Interval Tree and Its Application in Integer Factorization

Xingbo WANG

Correspondence: Department of Mechatronic Engineering, Foshan University, Foshan City, PRC; Guangdong Engineering Center of Information Security for Intelligent Manufacturing System, Foshan City, PRC; State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi City, PRC

Received: February 20, 2019 Accepted: March 12, 2019 Online Published: March 21, 2019

doi:10.5539/jmr.v11n2p103 URL: https://doi.org/10.5539/jmr.v11n2p103

Abstract

The paper first puts forward a way to study odd integers by placing the odd integers in a given interval on a perfect full binary tree, then makes an investigation on the odd integers by means of combining the original properties of the integers with the properties of the binary trees and obtains several new results on how an odd integer's divisors distribute on a level of a binary tree. The newly discovered law of divisors' distribution that includes common divisors between two symmetric nodes, genetic divisors between an ancestor node and its descendant node can provide a new and simple approach to factorize odd composite integers. Based on the mathematical deductions, numerical experiments are designed and demonstrated in the Maple software. All the results of the experiments are conformance to expectation and validate the validity of the approach.

Keywords: binary tree, integer factorization, genetic trait, algorithm

1. Introduction

In 2016, WANG X in article (WANG X, 2016(IJSIMR)) put forward an approach that studies integer by putting odd integers bigger than 1 on a full perfect binary tree from the top to the bottom and from the left to right. This approach then derived out many previously-unknown properties of the odd integers, such as properties of symmetric nodes and symmetric common divisors, properties of subtrees' duplication and transition, and properties of sum by level, root division and the genetic traits, as introduced in WANG's articles (WANG X, 2017(JM), 2017(GJPAM),2019(IJAPM)). It has known that, these new properties could be helpful in solving the problem of integer factorization, as probed in FU's paper (FU D,2017(JCE)),WANG's paper(WABG X, 2017(JCE)) and LI's paper (LI J., 2018(AJCM)), and they also would be helpful for knowing of the RSA modulus, as investigated in papers (WANG X, 2018(JMR), WANG X. 2018(IJMSS)).

In applying the T_3 tree to speed-up the computation of integers in a definite big interval, another tree-approach was found to be usful in understanding more properties of odd integers. This paper introduces the new approach and its traits in factoring odd composite integers.

2. Preliminaries

This section introduces symbols, definitions and lemmas that are necessary in later sections.

2.1 Symbols and Notations

Throughout this paper, an odd sequence is defined to be a sequence of odd numbers, *e.g.*, 13,15,19,23,31. An odd interval $[a, b]$ is a set of consecutive odd numbers that take a as their lower bound and b as their upper bound. For example, $[3, 11] = \{3, 5, 7, 9, 11\}$. Two odd intervals, I_1 and I_2, are said to have intersection and denoted by $I_1 \cap I_2 \neq \emptyset$ if they contain some common items. For example, $[3, 11] \cap [7, 19] \neq \emptyset$. Symbol $\lfloor x \rfloor$ is to express x's floor function defined by $x - 1 < \lfloor x \rfloor \leq x$, where x is a real number. The terms *binary tree* and its root, nodes, father, left-son, right-son as well as subtrees can be seen in school-books of data structure, for example, Dinesh's handbook [**?**]. This paper mainly concerns the *perfect full binary tree* that has $2^{n+1} - 1$ nodes with depth $n \geq 0$. Symbol $N_{(k,j)}$ is to denote the node at position j on level k of a tree T, where $k \geq 0$ and $0 \leq j \leq 2^k - 1$. On the same level k, two nodes $N_{(k,j)}$ and $N_{(k,2^k-1-j)}$ are called co-symmetric nodes because they station at the geometric symmetric positions. Symbol $T_{(k,j)}$ is to denote the subtree whose root is $N_{(k,j)}$. Symbol $\sum T$ means the sum of all node of T. Symbol $x \in T$ means number x is a node of T. Symbol $A \otimes B$ means A holds and B simultaneously holds; symbol $A \oplus B$ means A or B holds. Symbol $(a = b) > c$ means a takes the value of b and $a > c$. Symbol $A \Rightarrow B$ means conclusion B can be derived from condition A, and symbol $A \Leftrightarrow B$ means A is equivalent to B. Symbol $\mathbf{Z}^+$ means the set of positive integers.

2.2 Lemmas

Lemma 1(See in WANG X, 2016(IJSIMR)) Let p be a positive odd integer; then among p consecutive positive odd integers there exists one and only one that can be divisible by p. Let q be a positive odd number, $S = \{a_i | i \in Z^+\}$ be a set that is composed of consecutive odd numbers; then S needs at least $(n-1)q+1$ elements to have n multiples of q; if $a_\alpha \in S$ is a multiple of q, then so it is with $a_{\alpha+q}$.

Lemma 2(See in WANG X, 2014 & 2017(IOSR-JM)) Let x and y be real numbers and N be a positive integer with $N > 2$; then

(1) $\lfloor \log_2 N \rfloor \leq \log_2(1+N) \leq 1 + \lfloor \log_2 N \rfloor$;

(2) $x \leq y \Rightarrow \lfloor x \rfloor \leq \lfloor y \rfloor$.

