International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 3, Issue 4, April 2015, PP 1-8
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Analytic Formulas for Computing LCA and Path in Complete Binary Trees

WANG Xingbo

Department of Mechatronics
Foshan University, Foshan City
Guangdong Province, PRC
wxbmail@msn.com

Abstract: *The paper presents and proves several theorems that are related to calculating a lowest common ancestor (LCA) of nodes and a path connecting two nodes in a complete binary tree. The proved theorems involve in an analytic formula to calculate the LCA of two nodes, an analytic formula to compute a path connecting two nodes and a formula to estimate the bound of a path in a complete binary tree. Some other theorems and propositions that depict the distribution of nodes and their LCAs are also given. All the theorems and proposition are strictly deduced by means of mathematical deductions. The formulas and the related theorems can provide an analytic approach to analyze problems related to the LCA and the path in a complete binary tree.*

Keywords: *Discrete mathematics, Data structure, complete binary tree, Lowest common ancestor, Path*

1. Introduction

Computation of a path connecting two nodes in a graph or a tree has been a topic in both discrete mathematics and computer science even since 1950s. During the past more than half a century, people have developed a series of algorithms to find the shortest path, as the bibliographies [1]-[2] summarized and introduced, the longest path, as the bibliography [3] summarized, and other path-related problems [4]. Approximately estimating, there may be more than a thousand bibliographies that are involved in investigation of the issue.

When computing a path in a rooted-tree, one will inevitably met another old and fundamental problem: computation of an LCA of two or more nodes. Actually, computation of LCAs is the key to solve the problem of computing a path in a rooted-tree, as illustrated and stated in bibliographies [5]-[11].

Like the problem of path-computation, people have also developed different algorithms to LCA-computation, as introduced in bibliographies [12]-[14]. However, when one has a look into bibliographies of either path-computation or LCA computation, he will immediately see that, the a common trait appears in these bibliographies, that is to perform the computation by way of computer searching of recursions or iterations; few bibliographies give an analytic formula for the computations, even for that in a complete binary tree. This leaves inconvenience for us to evaluate and analyze the computing target by means of mathematics because computer searching is a post-known approach. That is, one cannot know of the computing result unless the search is done. Hence the computational target cannot be evaluated before a search. On the contrary, an analytic formula for a computation can provide a pre-known effect: an outline or even an exact range of the computational target can be possibly known by the formula.

It is just from this point of view that I present mathematically properties and analytic conditions for analyzing LCAs in bibliography [15] and I present in this paper further and more exact analytics formulas for concrete computations of LCAs and paths in a complete binary tree.

This paper is composed in 4 parts. The first part is this introduction, the second part contains the required preliminaries, the third part includes main results and their proofs, which involve in several theorems and propositions that illustrate detail mathematical relations related to LCAs and paths in a complete binary tree, and the last part is a conclusion of the whole paper.

2. PRELIMINARIES

We first need to introduce some notations and some lemmas.

2.1 Definitions and Symbols

Definitions related with binary trees can be seen by certain entries in books [8] and [9]. In this paper, a node of a binary tree refers to either a vertex or a leaf; we assume the depth of the complete binary tree we study is *h* and all the nodes we investigate are valid. An *h*-leveled full binary tree is a complete binary tree that has $2^h - 1$ nodes. If each node of a binary tree is encoded by natural number 1,2,…,*n* by the way from top to bottom and from left to right, then the code of the node is called a natural code (NC), and the tree is said to be a NC-coded tree.

We use symbol $N_{(k,j)}$ to express the node at the *j*-th position on the *k*-th level (*k*>0) of a binary tree, $G_{(k,i)}^{(l,j)}$ to express the lowest common ancestor (LCA) of $N_{(k,i)}$ and $N_{(l,j)}$, and $l(N)$ to be number of the level on which node *N* lies, for example, $l(N_{(k,j)}) = k$. Symbol T_N is to express the subtree with node *N* being its root. Two nodes $N_{(k,i)}$ and $N_{(l,j)}$ such that $l(N_{(l,j)}) > l(N_{(k,i)})$ are said to be co-path if they lie on the same path from the root of the binary tree to $N_{(l,j)}$ otherwise they are called un-co-path. Symbol $\lfloor x \rfloor$ is to express the floor function defined by $x-1 < \lfloor x \rfloor \le x$, where *x* is a real number, and symbol $\{x\}$ is the decimal function such $\{x\}=x-\lfloor x \rfloor$. Symbol $(\alpha)_{10}$ is the decimal representation of integer α and $(\alpha)_2$ is α's binary representation. Symbol $z(\alpha)$, called *z*-function, which is defined in [16] represents the position of the first 0-bit that occurs from the least significant bit (lsb) of α's binary representation, e.g., $z(0) = z((00000000)_2) = 1$; $z(1) = z((00000001)_2) = 2$; $z(83) = z((01010011)_2) = 3$. Symbols $\wedge$, <<, and >> are respectively operations of exclusive OR operation, left shift and right shift.

