Traits of Interval Tree in Solving Blind Search Problems of Finding a Term in an Ordered Data Set

Xingbo Wang and Jicong Wu

Abstract—**This paper investigates the traits of the interval tree in solving the blind-searched problems of finding uninformed terms in an ordered data set. It first proves several new properties of the interval tree and then shows that applying an interval tree to express data set results in half of the objective terms lying on the bottom level while another half lying on the levels over the bottom, and a bigger probability as well as half or less than half a amount of searching steps to find an objective term in comparison to conventional search strategies. Mathematical reasoning on the new properties of the interval tree plus conclusions related with the distribution of the objective terms on the interval tree is shown in detail and searching strategy is proposed in the end. The results in this paper are helpful for designing a searching algorithm.**

Index Terms—**Artificial intelligence, blind search, binary tree, probability, algorithm.**

I. Introduction

The concept of the interval tree was originally put forwards by X WANG in paper [1] to study the divisibility of odd integers. The tree is constructed with the terms in an integer-interval by picking a middle term to be the root to subdivide the interval into two subintervals, and picking the middle terms in the subintervals to be the left-son and right-son respectively and recursively. By means of the interval tree, properties of integers were demonstrated in a different point of view.

A recent study has revealed another new trait of the interval tree in solving the problems of so-called blind-searches, which was named in book [2], in a large integer interval or ordered data set. The background originates from a search problem in cryptography.

It is known that, guessing and searching a hidden number in a large data set is an ordinary task in study of network security or cryptography, as stated in [3], and most of such searches are blind ones. A typical problem is stated as follows.

Let *S* be a subinterval consisting of 2^{α} integers that belong to a large integer interval *I* that contains $2^{\alpha+\beta}$ integers, where $\alpha>0$ and $\beta>0$ are integers and the location of *S* is unknown in advance; find a term in *S* as fast as possible.

For example, the odd interval [2558595694593, 2558596743167] contains 2^{19} odd integers, and among these 2^{19} ones, there are $2^{4}=16$ consecutive odd integers $n,n+2,...,n+30$ that are calculated from *N*=78081683 by a certain rule. We do not where the 16 numbers lie, and the task is to find them as fast as possible.

Since the interval *I* is very big when β is big (normally more than 2^{200}) and *S*'s location is unknown in advance, it is hard to search in *I* one by one even with the fastest computer in the world. Considering that an interval tree is actually a bi-subdivision approach that is very like the binary-search method, it is found via test and theoretical reasoning that the interval tree could increase the probability and reduce the amount of the searching steps as well as be well incorporated with the Monte Carlo pseudorandom number generator [4], which has been widely applied in cryptography, to find out the objective. This paper shows the details.

II. Preliminaries

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

A. Symbols and Notation

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 $I_1\cap I_2\neq\varnothing$ if they contain some common terms. For example, $[3,11]\cap[7,19]\neq\varnothing$. 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 [5]. 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 $N_{(k,2^{k}-1-j)}$ 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 $x\in T$ means number x is a node of T. 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. Symbol $A\otimes B$ means *A* holds and simultaneously *B* 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 **Z+** means the set of positive

Manuscript received February 10, 2020; revised May 6, 2020.

Xingbo Wang is with Department of Mechatronic Engineering, Foshan University, Foshan, China (e-mail: 153668@qq.com).

Jicong Wu is with State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

doi: 10.18178/ijiet.2020.10.7.1417

integers.

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

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

2. The term $a_{2^{K-1}}$, the intermediate term of the $2^K - 1$ terms left to a_{2^K}, is set to the left son of $N_{(0,0)}$; the term $a_{2^K + 2^{K-1}}$, the intermediate term of the $2^K - 1$ terms 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 'rule of intermediate term' to finish constructing the whole tree $T_{[a_1, a_u]}$.

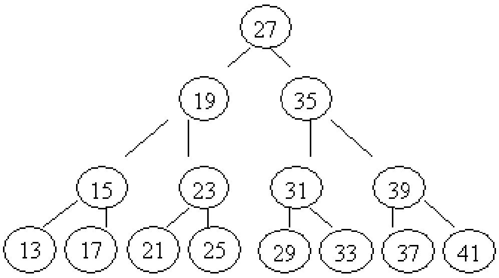


Fig. 1. A full perfect binary tree constructed from odd interval [13, 41].