3. Method and Main Results

3.1 Binary Tree Method

Let $K \geq 0$ be an integer, $u = 2^{K+1} - 1$ and $a_1, a_2, ..., a_u$ be $2^{K+1} - 1$ consecutive positive odd integers; construct a full perfect binary tree $T_{[a_1,a_u]}$ with $2^{K+1} - 1$ nodes by following way

1. The middle item a_{2^K} is set to the root $N_{(0,0)}$ of $T_{[a_1,a_u]}$.

2. The item $a_{2^{K-1}}$, the middle item of the $2^K - 1$ items left to a_{2^K}, is set to the left son of $N_{(0,0)}$; the item $a_{2^K+2^{K-1}}$, the middle item of the $2^K - 1$ items right to a_{2^K}, is set to the right son of $N_{(0,0)}$.

3. Recursively take each son's left son and right son by the above *middle item rule* to finish constructing the whole tree $T_{[a_1,a_u]}$.

For example, with $a_1 = 13, a_2 = 15, ..., a_{14} = 39$ and $a_{15} = 41$, setting $K = 3$, $T_{[13,41]}$ is constructed as figure 1.

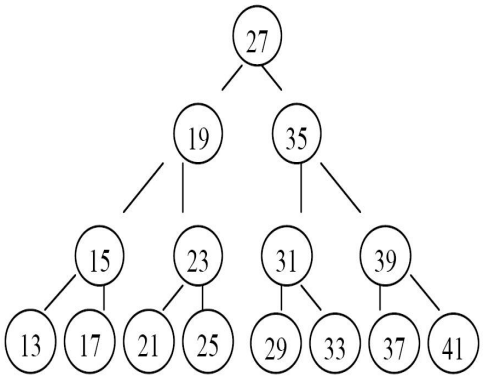

Figure 1. Interval tree constructed from odd interval $[13, 41]$

For convenience, the tree constructed above is called an *odd interval tree* or simply an *interval tree*. An interval tree can be denoted with an abstract symbol T_I, or an interval symbol $T_{[x,y]}$ for the case the interval $[x, y]$ is given or a root symbol $T_{N_{(0,0)}}$ for the case that $N_{(0,0)}$ is the root of the tree. The nonnegative integer K is the depth of the tree. A tree of depth $K = 0$ means it contains merely 1 node, the root. The left and the right subtrees of T_I are respectively denoted by T_{Il} and T_{Ir}. On level l with $l \geq 0$ there are 2^l nodes each of which can be a root of a subtree. Subtree $T_{(l,s)}$ is said left to subtree $T_{(l,t)}$ if $s < t$.

3.2 Main Results

Proposition 1 (Maximum Depth of Interval Tree) Let T_I be an odd interval tree constructed from odd interval $[a_1, a_u]$ with $a_1 > 0$; if $N_{(0,0)}$ is the root of T_I, then T_I contains at most $2^{\lfloor \log_2 N_{(0,0)} \rfloor + 1} - 1$ nodes. In another word, the maximum depth K_{max} of T_I is limited to $K_{\max} = \lfloor \log_2 N_{(0,0)} \rfloor$.

Proof. [Proof of Proposition 1] Among all the positive integers, there are $\frac{N_{(0,0)}+1}{2}$ positive odd integers from 1 to $N_{(0,0)}$. $N_{(0,0)}$ being the root of T_I means $N_{(0,0)} = a_{2^K}$ provided that the odd interval $[a_1, a_u]$ contains $2^{K+1} - 1$ consecutive positive odd integers. Therefore $2^{K_{\max}} \leq \frac{N_{(0,0)}+1}{2}$, which leads to

$$K_{\max} \leq \log_2(N_{(0,0)} + 1) - 1$$

Then by Lemma 2, it yields

$$K_{\max} \leq \lfloor \log_2 N_{(0,0)} \rfloor$$

□

Proposition 2 (In-order Traversal Restoration) Let $K \geq 0$, $u = 2^{K+1} - 1$ be an integer, $I = [a_1, a_u]$ be an odd interval and $T_{[a_1,a_u]}$ be the interval tree constructed from I; then the odd interval $I = [a_1, a_u]$ can be restored by applying the *in-order traversal* on $T_{[a_1,a_u]}$.

Proof. [Proof of Proposition 2] Referring to the definition of the in-order traversal of a binary tree, as seen in [**?**], and comparing it with the middle item rule to construct the tree $T_{[a_1,a_u]}$, it immediately knows that the proposition holds. □

Proposition 3 Let $K \geq 0$, $u = 2^{K+1} - 1$ be an integer, $I = [a_1, a_u]$ be an odd interval and $N_{(0,0)}$ be the root of the odd interval tree T_I that is constructed from I; then the items that satisfy $x \in I$ and $x < N_{(0,0)}$ lie in T_{Il} whereas the items that satisfy $x \in I$ and $x > N_{(0,0)}$ lie in T_{Ir} . Among a father and its two sons, the left son is the smallest, the father is the average of the two sons and the right son is the biggest. Consequently, for a node G and its two sons, S_l and S_r, if n_{ll} is a node in the left subtree of S_l and n_{lr} is a node in the right subtree of S_l, it holds $n_{ll} < S_l < G$ and $S_l < n_{lr} < G$; whereas, if n_{rl} is a node in the left subtree of S_r and n_{rr} is a node in the right subtree of S_r, it holds $G < n_{rl} < S_r$ and $G < S_r < n_{rr}$.