2.2 Lemmas

Lemma1 ([17]). For a NC coded complete binary tree *T*, node α's left son and right son are respectively 2α and $2\alpha+1$; the father of arbitrary node β (β>1) is $\lfloor \beta/2 \rfloor$; The *k*-th (*k*≥1) level of *T* has at most 2^{k-1} nodes; the code of the first node on the *k*-th level is 2^{k-1}. The level where node *N* lies is $\lfloor \log_2 N \rfloor + 1$. There is a unique path connecting two nodes in *T*. If *r* is the root of *T* and γ is a sibling node, then $Path(r, \gamma)=l(\gamma)$.

Lemma2 ([8][9][10][11]). Let *A* and *B* be two un-co-path nodes in a complete binary tree, then *LCA*(*A*,*B*) is the root of the minimum size subtree which contains *A* and *B* and it holds

$$Path(A,B) = Path(A, LCA(A,b)) + Path(LCA(A,B),B) .$$

Remark. The statement "*LCA*(*A*,*B*) is the root of the minimum size subtree which contains *A* and *B*" is seen in bibliography [11]. The formula "*Path*(*A*, *B*)=*Path*(*A*,*LCA*(*A*,*B*))+ *Path*(*LCA*(*A*,*B*),*B*)" is inferred from [8], [9] and [10].

Lemma3 ([18][19]). For any real *x* and integers *n*: $\lfloor x \rfloor + \left\lfloor x + \frac{1}{2} \right\rfloor = \lfloor 2x \rfloor$; for any integer *n* and real *x*: $\lfloor n + x \rfloor = n + \lfloor x \rfloor$. For any real *x* and *y*, $x \le y$ yields $\lfloor x \rfloor \le \lfloor y \rfloor$ and $x \ge y$ yields $\lfloor x \rfloor \ge \lfloor y \rfloor$.

Lemma4 ([19]). Total valid bits of positive integer α's binary representation is $\lfloor \log_2 \alpha \rfloor + 1$.

Lemma5 ([20]). The radix complement of an *n* digit number *y* in radix *b* is $b^n - y$, and its diminished radix complement is $b^n - 1 - y$.

Lemma6([15]). Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(1 < k \le h, 2^{k-1} \le \alpha < \beta < 2^k)$ be two nodes on the *k*-th level of a complete binary tree; if *I* is the smallest positive integer that fits the equation $\left\lfloor \frac{\alpha}{2^i} \right\rfloor = \left\lfloor \frac{\beta}{2^i} \right\rfloor$ of integer unknown *i* and $\sigma = \left\lfloor \frac{\alpha}{2^I} \right\rfloor$, then $N_{(k-I,\sigma)}$ is the LCA of $N_{(k,\alpha)}$ and $N_{(k,\beta)}$.

Lemma7([15]). Let $N_{(k,i)}$ and $N_{(l,j)}$ $(1<k<l\le h, 2^{k-1}\le i<2^k, 2^{l-1}\le j<2^l)$ be two nodes in a complete binary tree; then the two share a common direct ancestor (or on the same path) if and only if there exists a positive integer σ such that $i=\left\lfloor \frac{j}{2^{\sigma}} \right\rfloor$ and $k=l-\sigma$, and then $N_{(k,i)}$ is the LCA of $N_{(k,i)}$ and $N_{(l,j)}$.

Lemma8 ([15]). Let $N_{(m,\alpha)}$ and $N_{(n,\beta)}$ $(1<m\le n\le h, 2^{m-1}\le\alpha<2^m, 2^{n-1}\le\beta<2^n)$ be two nodes in a complete binary tree; then $G_{(m,\alpha)}^{(m,\gamma)}=G_{(m,\alpha)}^{(n,\beta)}$, where $\gamma=\left\lfloor \frac{\beta}{2^{n-m}} \right\rfloor$.

Lemma9 ([15]). Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(1<k\le h, 2^{k-1}\le\alpha<\beta<2^k)$ be two nodes on the *k*-th level of a complete binary tree, and σ, χ respectively satisfy $\sigma=\left\lfloor \frac{\alpha}{2^{z(\alpha)}} \right\rfloor$,

$$(\chi-\frac{1}{2})\times 2^{z(\alpha)}<\beta-\alpha\le(\chi+\frac{1}{2})\times 2^{z(\alpha)}$$

then $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ share their LCA with $N_{(k-z(\alpha),\sigma)}$ and $N_{(k-z(\alpha),\sigma+\chi)}$, namely, $G_{(k,\alpha)}^{(k,\beta)}=G_{(k-z(\alpha),\sigma)}^{(k-z(\alpha),\sigma+\chi)}$.

