



Exercise Sheet 1

complete until Monday, October 23rd

Exercise 1

Compute all occurrences of a three-letter pattern of your choice in the paper *Suffix Arrays: A New Method for On-Line String Searches*, by Udi Manber and Gene Myers, SIAM Journal of Computing 22(5):935–948, 1993. Follow these steps:

1. Download a PDF version of the paper from the seminar homepage: just add `manbermyers.pdf` to the URL in the header of this sheet. Convert the PDF to text with a tool like `pdftotext`, or download the text version directly by writing `.txt` instead of `.pdf` in the URL.
2. Write a program that takes two arguments: a file name and a pattern, and that outputs the set of *distinct* whole words in the file which contain your pattern. For example, a call to your program might look like this:

```
my_program manbermyers.txt ffi
```

and for that pattern the output should consist of the seven words `efficiency`, `efficient`, `efficiently`, `suffices`, `sufficient`, `suffix`, and `suffixes`.

Your program should build a suffix array as discussed in the lecture. For that it should consider the contents of the file as a single long string. The pattern search should be done via the simple binary search explained in the lecture. You will have to spend a little bit of extra thought on how to actually determine the *whole word* at each match.

3. Measure the time for the suffix array construction for your run from 2, and also measure the time for the construction for a *ten times* larger file, obtained by simply concatenating ten copies of the file used in 2.
4. Upload your source code, binary, and the output from 2 to the Wiki. Also enter the programming language you used (the choice is up to you), the two construction times from 3, your pattern and the number of distinct words matching it. Details will be given on the Wiki.

The pattern with the largest number of distinct words matching it wins! However, you may not post a pattern already posted by someone else.