Prof. Dr. Knut Reinert
Enrico Siragusa
Sascha Meiers
Christoph Hartmann

Institut für Informatik
AG Algorithmische Bioinformatik

Freie Universität Berlin

Algorithmen und Datenstrukturen in der Bioinformatik

Drittes Übungsblatt WS 11/12

Abgabe Montag, 07.11., 15:00 Uhr

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer:

Niveau I

Aufgabe 1: Repeats finden

Ein Teilwort $R = S[i \,..\, i+r-1]$ der Länge r wird als *repeat* bezeichnet, wenn es ein $j \neq i$ gibt mit $R = S[j \,..\, j+r-1]$. Man nennt es *maximales repeat*, wenn zusätzlich $S[i-1] \neq S[j-1]$ und $S[i+r] \neq S[j+r]$. Letztlich nennt man ein repeat mit maximaler Länge *längstes repeat*.

a) Zeigen Sie (am Beispiel) dass ein maximales repeat nicht zwingend ein längstes repeat ist, umgekehrt aber schon!

b) Wie findet man mithilfe des Suffixbaums das längste repeat in S, und in welcher Laufzeit? Es empfiehlt sich eine Traversierungsmethode wie z.B. DFS.

c) Finden Sie alle maximalen repeats in S mithilfe des Suffixbaums in $O(n)$.

Aufgabe 2: Needleman-Wunsch Algorithmus für globale Alignments

Berechnen Sie für die Sequenzen $S =$ `gtac` (horizontal) und $T =$ `agcta` (vertikal) ein globales Alignment nach dem Algorithmus von Needleman-Wunsch.

Ein Match trägt 3 zum Score des Alignments bei, ein Mismatch -5 und ein Space -2.

a) Berechnen Sie die DP-*Matrix*, und

b) tragen Sie dabei jeweils *alle* Traceback-Kanten ein (also eventuell mehrere pro Zelle).

c) Heben Sie *die beim Traceback benutzten* Kanten besonders hervor,und

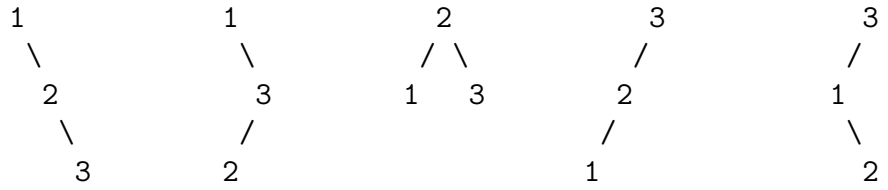
d) notieren Sie das resultierende *globale Alignment*.

Karopapier ... 2. Versuch ...

Niveau II

Aufgabe 3: Dynamische Programmierung

Wie viele verschiedene binäre Suchbäume mit den Schlüsseln $1 \dots n$ gibt es? Für $n = 3$ gibt es beispielsweise 5 verschiedene solcher Suchbäume:



a) Geben Sie eine rekursive Berechnungsvorschrift der Zahl $T(n)$ an. Nutzen Sie $T(0) = T(1) = 1$ als Rekursionsanker.

b) Skizzieren Sie einen Algorithmus, der dynamische Programmierung nutzt um $T(n)$ zu bestimmen.

c) Was ist dessen Laufzeit?

Zur Erinnerung: Ein binärer Suchbaum ist ein echter Binärbaum (jeder Knoten hat genau zwei Kinder oder ist ein Blatt) mit der Eigenschaft, dass alle Elemente im linken Teilbaum eines Knotens x kleiner oder gleich x sind und alle Elemente im rechten Teilbaum gräüer oder gleich x sind.

Tipp zu (a): Jedes der n Elemente kann die Wurzel des Suchbaums sein und durch die Sucheigenschaft werden die restlichen Elemente auf die beiden Teilbäume darunter verteilt.

Programmieraufgabe (Abgabe Montag, 14.11.2010, 15:00)

P-Aufgabe 2:

a) Implement in C++ a naive Suffix Array construction algorithm using `std::sort`.

b) Implement binary search with MLR heuristic.

c) Write a program that:

- loads the text T from an input file;
- accepts a list of patterns $P_0, \ldots, P_k$ from the command line and searches each pattern P_i in T using both Horspool algorithm (from P-Aufgabe 1) and MLR binary search;
- prints to standard output, interleaved by a comma: the name of the corresponding algorithm, the total time in milliseconds (including Suffix Array construction time for MLR) and the number of occurrences in T for each pattern P_i.

The first command line argument indicates the name of the input file. All other following arguments are taken as patterns. Example:

```
beispiel@musterstadt:~$ ./aufgabe2 english.50MB whatever reason anyway
Horspool, 181, 687, 2973, 76
MLR, 127, 687, 2973, 76
beispiel@musterstadt:~$
```

You can use the template at `https://svn.mi.fu-berlin.de/agbio/aldabi/ws11/documents/aufgabe2.cpp` to load an input file. To test your program, download and unpack the input file `http://pizzachili.dcc.uchile.cl/texts/nlang/english.50MB.gz`, then search the patterns *whatever*, *reason*, *anyway*. Remember the Praktikum hints at `https://www.mi.fu-berlin.de/w/ABI/AlDaBiWS11`.