2^{\alpha} + 2\left(2^{\alpha-1} - 2^{\alpha} + \frac{p-1}{2}\right) + 1 \\ &= 2^{\alpha}\left(\left(\underline{up+2}\right) - 1\right) + 2\left(2^{\alpha-1} - 2^{\alpha} + \frac{p-1}{2}\right) + 1 \end{aligned}$$

Let $n = up + 2$; by Theorem 2, $0 < 2^{\alpha-1} - 2^{\alpha} + \frac{p-1}{2} \le 2^{\alpha-1} - 1$; consequently:

$$N = N^{n}_{\left(\alpha, 2^{\alpha}+2^{\alpha-1}-\frac{p+1}{2}\right)}$$

That is to say, tracing upwards from N by α steps will reach n, the node left to up; then:

$$p = GCD(n-2, N)$$

The relations described in Corollary 7 among n, N and up are illustrated in Fig. 11.

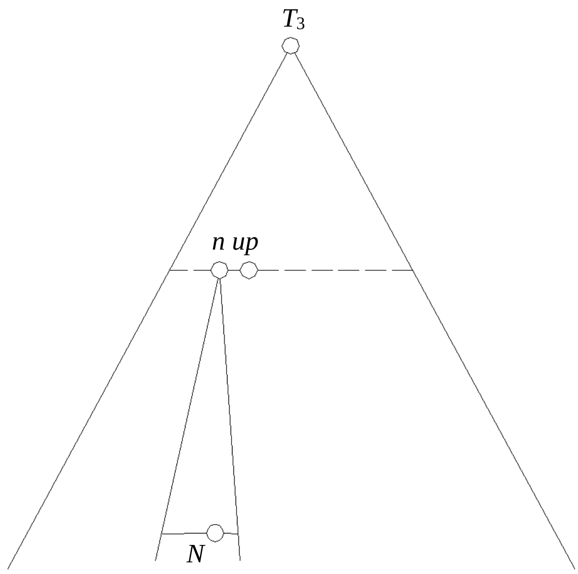

Fig. 10: Relations among n, N and up

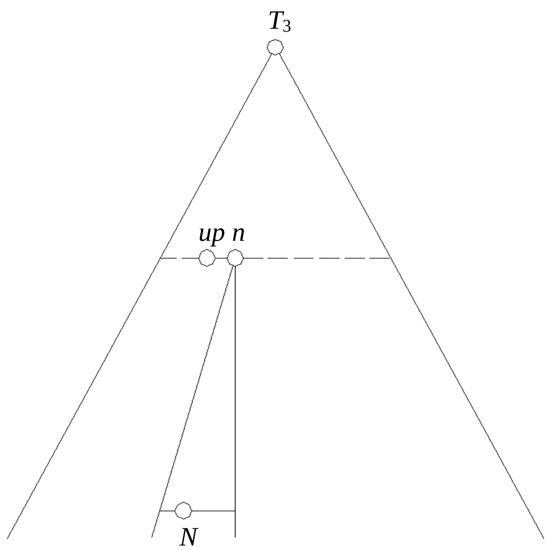

Fig. 11: Relations among n, N and up

Theorem 5

Let $N = pq$ be an odd integer with p and q being odd integers and $1 < p < q$; suppose $q = 2^{\alpha}u \pm 1$ with u being an old integer and α being an positive integer; if $2^{\alpha}+1 < p \le 2^{\alpha+1} - 1$ then N can be factorized in $3\lfloor \log_2 N \rfloor + 1$ searching steps or in $O((\log_2 N)^4)$ bit operations.

Proof

Summarizing Corollaries 6 and 7 yields to Table 2.

Table 2 shows that, N is a node of T_{up+2} or T_{up-2}. Hence it easy to trace upwards from N to $up+2$ or $up-2$ and then find out the divisor p. The time complexity is demonstrated in section 4.1.