Proof. [Proof of Proposition 3] The stated relationships are actually from the construction of the interval tree. □

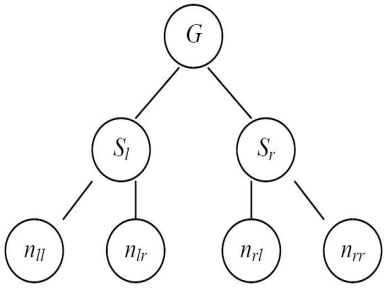

Figure 2. Relationships among a node and its descendants

Proposition 4 Let $I_1 = [a_1, a_u]$ and $I_2 = [b_1, b_v]$ be two odd intervals, T_1 and T_2 be the odd interval trees corresponding to the two intervals respectively; then $I_1 \cap I_2 \neq \emptyset \Leftrightarrow T_1 \cap T_2 \neq \emptyset$.

Proof. [Proof of Proposition 4] By Proposition 2, an odd interval is equivalent to its odd interval tree. □

Theorem 1 (Calculation of Nodes) Let $K \geq 0$, $u = 2^{K+1} - 1$ be an integer and $a_1, a_2, ..., a_u$ be $2^{K+1} - 1$ consecutive positive odd integers; assume $N_{(0,0)} = a_{2^K}$ is the root of $T_{[a_1,a_u]}$; then

$$N_{(i,\omega)} = 2\alpha - 1 + 2^{K-i+1}(1 + 2\omega)$$
$$i = 0, 1, ..., K; \omega = 0, 1, ..., 2^i - 1$$

or equivalently,

$$N_{(i,\omega)} = N_{(0,0)} - 2^{K+1} + 2^{K+1-i}(1 + 2\omega) = N_{(0,0)} - 2^{K+1-i}(2^i - 2\omega - 1)$$
$$i = 0, 1, ..., K; \omega = 0, 1, ..., 2^i - 1$$

Proof. [Proof of Theorem 1] Without loss of generality, assume $a_1 = 2\alpha + 1$ with $\alpha \geq 0$ being an integer; then

$$a_n = 2\alpha + 1 + 2(n - 1) \tag{1}$$

where $n = 1, 2, ..., u$.

By the rule of the construction, first considering the leftmost node on each level of $T_{[a_1,a_u]}$, it holds

$$N_{(0,0)} = a_{2^K} = 2\alpha + 1 + 2(2^K - 1) = 2\alpha + 2^{K+1} - 1$$

$$N_{(1,0)} = a_{2^{K-1}} = 2\alpha + 1 + 2(2^{K-1} - 1) = 2\alpha + 2^K - 1$$

$$N_{(2,0)} = a_{2^{K-2}} = 2\alpha + 1 + 2(2^{K-2} - 1) = 2\alpha + 2^{K-1} - 1$$

$$\cdots\cdots$$

$$N_{(i,0)} = a_{2^{K-i}} = 2\alpha + 1 + 2(2^{K-i} - 1) = 2\alpha + 2^{K+1-i} - 1$$

$$\cdots\cdots$$

$$N_{(K,0)} = a_{2^{K-K}} = 2\alpha + 1 + 2(2^{K-K} - 1) = 2\alpha + 1 = a_1$$

Now consider the nodes on the same level and take level i as a general case. Note that, on level i there are 2^i nodes distributed uniformly in terms of their indices. Since $N_{(i-1,0)} = a_{2^{K-(i-1)}}$, it knows $N_{(i,0)} = a_{2^{K-i}}$ and $N_{(i,1)} = a_{2^{K-(i-1)}+2^{K-(i-1)-1}} = a_{2^{K-i}(2+1)}$, which means the difference of the indices between two adjacent nodes is $d = 2^{K-i}(2+1) - 2^{K-i} = 2^{K+1-i}$. Consequently,

$$N_{(i,\omega)} = a_{2^{K-i}+2^{K+1-i}\omega} = a_{2^{K-i}(1+2\omega)} = 2\alpha + 2^{K-i+1}(1+2\omega) - 1$$

By $N_{(0,0)} = 2\alpha + 2^{K+1} - 1$, it yields

$$\begin{aligned} N_{(i,\omega)} &= 2\alpha - 1 + 2^{K+1} - 2^{K+1} + 2^{K-i+1}(1+2\omega) \\ &= N_{(0,0)} - 2^{K+1} + 2^{K+1-i}(1+2\omega) \\ &= N_{(0,0)} - 2^{K+1-i}(2^i - 2\omega - 1) \end{aligned}$$

□

Corollary 1 (Node in In-order Traversal Restoration) Let T_I be an $N_{(0,0)}$-rooted odd interval tree with depth $K \geq 0$, and $[a_1, a_u]$ be its in-order traversal restoration; then $N_{(i,\omega)} = a_{2^{K-i}(1+2\omega)}$ and there are $|2^{K-i}(2^i - 2\omega - 1)| + 1$ odd integers from $N_{(0,0)}$ to $N_{(i,\omega)}$ in the interval $[a_1, a_u]$, where $0 \leq i \leq K$ and $0 \leq \omega \leq 2^i - 1$.

