UMEÅ UNIVERSITET
Department of Computing Science
Assignment 3

February 20, 2023

Pretitle

5DV149 Datastrukturer och algoritmer

Assignment 3 — Comparison of Table implementations

version 1.0

Name Name
Username CS username

Graders
Graders

Contents

0 Changes

If this is a resubmission, include a list of changes with respect to the previous submission.

1 Introduction

Describe the problem to the reader. Assume that the reader does not know the assignment, what do they need to know? (E.g. data structure interfaces, algorithms, etc.). This section could be split into e.g. Intro + Theory, Intro + Background etc.}

2 Datatypes

What methods did you use? Describe the implementation details for your implementations of the datatype. Especially discuss how you handled duplicates. You may organize this section as below.

2.1 The Table user interface

Present each of the functions in the user interface.

2.2 Implementation details

2.2.1 Table

Description... The implementation uses (other data type)... Duplicates are handled...

2.2.2 MTFTable

2.2.3 ArrayTable

2.3 Complexity analysis

Provide a simplifed asymptotic complexity analysis for *one* call to each function in the user interface for each implementation of the datatype. The reported information will typically be $O(1)$, $O(n)$ or something similar, so a table is a good presentation. See section A.2 for a further discussion on tables.

Use the table to summarize the complexity. Furthermore, discuss in the text things that you want to highlight, e.g. why some values are equal and some or not.

Table 1: A good caption, e.g., asymptotic complexity for the table operations for the different table implementations.

	`table`	`mtftable`	`arraytable`	...
`empty`	$O(1)$			
`isempty`				
`insert`	$O(n)$			
`lookup`				
`remove`				
`choose_key`				
`kill`				
`print`				

3 Experiments

What did you do? How did you set up your experiment? (E.g. explain how you ran all k implementations on the same m problem sizes.) How many times did you repeat? What hardware did you run on? Under what operating system? If you used an external code package for the computations, e.g. Matlab, specify its name and version number. Anything interesting that could affect the results should be mentioned.

3.1 User instructions

Explain how the reader can replicate your experiments. As a start, describe exactly how to compile the source code[1], e.g.

```
gcc -o tabletest -I<path/to/include> <path/to>/table.c tabletest-1.9c
```

3.2 Test runs

If you did any test runs, they should be documented, e.g. by screen dumps. If they are tedious and not important for the general reader, this section may be deferred to the appendix.

3.3 The actual experiments

What commands did you use? What experiments were performed? What was computed? Explain such that the reader can understand the experiments without looking in the source code. "The time was computed for inserting n elements, for n=x, y, ..., z."

[1] on the linux system

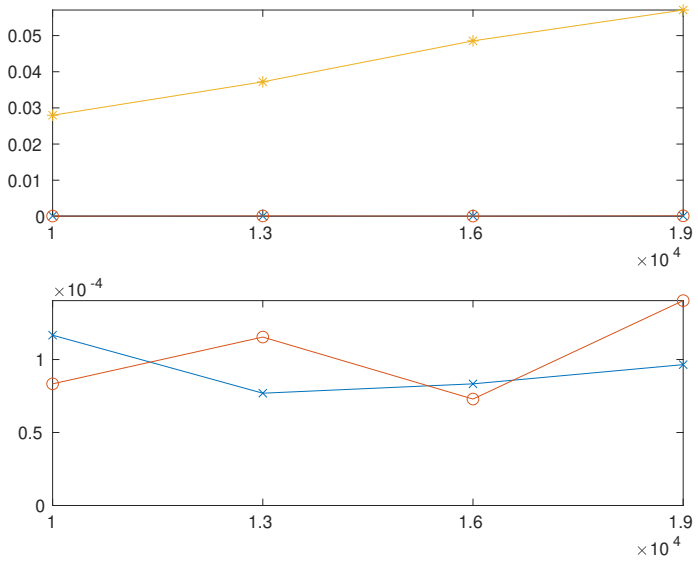

Figure 1: One

4 Results

What did you find? For this assignment, refer to slides 45-49 of lecture note package F06 (Complexity analysis). In particular, **each plot should be of the type at the bottom of page 48**. In the caption and labels, include units, number of runs, how the data was computed (e.g., mean or median of x runs), etc. In a real-world report, raw data would be presented in an appendix.

In particular, generate the following plots:

1. Two version of a plot for the `insert` operation. The first version should include timings for **all** implemented tables using the most common $g(n)$. Thus, some lines may behave different than the others. The second version of the plot should only include the implementations that have the typical $g(n)$. Be sure to anchor the bottom y scale to zero. Figure 1 is a poorly annotated version of such a plot.
2. One plot for the `remove` operation, similar to the one in Figure 2. Be sure to anchor the bottom y scale to zero.
3. One plot with the three `lookup` versions; non-existing, existing, skewed similar to Figure 3.

Help the reader to focus by pointing out interesting observations. However, the results section should be strictly objective, i.e., no evalution of what is, e.g., ”better”. That is to be left for the discussion. It is ok to say that ”on data of type x, method b is fastest”, but not ”method b is better than the others”.