DOI: 10.3844/jmssp.2020.24.34

Table 2: Summarized cases from Corollaries 6 and 7

p	q	J	N	
$2^{\alpha}+1 < p \le 2^{\alpha+1}-1$	$q = 2^{\alpha}u-1$	$2^{\alpha-1} \le 2^{\alpha-1}+2^{\alpha}-\frac{p+1}{2} < 2^{\alpha}-1$	$N = N^{up-2}_{\left(\alpha, 2^{\alpha}+2^{\alpha-1}-\frac{p+1}{2}\right)}$	$N \in r(T_{up-2})$
	$q = 2^{\alpha}u-1$	$0 < 2^{\alpha-1}-2^{\alpha}+\frac{p-1}{2} \le 2^{\alpha-1}-1$	$N = N^{up+2}_{\left(\alpha, \frac{p-1}{2}-2^{\alpha-1}\right)}$	$N \in l(T_{up+2})$

Table 3: Ten factorized samples

Odd Integers	Factorizaion
34639739	8191×4229
1159847279	131071×8849
10581684521	524287×20183
60782931320919664123	59649589127497217×1019
10263855667940024299	1256132134125569×8171
115271397873601774304441	2305843009213693951×49991
174538042279885450969073	2663848877152141313×65521
944515611538471874461691	3603109844542291969×262139
2732669846011417649053579	167988556341760475137×16267
5057672949897463733694209	18446744073709551617×274177

Algorithm and Numerical Experiments

Algorithm

Theorems 4 and 5 provide an approach to factorize rapidly a composite odd integer $N = pq$ if q is in the form $q = 2^{\alpha}u \pm 1$ and p satisfies $1 < p \le 2^{\alpha} \pm 1$ or $2^{\alpha}+1 < p \le 2^{\alpha+1}-1$. This section presents a factoring algorithm. The whole procedure includes two subroutines and a main routine as follows.

Algorithm 1 Father (Calculate the father of a node)

```
Input Parameters: Son;
Begin
if Son ≡ 1(mod4) then
    return (Son-1)/2;
else
    return (Son +1)/2;
end if
End
```

The main routine shows, it requires at most $3\lfloor \log_2 N \rfloor + 1$ searching steps to factorize N. Since at each searching step, it needs $O((\log_2 N)^3)$ bit operations to compute the GCD, it knows that the total computation can be completed in $O((\log_2 N)^4)$ bit operations.

Numerical Experiments with Maple 15

With the algorithm, programs in Maple are designed as list in the appendix. With the programs, ten odd integers are factorized in milliseconds in Maple. The ten numbers are list in Table 3. The biggest one is a 25 decimal-digit number 5057672949897463733694209.

Algorithm 2 gcdOnBorder

```
Comment: Calculate GCD along left border
Input Parameters: N, k;
Begin
for i = 1 to k do
    Calculate X = 2^i(N-1)+1;
    Calculate g_X = gcd(N, X);
    if (g_X > 1) then
        return g_X;
    end if
    Calculate Y = 2^i(N-1)-1;
    Calculate g_Y = gcd(N, Y);
    if (g_Y > 1) then
        return g_Y;
    end if
end for
End
```

Conclusion and Future Work

Looking through the theorems and corollaries proved in previous sections, one can easily know that, for an odd composite integer $N = pq$ with q being in the form of $2^{\alpha}u \pm 1$ and p satisfying $1 < p \le 2^{\alpha} \pm 1$ or $2^{\alpha}+1 < p \pm 2^{\alpha+1}-1$, it is easy to factorize N with the help of the valuated binary tree T_N. Actually, the factorization can be completed by just tracing and finding in T_N the GCD between N and N's ancestors or between N and the leftmost path P_0^N as well as the left side-path P_L^N. Since there are a lot of odd positive integers that fit the conditions, this paper surely solves part of the problem on factoring big odd integers.

Meanwhile, readers can see from the list of bibliographies and their related references that, the tree method is in deed a valid method to study integers. This leads to the future work. Hope more gougers join the study and solve the hard problem of integer factorization.

Acknowledgement

The research is supported by the Open Project Program of the State Key Lab of CAD&CG（Grant No. A2002）and by Foshan University and Foshan Bureau of Science and Technology under project that constructs Guangdong Engineering Center of Information Security for Intelligent Manufacturing System.

Author’s Contributions

Prof. Xingbo WANG contributes 95% of the work in this paper, including discovering and proving the corollaries and theorems as well as designing the algorithm. Mr. Junjian ZHONG contributes 5% of the work, mainly programs and does numerical experiments.

Ethics

The authors declare that there is no conflict of interests regarding the publication of this article.

