Scientific
Research
Publishing

Heru Search Method—Unique in the World that Uses Unprecedented Mathematical Formulas and Replaces the Binary Tree Breaking Various Paradigms Like $O(\log n)$

Carlos Roberto França

Federal University of Fronteira Sul, Chapecó, Brazil

Email: heru@heru.com.br

Abstract

This paper deals with the efficiency of the search, with a method of organization and storage of the information that allows better results than the research trees or binary trees. No one ever dared to present better results than $O(\log n)$ complexity, and when they wish to improve, they use balanced trees, but they continue to use principles that do not impact the pre-semantic information treatment. The Heru search method has as main characteristic the total or partial substitution of the use of the binary trees, enabling the elimination of the approximate results and informing the user the desired information instead of occurrences by sampling outside the desired information. The breakdown of the $O(\log n)$ paradigm and the refinement of the searches are achieved with the use of a set of unpublished mathematical formulas and concepts called Infinite Series with Multiple Ratios.

Keywords

Search Method, B-Tree, Innovations in Database, Infinite Series

1. Introduction

Sir Isaac Newton initiated the research involving Infinite Series with Multiple Ratios (SMRs), but he didn't conclude or published. Some citations in documentaries about the incomplete work of the brilliant physicist were the source of inspiration for two Brazilian researchers, and what served as an auxiliary at the beginning of the 90s, followed the search for practical applications to the set of 16 (sixteen) formulas. By insisting and dedicating SMRs for more than 20 years,

it has been successful with strong cryptography that interest from several universities, research centers, governments and large companies. Here is the new applicability that is mathematically ready and will modify and break paradigm in the matter of data storage and search.

SMRs complement and extend centennial physical/mathematical concepts such as those involving arithmetic and geometric progressions, Newtonian kinematics, Torricelli's equation, and others. They are all formulas that are based on singular events with a single ratio, or a speed or acceleration stipulated as a starting point to solve problems of the universes of physics and mathematics. The Infinite Series with Multiple Ratios (SMRs) works with a finite set of size (n) ratios, n being any number belonging to the set Z+ (positive integers).

Below are some concepts of the 16 SMR formulas:

According to the author and creator of the Infinite Series with Multiple Ratios, Professor Edgar Oliveira Rodrigues [1] the following basic concepts are essential for understanding these new mathematical elements. The concepts of the new Arithmetic Series formulas that extend and encompass every foundation of Geometrical Progressions (G.P) and Arithmetic Progressions (A.P), as well as Newtonian kinematics, require detailed discussions of their formation law. I would like to point out once more that mathematical formulas and concepts (SMRs) are unknown to the exact world science. The conceptual exposition begins below from the general term of this new approach. They are concepts that identify the elements that make up the basic law of formation of the same, and that do not invalidate the traditional formulas of the fields mentioned above, because in fact, it is a conceptual extension and the conventional formulas become subset of the SMRs.

1.1. Arithmetic Series in the Conceptual Approach of SMRs—General Terms

1ª) $x = 0$

X is the remainder of the division made by the number of terms in the series (n), and the amplitude of the interval (K).

For a better understanding, it is necessary to understand some concepts:

1ª) Homologous term (th) is defined by the module between the terms. If it is a multiple of K, they will be homologous;

Ex. We will take $K = 5$, arbitrarily, to facilitate the $a_7 tha_{22}$ because $|7 - 22| = 15$ which is a multiple of 5, arbitrated value for k.

Another important concept is the imagine or suppose a series with 23 terms (a_{23}) and $K = 5$, $23/5 = 4$ and with remainder 3, 4 is the integer quotient of the division (n/k) and represents the number of periods (y)

∴ $y = 4$ periods $[a_2, a_6], [a_7, a_{11}], [a_{12}, a_{16}], [a_{17}, a_{21}]$.

The rest of the division (n/k), represented by (x), locates the term in the series.

Examples:

If $x = 1$ we have the following relations in the 4 periods above, dividing the

last term by k (1st period the last term is a_6), in the second it is a_{11}, in the third it is a_{16}, and in the fourth period a_{21}.

Then for $x = 1$ we have: $6/5 = 1$ remainder 1, $11/5 = 2$ remainder 1, $16/5 = 3$ remainder 1 and $21/5 = 4$ remainder 1.

This fact indicates that the term homologous is always the last of each period and homologous to (a_1). Thus, a_1 th a_6, a_1 th a_{11}, a_1 th a_{16}, and a_1 th a_{21}.

If $x = 0$, we have the following relations in the 4 periods above.

$5/5 = 1$ rest 0, $10/5 = 2$ rest 0, $15/5 = 3$ rest 0 and $20/5 = 4$ rest 0.

