Algorithms & Data Structures II (course 1DL231)
Uppsala University – Autumn 2014
Report for Assignment n by Group t

Clara CLEVER Whiz KIDD

25th February 2015

This document shows the ingredients of a good homework report for the AD2 course. The LaTeX source code of this document exemplifies almost everything you need to know about LaTeX in order to typeset a professional-looking assignment report (for the AD2 course). Use it as a starting point for imitation and delete everything irrelevant. The usage of LaTeX is *optional*, but highly recommended, for reasons that will soon become clear to those who have never used it before; any learning time is *outside* the budget of this course, but will hugely pay off, if not in this course then in the next course(s) you take and when writing your BSc, MSc, or PhD thesis.

1 Insertion Sort

Insertion sort is an efficient algorithm for sorting a small number of elements. Insertion sort works the way many people sort a hand of playing cards. We start with a empty left hand and the cards face down on the table. We then remove one card at a time from the table and insert it into the correct position in the left hand. To find the correct position for a card, we compare it with each of the cards already in the hand, from right to left, as illustrated in. At all times, the cards held in the left hand are sorted, and these cards were originally the top cards of the pile on the table.

1.1 Specification and Program

1.2 Analysis

$$T(n) = \begin{cases} \Theta(1) & \text{if } n < 2 \\ T(n-1) + T_{\text{ins}}(n) & \text{if } n \geq 2 \end{cases}$$

Using recurrence (), we get the following time complexity results:

Theorem 1. *The following recurrence, for some* constants *a and b:*

$$T(n) = \begin{cases} \Theta(1) & \textit{if } n < b \\ a \cdot T(n-1) + \Theta(1) & \textit{if } n \geq b \end{cases}$$

has $\Theta(n)$ as closed form for $a = 1$, and $\Theta(a^n)$ as closed form for $a > 1$.

Proof. By induction (left as an exercise to the reader in the AD1 course). □

Intellectual Property

We certify that the material in this report is solely produced by its authors, except where otherwise indicated and clearly referenced.

Checklist before Submitting

In order to protect yourself against an unnecessary loss of points, and in order to show both self-respect and respect for the human reader of your report, please use the following checklist before submitting:

- Crosscheck your report against the homework instructions.
- Crosscheck against the technical writing and LaTeX advice below. The style manual and checklist at `http://www.it.uu.se/research/group/astra/checkList.pdf` offers many further pieces of advice. Common errors in English usage are discussed at `http://public.wsu.edu/~brians/errors/errors.html`. In particular, common errors in English usage by native Swedish speakers are listed at `http://www.cb.uu.se/~cris/English_language.html`.
- Spellcheck all documents, including the comments in the source code.
- Proofread, if not grammar-check, your report at least once per teammate.

2 More LaTeX and Technical Writing Advice

Unnumbered itemisation (only to be used when the order of the items does *not* matter):[1]

- Unnumbered displayed formula:
$$E = m \cdot c^2$$
- Numbered displayed formula (which is normally cross-referenced somewhere):
$$E = m \cdot c^2 \tag{1}$$
- Formula — the same as formula (1) — spanning more than one line:
$$E = m \cdot c^2$$

Numbered itemisation (only to be used when the order of the items *does* matter):

1. First do this.
2. Then do that.
3. If we are not finished, then go back to Step 2, else stop.

The typesetting of tables and elementary mathematics is exemplified in Table 1; see `ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf` for many more details.

Do *not* use programming-language-specific lower-ASCII notation (such as ! for negation, && for conjunction, || for disjunction, and the equality sign = for assignment) in algorithms or formulas (but rather $\neg$ or **not**, $\wedge$ or & or **and**, $\vee$ or **or**, and $\leftarrow$ or $:=$, respectively), as this testifies to a very strong confusion of concepts.

