Essay in Advanced Algorithmics course (2021)

Choose and read a paper (**1 out of 7**), write a concise summary of this paper. You can read more about summarising research papers here or here.

Make it exactly 2 pages - no exceptions! It can be 30 lines short, but not a single line longer...

Upload as a PDF only. No Word, RTF, etc.

Make absolutely clear in the abstract and text that it is **an overview of the published article**(s), citing all relevant papers. *Add enough relevant citations from the article (probably 3-5) to the most important other articles that are cited there*.

Add some illustration(s). They are worth thousands of words.

The essay has a **title**, **author** (you), author **affiliation** (Institute of Computer Science, University of Tartu, …), **abstract**, **introduction**, **body** (with subsections), **conclusions and references**. Acknowledge your funding.

Use a 2-column layout, this is much easier to read. I would **strongly recommend LaTeX styles**. They are nice, you do not need to worry about layout too much (although you may if you want to procrastinate). And as programmers, you are familiar with "compiling" code into the end product. If you plan to use LaTeX, there are **very convenient online tools** like **overleaf**, for which you don't have to install anything. You can just create a document online and even import the necessary layout (2-column) with minimal effort.

Avoid long sentences. Be concise. In the abstract there are usually no references; it has to be readable alone.

Citing: use numeric [2,3], author name (Kurzweil, 1979), or capitals [KTU76], [Kur76] styles. Citation is always **part of a sentence**, not outside of it [2]. And usually, sentences must be readable without citations as well. Not like: "In [1] new method was proposed", but rather "A new method was proposed [1], that ...".

Write to your peers. We will introduce a peer-review by co-students. **To be decided - how exactly...**

New Essay Topics

1) **The Case for Learned Index Structures**
   https://dl.acm.org/doi/10.1145/3183713.3196909

2) **Finding frequent items in data streams**
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.695&rep=rep1&type=pdf

3) **Heavy hitters via cluster-preserving clustering**
https://arxiv.org/abs/1604.01357

4) **Solving Sparse Linear Systems Faster than Matrix Multiplication**
https://arxiv.org/abs/2007.10254

5) **PSO+**: A new particle swarm optimization algorithm for constrained problems
https://www.sciencedirect.com/science/article/abs/pii/S1568494619306465

6) **Faster integer multiplication**
https://web.archive.org/web/20130425232048/http://www.cse.psu.edu/~furer/Papers/mult.pdf

7) **Loglog Counting of Large Cardinalities**
http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.2718

Previous Year's Topics

Choose one of the following articles: (2020)
(article PDF-s are here -
https://drive.google.com/drive/folders/1HdNDclaDZwp2a8gdwBEbjl6gDHFnitv7?usp=sharing)

Previous 2020 articles:

01: Roaring bitmaps: https://arxiv.org/abs/1709.07821 (from http://roaringbitmap.org/publications/)

02: HOT: A **Height Optimized Trie Index** for Main-Memory Database Systems
https://dl.acm.org/doi/abs/10.1145/3183713.3196896
https://dl.acm.org/doi/pdf/10.1145/3183713.3196896

03: Bloom Filter Trie: an alignment-free and reference-free data structure for pan-genome storage
https://almob.biomedcentral.com/articles/10.1186/s13015-016-0066-8

04: BB-Tree: A practical and efficient main-memory index structure for multidimensional workloads.
https://ieeexplore.ieee.org/abstract/document/8731440
https://www.informatik.hu-berlin.de/de/forschung/gebiete/wbi/research/publications/2019/edbt19.pdf

05: Theoretically-Efficient and Practical Parallel In-Place Radix Sorting
https://dl.acm.org/doi/abs/10.1145/3323165.3323198
https://dl.acm.org/doi/pdf/10.1145/3323165.3323198

06: Multidimensional segment trees can do range queries and updates in logarithmic time
https://arxiv.org/abs/1811.01226
https://cp-algorithms.com/data_structures/segment_tree.html

07: The Splay-List: A Distribution-Adaptive Concurrent Skip-List
https://arxiv.org/abs/2008.01009
https://deepai.org/publication/the-splay-list-a-distribution-adaptive-concurrent-skip-list