We can observe that all terms homologous to the 1st are the penultimate terms of each period. Since there is no (a_0) in the series, the 1st th is the penultimate of the period, causing the number of periods to be subtracted from a unit: ($y - 1$).

Let us see another important finding.

$a_5 tha_{20}$, therefore $20/5 = 4$ remainder 0.

Note that between a_5 and a_{20} there are only three complete periods, which are:

$$[a_{20}, a_{16}], [a_{15}, a_{11}], [a_{10}, a_6].$$

Then if $x = 0$ there will be ($y - 1$) periods $(a_x tha_n)$ for $x \geq 1$, for example $23/5 = 4$ remainder 3, then $x = 3$ which is the remainder and $a_3 tha_{23}$ since $|3 - 23| = 20$ which is a multiple of 5.

If $x \geq 2$, the term will be found in the incomplete period ($x - 1$) positions above the full period. Example: $23/5 = 4$ remainder 3, as x is the remainder of the division n/k we have $(3 - 1) = 2$ and really is two positions above the last period.

Generalizing: th for $x \geq 1$.

$R =$ That is, R is the sum of the multiple ratios of the set r that will have an amplitude (quantity of elements) represented by K.

From the explanations from the 1st formula, it is possible to deduce and use the others that make up the scope of the arithmetic series.

Below are some of the 16 SMRs formulas:

The variables below are the result of the concepts of the Infinite Series with Multiple Ratios and have the following meanings:

(X) is the remainder of the division of n/k, where n is the quantity of elements that compose the series and (k) is the quantity of elements of the set of reasons. We have (y) that is the integer quotient of n/k, (p1) which symbolizes the first period of the series, (R) which is the sum of the elements of the set k, and variable (a) represents the homologous terms, going from $(a_1$ to $a_x)$, where x is a number of terms that form the periods seen in the above concepts.

It is worth mentioning that SMRs were never published in books, and were approved and presented at meetings of the Brazilian Society for the Advancement of Science—SBPC in Portuguese [2] [3] [4] and validated by renowned Brazilian and Canadian mathematicians. This paper is not intended to discuss the SMRs, but to present the first computational application originated by them, which is the new

search method, here also called as Search Method Heru.

$$x = o \rightarrow s_n = y\left(\sum p1 - R\right) + \frac{KRy(y-1)}{2} \tag{1}$$

$$x = 1 \rightarrow s_n = a_1 + \sum p1 + \frac{KRy(y-1)}{2} \tag{2}$$

$$x \geq 2 \rightarrow s_n = \sum_1^x a + y\left[\sum p1 + (x-1)R\right] + \frac{KRy(y-1)}{2} \tag{3}$$

Multiple Ratios for Applying Instantaneous Motion Speed.
Space traveled

$$x = 0 \rightarrow s_n = y\left(\sum p1 - R\right) + \frac{KRy(y-1) - (v_0 + v_f)}{2} \tag{4}$$

$$x = 1 \rightarrow s_n = a_1 + y\sum p1 + \frac{KRy(y-1) - (v_0 + v_f)}{2} \tag{5}$$

$$x \geq 2 \rightarrow s_n = \sum_1^x a + y\left[\sum p1 + (x-1)R\right] + \frac{KRy(y-1) - (v_0 + v_f)}{2} \tag{6}$$

SMR is a Series or Numerical Successions, composed by a finite set of ratios. Different from known A.P and G.P (Arithmetic Progression and Geometric Progression) admit one single ratio, SMR solve any type of progression that allow multiple ratios, even progressions solved by the traditional A.P, G.P. Beyond that, SMR to kinematics, Torricelli's equation and all those Physics/Mathematical belonging are sub-cases or subsets of SMR.

Note [1]: Professor Hermann Hankel (Notorious German researcher from XIX century) in his thesis on the creation of rational numbers introduced the principle of extension (see Figure 1), divided into two parts:

1) The new numbers should be self-sufficient completely solve his own creation;

2) New numbers should not be inconsistent with existing ones, on the contrary, it must hold as special cases.

At this context SMR resolve and treat as special cases the Series quoted, well as linear motion and Torricelli's equations. We believe that the principle of Hermann Hangel applies to the Infinite Series with Multiple Ratios. Therefore, Uniform Rectilinear Movement (U.R.M), Varied Uniform Rectilinear Movement (V.U.R.M), Varied Rectilinear Movement (V.R.M), Geometrical Progressions (G.P) and Arithmetic Progressions (A.P), are subsets of SRMs, according to Figure 2 below.
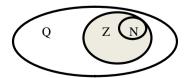

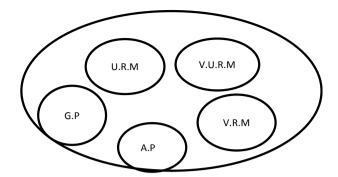Figure 1. Hermann Hankel's principle-source: own creation.