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 Fig. 1.

For convenience, the tree constructed above is called a ***set tree***, simply denoted by T_S. If the set consists of integers in an interval, it is also called 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. If the interval $[x, y]$ is an odd one, the tree $T_{[x,y]}$ or T_I is also called an odd interval 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 $s < t$. By default, interval tree or set tree means a perfect tree and the set is an ordered one in this whole paper.

B. Lemmas

Lemma 1 (**In-order Traversal Restoration, see in [1]**) 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 $T_{[a_1, a_u]}$.

Lemma 2 (Node in In-order Traversal Restoration, see in [1]) 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 $0 \leq \omega \leq 2^i - 1$.

Lemma 3(See in [1]) 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 left son S_l, right son 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 sub -tree of S_r, it holds $G < n_{rl} < S_r$ and $G < S_r < n_{rr}$.

Lemma 4 (Calculation of Nodes, see in [1]). Let $K \geq 0$, $u = 2^{K+1} - 1$ be integers 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

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

Lemma 5(See in [6]) In a binary tree, nodes $N_{(k+1,2j)}$ and $N_{(k+1,2j+1)}$ on the (k+1)th level are respectively left son and right son of node $N_{(k,j)}$ on the kth level.

Lemma 6 (**See in [8]**) Let $\{x_i\}$ be a sequence of non-negative integers generated by

$x_i \equiv ax_{i-1} + c \pmod m$

Then the sequence has full period m provided that

(1) c is relatively prime to m;

(2) $a \equiv 1 \pmod p$ if p is a prime factor of m;

(3) $a \equiv 1 \pmod 4$ if 4 is a factor of m.

Particularly, if m is a power of 2, it suffices to have $a \equiv 1 \pmod 4$ and c odd.

III. New Fundamental Properties

Property 1. Let $K \geq 0$ be an integer, $u = 2^{K+1} - 1$, $S = \{a_1, a_2, ..., a_u\}$ be a set consisting of $2^{K+1} - 1$ terms and $T_S = T_{N_{(0,0)}}$ be the $N_{(0,0)}$-rooted set tree constructed with S; then

$$N_{(0,0)} = a_{2^K}$$

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

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

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

Particularly,

$$N_{(K,0)} = a_1,\ N_{(K,1)} = a_3,\ N_{(K,2)} = a_5, ..., N_{(K,2^{K-1}-1)} = a_{2^K-1},$$
$$N_{(K,2^{K-1})} = a_{2^K+1}, ..., N_{(K,2^K-1)} = a_{2^{K+1}-1}$$

Proof. See in Lemma 2.

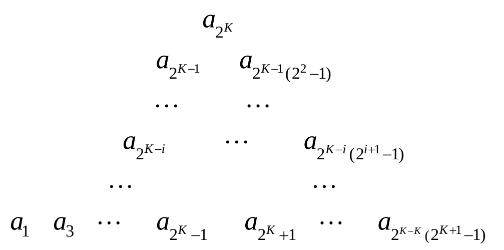


Fig. 2. A set tree constructed from $S = \{a_1, a_2, ..., a_u\}$.

Property 2. Let T_I be an odd interval tree with depth $K > 0$; then subtree $T_{N_{(i,\omega)}}$ contains $2^{K+1-i}-1$ nodes that originate from subinterval $[N_{(i,\omega)} - 2(2^{K-i}-1), N_{(i,\omega)} + 2(2^{K-i}-1)] \subset I$.

Proof. By definition, the depth of $T_{N_{(i,\omega)}}$ is $d = K - i$ because $N_{(i,\omega)}$ is on level i of T_I. Thereby, there are $2^{d+1}-1 = 2^{K-i+1}-1$ nodes on $T_{N_{(i,\omega)}}$. By Lemma 3 and Lemma 4, the smallest node and the biggest node are respectively

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

and

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

Since $T_{N_{(i,\omega)}}$ is also an odd interval tree, by Lemma 2, the proposition 1 holds.