Figures can be imported with `\includegraphics` (such as Figure **??**) or drawn inside the LaTeX source code using the highly declarative notation of the `tikz` package (see Figure 1 for

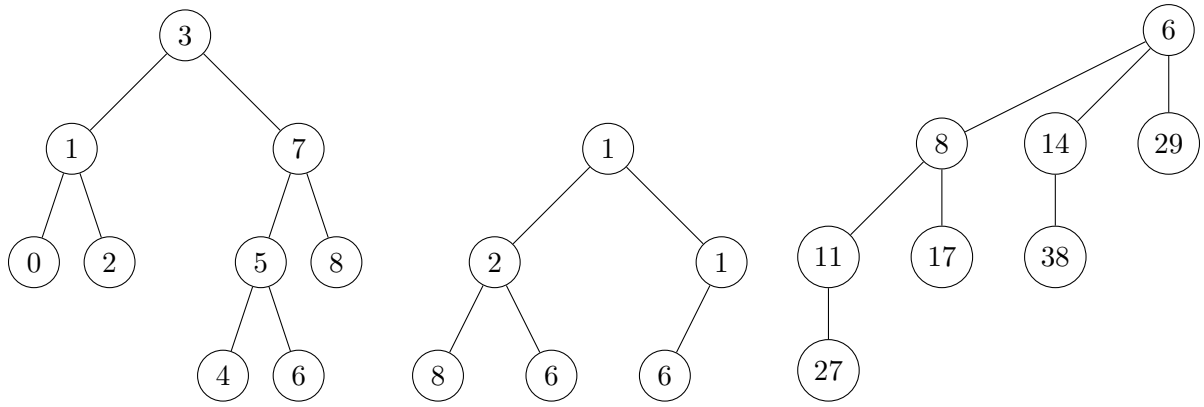

Figure 1: A binary search tree (on the left), a binary min-heap (in the middle), and a binomial tree of rank 3 (on the right).

sample drawings). It is perfectly acceptable in this course to include scans or photos of drawings that were carefully done by hand.

If you are not sure whether you will stick to your current choice of notation, then introduce a new (possibly parametric) command. For instance, upon

```
\newcommand{\Cardinality}[1]{\left\lvert#1\right\rvert}
```

the in-lined formula `$\Cardinality{S}$` typesets the cardinality of set S as $|S|$ with size-adjusted vertical bars and proper spacing, but upon changing the definition of that parametric command to

```
\newcommand{\Cardinality}[1]{\# #1}
```

and recompiling, the formula `$\Cardinality{S}$` typesets the cardinality of set S as $\#S$. You can thus obtain an arbitrary number of changes in the document with a *constant*-time change in its source code, rather than having to perform a *linear*-time find-and-replace operation within the source code, which is painstaking and error-prone. The source code of this document has some useful predefined commands about mathematics and algorithms.

Use commands on positioning (such as `\hspace`, `\vspace`, and `noindent`) and appearance (such as `\small` for reducing the font size, and `\textit` for italics) very sparingly, and ideally only in (parametric) commands, as the very idea of mark-up languages such as LaTeX is to let the class designer (usually a trained professional typesetter) decide on where things appear and how they look. For instance, `\emph` (for emphasis) compiles (outside italicised environments, such as `theorem`) into *italics* under the `article` class used for this document, but it may compile into **boldface** under some other class.

Lite centrerad text bara :))

Note that *no* absolute numbers are used in the LaTeX source code for any of the references inside this document. For ease of maintenance, `\label` is used for giving a label to something that is automatically numbered (such as an algorithm, equation, figure, footnote, item, line, section, subsection, or table), and `\ref` is used for referring to a label. An item in the bibliography file is referred to by `\cite` instead. Upon changing the text, it suffices to recompile, once or twice, and possibly to run BibTeX again, in order to update all references consistently.

[1] Use footnotes very sparingly, and remember that footnote pointers are never preceded by a space and always glued immediately *behind* the punctuation, if there is any.

Prefer `Section~\ref{sect:isort}` over `Section \ref{sect:isort}`, using the non-breaking space (given as ~) instead of the space, giving "Section 1" instead of "Section 1" and thereby avoiding that a cross-reference is spread across a line break, as happened in the previous two lines: this is considered poor typesetting.