Proof. [Proof of Corollary 1] By Theorem 1, it yields

$$N_{(i,\omega)} = 2\alpha - 1 + 2^{K-i+1}(1+2\omega) = 2\alpha + 1 + 2(2^{K-i}(1+2\omega) - 1)$$

Referring to (1), it knows

$$N_{(i,\omega)} = a_{2^{K-i}(1+2\omega)}$$

Since $N_{(0,0)} = a_{2^K}$, it is sure that there are $|2^{K-i}(1+2\omega) - 2^K| + 1 = |2^{K-i}(2^i - 2\omega - 1)| + 1$ odd integers from $N_{(0,0)}$ to $N_{(i,\omega)}$. □

Corollary 2 (Root Form vs Bottom Form). Let T_I be an $N_{(0,0)}$-rooted odd interval tree with depth $K \geq 1$. If $N_{(0,0)}$ is of the form $4k+1$, then all the nodes from level 0 to level $K-1$ are of the form $4k+1$, whereas every node on level K is of the form $4k-1$. If $N_{(0,0)}$ is of the form $4k-1$, then all the nodes from level 0 to level $K-1$ are of the form $4k-1$, whereas every node on level K is of the form $4k+1$.

Proof. [Proof of Corollary 2]Consider the case $N_{(0,0)} = 4k+1$. By Theorem 1,

$$N_{(i,\omega)} = N_{(0,0)} - 2^{K+1-i}(2^i - 2\omega - 1) = 4k + 1 - 2^{K+1-i}(2^i - 2\omega - 1)$$

it knows that $2^{K+1-i} = 2^2 \cdot 2^\alpha$ with integer $\alpha \geq 0$ when $i \leq K-1$. Hence it yields

$$N_{(i,\omega)} = 4k + 1 - 2^{K+1-i}(2^i - 2\omega - 1) = 4(k - 2^\alpha(2^i - 2\omega - 1)) + 1$$

When $i = K$, $N_{(i,\omega)} = 4k + 1 - 2^1(2^K - 2\omega - 1) = 4k - 2^{K+1} - 4\omega - 1$.

Similarly, the other conclusion holds.

□

Corollary 3 (Subtraction of Two Nodes) Let $N_{(i,\omega)}$ and $N_{(j,\vartheta)}$ be two nodes of an odd interval tree with depth $K \geq 0$; then

$$N_{(i,\omega)} - N_{(j,\vartheta)} = 2^{K+1-i}(2\omega + 1) - 2^{K+1-j}(2\vartheta + 1)$$

Particularly,

$$N_{(i,\omega)} - N_{(i,\vartheta)} = 2^{K+2-i}(\omega - \vartheta)$$
$$N_{(i,\omega)} - N_{(j,\omega)} = (2^{K+1-i} - 2^{K+1-j})(2\omega + 1)$$

Proof. [Proof of Corollary 3] Direct calculations immediately lead to the results. □

Corollary 4 (Multiples on One Level) Let T_I be an $N_{(0,0)}$-rooted odd interval tree with depth $K \geq 0$; if $N_{(0,0)} \geq 2^{K+2} - 3$ then on the same level of an interval tree there is not a node that is a multiple of another one.

Proof. [Proof of Corollary 4] Referring to Theorem 1, taking the smallest and the biggest nodes on level i yields

$$N_{(i,0)} = N_{(0,0)} - 2^{K+1} + 2^{K+1-i}$$
$$N_{(i,2^i-1)} = N_{(0,0)} + 2^{K+1} - 2^{K+1-i}$$

and

$$N_{(i,2^i-1)} - N_{(i,0)} = 2^{K+2} - 2^{K+2-i}$$
$$N_{(i,2^i-1)} - 2N_{(i,0)} = -(N_{(0,0)} - 3 \times 2^{K+1-i}(2^i - 1))$$
$$N_{(i,2^i-1)} - 3N_{(i,0)} = -2(N_{(0,0)} - 2^{K+2-i}(2^i - 1))$$

Obviously, when $N_{(0,0)} \geq 2^{K+2} - 3$ it holds $N_{(0,0)} > 2^{K+2} - 4$ and $N_{(i,2^i-1)} - 3N_{(i,0)} < 0$. That is to say, there is not a node 3 times bigger than the smallest node on the same level and thus it is natural that there is not a node that is a multiple of another one because all the nodes are odd integers. □

Corollary 5 (Subtraction of Two Trees) Let T_M and T_N be two odd interval trees of the same depth; then yields

$$M_{(i,\omega)} \geq N_{(i,\omega)}$$

where $M_{(i,\omega)}$ and $N_{(i,\omega)}$ are nodes at position ω on level i of T_M and T_N respectively.

Proof. [Proof of Corollary 5] Assume $K \geq 0$ is the depth of the two trees; then by Theorem 1

$$M_{(i,\omega)} - N_{(i,\omega)} = M - N \geq 0$$

□

Corollary 6 (Symmetric Common Divisors). Let $N_{(0,0)}$ be the root of an interval tree T_I ; then its two symmetric nodes,$N_{(i,\omega)}$ and $N_{(i,2^i-1-\omega)}$, satisfy

$$N_{(i,\omega)} + N_{(i,2^i-1-\omega)} = 2N_{(0,0)}$$

and thus

$$(N_{(i,\omega)}, N_{(i,2^i-1-\omega)})|N_{(0,0)}, (N_{(0,0)}, N_{(i,\omega)})|N_{(i,2^i-1-\omega)}, (N_{(0,0)}, N_{(i,2^i-1-\omega)})|N_{(i,\omega)}$$