Example 1. Take the odd interval tree in Fig. 1; it knows that

$$T_{19} = [19 - 2(2^{3-1}-1), 19 + 2(2^{3-1}-1)] = [13, 25]$$
$$T_{35} = [35 - 2(2^{3-1}-1), 35 + 2(2^{3-1}-1)] = [29, 41]$$
$$T_{15} = [15 - 2(2^{3-2}-1), 15 + 2(2^{3-2}-1)] = [13, 17]$$
$$T_{23} = [23 - 2(2^{3-2}-1), 23 + 2(2^{3-2}-1)] = [21, 25]$$
$$T_{31} = [31 - 2(2^{3-2}-1), 31 + 2(2^{3-2}-1)] = [29, 33]$$
$$T_{39} = [39 - 2(2^{3-2}-1), 39 + 2(2^{3-2}-1)] = [37, 41]$$

For a set tree, Property 2 is stated as the following Property 2*.

Property 2*. Let T_S be a set tree with depth $K > 0$; then subtree $T_{N_{(i,\omega)}}$ contains $2^{K+1-i}-1$ nodes.

Property 3. Let T_I be an odd interval tree with depth $K > 0$; then the node at position s on level l of the subtree $T_{N_{(i,\omega)}}$ is calculated by

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

Proof. Since $N_{(i,\omega)}$ is on level i of T_I, there are $K-i$ levels in $T_{N_{(i,\omega)}}$. Then referring to Lemma 4 directly yields the result.

Example 2. Look at the odd interval tree in Fig. 1 and take the node $N_{(1,0)} = 19$; it knows $i = 1, \omega = 0$ and

$$N_{(1,0)}^{19} = 19 - 2^{3+1-1-1}(2^1 - 2\times 0 - 1) = 15$$
$$N_{(1,1)}^{19} = 19 - 2^{3+1-1-1}(2^1 - 2\times 1 - 1) = 23$$
$$N_{(2,0)}^{19} = 19 - 2^{3+1-1-2}(2^2 - 2\times 0 - 1) = 13$$
$$N_{(2,1)}^{19} = 19 - 2^{3+1-1-2}(2^2 - 2\times 1 - 1) = 17$$
$$N_{(2,2)}^{19} = 19 - 2^{3+1-1-2}(2^2 - 2\times 2 - 1) = 21$$
$$N_{(2,3)}^{19} = 19 - 2^{3+1-1-2}(2^2 - 2\times 3 - 1) = 25$$

Property 3*. Let T_I be an odd interval tree with depth $K > 0$; then the node at position s on level l of the subtree $T_{N_{(i,\omega)}}$ is calculated by

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

Proof. By Property 3 and by Lemma 4, it holds

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

Again direct calculation by Lemma 4 yields

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

Therefore,

$$N_{(l,s)}^{N_{(i,\omega)}} = N_{(i+l,2^l\omega+s)}$$

Property 4. Let T_I be an interval tree with depth $K > 0$ and $N_{(i,\omega)}$ be a node of T_I with $i \geq 0$; then $N_{(i,\omega}$ is right next to subtree $T_{N_{(i+1,2\omega)}}$ and it left next to $T_{N_{(i+1,2\omega+1)}}$ in the interval restored from the in-order traversal restoration.

Proof. Considering by Lemma 5 that $N_{(i+1,2\omega)}$ and $N_{(i+1,2\omega+1)}$ are the left son and right son of $N_{(i,\omega)}$ respectively, as depicted with Fig. 3, and by Lemma 4, it knows the proposition holds.

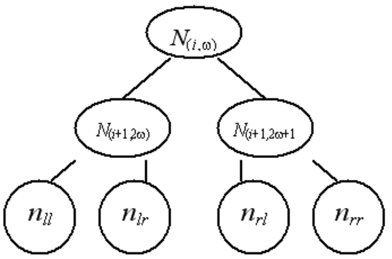


Fig. 3. Relationships among a node and its descendants.

