Sort Chinese String based on MSD Radix Sort

Zeen Wang, College of Engineering, Northeastern University, USA

Xiyue Suo, College of Engineering, Northeastern University, USA

Shuo Wang, College of Engineering, Northeastern University, USA

Abstract

Choosing a reasonable sorting method in the program design is conducive to forming the best algorithm, this includes multiple factors, such as performing time, storage space, and stability. This study majorly concerns radix sort, which assigns elements into buckets with different significant digits. This radix sort needs to work with other mechanisms to complete the sorting, especially sorting Chinese needs to convert Chinese character into pinyin, this process will need the external jar package. MSD and LSD accomplish this process in opposite ways. Also, HashMap stores pairs of elements to prevent the order problem. Comparing time and space complexity to other popular algorithms is a crucial part of this study.

Keywords

MSD radix sort, LSD radix sort, Dual-Pivot Quick Sort, Tim sort, Husky sort, Pinyin4j, UTF-8, Identity HashMap.

1 Introduction

Chinese is the most spoken language, so it makes sense to study efficient Chinese sorting algorithms. Usually when sorting Chinese, the order taken is the Pinyin (Use English letters to represent the pronunciation of Chinese characters) order. So, the first step to sort Chinese String is converting all Chinese characters in String to Pinyin. We can use a Java library named pinyin4j to do that. And before sorting Pinyin Strings, we need get ASCII code (Fig 1. ASCII code) of each character in Pinyin String, because sorting algorithm can only sort digits. And then use MSD Radix Sort Algorithm to sort Pinyin Strings (already get their ASCII code). The last step is converting Pinyin to Chinese. The flow of all steps is showing in the Fig 2. The flow of sorting Chinese String.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Fig 1. ASCII code

Fig 2. The flow of sorting Chinese String

2 Pinyin4j

Pinyin 4j is a popular Java library that supports conversion between Chinese characters and the most popular Pinyin system. There are two advantages of pinyin4j. The first one is that pinyin4j supports multiple pronunciations of a single Chinese character. For example, taking the Chinese character '和' as input, you will get eight Pinyin results (hé hè huó huò hú), depeding on different contexts, which is meaningful when sorting Chinese String[1]. The second one is that pinyin4j supports formatted output of Pinyin, such as tones of Chinese characters. For example, pinyin4j can convert “王泽恩” to “wang2 ze2 en1”, convert “王硕” to “wang2 shuo4”, convert ”索锡岳” to “suo3 xi1 yue4”, the number means the tone of each character in Chinese.

3 Identity HashMap

To set the different Chinese words with same pinyin and tone. As we know the HashMap only can keep one key towards one value. The HashMap can’t store the same key. So, the identity HashMap can solve this, in the identity HashMap we can store the serval values in one key. For example, the key is “a1 bin1”, it can store values “阿斌”, “阿彬”, “阿滨”.

4 Radix Sort

Radix Sort was invented by Herman Hollerith in 1887[2]. Radix Sort is a non-comparative sort[3]. It avoids comparison by assigning them to buckets with the ordered significant radix. Then repeat the

assigning process with multiple significant digits, recursively until the longest digit was taken. For this reason, radix sorting is also called Bucket Sorting and Digital Sorting.[2]

In radix sort, the elements with the same value on the current digit are "bucketed" together each time, and there is no need to exchange positions. So, Radix Sort is a stable algorithm.

The radix sorting method can be the LSD (Least significant digital) method or the MSD (Most significant digital) method.

1.LSD Radix Sort:

It starts sorting from the end of strings (leftmost), it shows in Fig 3. LSD. The time Complexity in Best and Worst-Case time complexity is O(N*M), where M = length of the longest string. And the auxiliary space is O(N + B).[4]

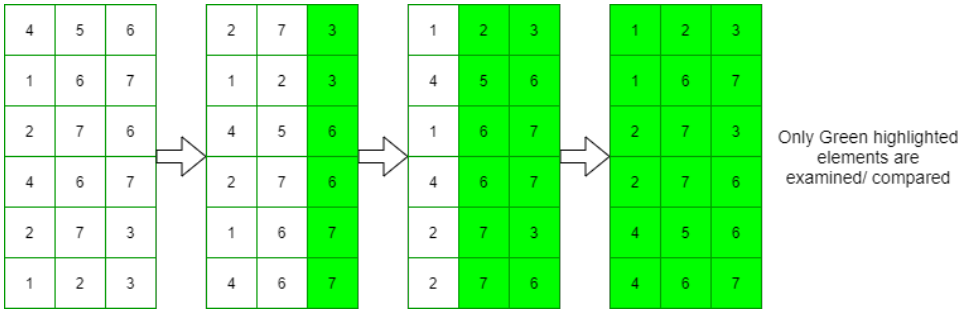

Fig 3. LSD

2.MSD Radix Sort:

It starts sorting from the beginning of strings (rightmost). It shown in Fig 4. MSD.