Figure 2. Infinite series with multiple ratios (SMRs) and their possible subsets. Source: own creation.

1.2. Replacing the Binary Tree Search Method

When we type any character in the computer, it codifies what has been typed as a number. This way, when type C the computer will register 67; $A = 65$; $R = 82$; $L = 76$ and so on. Then the decimals are transformed in binary and the task will be executed in a minimal period time. As an example, we have a Personnel Department File and we want to find a certain employee. The task will be executed through a process called search. The simplest search we have is the linear search. In this system, the computer finds a data and compares it with the one we asked for until a coincidence occurs. If the list is disorganized the linear search is probably the only possible way we have to find the record. We also have the binary search uses the B-tree. It also uses the comparison method. For example, we are looking for a certain record. In this method, the record will be sent to the root of the B-Tree in order to begin search. This search will be done through the right side of the root. If the record we are looking for is smaller than the record placed in the root, the search will be done through the left side. Every time a knot is visited, an empty knot is created, always by comparison. This operation will go on until the record we are looking for is found.

EXAMPLE: Suppose a database with some great mathematicians, where we want to find the registration number 67 (**Figure 3**). Even using a binary search with a recursive method, we have to make a comparison one by one.

In a Binary tree search, the comparisons are made between the nodes. The **Figure 4** shows the database used as an example represented by a binary tree to illustrate the formation of the search tree and the steps to find the desired key, in this case the search is by the register 67. This comparisons use a subroutine to calculate the total order between two values. The search for a specific value can be done with recursive or iterative process. It requires $0(\log n)$ time in common cases. The example was needed 4 steps after tree construction.

The Calculation of the Complexity $0(\log n)$, Which Is the Best Case Possible for a Search Tree

According to N. Wirth (1976) apud S.W. Song [5], given a value x, we want to find, if there is a node in the binary search tree whose key is equal to x. The complexity of the worst case equals the height of the tree. The worst case can be

Figure 3. Database used as an example (source: http://www.heru.com.br).
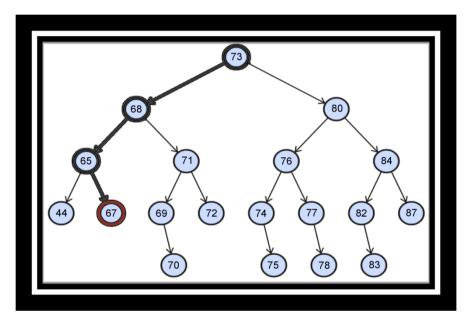

Figure 4. Database represented by a *B*-tree (source: http://www.heru.com.br).

$O(n)$, where *n* is the number of nodes in the tree. When the tree is balanced, the search is efficient and the worst case is $0(\log n)$. It is fundamental to know the average number of comparisons to locate a tree node. This average number is represented by an, and when the tree is balanced a_n is approximately $(\log n)$. If we are analyzing a degenerate (unbalanced) tree, where each node has only one child, we are faced with the worst case and a_n is $(n+1)/2$.

Still according to the same source, it is important to know the value of a_n for an average case. Let the keys $1, 2, \cdots, n$, each with the same probability of being the next one to be inserted, into a binary search tree. The built-in tree can have any of these keys at the root (**Figure 5**).

The probability that the *i* key is the first to be inserted into the tree, initially empty, is $1/n$.
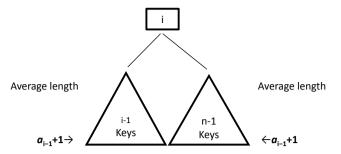
Figure 5. Analysis of the search algorithm in the middle case (source: Song [5]).

Let $a_n^{(i)}$ be the value of the average tree length with n nodes being i in the root. $a_n^{(i)}$ can be expressed in terms of the mean lengths of its subtrees.

$$a_n^{(i)} = \left(a_{i-1}+\right)\frac{i-1}{n}+1\times\frac{1}{n}+\left(a_{n-i}+1\right)\frac{n-i}{n} \tag{7}$$

$$a_n = \frac{1}{n}\sum_{i=1}^{n}\left[\left(a_{i-1}+1\right)\frac{i-1}{n}+\frac{1}{n}+\left(a_{n-1}+1\right)\frac{n-1}{n}\right]=\cdots=1+\frac{2}{n^2}\sum_{i=1}^{n-1}ia_i$$

To get the a_n we must write in terms of the harmonic function

$$H_n = 1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}$$

Results in the expression $a_n = 2\frac{n+1}{n}H_n-3$.