Property 4 can be directly applied on the set tree as the following Property 3*.

Property 4*. Let T_S be a set tree with depth $K>0$ and $N_{(i,\omega)}$ be a node of T_S with $i \geq 0$; then $N_{(i,\omega)}$ is right next to subtree $T_{N_{(i+1,2\omega)}}$ and it left next to $T_{N_{(i+1,2\omega+1)}}$.

Property 5. Let T_I be an interval tree with depth $K>0$ and $N_{(i,\omega)}$ be a node of T_I with $i \geq 0$; then node $N^{N_{(i+1,2\omega)}}_{(K-i-1,2^{K-i-1}-1)} = N_{(K,2^{K-i}\omega+2^{K-i-1}-1)}$, the rightmost node on the bottom level of $T_{N_{(i+1,2\omega)}}$, is left next to $N_{(i,\omega)}$, and node $N^{N_{(i+1,2\omega+1)}}_{(K-i-1,0)} = N_{(K,2^{K-i}\omega+2^{K-i-1})}$ the leftmost node on the bottom level of $T_{N_{(i+1,2\omega+1)}}$, is right next to $N_{(i,\omega)}$ in the interval restored from the in-order traversal restoration. In another word, $N^{N_{(i+1,2\omega)}}_{(K-i-1,2^{K-i-1}-1)} = N_{(K,2^{K-i}\omega+2^{K-i-1}-1)}$, $N_{(i,\omega)}$ and $N^{N_{(i+1,2\omega+1)}}_{(K-i-1,0)} = N_{(K,2^{K-i}\omega+2^{K-i-1})}$ are three consecutive integers in the interval *I* restored from T_I by the in-order traversal restoration.

Proof. This property is directly derived from Property 3, Property 3*, Property 4 and the in-order traversal restoration of $T_{N_{(i,\omega)}}$.

What the property 5 says can be illustrated with Fig. 4.

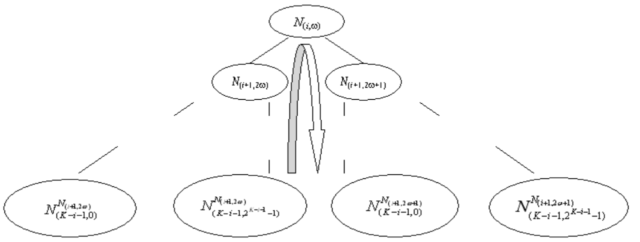

$N_{(i,\omega)}$
$N_{(i+1,2\omega)}$
$N_{(i+1,2\omega+1)}$
$N^{N_{(i+1,2\omega)}}_{(K-i-1,0)}$
$N^{N_{(i+1,2\omega)}}_{(K-i-1,2^{K-i-1}-1)}$
$N^{N_{(i+1,2\omega+1)}}_{(K-i-1,0)}$
$N^{N_{(i+1,2\omega+1)}}_{(K-i-1,2^{K-i-1}-1)}$

Fig. 4. Three consecutive nodes in the in-order traversal restoration of a tree.

Example 3. Look at the odd interval tree in Fig. 1 and take the node $N_{(1,0)}=19$ and it knows $i=1, \omega=0$, $K=3$; then

$$N_{(1+1,0)} = N_{(2,0)} = 15, N_{(1+1,2\times0+1)} = N_{(2,1)} = 23$$

$$N^{N_{(i+1,2\omega)}}_{(K-i-1,2^{K-i-2}-1)} = N^{N_{(2,0)}}_{(1,2^{3-1-1}-1)} = N^{N_{(2,0)}}_{(1,1)} = N_{(3,2^{3-1}\times0+2^{3-1-1}-1)} = N_{(3,1)} = 17$$

$$N^{N_{(i+1,2\omega+1)}}_{(K-i-1,0)} = N^{N_{(2,1)}}_{(1,0)} = N_{(3,2^{3-1}\times0+2^{3-1-1})} = N_{(3,2)} = 21$$

