

Übungen zur Vorlesung
Effiziente Algorithmen
SS 2002
Blatt 7

Aufgabe 7.1 (4 Punkte)

Die automatische Rechtschreibprüfung in einem bekannten Textverarbeitungssystem benutzt die *Editierdistanz* zwischen zwei Wörtern $x = (x_1, \dots, x_n)$ und $y = (y_1, \dots, y_m)$, um etwaige Alternativen zum eingegebenen Wort anbieten zu können. Die Definition der Editierdistanz lässt sich am besten anhand derjenigen Operationen verstehen, die in einem typischen Editor ausgeführt werden, um das Wort x in das Wort y zu ändern. Formal benötigen wir aber eine „Cursorposition“ i für das Wort x sowie eine „Cursorposition“ j für das zu erzeugende Wort y und einen zunächst leeren String y' . Anfangs ist $i := j := 1$. Nun darf in jedem Editierschritt eine der vier folgenden Operationen ausgeführt werden.

- Kopieren: Setze $y'_j := x_i$. Erhöhe i und j um jeweils 1.
- Ersetzen: Setze y'_j auf einen gewünschten Buchstaben. Erhöhe i und j um jeweils 1.
- Einfügen: Setze y'_j auf einen gewünschten Buchstaben. Erhöhe j um 1.
- Löschen: Erhöhe i um 1.

Eine gültige vom Wort x zum Wort y führende Folge von Editierschritten muss darüber hinaus die Eigenschaft besitzen, dass i nach deren Abschluss auf dem Wert $n + 1$ steht und $y' = y$ ist. Es ist klar, dass stets m Einfüge- und n Löschoptionen genügen. Abhängig von x und y können aber auch Kopier- und Ersetzoperationen möglich sein.

Angenommen, jede Ersetz-, Einfüge- und Löschoption verursacht 1 Kosteneinheit, während eine Kopieroperation kostenlos ist. Beschreiben Sie einen effizienten Algorithmus, der zu zwei vorgegebenen Strings (x_1, \dots, x_n) und (y_1, \dots, y_m) eine kostenminimale Folge von Editieroperationen angibt, durch die x in y überführt wird.

Aufgabe 7.2 (4 Punkte)

Wir betrachten die im Skript auf S. 39 geschilderte Aufgabe, ein global optimales Alignment im Falle affiner Lückenkosten, d. h., eine Lücke der Länge l in einer der Sequenzen verursacht eine Bewertung von $-c - ld$, zu bestimmen. Erweitern Sie den Hirschberg-Algorithmus dahin gehend, dass er ein global optimales Alignment zweier Strings (x_1, \dots, x_n) und (y_1, \dots, y_m) auch bei affinen Lückenkosten konstruktiv in Zeit $O(nm)$ auf Platz $O(n + m)$ berechnet.

Ausweichtermine für Übungsgruppen, die am 30. Mai nicht stattfinden können:

Gruppen 4 und 6: Mittwoch, 29. Mai, 08:30–10:00 Uhr, GB 5, 324

Gruppe 5: Montag, 27. Mai, 08:15–10:00 Uhr, OH 16, 205



Aufgabe 7.3 (4 Punkte)

In dieser Aufgabe betrachten wir eine Optimierungsvariante des Erfüllbarkeitsproblems der Aussagenlogik, das Problem MAXSAT:

Gegeben: Eine Menge $X = \{x_1, \dots, x_n\}$ von aussagenlogischen (oder booleschen) Variablen und eine Menge $\mathcal{C} = \{C_1, \dots, C_m\}$ von Klauseln in diesen Variablen. Hierbei ist eine Klausel eine Disjunktion von aussagenlogischen Variablen und negierten aussagenlogischen Variablen.

Zielsetzung: Gesucht ist eine Belegung der Variablen in X , unter der maximal viele Klauseln in \mathcal{C} erfüllt werden.

Entwerfen Sie einen Branch-and-Bound-Algorithmus für das Problem MAXSAT.

Aufgabe 7.4 (4 Punkte)

Das Problem MAXCUT ist das folgende Maximierungsproblem:

Gegeben: Ein endlicher (ungerichteter) Graph $G = (V, E)$.

Zielsetzung: Gesucht ist eine Partition der Knotenmenge V in zwei Mengen S und $V - S$ mit der Eigenschaft, dass die Anzahl der Kanten in E , die zwischen Knoten aus S und $V - S$ verlaufen, maximal ist.

Entwerfen Sie einen Branch-and-Bound-Algorithmus für das Problem MAXCUT.