3. MAIN RESULTS AND PROOFS

Main results of this paper include two parts: the first part is some theorems and propositions to describe distribution of nodes and their LCAs in a complete binary tree and the second part is several theorems that describe a path connecting two nodes.

3.1. Theorems of Nodes and LCAs

Theorem1. A full binary tree is a geometrically symmetric in a way that on the *k*-th level ($1<k$), two nodes $N_{(k,i)}, N_{(k,j)}$ that satisfy $i+j=2^k+2^{k-1}-1$ and $i\le j$ locate at symmetric positions, namely, the distance of $N_{(k,i)}$ from the leftmost node is equal to the distance of $N_{(k,j)}$ from the rightmost node, or vice versa.

Proof. By Lemma 1, $N_{(k,2^{k-1})}$ is the leftmost node and $N_{(k,2^k-1)}$ is the rightmost node on the *k*-th level. Let $i=2^{k-1}+\alpha, j=2^k-1-\beta$ where $0\le\alpha,\beta\le 2^{k-1}$ are integers; then $i+j=2^{k-1}+2^k-1+\alpha-\beta=3\times 2^{k-1}-1+\alpha-\beta$. Since by the given condition $i+j=2^k+2^{k-1}-1$, it yields $3\times 2^{k-1}-1+\alpha-\beta=2^k+2^{k-1}-1\Rightarrow\alpha-\beta=0$, namely, $\alpha=\beta$. That means that the distance from $N_{(k,i)}$ to the leftmost node is equal to the distance from $N_{(k,j)}$ to the rightmost node. Thus the lemma holds.□

Theorem2. The fathers of two symmetric nodes $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(k>1, 2^{k-1}\le\alpha,\beta<2^k)$ that lie on the *k*-th level of a complete binary tree are also respectively symmetric on the (*k*-1)-th level of the tree. Furthermore, let *T* be a complete binary tree; then the LCAs of *T*'s nodes that lie at symmetric positions are also symmetric.

Proof. We first prove the first conclusion since the second conclusion is just a proposition of the first one. Without loss of generality, we assume that $\alpha=2^{k-1}+\delta$ and $\beta=2^k-1-\delta$, where δ is an integer such that $0\le\delta<2^{k-1}$. By Lemma 1, fathers of $N_{(k,\alpha)}, N_{(k,\beta)}$ are respectively $N_{(k-1,\lfloor\alpha/2\rfloor)}$ and $N_{(k-1,\lfloor\beta/2\rfloor)}$. Then we need to prove that $N_{(k-1,\lfloor\alpha/2\rfloor)}$ and $N_{(k-1,\lfloor\beta/2\rfloor)}$ are symmetric. By Theorem 1, this only requires to prove that $\left\lfloor\frac{\alpha}{2}\right\rfloor+\left\lfloor\frac{\beta}{2}\right\rfloor=2^{k-1}+2^{k-2}-1$. In fact, we have (by Lemma 3)

$$\left\lfloor\frac{\alpha}{2}\right\rfloor+\left\lfloor\frac{\beta}{2}\right\rfloor=\left\lfloor\frac{2^{k-1}+\delta}{2}\right\rfloor+\left\lfloor\frac{2^k-1-\delta}{2}\right\rfloor=\left\lfloor\frac{2^{k-1}+\delta}{2}\right\rfloor+\left\lfloor\frac{2^{k-1}+\delta}{2}+\frac{1}{2}+2^{k-2}-1-\delta\right\rfloor$$
$$=\left\lfloor\frac{2^{k-1}+\delta}{2}\right\rfloor+\left\lfloor\frac{2^{k-1}+\delta}{2}+\frac{1}{2}\right\rfloor+2^{k-2}-1-\delta=\left\lfloor 2\times\frac{2^{k-1}+\delta}{2}\right\rfloor+2^{k-2}-1-\delta$$
$$=2^{k-1}+2^{k-2}-1$$

Hence Theorem 2 holds.

Theorem3. Let T be an h-leveled and NC-coded full binary tree and $N_{(m,\alpha)}$ ($m \geq 1$, $2^{m-1} \leq \alpha < 2^m$) be the root of subtree $T_{N_{(m,\alpha)}}$; suppose α's binary representation is $\alpha = (\alpha_m \alpha_{m-1} ... \alpha_2 \alpha_1)_2$, then the binary representations of the 2^{k-1} nodes on the k-th level of $T_{N_{(m,\alpha)}}$ are

$$(\alpha_m ... \alpha_2 \alpha_1 \underbrace{0...0}_{k})_2, (\alpha_m ... \alpha_2 \alpha_1 \underbrace{0...0}_{k-1}1)_2, (\alpha_m ... \alpha_2 \alpha_1 \underbrace{0...0}_{k-2}10)_2, (\alpha_m ... \alpha_2 \alpha_1 \underbrace{0...0}_{k-2}11)_2, ..., (\alpha_m ... \alpha_2 \alpha_1 \underbrace{1...1}_{k})_2$$

Where the positions of the 0s and the 1s subordinate to the following binomial array as figure 1 depicts.

