



BSc thesis

Batchelor's Programme in Computer Science

Tutkielman otsikko

Jussi Timonen

February 2, 2020

FACULTY OF SCIENCE
UNIVERSITY OF HELSINKI

Supervisor(s)

Prof. D.U. Mind, Dr. O. Why

Examiner(s)

Prof. D.U. Mind, Dr. O. Why

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki, Finland

Email address: info@cs.helsinki.fi

URL: <http://www.cs.helsinki.fi/>

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Faculty of Science		Batchelor's Programme in Computer Science	
Tekijä — Författare — Author			
Jussi Timonen			
Työn nimi — Arbetets titel — Title			
Tutkielman otsikko			
Ohjaajat — Handledare — Supervisors			
Prof. D.U. Mind, Dr. O. Why			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
BSc thesis	February 2, 2020	13 pages, 3 appendice pages	
Tiivistelmä — Referat — Abstract			
<p>Tämä dokumentti on tarkoitettu Helsingin yliopiston tietojenkäsittelytieteen osaston opinnäytteiden ja harjoitustöiden ulkoasun ohjeeksi ja mallipohjaksi. Ohje soveltuu kandidutkielmiin, ohjelmistotuotantoprojekteihin, seminaareihin ja maisterintutkielmiin. Tämän ohjeen lisäksi on seurattava niitä ohjeita, jotka opastavat valitsemaan kuhunkin osioon tieteellisesti kiinnostavaa, syvällisesti pohdittua sisältöä.</p> <p>Työn aihe luokitellaan ACM Computing Classification System (CCS) mukaisesti, ks. https://www.acm.org/about-acm/class, käyttäen komentoa <code>\classification{}</code>. Käytä muutamaa termipolkua (1–3), jotka alkavat juuritermistä ja joissa polun tarkentuvat luokat erotetaan toisistaan oikealle osoittavalla nuolella.</p> <p>ACM Computing Classification System (CCS) General and reference → Document types → Surveys and overviews Applied computing → Document management and text processing → Document management → Text editing</p>			
Avainsanat — Nyckelord — Keywords			
ulkoasu, tiivistelmä, lähdeluettelo			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsinki University Library			
Muita tietoja — övriga uppgifter — Additional information			
Algorithms specialisation line			

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Faculty of Science		Batchelor's Programme in Computer Science	
Tekijä — Författare — Author			
Jussi Timonen			
Työn nimi — Arbetets titel — Title			
Tutkielman otsikko			
Ohjaajat — Handledare — Supervisors			
Prof. D.U. Mind, Dr. O. Why			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
BSc thesis	February 2, 2020	13 pages, 3 appendice pages	
Tiivistelmä — Referat — Abstract			
<p>Use this otherlanguage environment to write your abstract in another language if needed.</p> <p>Topics are classified according to the ACM Computing Classification System (CCS), see https://www.acm.org/about-acm/class: check command <code>\classification{}</code>. A small set of paths (1–3) should be used, starting from any top nodes referred to by the root term CCS leading to the leaf nodes. The elements in the path are separated by right arrow, and emphasis of each element individually can be indicated by the use of bold face for high importance or italics for intermediate level. The combination of individual boldface terms may give the reader additional insight.</p> <p>ACM Computing Classification System (CCS) General and reference → Document types → Surveys and overviews Applied computing → Document management and text processing → Document management → Text editing</p>			
Avainsanat — Nyckelord — Keywords			
ulkoasu, tiivistelmä, lähdeluettelo			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsinki University Library			
Muita tietoja — övriga uppgifter — Additional information			
Algorithms specialisation line			

Contents

1	Introduction	1
2	Basic Concepts	2
3	Previous Work	3
4	Algorithm	4
5	Results	5
6	Conclusion	6
7	Future work	7
7.1	Figures	7
7.2	Tables	7
8	Citations	9
8.1	Citations to literature	9
8.2	Crossreferences	10
9	From tex to pdf	11
10	Conclusions	12
	Bibliography	i
A	Template instructions	
B	Sample Appendix	

1 Introduction

Enormous datasets are a common case in today's applications. Compressing the datasets is beneficial, because they naturally decrease memory requirements but also are faster when compressed data is read from disk (**Zob95**).

One of the leading methods of data compression is variable-length coding (**Sal99**), where frequent sequences of data are represented with shorter codewords. Because the sequences of data have different lengths when compressed, it is not trivial to determine the exact location of a certain element. If this is required, the usual data compression algorithms are inefficient. Fortunately this is not a requirement compression algorithms usually need to fulfill.

However, random access of compressed data is needed in compressed data structures. In most compression methods, the only way to this is to decompress data from the beginning. An integer compressing method with fast random access is explained and compared existing state-of-the-art methods.

2 Basic Concepts

Variable-byte encoding (**Wil99**) is a method of compressing integers. It splits an integer into blocks of b bits and then encodes the block as $b + c$. c is a bit denoting whether b has the most significant bits of the number or not. (example?) A good dataset for variable-byte encoding is a set with both very small and very large numbers, since the compression comes from omitting the leading zeros in the bit array. This method loses in compression performance to other methods (**Bri09**), but decoding is fast.

Rank and select are two functions that work on bit arrays. Rank(i) gives the sum of 1 bits from the beginning of the bit array and select(i) gives the index of i th 1 bit in the bit array. Both functions work in constant time (citation?) and they require only a few percents of extra space over the data. The extra bits c added by variable-byte encoding conveniently create a bit array, where 1's represent the endings of numbers. An effective version of random access has already been introduced (**Bri09**). It uses one rank call for each block after the first which makes the fetching very fast if the number in question fits to one block.