By Euler's formula:

$$H_n = \gamma + \ln(n)+\frac{1}{12n^2}+\cdots$$

$$\therefore \gamma \text{ It's worth approximately } 0,577$$

For high values of n $a_n \cong 2\left[\ln(n)+\gamma\right]-3 = 2\ln n - C$.

In the above expression, C is a constant value. Therefore, $a_n = 0(\log n)$.

It is important to note that for a balanced tree, I say, perfectly balanced, we have $a_n' = \log n$. *For high values of* n : $\lim_{n\to\infty}\frac{a_n}{a_n'}=\frac{2\ln n}{\log n}=2\ln 2 \cong 1,39$.

We have just verified that in the average case, there is an improvement in the number of comparisons, to find the key or desired information, of only 39% when the tree is perfectly balanced. This means that if we need to, we will have to create empty nodes to ensure the trees are balanced. It will not always be possible to get a balanced tree (same number of nodes for trees and sub-trees).

There are resources to make sure that a search is efficient, although in the worst case, like the use of AVL or red-black tree. The average case, according to Song [5], is encouraging and with complexity $0(\log n)$.

This paper has as one of the objectives to compare the efficiency of the Heru Search Method with an optimum binary search tree, which are those assembled from keys and information about the probability of access of each one. This type of tree is used for searching, without insertion or removal of information. The method created by me (the author of this work), is based on input files without

restrictions of size or type, but which will be remodeled by the Heru method and then made available for searching. If you have to enter or remove information from this input file, the remodeling process needs to be redone, just as the search tree would also have to be re-created. There are differences between the binary search trees and the heru method, and the most important is the paradigm break $0(\log n)$. Note that the method presented here guarantees a search with less pitch than the best case of the binary search. This fact places the heru method with performance superior to that obtained by the search *B*-tree and consequently the complexity $0(\log n)$.

The understanding of the method below is not trivial, since it requires in-depth knowledge of the Infinite Series with Multiple Ratios (SMRs) training law, presented in section 1.1. They are formulas and concepts totally unpublished and unknown to the Exact Sciences and it is not the intention or pretension to exhaust the subject and to make it very clear with this paper.

It is believed that this method will revolutionize several fields aimed at the treatment of information, and one of the intentions is to present to the scientific world that occurred the break of paradigm that humanity has used since World War II when linear programming emerged, according to some historians. This one permeates the use of the binary search trees.

2. Heru Search Method

The Heru search method eliminates the comparison one by one. A link between records will occur. This link will begin with the first ideal group of records. The ideal group of records is obtained by applying the formulas of SMRs (Infinite Series with Multiple Ratios) in the records that compose the file. An analogy can be made to how we naturally seek information, where we look at the entire set of values and begin mapping areas that tend to contain the object being searched. The Heru method performs these operations mathematically, eliminating the comparison one by one. The complementary understanding of the conceptualization of the SMRs, becomes fundamental for assimilation of the new reorganization of the dataset in time of storage. Each byte that composes the file is repositioned into idea groups and parallel groups are formed, which are incompatible data with the ideal groups of records. The data processing processes follow the following steps, which cannot be confused with the search processes seen below that only occur with the ordered data.

The Steps for Forming Ideal Groups

Step 1: Submit a file of any size and typing, apply mathematical formulas (from the SMR suite) to hegemonize the bytes of the input file, size *N*, going from B1 (first byte of the file) to Bn (last byte of the file). This treatment is done from the numerical values of each byte, recognized and manipulated by any operating system. The hegemonization process has as main objective to eliminate the dispersivity between bytes, making the sets more homogeneous in the item distancing, points of convergence and other mathematical characteristics that

will allow the organization of the original file in sub-files, when it is not possible to hegemonize in a single step.

Step 2: After processing the information in step 1, the distance between the bytes (applying formulas) is determined and the less dispersive groups are found. After the creation/separation of ideal groups (elements classified and organized according to the SMR concepts) minimally, the pertinent differentiations are made for each group and numerical sets derived from this information are stipulated.

Step 3: The elements that cannot position in the ideal groups form the waiting groups and the process is reapplied until there are no more waiting groups.

Step 4: Record the number of groups and information of the characteristics of each one, such as: first element, distances, data quantities of each group and mathematical information extracted from each group and any information peculiar to the processes of hegemonizations mentioned in Step 1.