Proof. [Proof of Corollary 6] By $N_{(i,\omega)} = N_{(0,0)} - 2^{K+1-i}(2^i - 1) + 2^{K+2-i}\omega$, direct calculation yields

$$\begin{aligned} N_{(i,2^i-1-\omega)} &= N_{(0,0)} - 2^{K+1-i}(2^i - 1) + 2^{K+2-i}(2^i - 1 - \omega) \\ &= N_{(0,0)} - 2^{K+1} + 2^{K+1-i} + 2^{K+2} - 2^{K+2-i} - 2^{K+2-i}\omega \\ &= N_{(0,0)} + 2^{K+1} - 2^{K+1-i} - 2^{K+2-i}\omega \\ &= N_{(0,0)} + 2^{K+1-i}(2^i - 1) - 2^{K+2-i}\omega \end{aligned}$$

which shows $N_{(i,\omega)} + N_{(i,2^i-1-\omega)} = 2N_{(0,0)}$, and thus validates the theorem By the Euclid division theorem. □

Corollary 7 (Sum Property) Let $N_{(0,0)}$ be the root of an interval tree T_I of depth $K \geq 0$; then

$$\sum T_I = (2^{K+1} - 1)N_{(0,0)}$$

Proof. [Proof of Corollary 7] (Omitted) □

Corollary 8 (Sum Subtraction of Subtrees). Let T_{Il} and T_{Ir} be respectively the left and right subtrees of an interval tree T_I with depth $K \geq 0$; then

$$\sum T_{Ir} - \sum T_{Il} = 2^{K+1}(2^K - 1)$$

Proof. [Proof of Corollary 8] By Corollary 7, it yields

$$\sum T_{Il} = (2^K - 1)N_{(1,0)}, \sum T_{Ir} = (2^K - 1)N_{(1,1)}$$

Hence

$$\sum T_{Ir} - \sum T_{Il} = (2^K - 1)(N_{(1,1)} - N_{(0,0)})$$

By Theorem 1, $N_{(1,0)} = N_{(0,0)} - 2^K$ and $N_{(1,1)} = N_{(0,0)} + 2^K$; hence it holds

$$\sum T_{Ir} - \sum T_{Il} = 2^{K+1}(2^K - 1)$$

□

Corollary 9 (Divisors of Root). Let $N_{(0,0)}$ be the root of an interval tree T_I and $N_{(0,0)} = p\alpha$ with $p > 1$ and $\alpha \geq 1$ being positive odd integers; then for arbitrary node $N_{(i,\omega)}$ of T_I,

$$p|N_{(i,\omega)} \Leftrightarrow p|(|2^i - 2\omega - 1|)$$

Proof. [Proof of Corollary 9] By Theorem 1,

$$N_{(i,\omega)} = N_{(0,0)} - 2^{K+1-i}(2^i - 2\omega - 1)$$

□

Theorem 2 (Multiples of Root) Let p be a positive odd integer, $\sigma = \lfloor \log_2 p \rfloor + 1$ and $N_{(0,0)} = p$ be the root of an interval tree T_I with depth $K \geq \sigma$; then

(1) On level upper than σ, there is not a multiple node of p except $N_{(0,0)}$ itself.

(2) On level σ there are exactly two multiples of p, and there are at least 2 p's multiples on each level after level σ.

(3) The two p's multiples on level σ are $N_{(\sigma,s)}$ and $N_{(\sigma,t)}$ with $s = 2^{\sigma-1} - (p+1)/2$ and $t = 2^{\sigma-1} + (p-1)/2$.

(4) On level χ that satisfies $\sigma < \chi \leq K$, p's multiples of are calculated by

$$N_{(\chi,l-\alpha p)}, N_{(\chi,r+\alpha p)}$$

where $l = 2^{\chi-1} - (p+1)/2$, $r = 2^{\chi-1} + (p-1)/2$ and $\alpha = 1, 2, ..., \left\lfloor \frac{l}{p} \right\rfloor$.

Proof. [Proof of Theorem 2] Without loss of generality, assume $N_{(0,0)} = p\alpha$ with $p > 1$ and $\alpha \geq 1$ being odd integers and consider p's multiples in T_{Ir}. By Lemma 1, T_{Ir} has at least p nodes if it contains a multiple of p. Since it totally contains $2^i - 1$ nodes from level 1 to level i, it yields

$$2^i - 1 \geq p$$

which is $i \geq \log_2(p+1)$. Consequently, T_{Ir} does not contain a multiple of p if $i < \log_2(p+1) \leq \lfloor \log_2 p \rfloor + 1$, by referring to Lemma 2.

Now consider the case $i = \lfloor \log_2 p \rfloor + 1$. Since $2^{\lfloor \log_2 p \rfloor + 1} - 1 \geq 2^{\log_2(p+1)} - 1 = p$ by Lemma 2, T_{Ir} contains at least one multiple of p by Lemma 1. Meanwhile, $i = \lfloor \log_2 p \rfloor + 1 \leq \log_2 p + 1$ yields $2^{\lfloor \log_2 p \rfloor + 1} - 1 \leq 2^{\log_2 p + 1} - 1 = 2p - 1$. This says by Lemma 1 that it is impossible for T_{Ir} to have 2 multiples of p when $i = \lfloor \log_2 p \rfloor + 1$. Now that, $i < \lfloor \log_2 p \rfloor + 1$

resulting in no p's multiples in T_{Ir} and $i = \lfloor \log_2 p \rfloor + 1$ leading to 1 p's multiple certainly validate that there is 1 multiple of p on level $i = \lfloor \log_2 p \rfloor + 1$.

