Global Journal of Computer Sciences: Theory and Research

Volume 06, Issue 2, (2016) 47-53

www.gjcs.eu

Searching Big Data via cyclic groups

Timur Karacay *, Faculty of Engineering, Program of Statistics and Computer Sciences, Baskent University, Baglica Campuss, 06810 Ankara, Turkey**.**

Suggested Citation:
Karacay, T. (2016). Searching Big Data via cyclic groups. *Global Journal of Computer Sciences: Theory and Research*. *6*(2), 47-53.

Received June 14, 2016; revised August 18, 2016; accepted October 29, 2016.
Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

Abstract

We look up for a certain information in big data. To achieve this task we first endow the big data with a group structure and partition it to it's cyclic subgroups. We devise a method to search the whole big data starting from the smallest subroup through the largest one. Our method eventually exhausts the whole big data.

Keywords: BigData, topological groups, dual groups, linear search.

* ADDRESS FOR CORRESPONDENCE: **Timur Karacay,** Faculty of Engineering, Program of Statistics and Computer Sciences, Baskent University, Baglica Campuss, 06810 Ankara, Turkey. *E-mail address*: tkaracay@baskent.edu.tr /
Tel.: +90-312 246 6666

1. Introduction

The term big data describes the large volume of data. Data may be structured or unstructured. It grows in a business day-to-day basis. Data comes in all types of formats from structured data in traditional databases, unstructured text documents, email, numeric data, video, audio, sale prices, financial transactions, etc.

Big data can be analyzed for insights that lead to better strategic decisions for a business [7] [8], [9], [10].

Organizations may collect data from a variety of sources, including regular databases alongside other data sources. It is what organizations do with the data that matters. For a particular application, usually a subset B of the whole big data is chosen. For instance the criminal department of police may search only the plates of cars, or pictures of suspected persons. A doctor may look for symptoms of his patients. Therefore the underlying set B may change from application to application. However the method we present here works for all underlying sets as well.

Some applications may require to extract data from various reservoirs to obtain a smaller underlying set B. This task is accomplished with various methods of data manipulation. The underlying set would be as big or as small as to fit the requirements of the application.

This paper deals with finding a certain information in the underlying set B, not how to constract it.

Any set consisting of finite number of finite sets is finite. Even the whole big data, the whole set of all existence data is a vast amount it is always finite. Our underlying set B will be a set whose elements are extracted from, so it will always be finite.

2. Searching in Ordered and Unordered Sets

2.1 Preliminaries

The following known resulsts are restated in order to make the paper self-sufficient.

Computer science give us useful methods to search ordered and unordered sets. It might be worthwile to give a short decription of the existence methods of searhing sets in order to compare them with our method.

Assume that a value **val** is searched.

2.2 Searching in Linearly Ordered Sets

If the set is linearly ordered, linear search is preferred for relatively small sets (say if the list has less than 100 elements). For larger linearly ordered sets binary search should be employed.

Since our underlying set B is finite its elements can be linearly ordered by extracting the elements one by one and attaching a numeral to each severed element till the set is exhausted. If B has n elements, the number of different ordering could be made is then n! This method can also be employed to construct an array from the unordered set. However this ordering is on the numerals attached to the elements of the set during the selection process, so the ordering has no relation with the actual values. Hence binary search cannot be employed to seek the value **va**l. If by chance there is a natural ordering on the set, that ordering can of course be employed for binary search.

In a linear search, assuming the underlying set B has n elements, the best case is when the **val** is equal to the first element of the list, in which case only one comparison is needed. The worst case is when the **val** is not in the list or occurs only once at the end of the list, in which case n comparisons are needed. In computer science the algorithms of linear search are very simple to implement.

2.3 Arrays

If the set forms an array, the search algorithms use the indices to visit all the elements of the array. However an array need not be linearly ordered relative to the values of the elements. Hence a search method must visit all the elements of the array through the indices implying that binary search is not suitable for arrays.

