



Blatt 11

Aufgabe 11.1 Programmieren Sie den Algorithmus für das optimale globale Alignment von zwei Strings. Die Parameter (Gewichte) der Kostenfunktion $\delta(a, b)$ für zwei Buchstaben a und b bzw. Leerstrings sollen dabei frei wählbar sein und die erhaltene Kostenmatrix $D(i, j)$ soll ausgegeben werden. (3 Punkte)

Testen Sie Ihren Algorithmus am Beispiel aus Aufgabe 10.1. und an einem weiteren Beispiel. Dabei soll sowohl die minimale Edit-Distanz (und die zugehörige Kostenfunktion) als auch ein zugehöriges optimales Alignment der beiden Strings ausgegeben werden (Leerstellen mit „-“ markieren). Zur Ausgabe eines optimalen Alignments muss also eine weitere Funktion programmiert werden, die ein solches aus der Kostenmatrix $D(i, j)$ mit Hilfe von zusätzlich gespeicherten Links von Einträgen von $D(i, j)$ zu den entsprechenden minimierenden Vorgängern extrahiert. Geben Sie die Testergebnisse an! (2 Punkte)

Aufgabe 11.2 Die Kostenmatrix $D(i, j)$ aus dem Algorithmus zum Finden optimaler globaler Alignments von zwei Strings enthält auch ohne zusätzlich abzuspeichernde Links zu den minimierenden Vorgängern alle Informationen über Alignments der beiden Strings. Geben Sie einen effizienten Algorithmus an, der **alle** k optimalen globalen Alignments der beiden Strings ausgibt, dabei nicht auf eventuell zusätzlich gespeicherte Links zurückgreift und Laufzeit $\mathcal{O}(k(n + m))$ benötigt. (4 Punkte)

Aufgabe 11.3 Entwickeln Sie einen „Dynamisches Programmieren“ Algorithmus, der die **Anzahl** aller optimalen globalen Alignments zweier Strings der Längen n und m in Zeit $\mathcal{O}(nm)$ berechnet und beweisen Sie dessen Korrektheit. (3 Punkte)

Abgabe per E-Mail an grau@informatik.uni-halle.de oder in Raum 4.12.