References

Calik, P., Yilgora P., Ayhanb P. and Demir A.S., 2004. Oxygen transfer effects on recombinant benzaldehydelyase production. Chem. Eng. Sci., 59: 5075-5083. DOI: 10.1016/j.ces.2004.07.070

Phulachand, K.S., 2016. An overview of cryptography. Innovat. IT, 3: 8-11.

Sarnaik, S., Gadekar D. and Gaikwad U., 2016. An overview to integer factorization and RSA in cryptography. Int. J. Adv. Res. Eng. Thechnol., 2: 21-26.

Wang, X., 2016a. Valuated binary tree: A new approach in study of integers. Int. J. Scientific Innovat. Math. Res., 4: 63-67. DOI: 10.20431/2347-3142.0403008

Wang X., 2016b. Amusing properties of odd numbers derived from valuated binary tree. IOSR J. Math., 12: 53-57.

Wang, X., 2017a. Two more symmetric properties of odd numbers. IOSR J. Math., 13: 37-40. DOI: 10.9790/5728-1303023740

Wang, X., 2017b. Genetic traits of odd numbers with applications in factorization of integers. Global J. Pure Applied Math., 13: 493-517.

Wang, X., 2018. T3 tree and its traits in understanding integers. Adv. Pure Math., 8: 494-507. DOI: 10.4236/apm.2018.85028

Wang, X., 2019. Brief summary of frequently- used properties of the floor function: updated 2018. IOSR J. Math., 15: 30-33.

Wang, X. and Guo H., 2019. Some divisibility traits on valuated binary trees. Int. J. Applied Phys. Math., 9: 1-15. DOI: 10.17706/ijapm.2019.9.4.173-181

Yan, S.Y., 2013. Computer Number Theory and Modern Cryptography. 1st Edn., John Wiley and Sons, Singapore, ISBN-10: 1118188586, pp: 432.

Appendix: Maple Programs and Running Results

```
#Subroutine Father: find the father of a node
Father := proc (S)
   local X, r;
   r := modp(S, 4);
   if r = 1 then X := (1/2)*S+1/2
   else X := (1/2)*S-1/2 end if;
end proc
#Subroutine gcdOnBorder
gcdOnBorder := proc (N, k)
   local X, g, i;
   for i to k do
        X = 2^i * (N-1)+1;
        g := gcd(N, X);
        if 1 < g then break end if;
        X = 2^i * (N-1)-1;
        g := gcd(N, X);
        if 1 < g then break end if;
   end do;
end proc
# Main routine
doit:=proc(N)
   local k,F,i,g;
   k := floor((log(N)) / (log(2)))+1;
   g:=gcdOnBorder(N,k):
   if g > 1 then return(g): fi;
   F:= N;
   for i from 1 to k do
        F := Father(F);
        g := gcd(N,F);
        if g > 1 then return(g): fi;
        g:= gcd(N,F-2);
        if g > 1 then return(g): fi;
        g:=gcd(N,F+2);
        if g > 1 then return(g): fi;
   od;
end proc
#tested numbers
ob := Array(1 .. 10, [34639739, 1159847279,
10581684521, 10263855667940024299,
60782931320919664123,
115271397873601774304441,
174538042279885450969073,
9445156115384718744461691,
27326698460114176490535 79,
5057672949897463733694209]);
```

DOI: 10.3844/jmssp.2020.24.34

```
  # test commands
  for i to 10 do
    d1 := doit(ob[i]);
    d2 := ob[i]/d1;
    lprint(ob[i], d1, d2)
  end do;
# Test results
```

```
                                                   4229
                                                   8191
34639739, 4229, 8191
                                                   8849
                                                   131071
1159847279, 8849, 131071
                                                   20183
                                                   524287
10581684521, 20183, 524287
                                                   8171
                                                   1256132134125569
10263855667940024299, 8171, 1256132134125569
                                                   1019
                                                   59649589127497217
60782931320919664123, 1019, 59649589127497217
                                                   49991
                                                   2305843009213693951
115271397873601774304441, 49991, 2305843009213693951
                                                   65521
                                                   2663848877152141313
174538042279885450969073, 65521, 2663848877152141313
                                                   262139
                                                   3603109844542291969
944515611538471874461691, 262139, 3603109844542291969
                                                   16267
                                                   167988556341760475137
2732669846011417649053579, 16267, 167988556341760475137
                                                   274177
                                                   18446744073709551617
5057672949897463733694209, 274177, 18446744073709551617
```