Research on Ordinal Properties in Combinatorics Coding Method

Lu Jun
College of Computer Science and Technology, Heilongjiang University, Harbin, China
College of Computer Science and Technology, Harbin Engineering University, Harbin, China
Email: lujun111@yahoo.com.cn

Wang Tong
Colledge of Information and Communication Engineering, Harbin Engineering University, Harbin, China
Email: wangtong@hrbeu.edu.cn

Liu Daxin
College of Computer Science and Technology, Harbin Engineering University, Harbin, China
Email: ldx12399@sina.com

***Abstract*—Combinatorics coding is a new data coding method based on combinatorics. Ordinal computing is the core technology of in the combinatorics coding. Ordinal properties of combinatorics coding is studied in this paper. Firstly, the combinatorics coding theory and the correlative basic concepts are described. Then, the properties of the ordinal are revealed. The study indicates that the value of ordinal have a close relationship with the selection of benchmark sequence. The size of ordinal is correlative with the distributing of character frequency in the sequence.The length of the sequence should be longer than the length of the ordinal and this conclude is proved by the theory. Moreover, the difference of the initial sequence length and the whole max ordinal length is increased along with the increase of the initial sequence length pvoved by the experiment and theory. The research on ordinal properties is of great significance to the development of the combinatorics coding technology.**

***Index Terms*—Data Coding, Source Coding, Combinatorics Coding, Benchmark sequence, Ordinal**

I. INTRODUCTION

With the development of the information technology, data coding technology, applied in various fields such as computer science, multimedia and communications. It has become a basic tool and one of the basic problems to study. It now plays an important role in all kinds of studies and applications.

There are three embranchments in coding technology: source coding, channel coding and secrecy coding. The main task of source coding is to compress data. Data compression technology is studied and applied widely in the information world[1-3]. The main aim of channel coding is to enhance the reliability of information transmission, such as error correcting code. Secrecy coding prevents the information being filched. It is usually implemented by the data encryption techniques. Today, data encryption techniques become the hot research[4,5].

Combinatorics coding is a new data coding method based on combinatorics. Due to its unique properties and advantages, it is applicable in various fields such as data compression, data secrecy, and so on[6-10]. Combinatorics coding is of great significance to information disposing, data storage, and data transmission.

II. COMBINATORICS CODING THEORY

Combinatorics coding makes use of the relation between the space of the character sequence and the space of the corresponding ordinal to code. There is a kind of question in combinatorics:

Suppose there is a sequence includes n elements and $a_1\ a_2\ \ldots\ a_n$ is different. Full permutation is made to this sequence and the number of different permutations is $n!$. If there is an element a_1 and a_1 repeats w_1 times, then there are the same results of permutations. In fact, the different permutation number should be $n!/w_1!$. That is to say, the repetition degree is $w_1!$. In the same way, a_1 repeats w_1, a_2 repeats w_2, ……, a_m repeats w_m, and $w_1+w_2+.....+w_m=n$, that is to say, n elements include repeating elements, the different elements number is only m. If full permutation is made to thus sequence, the number of real different permutations is $n!/(w_1!*w_2!*...*w_m!)$. $n!/(w_1!*w_2!*...*w_m!)$ is less than $n!$. Combinatorics coding method is enlightened by this kind of question.

doi:10.4304/jcp.6.1.51-58

In practice, because data is usually handled and stored by bytes, the number of different elements is 256( ). Thus, to the sequence, the number of the real different permutations can also be expressed by the following formula:

$$P = n!/(\prod_{i=1}^{m}(w_i!)). \quad (1)$$

III. BASIC CONCEPTIONS

In order to describe contents and deduce formulas expediently, the basic conceptions are defined firstly.

Definition 1: ordinal and max ordinal.

The set $S = \{x_1, x_2 \ldots x_m\}$, $x_i \neq x_j$, the element number of S is m. There is a sequence $a_1 a_2 \ldots a_n$ which length is n, $a_i \in S$. If n elements in this sequence are made full permutation as a certain order, the permutation number is SUM. If it counts from 0, this sequence position in the whole permutations is $0 \sim SUM - 1$. The position of the sequence $a_1 a_2 \ldots a_n$ is named ordinal. At the same time, the value of $SUM - 1$ is named max ordinal.

To the same sequence, if the reference benchmark (certain order) is different, then the corresponding ordinal is different. That is to say, the ordinal is not exclusive. So benchmark sequence is defined to express reference benchmark.

Definition 2: benchmark sequence.

The set $S = \{x_1, x_2 \ldots x_m\}$, $x_i \neq x_j$, the element number of S is recorded as $|S| = m$. There is a sequence $a_1 a_2 \ldots a_n$ which the length is n and $a_i \in S$. To the elements m in S, there are $m!$ kinds of permutation methods. The sequence $a_1 a_2 \ldots a_n$ can be computed to different ordinal based on different permutation methods as a benchmark. But once the reference benchmark is confirmed, the ordinal is exclusive. The reference benchmark is named benchmark sequence.