2.4 Other Types of Ordered Sets

In general our underlying set B is not of the forms Tree sets, Linked Lists, HashSets as they require special methods of construction to build these data types. Of course the underlying set may be reorganized as one of such data types. But this process needs some labouring. Should the content of the set B change frequently, repeated reorganization may be more trouble than it is worth.

2.5 Unordered Sets

If the set is not ordered and is not in the form of an array like type, some preprocesses are needed to employ a search method. Some programming languages like java or python have methods to order the set, to construct an array from the set or search directly the unorderd set for the sought value. We may employ those methods to search our cylic subgroups.

2.6 Root of Unity

A root of unity, called a de *Moivre number*, is any complex number that gives 1 when raised to some positive integer power n. Thus an n-th root of unity, where n is a positive integer (i.e. n = 1, 2, 3, …) is a complex number z satisfying the equation

$$z^n = 1$$

An n-th root of unity is primitive if it is not a k-th root of unity for some smaller k. Every n-th root of unity z is a primitive r-th root of unity for some r where

$$1 \leq r \leq n$$

If $z^1 = 1$ then z is a primitive first root of unity, otherwise if $z^2 = 1$ then z is a primitive second (square) root of unity, otherwise, . . . , and by assumption there must be $z^r = 1$ at or before the n-th term in the sequence.

If z is an nth root of unity and $a \equiv b (mod\ n)$ then $z^a = z^b$. By the definition of congruence, $a = b + kn$ for some integer k. But then,

$z^a = z^{b+kn} = z^b z^{kn} = z^b (z^n)^k = z^b 1^k = z^b$

Therefore, given a power z^a of z, it can be assumed that $1 \leq a \leq n$.

Any integer power of an nth root of unity is also an nth root of unity:

$(z^k)^n = z^{kn} = (z^n)^k = 1^k = 1$

Here *k* may be negative. In particular, the reciprocal of an n-th root of unity is its complex conjugate, and is also an n-th root of unity: $z^- = \frac{1}{z} = z^{-1} = 1.z^{-1} = z^n.z^{-1} = z^{n-1}$

$z^- = 1/z = z^{-1} = 1.z^{-1} = z^n.z^{-1} = z^{n-1}$

Let z be a primitive nth root of unity. Then the powers $z^1, z^2, ..., z^{n-1}, z^n$ are all distinct.

An n-th-degree polynomial equation can only have n distinct roots implying that the powers of a primitive root are all of the n-th roots of unity.

An n-th-degree polynomial equation can only have n distinct roots implying that the powers of a primitive root $z^1, z^2, ..., z^{n-1}, z^n = z^0 = 1$ are all of the n-th roots of unity.

3. Groups

Throughout the paper , , and will denote the sets of natural numbers, integers, real numbers and complex numbers, in their respective order. The unit circle in the complex plane is the set

$$T = \{ z \mid \quad z \in \mathbf{C} \quad |z| = 1 \mid \}$$

T is a locally compact topological group. Usually the exponential form

$$\mathbf{z} = e^{2\pi i t} \quad (t \in \mathbf{R})$$

is used to denote the elements of the unit circle T ([1], [2]).

Definition 1

A group G is said to be a finite group if the set G has a finite number of elements. In this case, the number of elements is called the order (size) of G, denoted by |G|.

The letter e will stand for the unit of the multiplicative groups.

Definition 2

Let a be an element of the group G. If there exists a positive integer n such that an = e, then a is said to have finite order, and the smallest such positive integer is called the order of a, denoted by o(a). If there does not exist a positive integer n such that an = e, then a is said to have infinite order.

Definition 3

Let G be a group, and let a be any element of G. The set

$$< a >= \{ x \in G \mid \quad x = a^n \quad for\ some\ \ n \in \mathbf{Z} \ \}$$

is called the cyclic subgroup generated by a. The group G is called a cyclic group if there exists an element such that G =< a >. In this case a is called a generator of G.

In what follows gcd stands for greatest common divisor and k|m means k is a divisor of m.