Step 5: The bytes that could not be allocated after going through the previous steps form a sub-file with identification of B1 (first byte, which is the first element that was not allocated) and Bn (last unallocated element). Having B1 and Bn, the process repeats itself until there are no misallocated elements left. The sub-files receive the same mathematical treatment given to the N-sized input file.

The Heru search method uses a byte-by-byte repositioning system of any type of file (sound, video, image, text, etc.) without loss of information or quality, since operations are applied for storage and retrieved at runtime. Below are some figures that represent a set of data processed, stored by the Heru method, where it is possible to observe that the performance is superior than the binary tree, even though it is a very small amount of records. The B-tree required 4 steps to find the requested record 67, and the Heru Search Method found in 3 steps: The first one which is the structure itself and its ideal queues, which are the data that have gone through the training process listed above and which compose the method, (**Figure 6** below), the second step is data reallocation, which is the repositioning of the keys or records that were not allocated in the
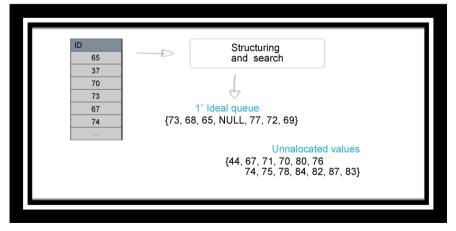

Figure 6. Representation of the first step of the heru method (source: http://www.heru.com.br).

ideal queue of the first step and will give rise to a second queue called the queue, represented by **Figure 7**. Finally, in this example, search data found at the first ideal queues and de process interrupted or finished (**Figure 8**). So, and even if we had repeated records we would only return the exact ones. It is worth mentioning that the complexity of formulas and processes are rewarded by an operationality and speed at the moment of searching the records.

STEP 1—The Structuring

The process in a cyclic way, allocating data until the desired value is found.

STEP 2—Reallocation

If the desired value is not present in the first ideal queue or groups, the non allocated values will be restructured.

Finalization the process

With the search object found in the second ideal queue, the cycle is interrupted. On this case, it was possible to reach the result in two steps, half of the steps needed in the Binary search tree.
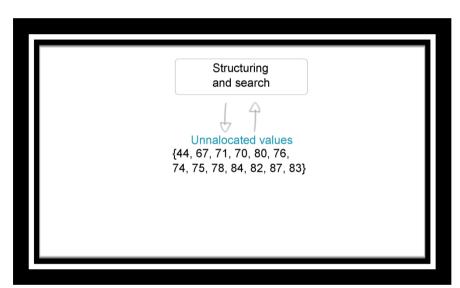

Figure 7. Representation of the second step to reallocation (source: http://www.heru.com.br).

Figure 8. The last step in this example (source: http://www.heru.com.br).

3. Results and Discussion

The binary search method is used on a large scale and probably the most popular of all with the best results obtained by the complexity $0(\log n)$. By getting a better performance than the more traditional search method used, the Heru search method has proved the applicability of Infinite Series with Multiple Ratios in the computational field such as search systems and database engines. The method will operate along the semantic treatment of the searched term in the lowest level of the search method, replacing the current binary search.

This replacement will result in efficient operations during a search and can be applied to any kind of search system.

Acknowledgements

To the students of the computer science course of the Federal University of Fronteira Sul, Rodrigo Bueno Tomiosso and Felipe Zanela Bourscheid, who assisted in the design of the figures.

Competing Interests

There are no competing interests in this work of any nature.

Authors' Contributions

This paper has a single author and all contributions are of the own. Some contributions have been received in relation to the figures, but are quoted in the acknowledgments.

References

[1] Rodrigues, E. (1995) Infinite Series with Multiple Ratios. *Journal of scientific divulgation*, **2**, 11-40.

[2] Horizonte, M.G. (1997) 49th Annual Meeting. *Proceedings of the Brazilian Society for the Advancement of Science* (*SBPC in Portuguese*), Federal University of Minas Gerais, Belo Horizonte, 715 p.

[3] Maringá, P.R. (1998) 6*th Special Meeting. Proceedings of the Brazilian Society for the Advancement of Science* (*SBPC in Portuguese*), Maringá, 404-405.

[4] Natal, R.N. (1998) 50th Annual Meeting. *Proceedings of the Brazilian Society for the Advancement of Science* (*SBPC in Portuguese*). Federal University of Rio Grande Do Norte, Natal, 1044 p.

[5] Song, S. (2008) Trees and Binary Trees.
https://www.ime.usp.br/~song/mac5710/slides/05tree

List of Abbreviations

SMRs: Infinite Series with Multiple Ratios.

Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: http://papersubmission.scirp.org/

Or contact ajcm@scirp.org