Combinatorics coding method requires a benchmark sequence that is agreed on by the coder and the decoder. Based on the same benchmark sequence, an ordinal corresponding to the character sequence is computed in the coding process and the primary character sequence is deduced from the ordinal in the decoding process. Ordinal computing is the core in the combinatorics coding technology. Research finds that the value of ordinal have an affinity for the selection of benchmark sequence.

The size of the ordinal lies in the distribution of the character frequency. An extremity case will be studied in this paper to make a research on correlative question. When the length of the sequence is n and the frequency of each element is the same, that is to say, in the worst case, max ordinal is studied. Max ordinal obtained in this case is named whole max ordinal. The definition is as follows.

Definition 3: whole max ordinal.

The set $S = \{x_1, x_2 \ldots x_m\}$, $x_i \neq x_j$, the element number of S is m. The permutation to each sequence $a_1 a_2 \ldots a_n$ whose length is n and $a_i \in S$, has a max ordinal. There is a biggest max value in these max ordinals. This biggest max value is named whole max ordinal.

To the character sequence whose length is n, there are 256 kinds of different characters. Each character may have different frequency. It means there are lots of permutation methods. Each sequence whose length is n has a max ordinal. To the sequence whose length is n, if n is the integral multiple of 256, when each character frequency is equal, the whole max ordinal is gained.

In fact, if n is not the integral multiple of 256, when the difference of each character frequency value is not more than one, that is to say, each character frequency value is $\lfloor n/256 \rfloor$ or $\lfloor n/256 \rfloor + 1$, the whole max ordinal can be gained.

To make the study convenient, we usually deals with the case when n is the integral multiple of 256. Because character sequence can be divided into subsections and the length of each subsection can be the integral multiple of 256, this method is reasonable. Of course, the last subsection is not handled. In this paper, the study aims at this case.

IV. RESEARCH ON ORDINAL PROPERTY

As to combinatorics coding method, an ordered space is contained in character frequency table of initial sequence which would be coded. This ordered space is made up of the sequences which corresponding character number is the same. For example, to all sequences, the number of character '0' is the same. The number of character '1' is same…… The number of character '255' is the same. That is to say, all sequences have the same corresponding character number with the sequence which would be coded. The difference lies in character position distribution. Moreover, each sequence is not repeated in this space. When the sequence is coded, the number of the sequences in front of each character is computed. At last, all sequence number is added and the position of this sequence in the whole space is obtained. That is to say, the ordinal of the sequence $a_1, a_2 \ldots\ldots a_n$ expresses the number of other sequences in front of this sequence under the destined benchmark sequence. Of course, different result is obtained if different benchmark sequence is adopted as permutation rule. But the ordinal corresponding to the sequence is exclusive if benchmark sequence is confirmed. For example, To the j^{th} character in the sequence which has n characters, if this character is the same as the i^{th} character in benchmark sequence, it is

predicated that there are existed permutation and combination sequences before the current sequence involved i-1 characters. To these i-1 characters, the permutation and combination number of the x[th] character (the character position in these i-1 characters is x) occupied the j[th] position can be expressed by $S_{j,x}$. It is shown in the equation (2).

$$S_{j,x} = \frac{(n-j)!}{(\prod_{q=1}^{x-1}(w_q!))*(w_x-1)*(\prod_{q=x+1}^{ii}(w_q!))} \quad (2)$$

w expresses the number of each character. To the j[th] character in the sequence which has n characters, the sequence sum of i-1 elements before the j[th] element can be computed by the formula (3).

$$\sum_{x=1}^{i-1} S_{j,x} \quad (3)$$

At last, the position (ordinal) in the whole space of the sequence which has n characters can be expressed by the formula (4).

$$\sum_{j=1}^{n}\sum_{x=1}^{i-1} S_{j,x} \quad (4)$$

The basic idea of combinatorics decompression (decoding) is that suppose the element in the inspected position is a certain element, the corresponding permutation and combination value p_1 is computed out based on benchmark sequence according to this suppose. Compared p_1 with ordinal, if ordinal is not less than p_1, then the permutation and combination value p_2 of the next element is computed out based on benchmark sequence. It needs to judge that ordinal is not less than $p_1 + p_2$ or not. ……Until it finds out that ordinal is less than $p_1 + p_2 + \ldots\ldots + p_r$. This moment, the element corresponding p_r should be the element in the inspected position. At last, the new ordinal needs to be computed to deduce the next position element according to the formula (5):

new ordinal= old ordinal-($p_1 + p_2 + \ldots\ldots + p_{r-1}$) (5)

In fact, the process of combinatorics decoding is the inverse process to combinatorics coding. Probing step by step method is adopted to parse correlative character. The most important step of probing method is also ordinal computing. In a word, the key of combinatorics coding and decoding is to compute ordinal.

There are three primary properties in the combinatorics coding method.

Property 1: The value of ordinal has a close relationship to the selection of benchmark sequence. To the same sequence, the ordinal corresponding to the sequence is different if the benchmark sequence selected is different. Moreover, the sum of the ordinals corresponding to the sequence is max ordinal if the order of the benchmark sequence selected is contrary.