If take the node $N_{(2,2)}=31$, then $K=3, i=2, \omega=2$

$$N^{N_{(i+1,2\omega)}}_{(K-i-1,2^{K-i-1}-1)} = N^{N_{(3,4)}}_{(0,0)} = N_{(K,2^{K-i}\omega+2^{K-i-1}-1)} = N_{(3,2^{3-2}\times2+2^{3-2-1}-1)} = N_{(3,4)} = 29$$

$$N^{N_{(i+1,2\omega+1)}}_{(K-i-1,0)} = N^{N_{(3,5)}}_{(0,0)} = N_{(K,2^{K-i}\omega+2^{K-i-1})} = N_{(3,2^{3-2}\times2+2^{3-2-1})} = N_{(3,5)} = 33$$

IV. New Extensive Properties

The following corollaries are derived from the fundamental properties of the interval tree.

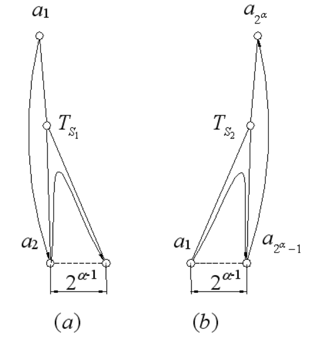
Corollary 1. Let *S* be a set consisting of 2^α consecutive integers (or odd integers) with $\alpha>0$ being an integer and *S* be embedded in an interval tree T_I whose depth is no less than α; then *S* occupies totally $\alpha+1$ levels in which α levels are consecutive from the bottom upwards and there are $2^{\alpha-1}$ terms of *S* on the bottom level of T_I.

Proof. Let

$$S=\{a_1, a_2, a_3, a_4 ..., a_{2^{\alpha-1}}, ..., a_{2^\alpha-1}, a_{2^\alpha}\}$$

and it be on an interval tree T_I (Notice: not T_S because S is embedded into T_I). Since $2^\alpha-1$ terms can form a perfect subtree of depth $\alpha-1$, there are several cases needed investigation by Properties 4 and 5.

Case 1. a_1 is an ancestor and $S_1=\{a_2, a_3, a_4 ..., a_{2^{\alpha-1}}, ..., a_{2^\alpha-1}, a_{2^\alpha}\}$ is a perfect right subtree such that a_2 is right next to a_1 in the in-order traversal restoration of T_I. This time, the depth of subtree T_{S_1} is $\alpha-1$ and there are $2^{\alpha-1}$ nodes on the bottom level of T_{S_1}, as depicted in Fig. 5(*a*), in which the curves with arrows mean the in-order traversal directions. Hence a_1 lies on a level upper than $\alpha-1$.



(*a*) (*b*)

Fig. 5. a_1 or a_{2^α} is ancestor.

Case 2. a_{2^α} is an ancestor and $S_2=\{a_1, a_2, a_3 ..., a_{2^{\alpha-1}}, ..., a_{2^\alpha-1}\}$ is a perfect left subtree such that $a_{2^\alpha-1}$ is left next to a_{2^α} in the in-order traversal restoration of T_I. This time, the depth of subtree T_{S_2} is $\alpha-1$ and there are $2^{\alpha-1}$ nodes on the bottom level of T_{S_2}, as depicted in Fig. 5(b). Hence a_{2^α} lies on a level upper than $\alpha-1$.

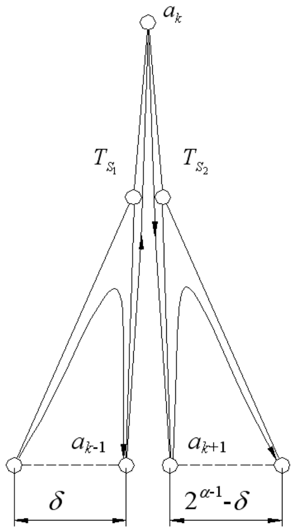


Fig. 6. a_k is ancestor.

Case 3. a_k with $k=2,3,...,2^\alpha-1$ is an ancestor and in the in-order traversal restoration of T_I, $S_1=\{a_1, a_2, a_3 ..., a_{k-1}\}$ is a left subtree (not necessarily perfect) such that a_{k-1} is left next to a_k while $S_2=\{a_{k+1},a_{k+2},...,a_{2^\alpha-1},a_{2^\alpha}\}$ is a right subtree (not necessarily perfect) such that a_{k+1} is right next to a_k, as illustrated in Fig. 6.

