Prof. Dr. Knut Reinert
Michael Bartel
Paul Pyl

Institut für Informatik
AG Algorithmische Bioinformatik

Freie Universität Berlin

Algorithmen und Datenstrukturen in der Bioinformatik

Zweites Übungsblatt WS 08/09

Abgabe einen Tag vor der Übung 12:00

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer: Ich kann Aufgabe _____ nicht vorrechnen.

Exercise 4: Algorithm illustration
For the following pattern and text:

$$P = \texttt{ababb}$$
$$T = \texttt{abaababbabbb}$$

Illustrate how the Horspool algorithm operates on this data. Give the number of necessary letter comparisons.

Exercise 5: Suffix trees and suffix arrays
Construct (a) the suffix tree, and (b) the suffix array, for the text $T = \texttt{aababbababbaab}$.

---

bitte wenden...

Programming Exercise 1: Exact string matching

a) Implement the Naive, Shift-Or, and Horspool algorithms from the lecture and the exercises. You can work in groups of at most three.

The executable program should be named `aldabism` and accept three arguments on the command line. For the code review, we will invoke your program like this:

```
aldabism algorithm patternfile textfile
```

The algorithm to be used is indicated by its first letter: `N`, `S`, or `H`.

The output should report all occurences (as start positions) and the total number of occurences. Be concise. No debug output in "productive" mode, please. You *may* support additional, optional options *after* the mandatory ones, e.g. for debugging output, like this:

```
aldabism algorithm patternfile textfile bla blabla blablabla
```

b) Evaluate the running times for different combinations of alphabet size, text length, and pattern length. The test data are provided on the web page. Read the README.txt for further instructions, e.g. which combinations of patterns and texts should be tried.

Prepare a report sheet in table format, preferably using or OpenOffice.org Calc (`.odt`, the older `.sxc` is also good) or MS Excel format (`.xls`). It has the following columns: (1) alphabet size, (2) pattern length, (3) text length, (4) running time Naive, (5) running time Shift-Or (6) running time Horspool.

You can measure the running time from within the program (using system calls) or using an external utility such as `time` (on Linux systems). Consider using Linux tools like `bash`, `time`, `grep`, `find`, `sed` to facilitate this task. Then interpret your findings and come up with some recommendations about which algorithm is suitable for which setting.

c) Send your solution via email.

```
To: {pyl,bartel}@inf.fu-berlin.de
Subject: [aldabism] <names of group members>
Attachments: P1.tar.bz2                          (or: P1.tar.gz, or: P1.zip)
```

Uncompressing the attachment shall create the following directory structure:

```
P1-Maier-Meyer-Mayr/
|-- code
|   |-- README
|   |-- main.C
|   |-- main.java
|   |-- what_ever.C
|   |-- what_ever.h
|   `-- what_ever.java
`-- results.odt
```

Please do *not* include *any* *.o, *.class, pattern- or textfiles. Too large emails may be rejected without consideration of their merits. Briefly describe how to build your program in the README file. Place your interpretation of these results in the body of the email.

d) For the code review, prepare printouts of the source code, the report sheet, and your interpretation (email).