Property 2: The size of ordinal is correlative with the distributing of character frequency in the sequence. The more differences of character frequencies are, the smaller ordinal is. The closer differences of character frequencies are, the bigger ordinal is. When the character frequencies are equal, the whole max ordinal is obtained.

Property 3: The length of whole max ordinal, the length of max ordinal and the length of ordinal are all less than the length of sequence.

Calculating the ordinal of the sequence is the core in the combinatorics coding method. Computing quantity is exponential increase along with the increase of the sequence length because combinatorics coding method is based on combinatorics. Arithmetic needs to be optimized. It is found in study that ordinal computing lies on max ordinal computing and max ordinal computing lies on whole max ordinal computing in the optimization arithmetic. The value of whole max ordinal is only relative to the length of sequence, so the whole max ordinal can be computed out beforehand and be stored in a file. Thereby, the speed of ordinal computing is improved. Furthermore, some useful properties of combinatorics coding can be obtained by the whole max ordinal, so the research on whole max ordinal is very important.

To the sequence whose length is , the first study is how to obtain the whole max ordinal.

At first, a theorem should be proved.

Theorem 1: If $\sum_{i=1}^{p} a_i = \sum_{j=1}^{q} b_j$, and p+q=z. a_i is an integer, $a_i \geq 0$. b_j is an integer, $b_j \geq 0$. Then to any integer x, $x \geq 0$, the following inequation is correct:

$$(x-a_1)!*(x-a_2)!*\ldots*(x-a_p)!*$$
$$(x+b_1)!*(x+b_2)!*\ldots*(x+b_q)! \geq (x!)^z \quad (6)$$

Prove: Look at the following formula:

$$((x-a_1)!*(x-a_2)!*\ldots*(x-a_p)!$$
$$*(x+b_1)!*(x+b_2)!*\ldots*(x+b_q)!)/(x!)^z$$
$$= ((x+b_1)(x+b_1-1)\ldots(x+1)(x+b_2)(x+$$
$$b_2-1)\ldots(x+1)\ldots(x+b_q)(x+b_q-1)\ldots(x+1))$$
$$/(x(x-1)\ldots(x-a_1+1)x(x-1)\ldots(x-a_2+1)$$
$$\ldots\ldots x(x-1)\ldots(x-a_p+1))$$

The numerator of the formula has $\sum_{j=1}^{q} b_j$ items. The denominator of the formula has $\sum_{i=1}^{p} a_i$ items. Because

$\sum_{i=1}^{p} a_i$ is equal to $\sum_{j=1}^{q} b_j$, the number of the items in the numerator is equal to the number of the items in the denominator. Each item value in the numerator is more than x or equal to x and each item value in the denominator is smaller than x or equal to x , so the value of the numerator is not smaller than the value of the denominator. That is to say, the following inequation is correct:

$$((x+b_1)(x+b_1-1)...(x+1)(x+b_2)(x+b_2-1)...(x+1)...(x+b_q)(x+b_q-1)...(x+1))/(x(x-1)...(x-a_1+1)x(x-1)...(x-a_2+1)...x(x-1)...(x-a_p+1)) \geq 1$$

Of course,

$$((x-a_1)!*(x-a_2)!*...*(x-a_p)!*(x+b_1)!*(x+b_2)!*...*(x+b_q)!)/(x!)^z \geq 1$$

So formula (2) is correct. The proof is over.

Theorem 1 indicates that the following result is correct:

To the character sequence whose length is n , the number of different characters is 256(z=256) and the frequency of each character is m , $m = x - a_i$ or $m = x + b_j$. x is the average frequency, that is to say, $x = n/256$. Further, $\sum_{i=1}^{p} a_i = \sum_{j=1}^{q} b_j$, $\sum_{i=1}^{p}(x-a_i) + \sum_{j=1}^{q}(x+b_j) = n$, p+q=256,based on the formula (1), the max ordinal to the sequence whose length is n can be computed by the following formula:

$$n!/(\prod_{i=1}^{256}(m_i!)) = n!/((x-a_1)!*(x-a_2)!*...*(x-a_p)!*(x+b_1)*(x+b_2)*...*(x+b_q)!)$$

Based on the inequation (2), it can be known:

$$(x-a_1)!*(x-a_2)!*...*(x-a_p)!*(x+b_1)!*(x+b_2)!*...*(x+b_q)! \geq (x!)^{256}$$

So,

$$n!/((x-a_1)!*(x-a_2)!*...*(x-a_p)!*(x+b_1)*(x+b_2)*...*(x+b_q)!) \leq \frac{n!}{(x!)^{256}} = \frac{n!}{((n/256)!)^{256}}$$