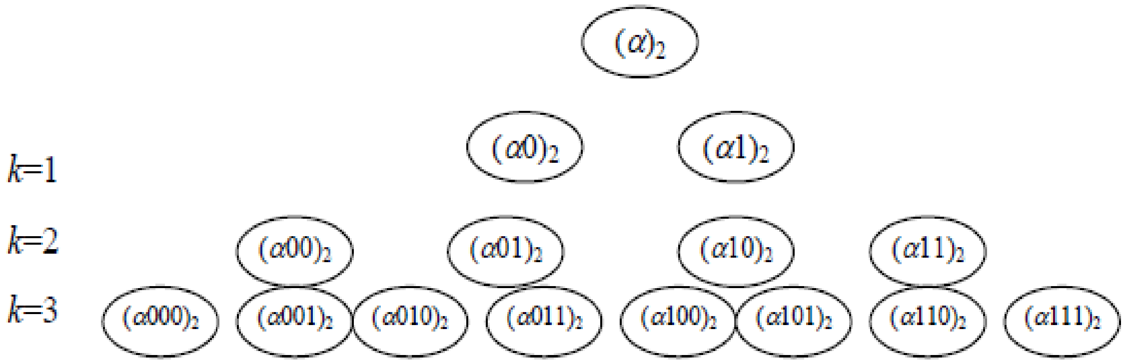

Figure1. *Binomial array of positions of the 0s and the 1s*

Proof. By Lemma 1, $N_{(m,\alpha)}$'s left son and right son are respectively 2α and $2\alpha+1$. Since 2α 's binary representation is a left shift of α's, namely, $2\alpha = \alpha << 1$, hence $(2\alpha)_{10} = (\alpha_m \alpha_{m-1} ... \alpha_2 \alpha_1 0)_2$ and $(2\alpha+1)_{10} = (\alpha_m \alpha_{m-1} ... \alpha_2 \alpha_1 1)_2$. Likewise, the others nodes are obtained in a recursive procedure.

Proposition1. Let T be an h-leveled and NC coded full binary tree and $N_{(m,\alpha)}$ ($m \geq 1$, $2^{m-1} \leq \alpha < 2^m$) be the root of subtree $T_{N_{(m,\alpha)}}$; then the 2^{k-1} nodes on the k-th level of $T_{N_{(m,\alpha)}}$ are $2^k \alpha, 2^k \alpha + 1, ..., 2^k \alpha + j, ..., 2^k \alpha + 2^{k-1} - 1$.

Proposition2. Two nodes in a complete binary tree share a common ancestor if and only if they have common bits in their binary representations from the most significant bit (msb). The more common bits they have, the lower level their common ancestors lie.

Proof. We first consider two arbitrary nodes on the same level of the tree. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(1 < k \leq h, 2^{k-1} \leq \alpha < \beta < 2^k)$ be two nodes on the k-th level of a complete binary tree; then Lemma 6 shows that the smallest positive integer I such that fits the equation $\left\lfloor \frac{\alpha}{2^i} \right\rfloor = \left\lfloor \frac{\beta}{2^i} \right\rfloor$ of integer unknown i and $\sigma = \left\lfloor \frac{\alpha}{2^I} \right\rfloor$ determine the node $N_{(k-I,\sigma)}$ to be the LCA of $N_{(k,\alpha)}$ and $N_{(k,\beta)}$. Let the binary representations of α and β are respectively $\alpha = (\alpha_k \alpha_{k-1} ... \alpha_2 \alpha_1)_2$ and $\beta = (\beta_k \beta_{k-1} ... \beta_2 \beta_1)_2$; then by meaning of right shift of bitwise operation it yields $\left\lfloor \frac{\alpha}{2^i} \right\rfloor = \alpha >> i = (\underbrace{0...0}_{i} \alpha_k \alpha_{k-1} ... \alpha_{i+1})_2$, $\left\lfloor \frac{\beta}{2^i} \right\rfloor = \beta >> i = (\underbrace{0...0}_{i} \beta_k \beta_{k-1} ... \beta_{i+1})_2$. Hence the equation $\left\lfloor \frac{\alpha}{2^i} \right\rfloor = \left\lfloor \frac{\beta}{2^i} \right\rfloor$ says α and β must have common bits in their binary representations from the msb. On the other hand, it is obvious that the number i plays two roles in the equation $\left\lfloor \frac{\alpha}{2^i} \right\rfloor = \left\lfloor \frac{\beta}{2^i} \right\rfloor$:one is to count the times of right shift operation, and the other is to record the number that $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ trace their LCA. Therefore, the smaller the number i is, the more common bits α and β have and the faster $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ find their LCA.

