Guidelines for DBL Algorithms reports

This document contains guidelines for the report that has to be written for the course *DBL Algorithms* (2IO90). It serves several purposes. First of all, it describes the structure your report should have: the sections it should contain, a description of the contents of those sections, and their maximum length. Second, it describes which sections should already be included in the intermediate report. Finally, it gives some hints on how to write certain sections. The main part of this document—everything except the current page—is written in the same format your paper should have.

You may use the LaTeX source file of this document as a template for your report. For your convenience I have already added commands for theorems, proof, etcetera, which can be used by writing, for example: `\begin{theorem} ... \end{theorem}`.

About the intermediate and the final report

The report for this DBL project should be written in the form of a research paper. This means that you should write the report as if you want to submit it to a scientific journal. Thus, do not write things like "For the course 2IO90 we had to write a report on labeling maps.". There are several requirements for the intermediate and final reports:

- The reports should be written using 11 pt font, using A4 paper with reasonable (not too large) margins and in single-column format. If you use LaTeX for preparing your paper—this is the preferred way of doing it—then use `\usepackage{a4wide}`.

- The maximum total length of the final report is 20 pages. This includes everything from the title and abstract up to (and including) the references. Do not use appendices. You do not need to use all 20 pages, although if you use significantly less than 20 pages it is likely that your paper is not sufficiently detailed or clear, that your experimental evaluation is not extensive enough, or that your paper could profit from more figures. The minimum number of pages is 16.

- The number of pages for the intermediate report should be at least 8 and at most 11. It should contain Section 1 (as far as possible), Section 2, and the references. You are expected to revise these sections for the final report, based on the feedback you get on the intermediate report and on your findings in the second half of the project.

Title of your DBL Algorithms Paper

S. Tudent1 S. Tudent2 S. Tudent3 S. Tudent4 S. Tudent5

Abstract

In the abstract you give an overview—typically one short paragraph—of the contents of your paper: you describe the problem you have studied, and what the main results are.

1 Introduction

> *Maximum size: 2 pages.*

The introduction usually starts with a description of and a motivation for the general problem area.

> *For the map-labeling problem: Start by describing the map-labeling in general terms, discuss applications and different variants of the problem, and so on.*

After introducing the general problem area, you zoom in to the specific problem studied in the paper. I like to already discuss previous work here. This way you can explain where the specific problem fits into the state-of-the-art and why it is interesting. Ideally, the discussion of the previous work culminates in a clear statement about what is still missing in the current state-of-the-art: namely an answer to the specific problem you study.

> *For the map-labeling problem: Here you state the variants of the problem that you study, and discuss the results (time bounds, theoretical guarantees on the quality of the results, experimental results, if applicable) from the literature on these variants. You may also want to describe in a few lines the approaches that are used in some of the papers.*

Then you give an overview of your results.

> *For the map-labeling problem: Try to give a high-level description of the approach(es) you have used and relate them to approaches found in the literature. State the theoretical guarantees (on running time, for instance, or on other aspects) that you may have proved for your algorithms and mention the main conclusions from the experiments.*

Finally, you can give an overview of the structure of the rest of your paper. Personally, however, I do not find these overviews very useful: I prefer to integrate this with the previous part of the introduction, where the overview of the results is given.

Note: I expect your introduction to contain five to ten references to papers describing related work.

2 The algorithms

> *Maximum size: 8 pages. Use subsections where appropriate. Often it helps to use figures to illustrate definitions and concepts used in the algorithm.*

The description of the algorithms should be such that a programmer can implement them without much difficulty. It is good practice to first explain the main ideas behind the algorithm at a more intuitive level, and then give a detailed description (for example using pseudo-code). Don't forget to describe which supporting data structures you use: linked lists, arrays, search trees, and so on. For standard data structures from the literature you do not need to explain how they work; a reference to the literature suffices.

