Advanced Algorithms in Bioinformatics (P4)

Sequence and Structure Analysis

Freie Universität Berlin, Institut für Informatik
Dr. Clemens Gröpl, Dr. Gunnar Klau, Andreas Döring
Sommersemester 2007

1. Review, 2007-05-30

Name, Vorname	Matrikelnummer

Zur Bearbeitung des Reviews stehen Ihnen 50 Minuten zur Verfügung. Jeder Punkt entspricht in etwa einer Minute.

Geben Sie auf diesem Titelblatt und auf allen eventuell zusätzlich abgegebenen Blättern ihren Namen und ihre Immatrikulationsnummer an.

Schreiben Sie ihre Lösungen direkt auf die entsprechenden Aufgabenbögen. Sollte dort der Platz nicht ausreichen und Sie weitere Blätter benötigen, vermerken Sie dies bitte, damit wir auch den Rest ihrer Antwort finden und bei der Bewertung berücksichtigen können. Am Ende des Reviews sind sämtliche Aufgabenblätter wieder abzugeben.

Ergebnis:

Aufgabe	maximal	erreicht
1	12	
2	12	
3	15	
4	10	
$\sum$	49	

Exercise 1. 3+9=12 Punkte

1. Given an example (actually, a series of examples) of a pattern P and a T such that the Horspool algorithm will run $\Omega(|P||T|)$ time. Argue why.
2. Explain in the form of pseudocode with comments, how the main loop of the Horspool algorithm can be "unrolled". Unrolling means that we can first shift the search window until its last character matches the last character of the pattern and then perform the verification.

Exercise 2. 12 Punkte

In the following, C is a dynamic programming matrix computed for two strings (say, P and T) using the edit distance. Assume that you already know that the following assertions hold for all indices i and j such that $i + j \leq k$, for some given number k.

$$\begin{array}{rrcl} \text{horizontal adjacency property} & \Delta h_{i,j} & = & C_{i,j} - C_{i,j-1} \in \{-1, 0, +1\} \\ \text{vertical adjacency property} & \Delta v_{i,j} & = & C_{i,j} - C_{i-1,j} \in \{-1, 0, +1\} \\ \text{diagonal property} & \Delta d_{i,j} & = & C_{i,j} - C_{i-1,j-1} \in \{0, +1\} \end{array}$$

Now prove that the *diagonal property* also holds for i, j such that $i + j = k + 1$.

Exercise 3. 3+9+3=15 Punkte

In the exercises we have developed an "index" data structure for (ungapped) Q-grams.

1. Explain the tables used and what they contain.
2. Explain how the data structure is initialized. Pseudocode may help, but is not required here.
3. Explain how, given a Q-gram, you can output (in optimal time) the list of all places in the text where it occurs.

Exercise 4. 4+6=10 Punkte

1. Find an example pattern $P = P[1 \,..\, m]$ and text $T = T[1 \,..\, n]$ such that the running time of *mlr*-based search is $\Omega(m \cdot \log n)$. Describe what is 'going wrong' in the *mlr*-based search. No formal proof is required here, but you should point out the principle.

2. Remember the algorithm of Kasai et al. Below you find a correct suffix array (`suftab`) and a would-be height array (`lcptab`) with exactly *one* error. Where is the error in the height array, and why cannot the height array be correct as it is written below? You do not need to know the underlying string in order to answer this question.

i	suftab[i]	lcptab[i]
0	5	-
1	3	2
2	0	2
3	6	1
4	4	0
5	2	2
6	1	1
7	7	0