It indicates when the frequency of each character is equal ($x = n/256$), the whole max ordinal can be obtained:

$$\frac{n!}{((n/256)!)^{256}} \qquad (7)$$

For example, to the character sequence whose length is 204800 bytes (n=204800), the average frequency of each character is 204800/256=800. Now x is 800. This is the extremeness case that the frequency of each character in the sequence is the average frequency. But in normal case, the frequency which is more than 800 is expressed as $x + b_j$ and the frequency which is less than 800 is expressed as $x - a_i$. Here, p+q=256, $z = 256$, $\sum_{i=1}^{p} a_i = \sum_{j=1}^{q} b_j$, $\sum_{i=1}^{p}(x-a_i) + \sum_{j=1}^{q}(x+b_j) = 204800$. It can be known based on the inequation (6):

$$(x-a_1)!*(x-a_2)!*...*(x-a_p)!*(x+b_1)!*(x+b_2)!*...*(x+b_q)! \geq (x!)^{256}$$

That is to say, to the sequence whose length is 204800 bytes, $\prod_{i=1}^{256}(m_i!) \geq (800!)^{256}$ is correct when m_i expresses character frequency in normal case.

It can be known based on formula (1) that to n=204800, the max ordinal can be expressed as $204800!/\prod_{i=1}^{256}(m_i!)$. When the frequencies of 256 characters are equal, the max ordinal is $204800!/\prod_{i=1}^{256}(800!)$ or $204800!/(800!)^{256}$. Because $\prod_{i=1}^{256}(m_i!) \geq (800!)^{256}$ is correct, $204800!/\prod_{i=1}^{256}(m_i!)$ should be less than $204800!/(800!)^{256}$. That is to say, to the sequence whose length is 204800 bytes, the whole max ordinal is $204800!/(800!)^{256}$.

In fact, to the sequence whose length is n ($n \geq 256$, n is multiples of 256), its whole max ordinal length must be less than n . This conclusion can be proved as follows.

Theorem 2: If the length of one sequence is n (≥ 256) and n is multiples of 256, then the length of the whole max ordinal corresponding to the sequence is less than n .

Prove: In order to prove theorem 2, it only needs to prove that the whole max ordinal is less than the smallest data whose length is n. The smallest data expressed by n bytes is 256^{n-1} . Then it needs to prove the following inequation is correct:

$$\frac{n!}{((n/256)!)^{256}} < 256^{n-1} \qquad (8)$$

Epagoge method is adopted:

(1) If n=256, the left of the inequation is $\frac{256!}{((256/256)!)^{256}} = 256!$ and the right of the inequation is 256^{255}. It is apparently $256! < 256^{255}$. So the inequation is correct.

(2) It is supposed that if n=k, the inequation is correct: $\frac{k!}{((k/256)!)^{256}} < 256^{k-1}$

Then if n=k+256, the left of the inequation:

$$\frac{(k+256)!}{(((k+256)/256)!)^{256}}$$

$$= \frac{(k+256)*(k+255)*\ldots\ldots*(k+1)*k!}{((k/256+1)!)^{256}}$$

$$= \frac{(k+256)*(k+255)*\ldots\ldots*(k+1)*k!}{(k/256+1)^{256}*((k/256)!)^{256}}$$

$$< (k+256)*(k+255)*\ldots\ldots* (k+1)/((k/256+1)^{256})*256^{k-1} \quad (9)$$

In fact, the follow inequation is correct apparently:

$$(k+256)*(k+255)*\ldots*(k+1) < (k+256)^{256}$$

So the following inequation is correct:

$$\frac{(k+256)*(k+255)*\ldots\ldots*(k+1)}{(k+256)^{256}} < 1$$

$$\Rightarrow \frac{(k+256)*(k+255)*\ldots*(k+1)}{(k+256)^{256}} * 256^{256} < 256^{256}$$

$$\Rightarrow \frac{(k+256)*(k+255)*\ldots\ldots*(k+1)}{((k+256)/256)^{256}} < 256^{256}$$

$$\Rightarrow \frac{(k+256)*(k+255)*\ldots\ldots*(k+1)}{((k/256+1))^{256}} < 256^{256} \quad (10)$$

Put formula (10) into formula (9), the following formula can be obtained:

$$\frac{(k+256)!}{(((k+256)/256)!)^{256}} < 256^{256} * 256^{k-1}$$

$$\Rightarrow \frac{(k+256)!}{(((k+256)/256)!)^{256}} < 256^{(k+256)-1}$$

So when n is k+256, the inequation (8) is also correct. Then theorem 2 is correct. Because of the length of ordinal is less than that of whole max ordinal, it is proved by the theory that the space of ordinal is less than the space of the initial sequence.

In fact, not only the whole max ordinal length must be less than the sequence length, but also the difference of the sequence length and the whole max ordinal length is increased along with the increase of the sequence length. Of course, the increase range become is less and less. The experiment data in table 1 incarnate this character.

TABLE I.
TABLE I. DIFFERENCE OF SEQUENCE LENGTH AND WHOLE MAX ORDINAL LENGTH

Sequence length	256	200k	1M	2M	3M	4M
Difference with ordinal length	46	195	233	248	258	264

Sequence length	5M	6M	7M	8M	9M	…
Difference with ordinal length	270	274	277	280	283	…

The experiment data in table 1 express that the difference range become less and less along with the sequence length increase. The difference range is possibly less than one bit. At last, perhaps the sequence length increases several millions, the difference increases one bit; the sequence length increases dozens millions, the difference increases one bit……But the difference increased along with the increase of the sequence length. This trend can also be expressed by Fig. 1.

