Saarland University
Jun.-Prof. Dr. Tobias Marschall
Dr. Marcel Schulz

5.Assignment - Algorithms for Sequence Analysis, SS 2018

Exercise 1: Local alignment (2 Theory)
Perform the local alignment algorithm (Smith-Waterman) introduced in the lecture for the two sequences $s =$ `CAGCCTCGCTTAG` and $t =$ `AATGCCATTGCCGG`. Use the score function

$$S(a,b) = \begin{cases} +3 & a = b \text{ and } a \neq \text{`-'} \text{ and } b \neq \text{`-'} \\ -3 & a \neq b \text{ and } a \neq \text{`-'} \text{ and } b \neq \text{`-'} \\ -2 & \text{otherwise (i.e. when there are gaps)} \end{cases}$$

Exercise 2: Number of optimal alignments (2 Theory)
Design an algorithm to find the number of optimal global alignments (in terms of edit distance) in $O(n \cdot m)$ time and $O(min(m, n))$ space where m and n are the lengths of the two input sequences. That is, your algorithm should compute how many different optimal alignments exist.

Exercise 3: Semiglobal alignment (3 + 3 = 6 Programming)
Implement a program that uses semi-global alignment with dynamic programming to find all matching positions of a give pattern with edit distance at most k. Your program should take three parameters, the pattern P, the name of an input file with the text T to be searched and the maximum edit distance k.

(a) implements the semiglobal alignment without Ukkonen's trick.

(b) implements the semiglobal alignment with Ukkonen's trick.

The output should be line separated for each hit with (startPos, endPos). Also, you should print the number of computed cells, that is, the number of DP table cells for which your algorithm has computed a value. An example for Part a, the program gets the pattern $P =$ `AATT` as the first parameter to search in the sequence `TAATAAAAAAT$` in the file `input.txt` and $k = 1$.

```
$ ./program AATT input.txt 1
(1,3)
(1,4)
(8,10)
60
```

Remarks:

- There are 10 points to be earned on each assignment sheet.
- 50% of programing points and 50% of theoretical exercises are necessary to take the exam.
- You are allowed to work in groups of two.
- Hand in your solutions on paper (except for source code) by putting it in the letter box of Tobias Marschall in E2.1 (ground floor).
- Programming code is to be sent as a *tar.gz* package by mail to:
    - aryan.3264@gmail.com
- Source code is only considered if
    - it is in one of the languages Python, C, C++, or Java,
    - it is reasonably documented, commented, and readable,
    - command-lines for compiling and calling are provided,
    - compilation does not fail, and
    - executables are named according to:

      `lastname1_lastname2_assignment1_exercise3`
- For each programing task, there will be three input files. One is for you for testing and two are for us for grading. The latter two are kept secret and points are awarded based on whether your program computes the right answer for these two input files.
- Do not forget to mention your names and matriculation numbers on your solutions!
- Copying between groups will result in zero points for all involved groups!

Hand in date: Tuesday, July 3, 2018, 4pm.