Next by Lemma 7 and in the same way, we can prove proposition holds in the case that two nodes are on different level of the tree. We just omit the details here.

Proposition3. Let $N_{(m,\alpha)}$ and $N_{(m,\beta)}$ $(1 < m \le n \le h, 2^{m-1} \le \alpha < \beta < 2^m)$ be two nodes on the *m*-th level of a complete binary tree; then $LCA(N_{(m,\alpha)}, N_{(m,\beta)}) = N_{(I,\gamma)}$, where $I = m - (\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1)$ and $\gamma = \alpha >> I$.

Proof. From the analytic process of proving Proposition 2 and by Theorem 3, it knows that the LCA of $N_{(m,\alpha)}$ and $N_{(m,\beta)}$ is actually determined by the maximal common bits that α and β have in their binary representations from the msb. Hence it is necessary to find out the number of the maximal common bits. Without loss of generality, we suppose that α and β have x common bits and write them by $\alpha = (c_1c_2...c_x\alpha_{m-x}...\alpha_2\alpha_1)_2$ and $\beta = (c_1c_2...c_x\beta_{m-x}...\beta_2\beta_1)_2$, where $c_i (1 \le i \le x)$ is the common bit. Then by definition of exclusive OR operation, it yields $\alpha \wedge \beta = (\underbrace{0...0}_{x}\chi_{m-x}\cdots\chi_2\chi_1)_2$, where $\chi_i = \alpha_i \wedge \beta_i$ $(1 \le i \le m-x)$ is 0 or 1. This and Lemma 4 lead to $x = m - (\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1)$. Since $(c_1c_2...c_x)_2 = \alpha >> (m-x)$, it holds $LCA(N_{(m,\alpha)}, N_{(m,\beta)}) = N_{(I,\gamma)}$, as the proposition claims. □

Proposition4. Let $N_{(m,\alpha)}$ and $N_{(n,\beta)}$ $(1 < m \le n \le h, 2^{m-1} \le \alpha < 2^m, 2^{n-1} \le \beta < 2^n)$ be two nodes in a complete binary tree; then $LCA(N_{(m,\alpha)}, N_{(n,\beta)}) = N_{(I,\chi)}$, where $I = m - (\lfloor \log_2(\alpha \wedge (\beta >> (n-m))) \rfloor + 1)$ and $\chi = \alpha >> I$.

Proof. By Lemma 1, it yields $m = \lfloor \log_2 \alpha \rfloor + 1$ and $n = \lfloor \log_2 \beta \rfloor + 1$ .Suppose the binary representations of α and β are respectively $\alpha = (\alpha_m\alpha_{m-1}...\alpha_2\alpha_1)_2$ and $\beta = (\beta_n\beta_{n-1}...\beta_2\beta_1)_2$. Let $\sigma = n - m$ and $\gamma = \left\lfloor \frac{\beta}{2^\sigma} \right\rfloor$; then $\gamma = \beta >> (n-m)$ $= (\underbrace{\beta_n\beta_{n-1}...\beta_{n-\sigma+1}}_{m})_2$ and $N_{(n-\sigma,\gamma)}$ is on the *m*-th level. By proposition 3, $LCA(N_{(m,\alpha)}, N_{(m,\gamma)}) = N_{(I,\chi)}$, where $I = m - (\lfloor \log_2(\alpha \wedge \gamma) \rfloor + 1)$ and $\chi = \alpha >> I$. Since by Lemma 8, $G_{(m,\alpha)}^{(m,\gamma)} = G_{(m,\alpha)}^{(n,\beta)}$, it yields $LCA(N_{(m,\alpha)}, N_{(m,\beta)}) = N_{(I,\chi)}$.

Theorem4. Two different nodes $N_{(k,\alpha)}$, $N_{(k,\beta)}$ $(k > 1)$ on the *k*-th level of a complete binary tree fit $\alpha \wedge \beta \le 2^{k-1} - 1$.