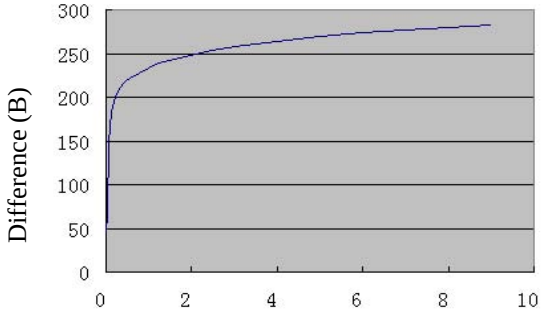

Figure 1. Differ d Whole Max Ordinal Length.

Above experiment conclusion can also be proved strictly by theory.

Theorem 3: The difference between the sequence length and the whole max ordinal length is increased along with the increase of the sequence length. Over here, the length of the sequence is the multiple of 256 and the step of increase is 256.

The idea of proving: Suppose the length of a sequence is A. The whole max ordinal of this sequence is Y. After the length of the sequence increases 256 bytes, the whole max ordinal of this sequence is Y_1. In order to prove theorem 3, it needs to prove that the difference between the smallest value $256^{(A-1)}$ expressed by the sequence whose length is A and Y is less than the difference between the smallest value expressed by the sequence whose length is A+256 and Y_1. That is to say, $256^{(A+256-1)} - Y_1 > 256^{(A-1)} - Y$ will be proved or the follow formula will be proved:

$$256^{(A+255)} - (A+256)!/(((A+256)/256)!)^{256}$$

$$> 256^{(A-1)} - A!/((A/256)!)^{256}$$

The formula can also be expressed by the formula (11).

$$256^{(A+255)} - (A+256)!/(((A+256)/256)!)^{256} - 256^{(A-1)} + A!/((A/256)!)^{256} > 0 \quad (11)$$

Epagoge method is still adopted:

Prove: (1) If the value of A is 256, then

$$256^{(A+255)} - (A+256)!/(((A+256)/256)!)^{256} - 256^{(A-1)} + A!/((A/256)!)^{256}$$
$$= 256^{511} - 512!/(2!)^{256} - 256^{255} + 256!/(1!)^{256}$$
$$= 256^{255}(256^{256} - 1) - 512!/(2!)^{256} + 256!$$
$$> 256^{255} * (256^{256} - 256^{255}) - (512 * 511 * 510 * 509 ..... * 258 * 257 * 256!)/(2 * 2..... * 2 * 2) + 256!$$
$$> 255 * (256^{255})^2 - (512 * 512 * 510 * 510 ..... * 258 * 258 * 256!)/(2 * 2..... * 2 * 2) + 256!$$
$$= 255 * (256^{127})^2 * (256^{128})^2 - (256 * 255 * .....129)^2 * 256! + 256!$$
$$> 255 * (256^{127})^2 * (256^{128})^2 - (256^{128})^2 * 256! + 256!$$
$$= (256^{128})^2 * (255 * 256^{254} - 256!) + 256!$$
$$> (256^{128})^2 * (2 * 256^{254} - 256!) + 256!$$
$$= (256^{128})^2 * (2 * 256^{254} - 256 * 255.....2 * 1) + 256!$$
$$= (256^{128})^2 * 2 * (256^{254} - 256 * 255..... * 3) + 256!$$
$$> 0$$

So when A is 256, formula (11) is right.

(2) Suppose when A=i, formula (11) is right:

$$256^{(i+255)} - (i+256)!/(((i+256)/256)!)^{256} - 256^{(i-1)} + i!/((i/256)!)^{256} > 0$$

That is to say, formula (12) is right:

$$256^{(i+255)} > (i+256)!/(((i+256)/256)!)^{256} + 256^{(i-1)} - i!/((i/256)!)^{256} \quad (12)$$

So if A=i+256, then

$$256^{(i+511)} - (i+512)!/(((A+512)/256)!)^{256} - 256^{(i+255)} + (i+256)!/(((i+256)/256)!)^{256}$$
$$= 256^{(i+255+256)} - 256^{(i+255)} - (i+512)!/(((i/256)+2)!)^{256} + (i+256)!/(((i/256)+1)!)^{256}$$
$$= 256^{(i+255)} * (256^{256} - 1) - (i+512)!/(((i/256)+2)!)^{256} + (i+256)!/(((i/256)+1)!)^{256}$$