For the case $i > \lfloor \log_2 p \rfloor + 1$, it knows $2^i - 1 \geq 2^{\lfloor \log_2 p \rfloor + 2} - 1 = 2 \times 2^{\lfloor \log_2 p \rfloor + 1} - 1 > 2p - 1 \geq 2p$. Hence there are at least 2 multiples of p on the level in T_{Ir}, which says there are at least 1 multiple of p on level $i > \lfloor \log_2 p \rfloor + 1$.

Now consider the two nodes on level $\lfloor \log_2 p \rfloor + 1$. Direct calculation shows

$$\begin{aligned} & N_{(\sigma,s)} = N_{(0,0)} - 2^{K+1} + 2^{K+1-\sigma}(1 + 2s) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K+1-\lfloor \log_2 p \rfloor - 1}(1 + 2^{1+\lfloor \log_2 p \rfloor} - p - 1) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K-\lfloor \log_2 p \rfloor}(2^{1+\lfloor \log_2 p \rfloor} - p) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K+1} - 2^{K-\lfloor \log_2 p \rfloor} p \\ & = N_{(0,0)} - 2^{K-\lfloor \log_2 p \rfloor} p \end{aligned}$$

Obviously, $p|N_{(0,0)}$ yields $p|N_{(\sigma,s)}$. Then by symmetric property it yields $p|N_{(0,0)} \Rightarrow p|N_{(\sigma,t)}$.

Likewise, direct calculation yields

$$\begin{aligned} & N_{(\chi,l)} = N_{(0,0)} - 2^{K+1} + 2^{K+1-\chi}(1 + 2l) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K+1-\chi}(1 + 2^{\chi} - p - 1) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K+1-\chi}(2^{\chi} - p) \\ & = N_{(0,0)} - 2^{K+1} + 2^{K+1} - 2^{K+1-\chi} p \\ & = N_{(0,0)} - 2^{K+1-\chi} p \end{aligned}$$

This and Lemma 1 as well as the symmetric property show $N_{(\chi,l-\alpha p)}$ and $N_{(\chi,r+\alpha p)}$ are p's multiples. □

Corollary 10 (Genetic Property) Let $N_{(0,0)}$ be the root of an interval tree T_I with depth $K \geq 0$ and p be an odd number; then $p|N_{(0,0)}$ yields $p|N_{(\chi,l-\alpha p)} \otimes p|N_{(\chi,r+\alpha p)}$, where $\lfloor \log_2 p \rfloor + 1 \leq \chi \leq K$, $l = 2^{\chi-1} - (p+1)/2$, $r = 2^{\chi-1} + (p-1)/2$ and $0 \leq \alpha \leq \left\lfloor \frac{l}{p} \right\rfloor$.

Proof. [Proof of Corollary 10] (Omitted) □

Remark 2. Corollary 10 shows that, taking $N_{(\chi,l-\alpha p)}$ or $N_{(\chi,r+\alpha p)}$ as root of an odd interval tree T_I^* with depth $K^* \geq K$, the nodes at the positions $l - \alpha p$ and $r + \alpha p$ on levels lower than $\lfloor \log_2 p \rfloor$ in T_I^* will have p as their divisors.

Example 1. Given an odd interval $[81, 109]$ and its interval tree, as seen in figure 3. It can see that, $5|(N_{(0,0)} = 95)$, $\chi = \lfloor \log_2 5 \rfloor + 1 = 3$, $l = 2^{\chi-1} - \frac{p+1}{2} = 2^2 - 3 = 1$, $r = 2^{\chi-1} + \frac{p-1}{2} = 2^2 + 2 = 6$ and $\alpha = \left\lfloor \frac{l}{5} \right\rfloor = 0$. Two nodes $N_{(\chi,l)} = N_{(3,1)} = 85$ and $N_{(\chi,r)} = N_{(3,6)} = 105$ are multiples of 5. It also can see that, the odd interval tree $T_{[71,99]}$ rooted with $N_{(3,1)} = 85$ also has two nodes, $N_{(3,1)} = 75$ and $N_{(3,6)} = 95$, divisible by 5, as shown in figure 4.

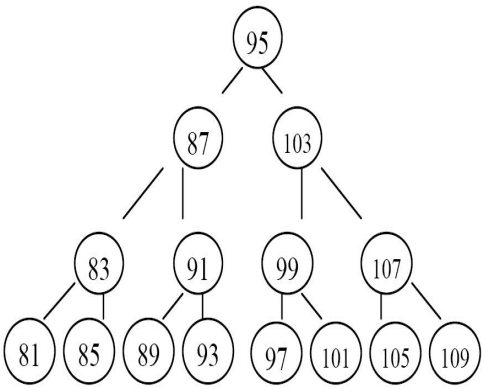

Figure 3. Interval tree constructed from odd interval $[81, 109]$

Corollary 11 (Factorization of Root) Let $N_{(0,0)}$ be the root of an interval tree T_I with depth no less than $\left\lfloor \frac{1}{2} \log_2 N_{(0,0)} \right\rfloor + 1$; if p is a divisor of $N_{(0,0)}$, then $N_{(0,0)}$ can be factorized in at most $\frac{p+1}{2}$ steps.