Proof. By Theorem 3, an arbitrary node $N_{(k,j)} (1 \le k \le h, 2^{k-1} \le j < 2^k)$ that lies on the *k*-th level implies that $j = 2^{k-1} + \delta$ where is an integer such that $0 \le \delta < 2^{k-1}$, hence *j*'s binary representation is $j = (0...01\underbrace{n_{k-1}\cdots n_2n_1}_{k-1})_2$, where $n_i (1 \le i \le k-1)$ is 0 or 1. Therefore, without loss of generality, we assume $\alpha = (0...01\underbrace{\alpha_{k-1}...\alpha_2\alpha_1}_{k-1})_2$ and $\beta = (0...01\underbrace{\beta_{k-1}...\beta_2\beta_1}_{k-1})_2$, where $\alpha_i, \beta_i (1 \le i \le k-1)$ is 0 or 1. Then it leads to $\alpha \wedge \beta = (0..00\underbrace{\chi_{k-1}\cdots\chi_2\chi_1}_{k-1})_2$ where $\chi_i = \alpha_i \wedge \beta_i (1 \le i \le k-1)$ is 0 or 1. Hence $\alpha \wedge \beta \le (0...00\underbrace{1\cdots11}_{k-1})_2 = 2^{k-1} - 1$.

Theorem5. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(k > 1, 2^{k-1} \le \alpha, \beta < 2^k)$ be two nodes in a complete binary tree; suppose $2^{k-1} \le \alpha < 2^{k-1} + 2^{k-2}$ and $2^{k-1} + 2^{k-2} \le \beta < 2^k$, then $\alpha \wedge \beta \ge 2^{k-2}$.

Proof. Without loss of generality, we assume that $\alpha = 2^{k-1} + \delta_\alpha$, $\beta = 2^{k-1} + 2^{k-2} + \delta_\beta$, where $0 \le \delta_\alpha, \delta_\beta < 2^{k-2}$; then the binary representations of δ_α and δ_β are respectively $\delta_\alpha = (0...0\underbrace{\delta_{k-3}^\alpha...\delta_2^\alpha\delta_1^\alpha}_{k-3})_2$, $\delta_\beta = (0...0\underbrace{\delta_{k-3}^\beta...\delta_2^\beta\delta_1^\beta}_{k-3})_2$. Consequently, it yields $\alpha = (0...010\underbrace{\delta_{k-2}^\alpha...\delta_2^\alpha\delta_1^\alpha}_{k-3})_2$, $\beta = (0...011\underbrace{\bar{\delta}_{k-2}^\beta...\bar{\delta}_2^\beta\bar{\delta}_1^\beta}_{k-3})_2$ and $\alpha \wedge \beta = (0..01\underbrace{\chi_{k-1}\cdots\chi_2\chi_1}_{k-3})_2$, where $\chi_i = \delta_i^\alpha \wedge \delta_i^\beta (1 \le i \le k-3)$ is 0 or 1. Obviously, $\alpha \wedge \beta = (0..01\underbrace{\chi_{k-1}\cdots\chi_2\chi_1}_{k-3})_2 = 2^{k-2} + \chi \ge 2^{k-2}$

Proposition5. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(k > 1)$ be two symmetric nodes on the *k*-th level of a complete binary tree, then $\alpha \wedge \beta = 2^{k-1} - 1$.

Proof. Without loss of generality, we assume that $\alpha = 2^{k-1} + \delta$, $\beta = 2^k - 1 - \delta$, $0 \le \delta < 2^{k-1}$ and δ's binary representation is $\delta = (0\underbrace{\delta_{k-1}...\delta_2\delta_1}_{k-1})_2$, where $\delta_i (1 \le i \le k-1)$; then α's binary representation is $\alpha = (0...01\underbrace{\delta_{k-1}...\delta_2\delta_1}_{k-1})_2$. Considering $\beta = 2^k - 1 - \delta = 2^{k-1} + 2^{k-1} - 1 - \delta$, by Lemma 5, the part $2^{k-1} - 1 - \delta$ is the one's complement of δ; hence $\beta = (0...01\bar{\delta}_{k-1}...\bar{\delta}_2\bar{\delta}_1)_2$, where $\bar{\delta}_i (1 \le i \le k-1)$ is the one's complement of δ_i Consequently, it yields $\alpha \wedge \beta = (0...01\underbrace{\delta_{k-1}...\delta_2\delta_1}_{k-1})_2 \wedge (0...01\underbrace{\bar{\delta}_{k-1}...\bar{\delta}_2\bar{\delta}_1}_{k-1})_2 =$ $(0...00\underbrace{1...11}_{k-1})_2 = 2^{k-1} - 1$.

3.2. Theorems of Path-computation

Theorem6. Let $N_{(m,\alpha)}$ and $N_{(n,\beta)}$ $(1 < m \le n \le h, 2^{m-1} \le \alpha < 2^m, 2^{n-1} \le \beta < 2^n)$ be two co-pathed nodes in a complete binary tree, then $Path(N_{(m,\alpha)}, N_{(n,\beta)}) = l(N_{(n,\beta)}) - l(N_{(m,\alpha)}) = n - m$.