Based on formula (12), above formula can be educed as follow:

$$> (256^{256} - 1) * ((i+256)!/(((i+256)/256)!)^{256} + 256^{(i-1)} - i!/((i/256)!)^{256}) - (i+512)!/(((i/256)+2)!)^{256} + (i+256)!/(((i/256)+1)!)^{256}$$
$$= 256^{256} * (i+256)!/(((i+256)/256)!)^{256} + 256^{256} * 256^{(i-1)} - 256^{256} * i!/((i/256)!)^{256} - (i+256)!/(((i+256)/256)!)^{256} - 256^{(i-1)} + i!/((i/256)!)^{256} - (i+512)!/(((i/256)+2)!)^{256} + (i+256)!/(((i/256)+1)!)^{256}$$
$$= 256^{256} * (i+256)!/(((i+256)/256)!)^{256} + 256^{(i+255)} - 256^{256} * i!/((i/256)!)^{256} - 256^{(i-1)} + i!/((i/256)!)^{256} - (i+512)!/(((i/256)+2)!)^{256}$$
$$> 256^{256} * (i+256)!/(((i+256)/256)!)^{256} + (i+256)!/(((i+256)/256)!)^{256} + 256^{(i-1)} - i!/((i/256)!)^{256} - 256^{256} * i!/((i/256)!)^{256} - 256^{(i-1)} + i!/((i/256)!)^{256} - (i+512)!/(((i/256)+2)!)^{256}$$
$$= (i+256)!/(((i+256)/256)!)^{256} * (256^{256} + 1 - 256^{256} * ((i/256)+1)^{256}/((i+256)(i+255).....(i+1))) - (i+512)(i+511).....(i+257)/((i/256)+2)^{256})$$
$$= (i+256)!/(((i+256)/256)!)^{256} * (256^{256} + 1 - (i+256)^{256}/((i+256)(i+255).....(i+1)) - 256^{256} * (i+512)(i+511).....(i+257)/((i+512)^{256})$$
$$= [(i+256)!/(((i+256)/256)!)^{256} / ((i+512)^{256}(i+256)(i+255).....(i+1))] * [256^{256}(i+256)(i+255).....(i+1)(i+512)^{256} + (i+512)^{256}(i+256)(i+255).....(i+1) - (i+256)^{256}(i+512)^{256} - 256^{256}(i+512)(i+511) .....(i+257)(i+256)(i+255).....(i+1)] \quad (13)$$

In order to be convenient for educing, suppose

$$M = [(i+256)!/(((i+256)/256)!)^{256} / ((i+512)^{256}(i+256)(i+255).....(i+1))]$$

Then the formula (13) become the formula (14)

$$M * [256^{256}(i+256)(i+255).....(i+1)(i+512)^{256} + (i+512)^{256}(i+256)(i+255).....(i+1) - (i+256)^{256}(i+512)^{256} - 256^{256}(i+512)(i+511) .....(i+257)(i+256)(i+255).....(i+1)] \quad (14)$$

$(a-h)(a+h) <= a^2$ is apparently, so

$(i+511)(i+510).....(i+257) < (i+384)^{255}$ is right. Thus the formula (14) can be educed as follow:

$$
\begin{aligned}
&> M*[256^{256}(i+512)^{256}(i+256)(i+255).....(i+1)\\
&+(i+512)^{256}(i+256)(i+255).....(i+1)\\
&-(i+256)^{256}(i+512)^{256}\\
&-256^{256}(i+512)(i+384)^{255}(i+256)(i+255).....(i+1)]
\end{aligned}
$$

$$
\begin{aligned}
&= M*[256^{256}(i+512)(i+256)(i+255).....(i+1)\\
&((i+512)^{255}-(i+384)^{255})\\
&+(i+512)^{256}(i+256)(i+255).....(i+1)\\
&-(i+256)^{256}(i+512)^{256}]\\
&> M*[256^{256}(i+512)(i+256)(i+255).....(i+1)\\
&((i+512)^{255}-(i+384)^{255})\\
&+(i+512)^{256}(i+1)^{256}-(i+256)^{256}(i+512)^{256}]\\
&= M*[256^{256}(i+512)(i+256)(i+255).....(i+1)\\
&((i+512)^{255}-(i+384)^{255})\\
&-(i+512)^{256}((i+256)^{256}-(i+1)^{256})]\\
&= M*[256^{256}(i+512)(i+1)^{256}\\
&(C_{255}^{1}i^{254}*(512-384)+C_{255}^{2}i^{253}(512^2-384^2)\\
&+.....+C_{255}^{254}i*(512^{254}-384^{254})\\
&+(512^{255}-384^{255}))
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
&-(i+512)^{256}(C_{256}^{1}i^{255}(256-1)\\
&+C_{256}^{2}i^{254}(256^2-1)+.....\\
&+C_{256}^{255}i*(256^{255}-1)+(256^{256}-1))]
\end{aligned}
$$