Theorem 1

Every subgroup of a cyclic group is cyclic.

Theorem 2

In a finite cyclic group, each subgroup has size dividing the size of the group. Conversely, given a positive divisor of the size of the group, there is a subgroup of that size.

Theorem 3

In a finite cyclic group, two elements generate the same subgroup if and only if the elements have the same order.

Theorem 4

Let $G =< a >$ be a cyclic group with $|G| = n$.

a. If $m \in \mathbf{Z}$, then $< a^m >=< a^{gd} >$, where $d = gcd(m, n)$, and a^m has order $n|d$.

b. The element a^k generates G if and only if $gcd(k, n) = 1$.

c. The subgroups of G are in one-to-one correspondence with the positive divisors of n.

d. If m and k are divisors of n, then

$$< a^m > \subseteq < a^k > \quad if\ and\ only\ if \quad k \mid m$$

Since the cyclic groups are abelian, they are often written additively and denoted $\boldsymbol{Z_n}$ with the identity written 0. All subgroups and quotient groups of $\boldsymbol{Z_n}$ are cyclic and abelian. Specifically, all subgroups of $\boldsymbol{Z}$ are of the form $m\boldsymbol{Z}$, with an integer ≥ 0. The quotient notations $\boldsymbol{Z_n}$, $\boldsymbol{Z}/n$, $\boldsymbol{Z}/(n)$, and $\boldsymbol{Z}/n\boldsymbol{Z}$ are standard alternatives.

Theorem 5

The direct product of two cyclic groups, $\mathbf{Z}_m \times \mathbf{Z}_n$ is cyclic if and only if the $gcd(m; n) = 1$.

Theorem 6

Let $\mathbf{Z}_m$ be cyclic group of order m. Then $\mathbf{Z}_m \times \mathbf{Z}_n = \mathbf{Z}_{mn}$ where $gcd(m, n) = 1$.

A primary cyclic group is one whose order is a power of a prime. The invariant factor decomposition states that every finitely generated abelian group G is isomorphic to a direct sum of primary cyclic groups and infinite cyclic groups. That is, every finitely generated abelian group is isomorphic to a group of the form

$$\mathbf{Z}^n \times \mathbf{Z}_{q_1} \times \mathbf{Z}_{q_2} \times \ldots \times \mathbf{Z}_{q_t}$$

where the rank $n \geq 0$, and the numbers $q_1, q_2, \ldots, q_t$ are powers of (not necessarily distinct) prime numbers. In particular, G is finite if and only if n = 0. The values of $n, q_1, q_2, \ldots, q_t$ are uniquely determined by G with a unique order and q_1 divides q_2, which divides q_3 and so on up to q_t.

Theorem 7

$$\mathbf{Z}_n^* = \{ a \in \mathbf{Z}_n \mid \quad \gcd(a,n) = 1 \ \}$$

is a group under multiplication.

Every finite cyclic group of order n is isomorphic to the additive group of $\mathbf{Z}_n$, the integers modulo *n*. Every cyclic group is an abelian group, and every finitely generated abelian group is a direct product of cyclic groups.

Euler's totient function (Euler's ϕ function) counts the positive integers up to a given integer *n* that are relatively prime to *n* ([4], [5]). Euler's product formula states that

$$\phi(n) = n \prod_{p|n} (1 - \frac{1}{p})$$

Here the product is over distinct prime numbers p dividing n. It is known that Euler function is multiplicative in the sense that

$$\phi(nm) = \phi(n)\phi(m), \quad \gcd(m,n) = 1$$

This leads the fundamental theorem of arithmetic;

$$n = p_1^{k_1}.p_2^{k_2}...p_r^{k_r} \qquad (2)$$

Where p_i are primes, k_i are integers which says that every natural number *n* greater than one has a unique factorization in terms of prime numbers.