> *Example: We store the set P of points in a red-black tree [2], using the x-coordinate of each point as its key.*

Try to theoretically analyze the worst-case running time of your algorithms and the amount of storage they use, and argue that it satisfies the requirements. Also try to say something about the quality of your algorithm: you might be able to prove that the algorithm is guaranteed to find an optimal solution, or you might be able to prove that the result of your algorithm is within a certain factor from optimum. (*Note:* If you are using an algorithm from the literature, you should not copy the proofs from the paper describing the algorithm. Instead, give a short sketch that summarizes the main ideas.) It is also useful to show example for which the algorithm does not perform well.

3 Experimental evaluation

> *Maximum size: 8 pages. Use subsections where appropriate.*

Describe the experiments, give the results of the experiments (in the form of tables or graphs), and discuss the results. Here you can also include pictures of your output for a few tests. Relate the outcome of the experiments to your theoretical analysis.

When you investigate running times, you should say on which machine you ran the experiments. You may also want to investigate the running time in a machine-independent manner, for example, by counting the number of calls to some basic procedure. It is important that you describe carefully how you generated the data sets.

> *For the map-labeling problem: Possible topics to investigate include: How does the running time of your algorithm scale when the number of points increases? Compare the number of labels placed by your algorithms for the different placement models. Are the results influenced by the distribution of the input points? How far from optimal are your results? (For the latter you would need to know an optimal solution. For this, you may be able to generate test instances in such a way that the maximum number of labels is known. If your algorithm contains certain parameters: investigate how the values for the parameters influence the results.*

Note that the experimental data in itself are not the main result: the discussion and the conclusions you can draw are what makes the data interesting.

4 Concluding remarks

Maximum size: 1 page.

Give a short overview of the main results, discussing both the strong points of your algorithms as well as their weak points, and ideas for improvements ("future work").

References

[1] S. Albers. On randomized online scheduling. In *Proc. 34th ACM Symp. Theory Comput.*, pages 134–143, 2002.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. *Introduction to Algorithms* (2nd edition). MIT Press, 2001.

[3] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM* 30: 852–865 (1983).

Make sure your references are well-polished and complete. References are typically to books and to papers in scientific journals and conference proceedings. References to web pages are hardly ever appropriate. The list of references is usually ordered alphabetically by first author (although some journals list them in the order they are cited for the first time in the paper). Only put in references that are actually cited in your paper. Formatting can be done as in the example above—note that these references have nothing to do with the topic of the DBL-project—, where the following formatting rules have been used:

- for journals: Authors. Title of paper. *Journal Name (italic)* volume: page numbers (year). See reference [3].
- for conference proceedings: Authors. Title of paper. In *Proc. Conference Name and number (italic)*, pages xxx–yyy, year. See reference [1]
- for books: Authors. *Book title (italic).* Publisher, year. See reference [2]

Note that names of journals and conferences are usually abbreviated. There is a more or less standard way of doing this (for example, *J. ACM* stands for *Journal of the ACM*, but how you do it exactly is less important than that you are consistent and list all the necessary information. Other things to consider:

- Make sure you are consistent in either writing first names in full (Susanne Albers) or only using initials (S. Albers). The latter is preferred. Some people argue that one should write names in the same way as in the original publication. For example, when the title page of reference [1] would say "Susanne Albers", then you should also write the first name in full in your list of references. I find consistency more important, especially since the same author may appear in different ways on different papers. Moreover, writing first names in full can lead to long references, which is why initials are preferred.
- Sometimes the location where a conference took place, or where a publisher is based, is also given in the references. I find this useless.

- In theoretical computer science it is common to use numbers as labels when citing a paper. (In LaTeX, use `\bibliographystyle{plain}`.)

  > *Example: We store the set P of points in a red-black tree [2], using the x-coordinate of each point as its key.*

  In other areas it may be more common to use (author names + year of publication) as a label, as in

  > *Example: We store the set P of points in a red-black tree (Cormen et al. 2001), using the x-coordinate of each point as its key.*

Appendix A

Appendices are not allowed in the report for this DBL project.