Proof. (omitted).

Theorem7. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ be two nodes on the *k*-th level of a complete binary tree, then $l(LCA(N_{(k,\alpha)}, N_{(k,\beta)})) = k - (\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1)$, and hence

$$Path(N_{(k,\alpha)}, LCA(N_{(k,\alpha)}, N_{(k,\beta)})) = Path(N_{(k,\beta)}, LCA(N_{(k,\alpha)}, N_{(k,\beta)})) = \lfloor \log_2(\alpha \wedge \beta) \rfloor + 1,$$

and

$$Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2(\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1).$$

Proof. The first formula is shown in proposition 3. The second one is drawn from the first one and Lemma 2.

Proposition6. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(1 < k \le h, 2^{k-1} \le \alpha < \beta < 2^k)$ be two nodes on the *k*-th level of a complete binary tree; then $2z(\alpha) \le Path(N_{(k,\alpha)}, N_{(k,\beta)}) \le 2(k-1)$.

Proof. First, by Lemma 2 and Lemma 9, it knows that,

$$Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2Path(N_{(k,\alpha)}, LCA(N_{(k,\alpha)}, N_{(k,\beta)})) \ge 2z(\alpha).$$

Then by Theorem 7, it yields

$$Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2(\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1).$$

Now the Proposition 5 says $\alpha \wedge \beta \le 2^{k-1} - 1$, which means $\alpha \wedge \beta$ must be identical to a node on the ρ-th level such that $1 \le \rho \le (k-1)$. By Lemma 1, it yields

$\lfloor \log_2(\alpha \wedge \beta) \rfloor \le k-2$. Hence $Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2(\lfloor \log_2(\alpha \wedge \beta) \rfloor + 1) \le 2(k-1)$

Proposition7. Let $N_{(k,\alpha)}$ and $N_{(k,\beta)}$ $(k > 1, 2^{k-1} \le \alpha, \beta < 2^k)$ be two nodes in a complete binary tree; suppose $2^{k-1} \le \alpha < 2^{k-1} + 2^{k-2}$ and $2^{k-1} + 2^{k-2} \le \beta < 2^k$; then $Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2(k-1)$.

Proof. First we know, by Theorem 4, that $\alpha \wedge \beta \le 2^{k-1} - 1$ holds for arbitrary $N_{(k,\alpha)}$ and $N_{(k,\beta)}$. And we also know, by Theorem 5, that $\alpha \wedge \beta \ge 2^{k-2}$. That is to say, $\alpha \wedge \beta$ must be identical to a node on the (*k*-1)-th level of the tree. Since $\alpha \ne \beta$, by Lemma 1, it yields $\lfloor \log_2(\alpha \wedge \beta) \rfloor = k-2$. This immediately leads to $Path(N_{(k,\alpha)}, N_{(k,\beta)}) = 2(k-1)$.

Theorem8. Let $N_{(m,\alpha)}$ and $N_{(n,\beta)}$ $(1 < m \le n \le h, 2^{m-1} \le \alpha < 2^m, 2^{n-1} \le \beta < 2^n)$ be two arbitrary un-co-pathed nodes in a complete binary tree; then

$$Path(N_{(m,\alpha)}, N_{(n,\beta)}) = n - m + 2(\lfloor \log_2(\alpha \wedge (\beta >> (n-m))) \rfloor + 1).$$

Proof. (Omitted).

4. CONCLUSION

The proved 8 theorems together with their 7 propositions form a complete system to calculate and estimate the LCAs and the paths in a complete binary tree. Theorem 1 and 2 establish the symmetric characteristics of a complete binary tree; theorem 3 presents the intrinsic relation between nodes' binary representations and their LCAs, showing that the LCA of two nodes is intrinsic and invariant in a complete binary tree; theorems 4 and 5 point out the distributive range of two nodes' LCA, laying a way to estimate a path connecting two nodes. Theorems 6,7 and 8 provide an approach to evaluate by means of mathematic deduction a path connecting two nodes in a complete binary tree. With these theorems and propositions, one can exactly calculate the LCAs and paths in a complete binary tree. Since a complete binary tree is an important content of discrete mathematics and graph theory, the study of this article can be a reference to discrete mathematics and graph theory.

In the end, it is worth to point out that, the most valuable contribution of this paper and bibliography [15] is that, I try to form an idea of *rational algorithm design*: design of algorithms based on mathematical deductions, not by some *observational* results like that in many books such as in [1], [21]-[23]. As a professor of engineering, I know very well that students of engineering always make such a mistake that they take their observations as a fundamental principle (or axiom) to set up the frame of their studies. Therefore, I prove each theorem and each proposition through strict mathematical deductions so as to set up a sample for students of engineering.