A number k is called a cyclic number if it has the property that $\mathbf{Z}_k$ is the only group of order *k*, which is true exactly when *gcd(k,φ(k)) = 1* ([6]). The cyclic numbers include all prime numbers, but also include some composite numbers such as *15, 35, 65,…* . All cyclic numbers are odd except *2*. Some of the cyclic numbers are:

1, 2, 3, 5, 7, 11, 13, 15, 17, 19, 23, 29, 31, 33, 35, 37, 41, 43, 47, 51, 53, 59, 61, 65, 67, 69, 71, 73, 77, 79, 83, 85, 87, 89, 91, 95, 97, 101, 103, 107, 109, 113, 115, 119, 123, 127, 131, 133, 137, 139, 141, 143, …

Sequence A003277 in the OEIS ([3]):

4. Implementation

Now let us return to our big data, the underlying set *B*. In order to search it for the sought value **val** we follow the steps:

1. The underlying set *B* is unordered containers that store unique elements in no particular order.
2. There are plenty of methods to count the elements of a set. For instance, the method *len(B)* in python will count the elements of *B* and give us the cardinal number *|B| = b* of our underlying set.
3. Once the number *b* is obtained, we look up for the least primary number *p* with $b \leq p$. The prime number *p* can be determined easily with a suitable computer program. Of course *p = b* if *b* is prime.

4. Determine the primes less then *p* and order them as

$1 < p_1 < p_2 < \ldots < p_k < p.$

5. Define a one to one map from $B \cup P$ to the cyclic group $\mathbf{Z}_p$, where *P* is the set consisting of the elements{ b+ 1, b + 2,..., p } joined to *B* to make possible the one-to one mapping, if necessitates. However no necessity occurs if P is empty (i.e., b is prime). Of course such a map is not unique, but this causes no trouble.

6. Represent the cyclic group $\mathbf{Z}_p$ as

$$\mathbf{Z}_p = \mathbf{Z}_{p_1} \times \mathbf{Z}_{p_2} \times \ldots \times \mathbf{Z}_{p_k} \qquad (3)$$

7. All nodes of $\mathbf{Z}_{p_i}$ *(i = 1, 2,..., k)* may be visited by starting from the least order to the highest order of cyclic groups (3), in their respective order. This procedure makes the visitor touch at all the elements of $\mathbf{Z}_p$.

8. While at a node gr in $\mathbf{Z}_{p_i}$ look at back to the underlying set *B* using the inverse map $^{-1}$(g^r) for the value **val** sought.

9. If *b < p* the search method may be made to skip the nodes g^r for *r > b*.

5. Conclusion

This method can be applied to all unordered set *B*. It has less labouring than linearly ordering *B* and then searching it.

References

[1] Hewitt, E. and Ross, K.A. *Abstract Harmonic Analysis I-II*. Berlin, Springer-Verlag, 1970.

[2] Rudin, W. *Fourier Analysis on Groups*. New York: Interscience, 1962.

[3] OEIS. 2016. Available from: http://oeis.org/.

[4] Calvin T. L. *Elementary Introduction to Number Theory*, 2nd ed., Lexington: D. C. Heath and Company, LCCN 77-171950, 1972.

[5] Anthony J. P. and Donald R. B. *Elements of Number Theory*. Englewood Cliffs: Prentice Hall, LCCN 77-81766, 1970.

[6] Abramowitz, M. The Euler Totient Function. In Stegun, C. A. (ed.). 24.3.2 *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th edition. New York: Dover, p. 826, 1972.

[7] Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D. And Tufano, P. Analytics: The real-world use of big data. How innovative enterprises extract value from uncertain data IBM Institute for Business Value, 2012.

[8] Diebold, F.X. A personal perspective on the origin(s) and development of big data: The phenomenon, the term, and the discipline (Scholarly Paper No. ID2202843)

[9] Chung, W. BizPro: Extracting and categorizing business intelligence factors from textual news articles. *International Journal of Information Management*, 2014, 34 (2), pp. 272-284

[10] Jiang, J. Information extraction from text. In: Aggarwal, C.C. and Zhai, C. (ed.) *Mining text data* United States: Springer, 2012, pp. 1141