

August 2, 2021

## Review of “Complexity of Dynamic Data Structures” (thesis by Karel Král)

This thesis is organized around the theme of the Integer Sorting problem, whose study has long been of major importance for computer science. Chapter 1 provides historical background and literature review (not exhaustive of this huge area, but sufficient for present purposes).

There then follow three chapters giving original research contributions of different kinds. In Chapter 2, a new construction is given of Sorting circuits for lists of “short” integers (of sufficiently small bitlength  $m \ll \log n$  compared to the length,  $n$ , of the integer sequence presented). Such an input list necessarily contains many repetitions, and so the task seems morally similar to sorting a smaller list; it is shown here how to efficiently carry out the task using a variety of natural subroutines including ones for counting, compressing, and decompressing relevant information, as well as previously-constructed efficient sorting networks. This chapter improves on recent work of Asharov et al [2021], using a different approach.

Chapter 3 studies the problem of Oblivious Random-Access Memory (ORAM), whose study was pioneered by Goldreich and Ostrovsky. Informally, an ORAM is a special protocol for memory access where, to guard against certain types of eavesdropping, the access pattern of the memory locations should not reveal anything about the data being processed or, in certain scenarios, about the algorithm being run. There are simple but inefficient (slow) ways to do this, and a research focus is understanding how much overhead is required to convert general Random-Access programs/instruction sequences into oblivious ones.

The relevance of this subject to Sorting is (in part) as follows: first, comparison-based sorting networks are an important example of a class of input-oblivious algorithm. For this model, closely matching upper and lower bounds are known for Sorting, but for more general models which treat data in input-oblivious ways (e.g. Boolean circuits of restricted depth), the complexity of Sorting is in fact still open and appears beyond current techniques to resolve. Also, Boyle and Naor have recently shown that sufficiently strong lower bounds on the overhead required for so-called “offline” ORAM, would imply breakthrough lower bounds for such Sorting circuits.

Larsen and Nielsen [LN18] made recent strong progress on lower bounds for “online” ORAM using the so-called “information transfer” technique. In Chapter 3, a similar

lower bound is proved “in a relaxed model without any restriction on the format of the access sequence to server memory.” Basically, the limitation exposed in the earlier work is shown to apply more broadly. The techniques used build on [LN18] but make meaningful adaptations to the broader setting, and an interesting combinatorial study is made of so-called “access graphs” associated with ORAM computation; a natural kind of graph “richness” is identified, shown to hold in these graphs, and shown to require a large number of edges in the graph, leading to a lower bound. This is good work in a grand tradition relating graph structure to computational properties (a theme going back at least to Valiant’s seminal work in the 1970s). In slightly more detail, the work in this chapter actually draws a new distinction between weak and strong forms of ORAM security, providing a distinct analysis and lower bound for each; and also points out (for the first time) a serious flaw in a previously-published definition of ORAM security.

In Chapter 4, a new randomized algorithm for Sorting of  $n$  integers in the ordinary RAM model of computation is presented. This is a powerful model which can “beat” the lower bounds which obtain for comparison-based sorting networks. The algorithm given here is simple, slick, and achieves  $O(n \log \log n)$  runtime in expectation. Roughly, and oversimplifying, it’s based on the idea of sorting a random subsample of the list, using this subsample as a “backbone” around which to organize the whole list as a sequence of small buckets, and sorting the buckets individually. Cool! This is not a space-efficient algorithm, nor is it the fastest-known such algorithm;  $O(n \sqrt{\log \log n})$  has been achieved as the result of a sequence of earlier papers, but the present algorithm is significantly simpler to analyze.

I feel these are each nice additions to the Sorting literature and collectively form a good contribution. The mathematical arguments are well-conceived at a high level and laid out clearly for the reader; while I have not checked every detail, they appear totally solid. The writing is acceptable; some small comments below.

Sorting is of course very broadly applicable as a subroutine and, as discussed, its study also connects to issues in graph theory, information security, and frontier questions in circuit complexity. Finally, this thesis does indeed demonstrate the author’s ability to do creative scientific work. It is definitely a sufficient basis for awarding a Ph.D. in Computer Science.

\*\*\*\*\*

Small comments to the author (these do not change the fact that the thesis is acceptable in its present form):

p. 6 “As proven by [REF] [,] pending some [better: “barring an”] unexpected breakthrough [in what?]... this size seems” I would explain this point more directly and immediately

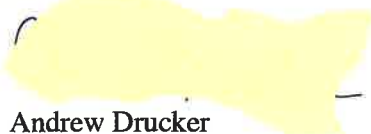
p. 7, examples of small grammatical or writing improvements that can (time allowing) be made in various places:

“solve even a more general” —> “solve an even more general”, “they use [a] substantially different approach”, comma after “However” or “On the other hand”. I recommend using dashes to make sentences easier to parse when you are using a phrase as an adjective. For example, “balls-and-bins lower bounds”, “comparison-based sorting”, “three-tape Turing machine”

There are occasionally citations made awkward by the format in which they appear, e.g. p. 10: “Since our paper Hubacek et al. [2019] there have been papers showing lower bounds for different ranges of parameters Komargodski and Lin [2020], for multi- server Larsen et al. [2020] setting, and many more.” This could be fixed e.g. with parentheses or with a more flexible presentation format.

\*\*\*\*\*

Sincerely

A yellow rectangular redaction box covers the signature of Andrew Drucker.

Andrew Drucker

Assistant Prof. of Computer Science  
University of Chicago