Algorithmic Engineering (PhD reading course, 7.5hp)

Course Description:

The Random Access Machine (RAM) model is the standard model of computation used in most algorithm analysis. It assumes that every memory access takes O(1) time, regardless of data size or memory location. This assumption works well for algorithms that fit entirely in main memory (RAM). However, when data sets exceed available memory, the O(1) access assumption becomes unrealistic. The time spent transferring data between fast main memory and slower storage (e.g., disk or SSD) dominates the total runtime. The External Memory (EM) model extends the RAM model by explicitly accounting for the cost of moving data between internal memory and external storage. Data is transferred in blocks of size B, and the algorithm's efficiency is analyzed in terms of the number of I/O operations rather than CPU steps. This course explores classical algorithmic problems through the lens of both theoretical models and practical performance. We will analyze the I/O complexity of algorithms under the EM model and discuss algorithm engineering techniques that bridge the gap between asymptotic theory and real-world efficiency on large-scale data.

Course coordinator: Kristoffer Sahlin (ksahlin@math.su.se)

Meeting plan: Course pace 25%. Course period covers VT 2026 (Weeks 5-19). Meetings are once a week. Each meeting lasts 2 to 3 hours.

Learning goals:
1. Understand the external-memory (EM) model, and be able to reason around an algorithm's I/O complexity under the EM model.
2. Learn a set of advanced algorithmic problems and their solutions, focusing on I/O complexity. For a more detailed list, see the preliminary course plan below.
3. Present and summarize in a clear and structured way the acquired knowledge by delivering lecture-style presentations of the reading material.

Prerequisites:
To qualify for the course, you should have a basic knowledge in discrete mathematics, probability theory, and statistics, as well as intermediate knowledge of algorithms, data structures, and complexity analysis.

This means knowledge roughly equivalent to Mathematics I, 30 hp (MM2001), Programming techniques, 7.5 hp (DA2005), Probability theory 1, 7.5 hp (MT3001), Statistical analysis, 7.5 hp (MT4001), Computer Science for Mathematicians, 7.5 hp (DA3018), Algorithms and Complexity, 7.5 hp (DA4005), Data structures and algorithms, 7.5 hp (DA4006).

Assessments:

1. Hold formal presentations based on the material presented in the book.
2. Prepare questions/insights/remarks on the material and presentation when not presenting.
3. Write a short 1-2 page proposal for how at least one of the EM versions of an algorithm or a data structure presented in the course could help your research.

Literature:

1. Paper: Vitter, J.S. (2001). "External Memory Algorithms and Data Structures: Dealing with Massive Data." *ACM Computing Surveys, 33(2):209–271.* https://dl.acm.org/doi/10.1145/384192.384193
2. Book: Ferragina, P. (2023). *Pearls of Algorithm Engineering*. Cambridge University Press. https://doi.org/10.1017/9781009128933

Preliminary Plan:

1. Paper 1 (up to, but not including, section 2.3), Book chapters 1 and 2: Introduction to the RAM and EM models and a warm-up problem.
2. Chapter 3: Random Sampling
3. Chapter 4: List ranking
4. Chapter 5: Sorting atomic items (Only sections 5.1-5.3)
5. Chapter 6: Set intersection
6. Chapter 7: Sorting strings (RadixSort, Multi-key QuickSort)
7. Chapter 8: Hashing and hash tables (Sections 8.1-8.4, direct-address and hash tables, Universal hashing, Perfect hashing)
8. Chapter 8: Hashing and hash tables (Sections 8.5-8.7, Cuckoo Hashing, MPHF, Bloom filters)
9. Chapter 9: Searching strings by prefix (pointer arrays, interpolation search, Tries)
10. Chapter 10: Searching strings by substring (Suffix arrays, suffix trees)
11. Chapter 11: Integer coding (Elias, Rice, PForDelta, Variable-byte, (s,c)-dense, interpolative, and Elias-Fano codes).
12. Chapter 12: Statistical coding (Only sections 12.1-12.2; Huffman and arithmetic coding)
13. Chapter 13: Dictionary-based compressors: (LZ77, LZ78, LZW, optimality)
14. Chapter 14: Block sorting (Only sections 14.1-14.3; Burrows-Wheeler and other transforms, the bzip compressor)
15. Chapter 15: Compressed representation of binary arrays, Trees, and Graphs.