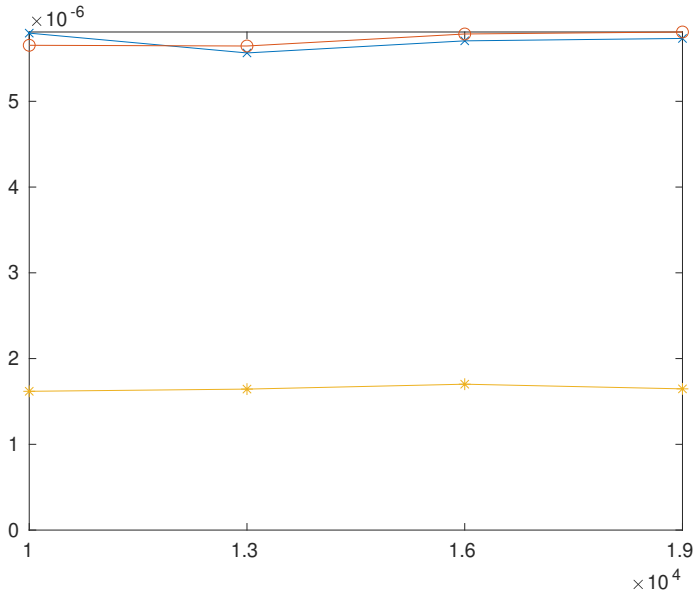

Figure 2: Two

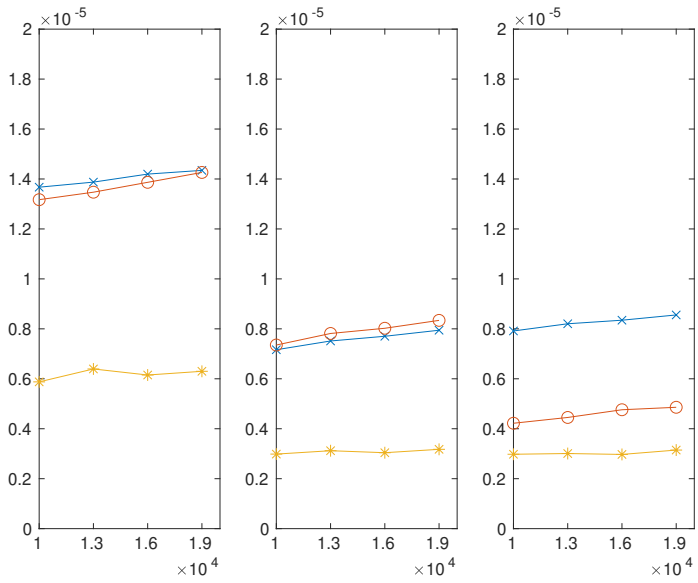

Figure 3: Three

5 Discussion

5.1 Discussion

What does it mean? Discuss whether or not the results make sense. Discuss your methods and their implication on the results (e.g. are any of the tests misleading?). What limitations does your experiment have? Are the results in agreement with expectations, or are any results surprising? Suggest recommendations for future use.

In particular, compare the two list-based implementations. Are the results similar where the theory suggests they should? Are the results different where the theory suggests they should?

Do another comparison between the list-based and array-based implementations.

Do a third comparison between the unsorted and sorted implementations (if working in pairs).

5.2 Conclusion

If your results and recommendations can be briefly summarized, write a short Conclusion section that expresses your key findings.

5.3 Future work (optional)

Did you think of anything interesting to try that you did not have time to include? If yes, this is the place to

5.4 Reflections

Any personal reflections you did during your work with this assignment. Anything that was particularly fun, challenging, poorly specified, surprising, etc.

Figure 4: A figure/image caption should provide sufficient information to make the figure/image as self-explanatory as possible. The caption should be placed under the figure.

A Useful LaTeX examples

Stuff that may be important to some readers, but not all, may be deferred to an appendix. The same is true for lengthy material that would disrupt the flow of the document if placed immediately where it is first referenced. Examples include code listings, file formats, standards, complete tables of all experiments, etc.

A.1 Figures

Figure 4 shows an example of a figure. Exactly *where* (at top or bottom of a page, on a separate page, or "here" in the text) to put figures/tables is a matter of style. The author of this document is of the opinion that "here" should be avoided at all cost. It might seem advantageous to have the figure close to the text that describes it. However, the figure/table should be as self-contained as possible. In general, it should be possible to read and understand the body text *without* having to look at the figure. Thus, if you are forced to write the body text and present the figure such that they will work independently, your report and writing style will benefit.

As the placement of figures and other floats in LaTeX may shift due to changes in text, you are encouraged to leave the fine-tuning of image placement **until your document is complete**.

If you want to annotate figures from Matlab in LaTeX, or generally generate figures to impress your mates, the Matlab package matlab2tikz could be used in conjunction with the LaTeX package PGF/TikZ (page on ikibooks, manual).

A.2 Tables

Tables are often used to present tabulated (no sh*t, sherlock?) data about the experiment setup, test data, etc., or with results of the experiments. In the former case, the body text would typically describe what is common with the data sets and then refer to a table with detailed information. In the latter case,

Table 2: A table caption should provide information that helps the reader to understand what data is in the table. Some additional information, e.g., units can also be part of the caption. A table caption should be placed above the table proper. Use as few borders in the table as possible! For instance, adding left and right borders to the table below would make it harder to read.

Table type	Lookup speed (ms)
MTFTable	x
Arraytable	y
DListTable	z

do not discuss the structure of the table in the body text! That would just confuse the reader. Such information belongs to the caption. In general, do not refer to the table such that the reader cannot continue without inspecting the table. Instead, summarize enough of the content of the table to allow the reader to continue to the next paragraph. Also see Section 4 about how to refer to results.

Data that can better be summarized in the body text should so appear, e.g., "The execution time for experiment x was below 2ms. The other execution times are given in Table x."

In all cases, consider the number of significant digits! Do not put a gazillion decimals in your tables just because your code spits it out! Make the table as easy to read as possible. An example of a stub of a results table is given in Table 2.

A.3 Source code

If you wish to include any source code in this report, you may use the *minted* or *listings* packages. The example below shows *minted*.

```
1 /* Example main */
2 int main(void) {
3     return 0;
4 }
```