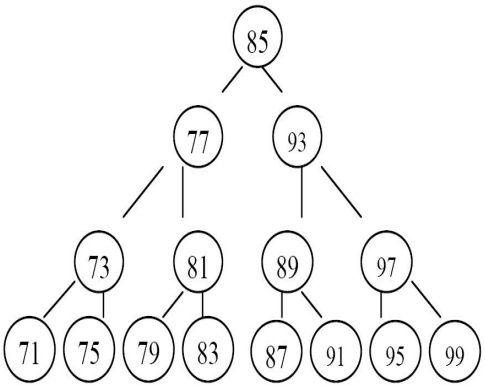

Figure 4. Interval tree constructed from odd interval [71, 99]

Proof. [Proof of Corollary 10] Without loss of generality, assume $N_{(0,0)} = pq$ with $p \leq \sqrt{N}$ and T_{Il} is the left subtree of T_I. Then $\lfloor \log_2 p \rfloor + 1 \leq \left\lfloor \frac{1}{2}\log_2 N \right\rfloor + 1$. Referring to Corollary 10 and taking $\sigma = \left\lfloor \frac{1}{2}\log_2 N \right\rfloor + 1$, it can see that, on each level χ with $\chi \geq \sigma$, there is at least one p's multiple-node $N_{(\chi,l)}$ with $l = 2^{\chi-1} - (p+1)/2$. It can see that, $N_{(\chi,l)}$ is the node that is $\frac{p+1}{2}$ nodes away from the rightmost node on level χ of T_{Il}. □

Theorem 3 Let $T_{N_{(0,0)}}$ be the odd interval tree rooted with $N_{(0,0)} > 1$ and

$$P = \begin{cases} \left\lfloor \sqrt{N_{(0,0)}} \right\rfloor, \left\lfloor \sqrt{N_{(0,0)}} \right\rfloor \text{is odd} \\ \left\lfloor \sqrt{N_{(0,0)}} \right\rfloor - 1, \left\lfloor \sqrt{N_{(0,0)}} \right\rfloor \text{ is even} \end{cases}$$

then the depth of $T_{N_{(0,0)}}$ is no less than $\lfloor \log_2 N_{(0,0)} \rfloor - 1$ if $P \in T_{N_{(0,0)}}$.

Proof. [Proof of Theorem 3] Assume $T_{N_{(0,0)}}$ is constructed from odd interval $[a_1, a_u]$ with $u = 2^{K+1} - 1$; then $N_{(0,0)}$ is the middle item of the interval and there are $u = 2^K - 1$ items left to $N_{(0,0)}$. By definition, P is surely an odd integer and is probably one of those left items. Not that, there are $\frac{N_{(0,0)}-P}{2} + 1$ items from P to $N_{(0,0)}$; hence in order to ensure P is one of the left items, it must hold

$$2^K - 1 \geq \frac{N_{(0,0)} - P}{2}$$

Namely,

$$2^{K+1} \geq N_{(0,0)} - P + 2 > N_{(0,0)} - P$$

Since $P \leq \left\lfloor \sqrt{N_{(0,0)}} \right\rfloor \leq \sqrt{N_{(0,0)}}$, it knows $N_{(0,0)} - P \geq \sqrt{N_{(0,0)}}(\sqrt{N_{(0,0)}} - 1) > (\sqrt{N_{(0,0)}} - 1)^2$, which leads to

$$K > 2\log_2(\sqrt{N_{(0,0)}} - 1) - 1$$

Note that, the inequality $\log_2(\sqrt{x} - 1)^2 - \log_2 x = \log_2(1 - \frac{2}{\sqrt{x}}(1 - \frac{1}{2x})) < 0 (x \geq 3)$ indicates $\log_2 N_{(0,0)} > 2\log_2(\sqrt{N_{(0,0)}} - 1)$, and thus $\lfloor \log_2 N_{(0,0)} \rfloor \geq \left\lfloor 2\log_2(\sqrt{N_{(0,0)}} - 1) \right\rfloor$ by Lemma 2, it knows that, taking $K = \lfloor \log_2 N_{(0,0)} \rfloor - 1$ can ensure $P \in T_{N_{(0,0)}}$. □

4. Application in Factoring Integers

Let $N = pq$ be a positive odd integer with divisors p and q satisfying $p < q$; construct an N-rooted odd interval tree T_N with depth $\sigma = \left\lfloor \frac{1}{2}\log_2 N \right\rfloor + 1$; then check on level σ in the left subtree of T_N the *greatest common divisor* (GCD) between each node and N. By Corollary 10, there must be one or more nodes containing p's or q's multiples on the level. It is sure that N can be always factorized. The approach can be summarized to be the procedure list below. Programming with the procedure in *Maple software*, picking randomly some odd composite integers in the form of Maple array and running the program on a PC with E5450 CPU and 4.0GB memory with Maple V15.0 obtain the results as figure 5 shows. The experiments show that the approach is valid for factoring odd integers.