The time complexity in Best Case is O(N) and in the worst case is O(N*M), where M = the average length of strings. And the auxiliary space is O(N + MB), where M = length of the longest string and B = size of radix.

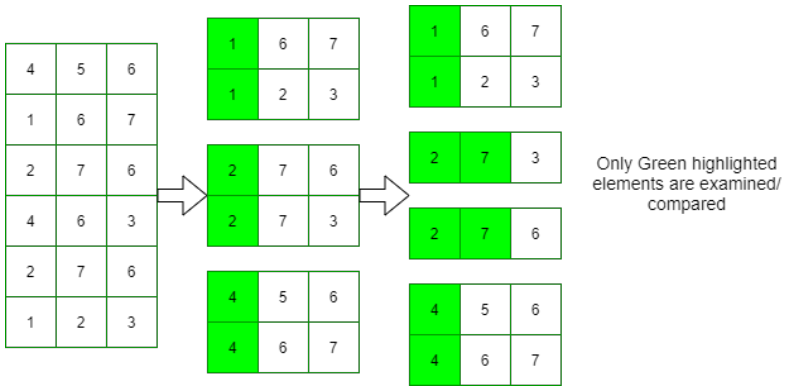

Fig 4. MSD

The biggest problem with LSD radix sorting is that it starts with the number with the smallest difference. If we can start with the most significant number, the first pass will be of great help in sorting the entire range, and each subsequent pass will only deal with details. The idea of MSD radix sort is to divide all numbers with equal values into their respective buckets, and then perform the same operation on all buckets until the array is sorted. Naturally, this suggests using a recursive algorithm, but it also means that we can now sort items of variable length without having to touch all the numbers to get a sorted array. This makes the classification of MSD bases much faster and more useful.But MSD uses recursion, so it requires more space than LSD. This means that MSD is much slower than LSD when working with a few inputs.[4]

5 Compare with Other Sorting Algorithms

There are some other sort algorithms may have good performance on sorting String, such as Dual pivot Quicksort, Timsort, and Husky sort.

Dual pivot Quicksort is a little bit faster than Quicksort which time performance is about 1.39NlnN. About Timsort, in Best Case the time complexity is O(N), and in worst case is O(NlogN). The time complexity of Huskysort is O(NlogN).

MSD radix sort, LSD radix sort and Timsort is stable, Dual Pivot Quicksort is not stable, Huskysort is not stable unless using the merge sort variation [5].

We get experimental data, as shown in Table 1. Sort times for MSD, LSD, dual-pivot quicksort, Timsort and Huskysort. It can be found that Huskysort is the fasted one. MSD radix sort and Dual Pivot Quicksort are also performed well, but when data size is big, MSD radix sort is better.

Table 1. Sort times for MSD, LSD, dual-pivot quicksort, Timsort and Huskysort

Size/Time	MSD	LSD	Quick_Dual	Tim	Husky
250k	392.865875	543.119088	382.740971	398.757271	375.495275
500k	815.630338	1041.99552	798.450225	830.409767	778.58295
1M	1681.25228	2081.67405	1630.39014	1759.8964	1604.40378
2M	3288.14192	4564.3655	3466.55106	3740.55863	3401.49164
4M	7891.54064	10954.4722	7973.06744	8977.23849	7653.35618

Conclusion

In this paper, we use some different algorithms to sort Chinese String. First, we make some conversions from Chinese character to pinyin that make Chinese String can be sorted. And then we compare these different algorithms by counting sorting time with Benchmark, the result shows that Huskysort is the fastest sort algorithm when sorting Chinese Strings, but it is unstable. MSD radix sort also performs well, and it is stable which makes it easy to convert pinyin back to Chinese Strings. In a word, MSD radix sort is a good algorithm to sort Chinese Strings.

References

[1] *Overview of PINYIN4J*. pinyin4j website. (n.d.). Retrieved December 1, 2021, from http://pinyin4j.sourceforge.net/.

[2] Wikimedia Foundation. (2021, December 1). *Radix sort*. Wikipedia. Retrieved December 1, 2021, from https://en.wikipedia.org/wiki/Radix_sort.

[3] Haijun Zhang, Shumin Shi. (2013, August 17). An Efficient Algorithm of Chinese String Sort in User-defined Sequence. Retrieved December 1, 2021, from https://ieeexplore.ieee.org/document/6646048 .

[4] *MSD (most significant digit) radix sort*. GeeksforGeeks. (2021, August 11). Retrieved December 1, 2021, from https://www.geeksforgeeks.org/msd-most-significant-digit-radix-sort/.

[5] Hillyard, R. C., Liaozheng, Y., & R, S. V. K. (2020, December 1). *Huskysort*. arXiv.org. Retrieved December 1, 2021, from https://arxiv.org/abs/2012.00866.