An Efficient Algorithm of Chinese String Sort in User-defined Sequence

Haijun Zhang
School of Computer Science and Technology
Xinjiang Normal University
Urumqi China 830054
ustczhj@mail.ustc.edu.cn

Shumin Shi
School of Computer Science and Technology
Beijing Institute of Technology
Beijing China 100081
nmssm@126.com

***Abstract*--Existing sort algorithms are difficult to implement Chinese string sort in user-defined sequence. This paper proposes an efficient string sort method in user-defined character order. On the basis of the consecutive numbers which used to define the custom order of characters, the hash table structure is employed to convert each string into corresponding array of integers. By taking the maximum number of characters as the new radix, the Radix sort algorithm is used to implement fast sort of strings in user-defined order. Theory analysis and experiments show that the sort algorithm of this paper can easily achieve Chinese string sort in user-defined order in linear time and space complexity. This sort algorithm has a better time performance *than* that of Quick Sort algorithm, and it can effortlessly extend to string sort applications of other languages.**

Keywords-Radix sort; string sort; user-defined order; hash table structure

Ⅰ. INTRODUCTION

Sort is a basic operation in the field of data processing, and its main role is to accelerate the speed of retrieving and merging of data. String sort is an important foundation for natural language processing technology, which has important applications in the fields such as new words extraction from large-scale network corpus, meaningful string detection, public opinion monitoring, etc. In the process of external sort of large-scale strings, the internal sort algorithm of string has played a crucial role.

In this paper, we will study sort algorithms thoroughly, especially Radix sort algorithm and propose an algorithm of Chinese string sort in user-defined order. The paper is organized as follows: Section Ⅱ gives the relative studies; Section Ⅲ describes the proposed user-defined order Chinese string sort algorithm and theoretically analyzes the algorithm's time and space complexity; Section Ⅳ conducts the experiments and discussions; Finally, it gives the conclusions and the direction of further study.

Ⅱ. RELATIVE STUDIES

Until now, people have found lots of efficient sort algorithms. Among them, the most are comparison-based sorting algorithm, such as Insertion sort, Bubble sort[1] as well as Quick sort[2] etc. The time complexity of Insertion sort and Bubble sort is $O(n^2)$,and the time complexity of Quick sort algorithm is $O(nlgn)$ in average cases, $O(n^2)$ in the worst cases. Bucket sort[3,4] is a special sort method, which has a very high efficiency (with time complexity $O(n)$) for uniform data, but degenerates into $O(n^2)$ in a very uneven data. Radix sort[5] algorithm is a non comparison-based sorting method whose time complexity is $O(dn)$. In theory, Radix sort is the fastest sort algorithm, but generally used in the field of integer with identical digits, which to some extent limits its application scope. He[6] proposed a grading sort method for complex data, which can effectively enhance the efficiency of some particular kinds of data, but sorting for the Chinese strings, there is a certain rooms for performance to be improved.

The above sort algorithms are conducted basing on the code sequence, resulting in the sort results nothing more than the positive-sequence and invert-sequence[8]. For example, four characters: "A", "B", "C", "D" by sorting only produce "ABCD" or "DCBA" two specific sequences and cannot generate the results according to user-defined order, such as: "DBAC" and "DABC" etc. After all, in some cases, there need to sort the data according to user-defined order, such as searching, may need to output the results in a particular order, but the

978-0-7695-5063-3/13 $26.00 © 2013 IEEE
DOI 10.1109/IALP.2013.33

current sorting methods are difficult to meet this requirement. In this paper, we will provide an efficient Chinese string algorithm of sort in user-defined sequence, and analyze it by theory and experiments. The important feature of this algorithm is that it can achieve user-defined sequence string sort, and the time and space complexity of it is linear.

III. AN ALGORITHM OF STRING SORT IN USER-DEFINED SEQUENCE

A string sort in user-defined sequence needs to reset the orders of characters in alphabet, and to achieve sorting in new order. In general, assuming a string sequence $\{S_1, S_2, \ldots\ldots S_n\}$ including n strings, and each string including d characters ($C_i^0, C_i^1, \ldots C_i^{d-1}$), the string sort in user-defined order is defined as any two strings in the string sequence satisfying the following orderly relationship: $(C_i^0, C_i^1, \ldots C_i^{d-1}) \le (C_j^0, C_j^1, \ldots C_j^{d-1})$, where there is $C_i^0 \le C_j^0$, $C_i^1 \le C_j^1 \ldots\ldots C_i^{d-1} \le C_j^{d-1}$, rather than the original machine code character order.

However, it is very troublesome to reset character order of machine code and comparison-based sort algorithms are difficult to be adapted to meet above requirements. As Radix sort is a non comparison-based algorithm and has the best sorting efficiency, we think to take Radix sort as the base of study.