Od:=Array([16637, 2129189,
4538873, 8772041,
1035918371, 2512642129,
5783560579, 9048212729,
80735174503, 211041144109,
170442776634553, 1808898276844231,
35249679931198483, 37522676526028537,
556499304645216091, 1123877887715932507,
1129367102454866881, 1902408569846737793,
10188337563435517819, 24928816998094684879]);

Procedure To Factorize Odd Integers

Procedure FactoringOdd
Input: N;
Step 1. Calculate $l = \left\lfloor \frac{1}{2}\log_2 N \right\rfloor + 1$;
Calculate $s = 2^{l-1} - 1$;
Step 2. For $i = 1$ to $i = s$ do
$N_i = N - 2^{l+1} + 4i + 2$;
$d = \gcd(N_i, N)$;
if($d > 1$) break;
EndFor
Step 3. Output d;
EndProcedure

Maple Programming Codes

```
f := proc (N)
local s, l, M, X, d, g, i;
   l := floor(0.5*log(N)/log(2))+1;
   s := 2^(l-1)-1; M := N-2^(l+1)+2;
   for i from 0 to s do
      X := M+2*i;
      d := gcd(X, N);
      if 1 < d then break end if
   end do;
g:=N/d;
printf("With %d steps, %d=%d*%d",I, N,d,g);
end proc
```

5. Conclusions

Both theoretic deductions and experiments show that, the odd interval tree is another new approach in knowing the integers. The properties discovered in this paper can surely provide a new way to demonstrates the genetic traits of the odd integers and a practical way in exploring solution of factoring of integers. In the end, reader can see that, the approach provided and tested with Maple software can be improved a lot, especially in parallel computing. Hope more followers can show more profitable results.

Acknowledgments

The research work is supported by the State Key Laboratory of Mathematical Engineering and Advanced Computing under Open Project Program No.2017A01, Department of Guangdong Science and Technology under project 2015A010104011, Foshan Bureau of Science and Technology under projects 2016AG100311, Project gg040981 from Foshan University. The authors sincerely present thanks to them all.

```
for k from 1 to 20 do
 f(Od(k));
 end do
With 62 steps, 16637=131*127
With 22 steps, 2129189=2003*1063
With 929 steps, 4538873=2237*2029
With 398 steps, 8772041=3299*2659
With 25 steps, 1035918371=32717*31663
With 7232 steps, 2512642129=51071*49199
With 25027 steps, 5783560579=81017*71387
With 15600 steps, 9048212729=99871*90599
With 106447 steps, 80735174503=311393*259271
With 6504 steps, 211041144109=511279*412771
With 49124 steps, 170442776634553=228479*745988807
With 819186 steps, 1808898276844231=2424833*745988807
With 45509976 steps, 35249679931198483=59138501*596052983
With 37363867 steps, 37522676526028537=193707721*193707697
With 163876505 steps, 556499304645216091=745988813*745988807
With 88137066 steps, 1123877887715932507=299155897*3756830131
With 6538187 steps, 1129367102454866881=25869889*43655660929
With 700747420 steps, 19024085698467377793=745988807*2550183799
With 55879043 steps, 10188337563435517819=70901851*143696355169
With 293968681 steps, 249288169980946848879=347912923*71652460573
```

Figure 5. Screenshot of Maple computing results

References

WANG, X. (2016). Valuated Binary Tree: A New Approach in Study of Integers, *International Journal of Scientific and Innovative Mathematical Research*, *4*(3), 63-67.

WANG, X. (2017). Amusing Properties of Odd Numbers Derived From Valuated Binary Tree, *IOSR Journal of Mathematics*, *12*(6), 53-57.

WANG, X.(2017). Genetic Traits of Odd Numbers With Applications in Factorization of Integers, *Global Journal of Pure and Applied Mathematics*, *13*(2), 493-517.

WANG, X., & GUO, H. (2019). Some Divisibility Traits on Valuated Binary Trees, *International Journal of Applied Physics and Mathematics*, *9*(1), 1-15. https://doi.org/10.17706/ijapm.2019.9.1.1-11

FU, D. (2017). A Parallel Algorithm for Factorization of Big Odd Numbers, *IOSR Journal of Computer Engineering*, *19*(2,v5), 51-54.

WANG, X. (2017) Strategy For Algorithm Design in Factoring RSA Numbers, *IOSR Journal of Computer Engineering*, *13*(3,v2), 1-7.

LI, J. (2018). A Parallel Probabilistic Approach to Factorize a Semiprime, *American Journal of Computational Mathematics*, *8*(2), 175-183. https://doi.org/10.4236/ajcm.2018.82013

WANG, X. (2018). Influence of Divisor-ratio to Distribution of Semiprime's Divisor, *Journal of Mathematics Research*, *10*(4), 54-61. https://doi.org/10.5539/jmr.v10n4p54

WANG, X., & SHEN, Z. (2018). Traits of a RSA Modulus on T3 Tree,*Journal of Mathematics Research*, *10*(6), 15-29. https://doi.org/10.5539/jmr.v10n6p15

WANG, X. (2018). Divisors' Distribution of A RSA Modulus On T_3 Tree, *International Journal of Mathematics and Statistics Studies*, *6*(4), 15-32.

Mehta, D. P., & Sartaj, S. (2005). *Handbook of Data Structures and Applications*,Chapman & Hall/CRC.

WANG, X. (2014). Some supplemental properties with appendix applications of floor function, *Journal of Science of Teachers College and University(in Chinese)*, *34*(3), 7-9.

WANG, X. (2017). Brief Summary of Frequently-Used Properties of the Floor Function, *IOSR Journal of Mathematics*, *13*(5), 46-48.