Because $a^m - b^m >= (a-b)^m$ (This can be proved by Epagoge method) and $256^m - 1 <= 256^m$, thus the formula (15) can be educed as follow:

$$
\begin{aligned}
&> M*[256^{256}(i+512)(i+1)^{256}(C_{255}^{1}i^{254}*128\\
&+C_{255}^{2}i^{253}*128^2+.....+C_{255}^{254}i*128^{254}+128^{255})\\
&-(i+512)^{256}(C_{256}^{1}i^{255}*256+C_{256}^{2}i^{254}*256^2+\\
&.....+C_{256}^{255}i*256^{255}+256^{256})]
\end{aligned}
$$

$$
\begin{aligned}
&= M*[2^{256}*128^{256}*(i+512)(i+1)^{256}*(C_{255}^{1}i^{254}*128\\
&+C_{255}^{2}i^{253}*128^2+.....+C_{255}^{254}i*128^{254}+128^{255})\\
&-(i+512)^{256}(C_{256}^{1}i^{255}*256+C_{256}^{2}i^{254}*256^2+\\
&.....+C_{256}^{255}i*256^{255}+256^{256})]
\end{aligned}
$$

$$
\begin{aligned}
&= M*[128^{256}*(i+512)(i+1)^{256}*(C_{255}^{1}i^{254}*128*2^{256}\\
&+C_{255}^{2}i^{253}*128^2*2^{256}+.....+C_{255}^{254}i*128^{254}*2^{256}\\
&+128^{255}*2^{256})-(i+512)^{256}(C_{256}^{1}i^{255}*256\\
&+C_{256}^{2}i^{254}*256^2+...+C_{256}^{255}i*256^{255}+256^{256})]
\end{aligned}
$$

$$
\begin{aligned}
&= M*[(i+512)(128i+128)^{256}*(C_{255}^{1}i^{254}*256*2^{255}\\
&+C_{255}^{2}i^{253}*256^2*2^{254}+.....+C_{255}^{254}i*256^{254}*2^2\\
&+256^{255}*2)-(i+512)^{256}(C_{256}^{1}i^{255}*256\\
&+C_{256}^{2}i^{254}*256^2+.....+C_{256}^{255}i*256^{255}+256^{256})]
\end{aligned}
\tag{16}
$$

Because $(128i+128)^{256} > (i+512)^{256}$ (This can be proved by Epagoge method) and the follow formula is right:

$$
\begin{aligned}
&(i+512)(C_{255}^{1}i^{254}*256*2^{255}+C_{255}^{2}i^{253}*256^2\\
&*2^{254}+.....+C_{255}^{254}i*256^{254}*2^2+256^{255}*2)
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
&> C_{255}^{1}i^{255}*256*2^{255}+C_{255}^{2}i^{254}*256^2*2^{254}\\
&+C_{255}^{3}i^{253}*256^3*2^{253}+C_{255}^{4}i^{252}*256^4*2^{252}\\
&+C_{255}^{5}i^{251}*256^5*2^{251}+C_{255}^{6}i^{250}*256^6*2^{250}\\
&+.....+C_{255}^{254}i^2*256^{254}*2^2+i*256^{255}*2
\end{aligned}
\tag{18}
$$

Moreover, because $C_{255}^{m}/C_{256}^{m} = (256-m)/256$ is right, thus the formula (18) can be educed as follow:

$$
\begin{aligned}
&= C_{256}^{1}*(255/256)*i^{255}*256*2^{255}\\
&+C_{256}^{2}*(254/256)*i^{254}*256^2*2^{254}\\
&+C_{256}^{3}*(253/256)*i^{253}*256^3*2^{253}\\
&+C_{256}^{4}*(252/256)*i^{252}*256^4*2^{252}\\
&+C_{256}^{5}*(251/256)*i^{251}*256^5*2^{251}\\
&+C_{256}^{6}*(250/256)*i^{250}*256^6*2^{250}\\
&+.....+C_{256}^{250}*(6/256)*i^6*256^{250}*2^6\\
&+C_{256}^{251}*(5/256)*i^5*256^{251}*2^5\\
&+C_{256}^{252}*(4/256)*i^4*256^{252}*2^4\\
&+C_{256}^{253}*(3/256)*i^3*256^{253}*2^3\\
&+C_{256}^{254}*(2/256)*i^2*256^{254}*2^2\\
&+i*256^{255}*2
\end{aligned}
\tag{19}
$$

The set W expresses the corresponding items in the formula (16):

$$
\begin{aligned}
&W = C_{256}^{1}i^{255}*256+C_{256}^{2}i^{254}*256^2+.....\\
&+C_{256}^{255}i*256^{255}+256^{256}
\end{aligned}
$$

$$= C_{256}^{1} i^{255} * 256 + C_{256}^{2} i^{254} * 256^{2} + \ldots\ldots$$
$$+ C_{256}^{250} i^{6} * 256^{250} + C_{256}^{251} i^{5} * 256^{251} + C_{256}^{252} i^{4} * 256^{252}$$
$$+ C_{256}^{253} i^{3} * 256^{253} + C_{256}^{254} i^{2} * 256^{254} + C_{256}^{255} i * 256^{255} + 256^{256}$$