A. THE DESCRIPTION OF THE PROPOSED ALGOTITHM

Radix sort conducts each keyword sorting by Counting sort, and the premise of Counting sort is that each keyword is an integer. However, a string is a sequence of characters, which is difficult for Radix sort to process. In theory, if a string can be converted into a sequence of number, the Radix sort for strings can be realized. The conversion between string and sequence of number must be fast enough, otherwise it will affect the overall performance of the string sort.

In view of the above requirements, we designed a quick conversion method between string and numeric sequence. First of all, on the basis of Chinese alphabet, each Chinese character is given a number according to the user-defined order to ensure that each Chinese character has a unique integer number. And then, the characters and the corresponding number are loaded into hash table, so that the corresponding number of each character can be retrieved quickly. In order to achieve the Radix sort for Chinese strings, we propose to use an integer array which having the same length with string to represent the string. It is very quickly for converting a string to the corresponding integer array by retrieving the number of character from hash table structure. As a result, the string and the corresponding integer array are one-to-one mapping because each character and a unique number are one-to-one correspondence. After sorting integer arrays by Radix sort, the ordered strings can be obtained by converting the sorted integer arrays into corresponding strings. According to the above correspondence, if the integer arrays are sorted, the string sorting can also be achieved at the same time. The flow of above algorithm is shown in Figure 1.

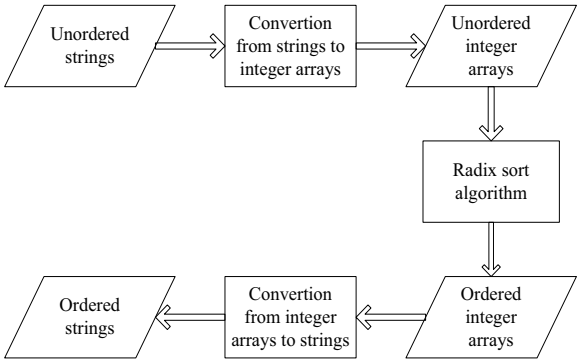

Figure. 1. The flow of the proposed sort algorithm

As the treated objects of Radix sort is integer with identical digits, the each element in integer array is treated as one digit in radix sort. However, there are big differences of length among strings which can cause inconsistent length of integer arrays and cannot employ Radix sort for strings. In order to facilitate the implementation of Radix sort for strings, this paper proposes to use identical length array to represent different length strings. The corresponding integer arrays are left-aligned, and for short string, the right vacant elements are padded with zeros. Here are some string and integer array conversion examples, as shown in Table Ⅰ.

Table Ⅰ.Table of conversion between strings and integer arrays

Strings	Integer arrays
阿贾克斯	[2][2093][2558][4570]
中国财经	[6422][1643][384][2314]
中国财税	[6422][1643][384][4545]
注册公司	[6494][418][1486][4565]
注明出处	[6494][3363][635][655]
电脑	[945][3475] [0][0]
自己想	[6593][2028][5313][0]

The above table lists some samples of strings and the corresponding integer arrays. This sort process is actually a change in the "radix" as used in the ordinary Radix sort, which radix is 10 and the radix in above algorithm becomes the sum of characters in alphabet (for example, English is 26, GB2312-80 Chinese character set is 6763).

B. Analyses

For time complexity, according to Figure.1, assuming the string data set size being *n*, the maximum length of the strings being *d*, and the retrieval efficiency of hash table is *O(1)*, then the complexity of conversion from strings to integer arrays is *O(dn)* and the complexity of Radix sort for integer arrays is *O(dn)*. The process of conversion from integer arrays to strings is similar to the conversion from strings to arrays and its complexity is also *O(dn)*. According to above analysis, the total complexity of the proposed algorithm is *O (dn) + O (dn) + O (dn) =O (dn).*

For space complexity, the space usage of Chinese character (encoded in GB2312) is *2* bytes and the space for corresponding integer of character also occupies *2* byte. Therefore, the space complexity of the proposed algorithm for Chinese strings is *O(2n+M)*, where *M* is the largest user-defined number of Chinese characters.

Based on the above analysis, compared with the original radix sort algorithm, the proposed algorithm has no increase in time and space complexity, but it can achieve Chinese string sort with *O(dn)*, which is a linear time and space sort algorithm.

Ⅳ. Experiments and Discussions

A. Experiments

The conditions used in experiments are as below: the clock frequency of CPU being *2.66 GHz*, memory capacity being *2 GB*, and the operating system being Windows XP with SP3. The test data contains 16 million strings extracted from large-scale corpus provided by Sogou Labs[10]. In order to detailed analyze the performance of the proposed algorithm, lots of comparison experiments are carried out. The experimental data are shown in table Ⅱ.