An effective version with select is also possible. The index of the block where i th number starts is obtainable with $\text{select}(i-1)+1$, when $i > 0$ and 0 when $i=0$. By separating the c bit array and b block array, b now contains variable-byte integers in readable form. Another upside is that functions next(i) and previous(i) are conveniently available without any extra calculation and the runtime is not dependent on the size of the value.

- explain how random access is good

3 Previous Work

bl

4 Algorithm

- modifications to basic implementation

5 Results

- comparison to basic implementation + Bri09 -

6 Conclusion

- here

7 Future work

- something to improve / research?

7.1 Figures

Figure 7.1 gives an example how to add figures to the document. Remember always to cite the figure in the main text.



Figure 7.1: University of Helsinki flame-logo for Faculty of Science.

7.2 Tables

Table 7.2 gives an example how to report experimental results. Remember always to cite the table in the main text.

Table 7.1: Results with 100k entries (in milliseconds).

Experiment	128	256	32768	65536	2^{24}	$2^{32} - 1$
<i>7bitVByteencoding</i>	34.97	49	53.04	52.18	53.08	76.21
<i>8bitVByteencoding</i>	33.57	32.47	42.96	43.11	46.15	65.14
<i>7bitVByteencodingwitharray</i>	33.39	46.85	51.24	49.03	48.93	66.84
<i>8bitVByteencodingwitharray</i>	32.52	31.88	41.54	39.94	41.15	52.86

Table 7.2: Results with 1M entries (in milliseconds).

Experiment	128	256	32768	65536	2^{24}	$2^{32} - 1$
<i>7bitV Byteencoding</i>	38.17	55.09	64.38	65.36	68.08	159
<i>8bitV Byteencoding</i>	37.09	37.75	53.44	54.6	59.32	148.7
<i>7bitV Byteencodingwitharray</i>	38.09	55.42	62.22	61.25	71.72	135.01
<i>8bitV Byteencodingwitharray</i>	36.13	36.83	50.58	50.73	56.93	103.18

8 Citations

8.1 Citations to literature

References are listed in a separate .bib-file. In this case it is named `bibliography.bib` including the following content:

```
@article{einstein,
  author =      "Albert Einstein",
  title =       "{Zur Elektrodynamik bewegter K{\o}rper}. ({German})
    [{0n} the electrodynamics of moving bodies]",
  journal =     "Annalen der Physik",
  volume =     "322",
  number =     "10",
  pages =      "891--921",
  year =       "1905",
  DOI =        "http://dx.doi.org/10.1002/andp.19053221004"
}

@book{latexcompanion,
  author      = "Michel Goossens and Frank Mittelbach and Alexander Samarin",
  title       = "The \LaTeX\ Companion",
  year        = "1993",
  publisher   = "Addison-Wesley",
  address     = "Reading, Massachusetts"
}

@misc{knuthwebsite,
  author      = "Donald Knuth",
  title       = "Knuth: Computers and Typesetting",
  url         = "http://www-cs-faculty.stanford.edu/%7Eknuth/abcde.html"
}
```

In the last reference url field the code %7E will translate into ~ once clicked in the final pdf.

References are created using command `\cite{einstein}`, showing as (einstein). Other examples: (latexcompanion; knuthwebsite).

Citation style can be negotiated with the supervisor. See some options in https://www.sharelatex.com/learn/Bibtex_bibliography_styles.

8.2 Crossreferences

Appendix B on page i contains some additional material.

9 From tex to pdf

In Linux, run `pdflatex filename.tex` and `bibtex filename.tex` repeatedly until no more warnings are shown. This process can be automatised using `make-command`.

10 Conclusions

It is good to conclude with a summary of findings. You can also use separate chapter for discussion and future work. These details you can negotiate with your supervisor.

Appendix A Template instructions

In the HY-CS-main.tex file you will find the following STEPS 0–5. Bellow instructions what each STEP means and how to set up your thesis by following these STEPS.

STEP 0 – Clone the thesis template

- One template for all thesis types: <https://www.overleaf.com/read/hzgngkgshqwh>

STEP 1 – BSc or MSc thesis?

1. Select whether your are writing BSc (tkf for new, tkfl for old) or MSc (csm for new, cs for old, dsm for data science) thesis.
2. Select your language: finnish, english, or swedish.
3. If you are writing MSc select your line / track.

STEP 2 – Set up your personal information

1. Write the working title of your thesis.
2. Write your name to the author field.
3. Write the names of your supervisors and examiners of the thesis.

STEP 3 – Write your abstract here

- You can also have the abstract in multiple languages with otherlanguages-environment. Bellow example how to add an english abstract to a thesis written in some other language than english:

```
\begin{otherlanguage}{english}
\begin{abstract}
Your abstract text goes here.
\end{abstract}
\end{otherlanguage}
```

STEP 4 – Writing your thesis

1. There are some writing instructions in `[bsc/msc]-[finnish/english]-contents.tex` files.
2. You can delete the contents of `[bsc/msc]-[finnish/english]-contents.tex` file and write your thesis inside that file.

STEP 5 – Set your bibliography style

- The default is Numbering alphabetic order, which should be used in most cases.

Appendix B Sample Appendix

usually starts on its own page, with the name and number of the appendix at the top. The appendices here are just models of the table of contents and the presentation. Each appendix Each appendix is paginated separately.

In addition to complementing the main document, each appendix is also its own, independent entity. This means that an appendix cannot be just an image or a piece of programming, but the appendix must explain its contents and meaning.