This time, it can prove that there are $2^{\alpha-1}$ nodes on the bottom of T_I and one of T_{S_1} and T_{S_2} has a depth no less than $\alpha-1$. This proof is established with three steps.

Firstly, apart from a_k, the other $2^\alpha-1$ nodes can form a perfect tree.

Secondly, consider the set $S^*=\{a_{k+1},a_{k+2},...,a_{2^\alpha-1},a_{2^\alpha},b_1,b_2,...,b_{k-1}\}$ that contains $2^\alpha-1$ terms, where $B=\{b_1,b_2,...,b_{k-1}\}$ are $k-1$ consecutive integers (or odd integers) following a_{2^α}, form a perfect interval tree as shown in Fig. 7. In the figure area *A* means the subtree for $A=\{a_{k+1},a_{k+2},...,a_{2^\alpha-1},a_{2^\alpha}\}$ and area B for $B=\{b_1,b_2,...,b_{k-1}\}$. For this reason, *B* is said to be a complementary of *A*. Of course, *A* can be said to be a complementary of *B*.

Thirdly, the subtree for $S=\{a_1,a_2,...,a_{k-1},a_k,a_{k+1,},...,a_{2^\alpha}\}$ is sure as Fig. 8, where the graph structure of T_1 is exact to be the graph structure of the complementary of T_2 .In another word, the tree T_1 is geometrically a transition of T T_2 's complementary and re-valuated.

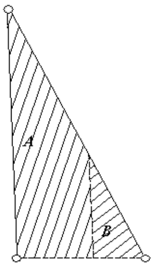


Fig. 7. Subtree and its complementary parts.

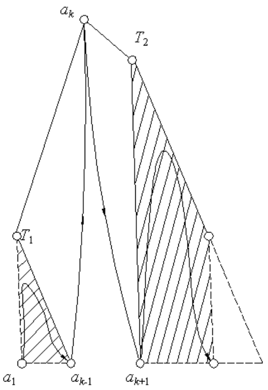


Fig. 8. Transition of a complementary subtree.

Example 2. See from Fig. 4, one can see that, in tree $T_{[65,125]}$, 4 consecutive odd integers, 65, 67, 69 and 71, occupy 3 levels with 65, 69 on the bottom level and 65, 67,69 on 2 consecutive levels while 4 consecutive odd integers, 91,93, 95, and 97 occupy 3 levels with 93, 97 at the bottom level and 91, 93 ,97 on 2 consecutive levels. It also can see that, 8 consecutive odd integers, 69,71,73,75,77,79,81 and 83 take 4 levels with 65,69,73,77 at the bottom level and 69,71,73,75, 77 on 3 consecutive levels; 8 consecutive odd integers, 89,91,93,95,97,99,101 and 103, occupy 4 levels with 89,93,97,101 at the bottom level and 89,93,97,101,91,99, 103 on 3 consecutive levels.

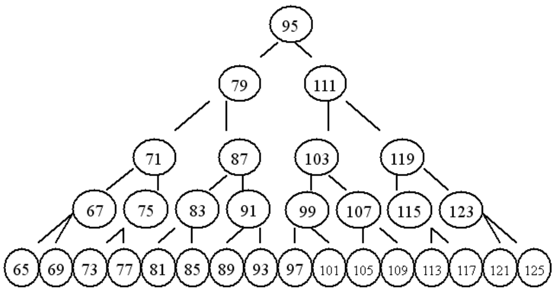


Fig. 9. Interval tree constructed from odd interval [65, 125].

Corollary 2. Let *S* be a subinterval consisting of 2^α terms that belong to an interval *I* that contains $2^{\alpha+\beta}-1$ terms, where $\alpha>0$ and $\beta>0$ are integers; then there must be a term of *S* lying on level $\beta-1$ or a upper level. Let *p* be the probability to pick successfully one term of *S*; then $\frac{1}{2^\beta}\le p\le\frac{1}{2^\beta-1}$.