The rules of English for how many spaces to use before and after various symbols are given in Table 2. Beware that they may be very different from the rules in your native language.

Topic	LaTeX code	Appearance
Greek letter	`$\Theta,\Omega,\epsilon$`	Θ, Ω, ϵ
multiplication	`$m \cdot n$`	$m \cdot n$
division	`$\frac{m}{n}, m \div n$`	$\frac{m}{n}, m \div n$
rounding down	`$\lfloor n \rfloor$`	$\lfloor n \rfloor$
rounding up	`$\lceil n \rceil$`	$\lceil n \rceil$
binary modulus	`$m \bmod n$`	$m \bmod n$
unary modulus	`$m = n \mod \ell$`	$m = n \mod \ell$
root	`$\sqrt{n},\sqrt[3]{n}$`	$\sqrt{n}, \sqrt[3]{n}$
exponentiation, superscript	`$n^{i}$`	n^{i}
subscript	`$n_{i}$`	n_{i}
overline	`$\overline{n}$`	$\overline{n}$
base 2 logarithm	`$\lg n$`	$\lg n$
base b logarithm	`$\log_b n$`	$\log_b n$
binomial	`$\binom{n}{k}$`	$\binom{n}{k}$
sum	`\[ \sum_{i=1}^n i\]`	$\sum_{i=1}^{n} i$
numeric comparison	`$\leq,<,=,\neq,>,\geq$`	$\leq, <, =, \neq, >, \geq$
non-numeric comparison	`$\prec,\nprec,\preceq,\succeq$`	$\prec, \nprec, \preceq, \succeq$
extremum	`$\min,\max,+\infty,\bot,\top$`	$\min, \max, +\infty, \bot, \top$
function	`$f\colon A\to B,\circ,\mapsto$`	$f\colon A \to B, \circ, \mapsto$
sequence, tuple	`$\langle a,b,c \rangle$`	$\langle a, b, c \rangle$
set	`$\{a,b,c\},\emptyset,\mathbb{N}$`	$\{a, b, c\}, \emptyset, \mathbb{N}$
set membership	`$\in,\not\in$`	$\in, \notin$
set comprehension	`$\{i \mid 1 \leq i \leq n\}$`	$\{i \mid 1 \leq i \leq n\}$
set operation	`$\cup,\cap,\setminus,\times$`	$\cup, \cap, \setminus, \times$
set comparison	`$\subset,\subseteq,\not\supset$`	$\subset, \subseteq, \not\supset$
logic quantifier	`$\forall,\exists,\nexists$`	$\forall, \exists, \nexists$
logic connective	`$\land,\lor,\neg,\Rightarrow$`	$\land, \lor, \neg, \Rightarrow$
logic	`$\models,\equiv,\vdash$`	$\models, \equiv, \vdash$
miscellaneous	`$\&,\#,\approx,\sim,\ell$`	$\&, \#, \approx, \sim, \ell$
dots	`$\ldots,\cdots,\vdots,\ddots$`	$\ldots, \cdots, \vdots, \ddots$
dots (context-sensitive)	`$1,\dots,n; 1+\dots+n$`	$1, \dots, n; 1 + \dots + n$
parentheses (autosizing)	`$\left(m^{n^k}\right),(m^{n^k})$`	$\left(m^{n^k}\right), (m^{n^k})$
identifier of > 1 character	`$\mathit{identifier}$`	*identifier*
hyphen, n-dash, m-dash, minus	`-`, `--`, `---`, `$-$`	-, –, —, −

Table 1: The typesetting of elementary mathematics. Note very carefully when italics are used by LaTeX and when not, as well as all the horizontal and vertical spacing performed by LaTeX.

		number of spaces after	
		0	1
number of spaces before	0	/ -	, : ; . ! ?)] } ’ ” %
	1	([{ ‘ “	– (n-dash) — (m-dash)

Table 2: Spacing rules of English