Table Ⅱ. Experiments among different size of data

Data size (ten thousand)	Sort time consumption(MS)				
	A	B	C	D	E
100	1266	1266	1266	1266	1266
200	2703	2641	2640	2641	2594
400	5406	5422	5438	5422	5391
800	11938	12015	11406	11406	11234
1100	15985	16062	16031	15938	17063

B. Discussions

In order to reduce accidental impacts, the same experiment for the same data is employed for five times; and to study the time complexity of the proposed algorithm, the incremental size data are used to facilitate comparisons. To illustrate the relationship between sorting time consumption and data size, the diagram between them are drawn, shown in Figure 2.

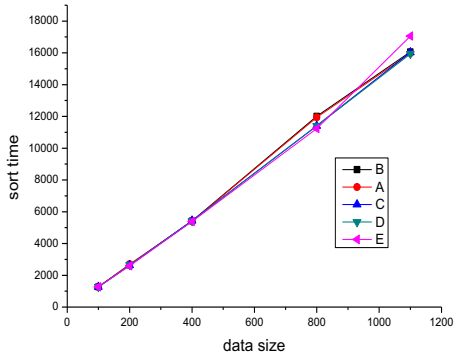

Figure 2. Diagram between sorting time and data size

From Figure 2, it is concluded that the relationship between sorting time and data size is linear, which further validates the conclusions in Section Ⅲ, and demonstrates that the algorithm is a linear time sorting algorithm.

C. Comparisons with Quick Sort

To further study the performance of the proposed algorithm, the comparative experiments between it and Quick sort are employed, and the data shown in table Ⅲ.

Table Ⅲ. Comparative experiments between algorithms

Data size (ten thousand)	800	1100	1600
Proposed algorithm(MS)	11406	15938	23797
Quick sort(MS)	36797	51875	76390

From table Ⅲ, it can be seen that the speed of the proposed sort algorithm is much faster than that of Quick sort for incremental data size. With the growth of size of sorting data, the time consumption gap between the two algorithms is increasing. From the comparative experiments, it can also validate the aforementioned theoretical analysis result in section Ⅲ,that is: the time complexity of the proposed sort algorithm is a linear time sort algorithm and it achieves much higher speed than that of Quick sort algorithm.

Ⅴ. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose an algorithm of Chinese string sort in user-defined order, in which we take Radix sort algorithm as the sorting basis and employ identical length integer arrays to represent strings to implement the Radix sort for strings. To achieve higher efficiency, we design a method to convert between strings and corresponding integer arrays by hash table structure. The most prominent feature of this algorithm is that it can achieve user-defined order string sorting. The theoretical analysis and experimental results show that this sort algorithm is a linear time sort and is much faster than that of Quick sort algorithm. Though the experiments are for Chinese strings, it can be extended to other languages because the idea is generic.

For the length of integer arrays are identical, this algorithm is more suitable for sorting strings with identical or even length, and it will result in storage space wasting for strings with different lengths. As for the further studies, we plan to apply it in external sort for large-scale candidate strings to improve the extraction efficiency of new word identification basing on repeats in large-scale corpus.

ACKNOWLEDGMENT

This paper is funded by Xinjiang Uygur Autonomous Region Natural Science Foundation of China (2012211A057).

REFERENCES

[1]Owen A. Bubble sort: An archaeological algorithmic analysis[C]. In: Proc of the 34th SIGCSE Technical Symp on Computer Science Education; 2003; New York: ACM Press; 2003. p. 1-5.

[2]Hore C. Quicksort [J]. The Computer Journal 1962, 5(1): 10-16.

[3] Yang Lei , Huang Hui , Song Tao The sample separator based distributing scheme of the external bucket sort algorithm[J].Journal of Software , 2005 , 16 (5) : 643 - 651 (in Chinese)

[4] Yang Lei and Song Tao. The Array Based Bucket Sort Algorithm[J]. Journal of Computer Research and Development, 2007,44(2):341-347.

[5]Cormen TH, Leiserson CE, Rivest RL *et al.* Introduction to algorithms (2nd Ed). Cambridge MA: MIT Press; 2001.

[6]He Wenming and Cui Junzhi. New High Efficient Sorting Algorithm by Grading to Character String Data[J]. MINI-MICRO SYSTEMS, 2004, 25(4): 698-701.

[7] Wang Xiangyang1 A new sorting method by base distribution and linking [J]. Chinese Journal of Computers, 2002 , 23 (7) :774 – 777.

[8] Zhang haijun, Pan weimin, Munina. A algorithm of string sort in custom order[J]. MINI- MICRO SYSTEMS, 2012, 33(9):1968-1971.

[9]Yan Weimin and Wu Weimin. Data structure(C language version). Beijing: Tsinghua University Press, 1997.

[10]Sogou Lab. The Internet Corpora(SogouT) Ver.2008, 2009, (Accessed 2009-7-25, at http://www.sogou.com/labs/dl/t.html).