AG Genominformatik, Technische Fakultät, Universität Bielefeld
Grundlagen der Sequenzanalyse WS 2003/04
1. Übungen zur Vorlesung: Ausgabe am 27.10.2003
Klaus-Bernd Schürmann und Jens Stoye

1 Aufgabe 1.1

Zeigen Sie, dass für beliebige Sequenzen u und v die Gleichung $edist_\delta(u, v) = edist_\delta(u^{-1}, v^{-1})$ gilt. Dabei ist u^{-1} die Sequenz u in umgekehrter Reihenfolge, d.h. falls $u = u_1u_2 \dots u_m$, dann ist $u^{-1} = u_m u_{m-1} \dots u_1$.

Zur Erinnerung einige Definitionen, die Sie für diese Aufgabe benötigen:

- Eine Kostenfunktion δ ordnet jeder Edit-Operation $\alpha \to \beta$ Kosten $\delta(\alpha \to \beta) \in \mathbf{R}_0^+$ zu, so daß gilt $\delta(\alpha \to \beta) = 0 \Leftrightarrow \alpha = \beta$.
- δ ist symmetrisch, falls $\delta(\alpha \to \beta) = \delta(\beta \to \alpha)$ für alle Edit-Operationen $\alpha \to \beta$.
- Die Kosten $\delta(A)$ eines Alignments A ist die Summe der Kosten der Edit-Operationen in A.
- $edist_\delta(u, v) := \min\{\delta(A) \mid A \text{ ist ein Alignment von } u \text{ und } v\}$ ist die Edit-Distanz von u und v.

2 Aufgabe 1.2

Seien u und v zwei beliebige Sequenzen der Länge m bzw. n. Für alle $i \in [0, m]$ und $j \in [0, n]$ bezeichne $Aligns(i, j)$ die Anzahl der Alignments von $u_1 \dots u_i$ und $v_1 \dots v_j$. In der Vorlesung wurde die folgende Rekursionsvorschrift für $Aligns$ angegeben:

$$\begin{aligned}
Aligns(0,0) &= 1 \\
Aligns(m,0) &= 1 \\
Aligns(0,n) &= 1 \\
Aligns(m,n) &= Aligns(m-1,n) + Aligns(m,n-1) + Aligns(m-1,n-1), \text{ falls } n, m > 0
\end{aligned}$$

1. Implementieren Sie in einer Programmiersprache Ihrer Wahl ein Programm, das direkt nach obiger Rekursionsvorschrift den Wert $Aligns(m, n)$ berechnet. D.h. schreiben Sie $Aligns$ direkt als rekursive Funktion auf. Die Parameter m und n soll das Programm als Argumente bekommen oder vom Benutzer erfragen. Die maximal möglichen Eingaben für m und n sollen jeweils 15 sein.

2. Berechnen Sie $\mathit{Aligns}(15, 11) = 921406335$. Wieviele Aufrufe der Funktion *Aligns* benötigt Ihr Programm, um diesen Wert zu bestimmen?

3. Implementieren Sie nun ein Programm, dass $\mathit{Aligns}(m, n)$ nach obiger Rekusionsvorschrift mit Hilfe einer $(m + 1) \times (n + 1)$-Matrix *AlignsTab* berechnet. Dabei gilt für alle $(i, j) \in [0, m] \times [0, n]$ die Gleichung

$$\mathit{AlignsTab}[i, j] = \mathit{Aligns}(i, j)$$

4. Berechnen Sie $\mathit{Aligns}(15, 11)$ mit Ihrem zweiten Programm. Wieviele Zugriffe auf die Matrix *AlignsTab* benötigt Ihr Programm, um obigen Wert zu bestimmen?

5. Welches Problem tritt auf, wenn Sie $\mathit{Aligns}(14, 14)$ berechnen wollen? Wie könnte der Benutzer Ihres Programms auf dieses Problem aufmerksam gemacht werden?