Each item in the formula (19) is more than each corresponding item in W, except that the sum of the last five items in the formula (19) is less than the sum of the last six items in W. Several items in the front of the formula (19) can be split as follow:

$$\begin{aligned}
&= C_{256}^{1} * i^{255} * 256 + C_{256}^{1} * i^{255} * 256 * (255 * 2^{247} - 1) \\
&+ C_{256}^{2} * i^{254} * 256^{2} + C_{256}^{2} * i^{254} * 256^{2} * (254 * 2^{246} - 1) \\
&+ C_{256}^{3} * i^{253} * 256^{3} + C_{256}^{3} * i^{253} * 256^{3} * (253 * 2^{245} - 1) \\
&+ \ldots\ldots + C_{256}^{250} * (6/256) * i^{6} * 256^{250} * 2^{6} \\
&+ C_{256}^{251} * (5/256) * i^{5} * 256^{251} * 2^{5} \\
&+ C_{256}^{252} * (4/256) * i^{4} * 256^{252} * 2^{4} \\
&+ C_{256}^{253} * (3/256) * i^{3} * 256^{253} * 2^{3} \\
&+ C_{256}^{254} * (2/256) * i^{2} * 256^{254} * 2^{2} \\
&+ i * 256^{255} * 2
\end{aligned} \tag{20}$$

Apparently, compared to the items in the front of W, the sum of the redundant items split and the last five items in the formula (20) must be more than the sum of the six items of W. So the value of the formula (17) is more than W. Thus, the value of the formula (16) is more than 0.

So if A=i+256, then the formula (11) is correct. The proof is over. Prepare your CR paper in full-size format, on A4 paper (210 x 297 mm, 8.27 x 11.69 in).

V. CONCLUSION

This article studies the properties of combinatorics coding ordinal. The study indicates that the value of ordinal have a close relationship to the selection of benchmark sequence. To the same sequence, the ordinal of the sequence is different if the benchmark sequence selected is different; the sum of the ordinals of the sequence is max ordinal if the order of the benchmark sequence selected is contrary. The size of ordinal is correlative with the distribution of character frequency in the sequence. When the character frequencies are equal, the whole max ordinal is obtained. The length of the sequence is longer than that of the ordinal. Moreover, the difference of the initial sequence length and the whole max ordinal length is increased along with the increase of the initial sequence length by the experiment and theory. The research on ordinal properties is of great significance to the development of the combinatorics coding technology.

This paper is supported by the National Postdoctoral Science Foundation under Grant No. 20080440840. We are grateful to all those who made constructive comments.

REFERENCES

[1] A Jas,J Ghosh-Dastidar. “An efficient test vector compression scheme using selective huffman coding,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, pp.797- 806, Jun 2003.

[2] Nourani M, Tehranipour M H, ”RL-Huffman encoding for test compression and power reduction in scan applications,”ACM Transitions on Design Automation of Electronic Systems, vol. 10, pp. 91-115, Jan 2005.

[3] Lan Bo,Lin Xiao-Zhu,Ji Jun-Wei, “Application of an improved LZW algorithum in image coding,” Computer Engineering And Science, vol. 28, pp.101-110, Jun 2006.

[4] Yang Huaqian, Liao Xiaofeng, et al. “A new block cipher based on chaotic map and group theory,” Chaos, Solitons & Fractals. pp. 10-16, Oct 2005.

[5] Wei Jun, Liao Xiaofeng, Wong Kwok-wo, et a1, “A new chaotic cryptosystem,” Chaos, Solitons&Fractals,vol.30, pp. 1143-1252, 2006.

[6] Lu Jun, Wang Tong, Li Yibing, “Study on Optimization Technology in Computing Ordinal Number,” Journal of Computers, vol. 5, pp. 210-217, Feb 2010.

[7] Lu Jun, Liu DaXin, Xie XinQiang, “Selection of the Smallest Compression Subsection in Constant Grade Compression,” Journal of Information and Computational Science, vol. 5, pp.1545-1550, April 2008.

[8] Lu Jun, Liu DaXin, “Optimization on Computing Base-data in Constant Grade Compression,” 2009 International Forum on Information Technology and Applications (IFITA 2009), pp. 68-71, May 2009.

[9] LU Jun, LIU Da-xin ,CHEN Li-yan, “Compression method with constant degree based on permutation and combination,” Journal of Dalian Maritime University, vol. 34, pp28-32, April 2008.

[10] Lu Jun, Liu Daxin, “Optimization of Constant Grade Compression Method,” Journal of Jiangsu University(Natural Science Edition). vol. 31, pp. 78-81. Jan 2010.

Lu Jun was born in 1971. She obtained her master's degree from College of Computer Science and Technology of Heilongjiang University in 2000 and the study filed is database. She works at College of Computer Science and Technology, Heilongjiang University, Harbin, China. Now, she is studying in Harbin Engineering University, Harbin, China. Her study field is database and repository, network and information security etc.

Wang Tong was born in 1977. Ph.D. associated Professor. He got his Ph.D. of computer science from Harbin Engineering University in 2007.And now he works at Colledge of Information and Communication Engineering, Harbin Engineering University. His study field is data mining, computer network, artifical intelligence etc.

Liu Daxin was born in 1941. Professor, Ph.D supervisor. Now he works at College of Computer Science and Technology, Harbin Engineering University. His study field is database and repository.