Proof. Three are five ways to pick randomly the terms in *I*. The first one is to pick directly and randomly in the interval *I*. This time,

$$p_1=\frac{2^\alpha}{2^{\alpha+\beta}-1}>\frac{2^\alpha}{2^{\alpha+\beta}}=\frac{1}{2^\beta}$$

The other ways are based on constructing an interval tree T_I because $2^{\alpha+\beta}-1$ terms can form an interval tree of depth $\alpha+\beta-1$. By Corollary 1, one term of *S* is on level $\beta-1$ or an upper level of T_I and the other $2^\alpha-1$ ones are on the lower levels, as illustrated in Fig. 10. In the figure, the hatched area is the area where the ancestor of T_S possibly lies, and the triangular area S_ means the area where the other $2^\alpha-1$ terms of S lie.

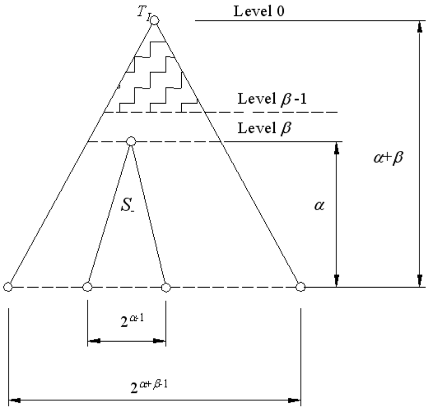


Fig. 10. 2^α terms are embedded in an interval tree of depth α+β-1.

Accordingly, the second way is to pick randomly on level $\beta-1$ or an upper level of T_I. Since level $\beta-1$ and its upper levels has $1+2+...+2^{\beta-1}=2^\beta-1$ nodes, it yields

$$p_2=\frac{1}{2^\beta-1}>\frac{1}{2^\beta}$$

Note that,

$$p_1 - p_2 = \frac{2^{\alpha}}{2^{\alpha+\beta}-1} - \frac{1}{2^{\beta}-1} = \frac{1-2^{\alpha}}{(2^{\alpha+\beta}-1)(2^{\beta}-1)}$$

which says $p_1 < p_2$ when $\alpha \ge 1$.

The third way is to pick randomly on level β. This time since there is merely one term on the level, it holds

$$p_3 = \frac{1}{2^{\beta}}$$

The fourth way is to pick randomly from level β to the bottom level. Since the number of total nodes of T_I from level β to the bottom is

$$2^{\beta} + 2^{\beta+1} + ... + 2^{\beta+\alpha-1} = 2^{\beta}(1+2+...+2^{\alpha-1}) = 2^{\beta}(2^{\alpha}-1)$$

and there are $2^{\alpha}-1$ terms in S, it holds

$$p_4 = \frac{2^{\alpha}-1}{2^{\beta}(2^{\alpha}-1)} = \frac{1}{2^{\beta}}$$

The last way is to pick randomly on level bottom of T_I. This time, the total nodes on the bottom level $2^{\alpha+\beta-1}$ and among the nodes there are $2^{\alpha-1}$ ones in S, accordingly

$$p_5 = \frac{2^{\alpha-1}}{2^{\alpha+\beta-1}} = \frac{1}{2^{\beta}}$$

Summarizing the above cases yields the corollary.

V. Searching Strategy

Now we look back at the problem raised in the introductory section. Conventionally, we have to subdivide the large interval into many small subintervals to perform the parallel search. For example, we can subdivide the $2^{\alpha+\beta}$ terms into 2^{β} subintervals each of which contains 2^{α} terms, and then search on each subinterval one by one. This strategy is a resource-consuming one because there are at most two subintervals containing the objective terms.

Fortunately, Corollary 2 tells us that, an interval tree can deposit half of the objective terms on the bottom level and another half on the levels over the bottom. Since each level except for the root contains 2^{χ} ($\chi > 0$) terms, it is certainly suitable to apply the Monte Carlos algorithm, as introduced in [4]. Of course, parallel computing can be surely performed on each independent level.

