Building and In-Place Checking Suffix Array in External Memory

Yi Wu, Ge Nong, Wai Hong Chan, and Bin Lao

Abstract—The induced sorting (IS) method has been used to design disk-based algorithms for suffix array (SA) construction. A recent engineering of these algorithms can achieve nearly optimal disk space and has a higher scalability over the state-of-the-art suffix sorters in terms of time and I/O complexities. In this paper, we propose a checking method that enables any IS suffix sorting algorithms to verify an SA when it is being built. For performance analysis, we integrate the checking method into the IS suffix sorting algorithm DSA-IS and implement the adapted algorithm to investigate the checking overhead. The experimental results demonstrate that the time, space and I/O consmputions for verification is negligible in comparison with that for construction, indicating that the combination of the proposed checking and the IS building methods can constitute an efficient solution for the situations where building and checking must be done simultaneously.

Index Terms—Suffix array, in-place verification, external memory.

1 Introduction

For any text drawn from a constant or integer alphabet, its suffix array can be built on RAM in linear time and space by the SA-IS algorithm [1], which adopts the IS method to induce the lexical order of unsorted suffixes from those already sorted. In the past five years, the IS method has been reused to design three disk-based suffix sorting algorithms eSAIS [2], DSA-IS [3] and SAIS-PQ [4] that employ different approaches to retrieve the heading characters of unsorted suffixes by efficient I/O operations. These algorithms are competitive with the existing alternatives in terms of time and I/O complexities, but their naive implementations consume around 20 times the input size, leading to a performance bottleneck. A recent work in [**?**] presented a careful engineering of these algorithms, which takes less than $8n$ disk space for an input consisting of $n \leq 2^{40}$ characters and runs faster than the state-of-the-art suffix sorter pSAscan [5] when n is considerably larger than the available RAM space. This reveals that the IS method can serve as a basis for potentially fastest suffix sorting solutions on EM models.

An SA should be verified before use to detect potential computation errors caused by hardware malfunctions or implementation bugs. Currently, the existing SA checker performs two runs of integer sorts and runs slowly in external memory. We propose in this paper a lightweight checking method that enables any disk-based IS suffix sorting algorithms to verify an SA when it is being built. From our experiments, the time, space and I/O volume for verification is negligible in comparison with that for construction.

The rest of this paper is organized as following. We first introduce the preliminaries for this paper in Section 2 and describe the idea of the proposed checking method in Section 3. Then, we demonstrate our experimental results in Section 4 and conclude the remarks in Section **??**.

2 Preliminaries

3 Checking Idea

4 Experiments

5 Conclusion

References

[1] G. Nong, S. Zhang, and W. H. Chan, "Two Efficient Algorithms for Linear Time Suffix Array Construction," *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1471–1484, October 2011.

[2] T. Bingmann, J. Fischer, and V. Osipov, "Inducing Suffix and LCP Arrays in External Memory," in *Proceedings of the 15th Workshop on Algorithm Engineering and Experiments*, 2012, pp. 88–102.

[3] G. Nong, W. H. Chan, S. Q. Hu, and Y. Wu, "Induced Sorting Suffixes in External Memory," *ACM Transactions on Information Systems*, vol. 33, no. 3, pp. 12:1–12:15, March 2015.

[4] W. J. Liu, G. Nong, W. H. Chan, and Y. Wu, "Induced Sorting Suffixes in External Memory with Better Design and Less Space," in *Proceedings of the 22nd International Symposium on String Processing and Information Retrieval*, London, UK, September 2015, pp. 83–94.

[5] J. Kärkkäinen, D. Kempa, and S. J. Puglisi, "Parallel External Memory Suffix Sorting," in *In proceedings of the 26th Annual Symposium on Combinatorial Pattern Matching*, Ischia Island, Italy, July 2015, pp. 329–342.

- *Y. Wu, G. Nong (corresponding author) and B. Lao are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China. E-mails: wu.yi.christian@gmail.com, issng@mail.sysu.edu.cn, Laobin@mail3.sysu.edu.cn.*
- *Wai Hong Chan (corresponding author) is with the Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong. E-mail: waihchan@ied.edu.hk.*