ACKNOWLEDGEMENTS

The research work is supported by the national Ministry of science and technology under project 2013GA780052, Department of Guangdong Science and Technology under projects 2012B011300068, Foshan Bureau of Science and Technology under projects 2013AG10007, Special Innovative Projects from Guangdong Education Department, and Chancheng government under projects 2013B1018 and 2013A1021. The authors sincerely present thanks to them all.

REFERENCES

[1] Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Introduction to Algorithms(3rd edtion), MIT Press,2009,ch6, pp588-683

[2] SONG Qing WANG Xiao-fan, Survey of Speedup Techniques for Shortest Path Algorithms, Journal of University of Electronic Science and Technology of China, 41(2), 176(2012)

[3] WANG Jian-xin, YANG Zhi-biao, CHEN Jian-er, Algorithms for Longest Path :A Survey, Computer Science(In Chinese),36(12), 1(2009)

[4] Xinbo Gao,Bing Xiao,Dacheng Tao,Xuelong Li, A survey of graph edit distance,THEORETICAL ADVANCES,Pattern Anal Applic, 13,113 (2010)

[5] Toru Kasai,Hiroki Arimura,Setsuo Arikawa,Efficient Substring Traversal with Suffix Arrays,DOI Technical Report. 185, 2001. Department of Informatics, Kyushu University

[6] Jeanne Ferrante,Karl J. Ottenstein, Joe D. Warren, The Program Dependence Graph and Its Use in Optimization,ACM Transactions on Programming Languages and Systems, 9(3),319(1987)

[7] Anxiao Jiang, Network Coding for Joint Storage and Transmission with Minimum Cost,Proc. IEEE International Symposium on Information Theory (ISIT'2006), pp1359-1363

[8] Dinesh P. Mehta, Sartaj Sahni, Handbook of Data Structures and Applications, Chapman & Hall/CRC Computer and Information Science Series,pp3.1-3.18,pp11.4,(2004)

[9] Michael Gertz, April Kwong, Charles U. Martel, Glen Nuckolls, Databases that tell the Truth: Authentic Data Publication,Bulletin of the Technical Committee on Data Engineering, 27(1),1(2004)

[10] Michel Habib, Christophe Paul, A Simple Linear Time Algorithm for Cograph Recognition, Discrete Applied Mathematics, 145, 183(2005)

[11] Maxime Crochemore, Costas S. Iliopoulos, Manal Mohamed,Marie-France Sagot,Longest repeats with a block of k don't cares,Theoretical Computer Science 362,248 (2006)

[12] Fischer J., Heun V., Theoretical and Practical Improvements on the RMQ-Problem, with Applications to LCA and LCE, Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Computer Science, Springer-Verlag, 4009, 36 (2006).

[13] Czumaj A., Kowaluk M., Lingas A., Faster algorithms for finding lowest common ancestors in directed acyclic graphs, Theoretical Computer Science, 380(1-2), 37(2007).

[14] Stefan Eckhardt, Andreas Michael Mühling and Johannes Nowak, Fast Lowest Common Ancestor Computations in Dags, Lecture Notes in Computer Science, Volume 4698, Pp705-716, Springer Berlin Heidelberg, (2007).

[15] WANG Xingbo, Properties of the Lowest Common Ancestor in a Complete Binary Tree, International Journal of Scientific and Innovative Mathematical Research, 3(3), 12(2015)

[16] Wang Xingbo. Functions Related to Binary Representation of Integers. Journal of Inequalities and Special Functions, 2(2),8(2011)

[17] Paul E. Black, Dictionary of Algorithms and Data Structures, US National Institute of Standards and Technology, 2004. pp.131,pp.257.

[18] Graham R. L., Knuth D. E., Patashnik O., Concrete Mathematics: A Foundation for Computer Science, Addison-Wesley, 1994,ch3,pp67-101.

[19] WANG Xingbo, Some Supplemental Properties with Appendix Application of Floor Function, Journal of Science of Teacher's College and University (In Chinese), 34(3),7(2014)

[20] John F. Wakerly, Digital Design Principles And Practices(4th edition),Prentice Hall division of Pearson Education,2005

[21] Rance D Necaise, Data Structures and Algorithms Using Python, John Wiley & Sons,2011

[22] Peter Brass, Advanced Data Structures, Combridge University Press, 2008

[23] DENG J H, Data Structure and Algorithms in Java Description, China Machine Press, 2005

AUTHOR'S BIOGRAPHY

WANG Xingbo was born in Hubei, China. He got his Master and Doctor's degree 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, still in charge of researching and developing CAD/CAM/NC technologies. Wang has published 8 books, over 70 papers and obtained more than 20 patents in mechanical engineering.