Meanwhile, by Property 1, it can see that, the objective terms are distributed as described in Table I.

TABLE I: Distribution of Objective Terms on Different Levels

Level	Total Terms	Objective Terms	Index trait
K	$2^{\alpha+\beta-1}$	$2^{\alpha-1}$	$a_{2\omega+1}$
K-1	$2^{\alpha+\beta-2}$	$2^{\alpha-2}$	$a_{4\omega+2}$
K-2	$2^{\alpha+\beta-3}$	$2^{\alpha-3}$	$a_{8\omega+4}$
……	……	……	……
$K-\alpha-\beta+1$	2^{β}	1	$a_{2^{\alpha+\beta-1}(2\omega+1)}$

It can see that, with the same probability, the total searched terms are reduced by 2’s power. This provides more possibilities for us to design the search algorithm. Our team have designed two different probabilistic searching algorithms on large interval searches, as shown in [7] and [8]. Here roughly introduce a search-by-level method as follows.

1) Search from level 1 to level *k* by parallel computing, where k is determined with the predefined Tolerance Time as proposed in [7].

2) From level *k* to level bottom – 1, perform the deterministic-embedded Monte Carlo approach, which was introduced in [8].

3) Choose a proper level, perform the deterministic -embedded Monte Carlo approach, which was introduced in [8].

VI. Conclusion and Future Work

A blind search or uninformed search, like the search problem that is raised in the introductory section, frequently occurs in cryptography and almost all of such searches require new theory and algorithm to realize. By disclosing the new properties of the interval tree, this paper shows that, the interval tree method might be a helpful attempt. However, frankly speaking, this does not eventually solve the problem because the attempt is still a resource-consuming one for a very large interval according our experiments although our method can reduce the number of searching steps by a half. How to improve the searching efficiency will be the future work. Hope more young people join this work.

Conflict of Interest

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

Author 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. Jicong Wu contributes 5% of the work.

Acknowledgment

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.

References

[1] X. Wang, “Interval tree and its application in integer factorization,” *Journal of Mathematics Research*, vol. 11, no. 2, pp. 103-113, 2019.

[2] W. Ertel, *Introduction to Artificial Intelligence*, Springer, 2017.

[3] X. Wang, “Two number-guessing problems plus applications in cryptography,” *International Journal of Network Security*, vol. 21, no. 3, pp. 498-504, 2019.

[4] T. E. Hull and A. R. Dobell, “Random number generator,” *SIAM Review*, vol. l4, no. 3, pp. 230-254, 1962.

[5] D. Mehta and S. Sahni, *Handbook of Data Structures and Applications*, Chapman & Hall/CRC, 2005.

[6] X. Wang, "Analytic formulas for computing LCA and path in complete binary trees," *International Journal of Scientific and Innovative Mathematical Research*, vol. 3, no. 4, pp. 1-8, 2015.

[7] J. Li, "A parallel probabilistic approach to factorize a semiprime," *American Journal of Computational Mathematics*, vol. 8, no. 2, pp. 175-183, 2018.

[8] X. Wang and J. Guo, "Deterministic-embedded monte carlo approach to find out an objective item in a large number of data sets," *International Journal of Applied Physics and Mathematics*, vol. 9, no. 4, pp. 173-181, 2019.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).

Xingbo Wang was born in Hubei, China. He got his master and doctor's degrees at National University of Defense Technology of China and had been a staff in charge of researching and developing CAD/CAM/NC technologies in the university. Since 2010, he has been a professor in Foshan University with research interests in computer application and information security. He is now the chief of Guangdong engineering center of information security for intelligent manufacturing system. Prof. WANG was in charge of more than 40 projects including projects from the National Science Foundation Committee, published 8 books and over 100 papers related with mathematics, computer science and mechatronic engineering, and invented 30 more patents in the related fields

Jicong Wu was born in Guangdong, China. He is a graduate student of Foshan University with research interests in mechatronics. Wu was in charge of 1 project from Foshan University, won the National fourth prize, and invented 2 patents in the related fields.