Universität Konstanz
Fachbereich Informatik & Informationswissenschaft
Prof. Dr. Ulrik Brandes / Dr. Andreas Karrenbauer / Dr. Jürgen Lerner / Martin Mader

Algorithmen und Datenstrukturen
WS 2010/2011

11. Übungsblatt

Ausgabe: 12. Dezember 2010 **Abgabe:** 19. Januar 2011, 10 Uhr
Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Aufgabe 1: Ausrichtung von Sequenzen — 5 Punkte

Berechnen Sie eine optimale Ausrichtung der Sequenzen *GAATTCAGTTA* und *GGATCGA* mit Algorithmus 25. Skizzieren Sie die verwendete Matrix D und den Verlauf des Algorithmus.

Aufgabe 2: `KNAPSACK` – dynamische Programmierung — 5 Punkte

Betrachten Sie folgendes Problem (0,1-`KNAPSACK`-Problem):

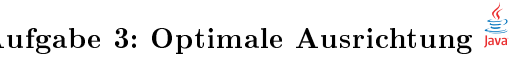
Seien n Objekte gegeben. Jedes Objekt i besitzt ein Gewicht $w_i \in \mathbb{N}$ und einen Wert $v_i \in \mathbb{N}$. Es soll nun eine Teilmenge der Objekte in einen Rucksack gepackt werden, so dass die Summe aller Werte der enthaltenen Objekte maximal ist, die Summe aller Gewichte der Objekte jedoch eine gegebene Schranke W nicht übersteigt. Es wird also nach einem Vektor $x \in \{0,1\}^n$ gesucht, so dass $\sum_{i=1}^{n} v_i x_i$ maximal ist unter der Bedingung $\sum_{i=1}^{n} w_i x_i \leq W$.

Geben Sie einen naheliegenden Algorithmus zur Berechunng des maximalen Wertes $V_{\max} = \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} v_i x_i$ an, dessen Laufzeit unabhängig von W ist (jedoch eventuell eine sehr schnell wachsende Funktion von n). Was ist die asymptotische Laufzeit dieses Algorithmus?

Geben Sie an wie unter Verwendung von dynamischer Programmierung $V_{\max}$ in $\mathcal{O}(n \cdot W)$ berechnet werden kann. Begründen Sie die Korrektheit Ihres Algorithmus.

Tipp: Verwenden Sie eine Matrix deren Zeilen mit $i = 1, \ldots, n$ und deren Spalten mit $w = 1, \ldots, W$ indiziert sind um die Ergebnisse von Teilproblemen zu speichern. Der Eintrag in Zeile i und Spalte w entspricht hierbei der Lösung des 0,1-`KNAPSACK`-Problems gegeben durch die ersten i Objekte und obere Schranke w.

[Bitte wenden]

Aufgabe 3: Optimale Ausrichtung  5 Punkte

Implementieren Sie das Interface `material.u11.IOptimalAlignment` durch eine Klasse `u11.name.OptimalAlignment`, die für zwei Strings eine optimale Ausrichtung und die zugehörige minimale Bewertung mittels dynamischer Programmierung bestimmt. Verwenden Sie als Bewertung die Editierdistanz. Als Platzhalter soll das Symbol "-" verwendet werden.

Aufgabe 3: `KNAPSACK`  5 Punkte

Implementieren Sie das Interface `material.u11.IKnapsack` durch eine Klasse `u11.name.Knapsack`, die das 0,1-`KNAPSACK`-Problem aus Aufgabe 2 löst. Zusätzlich zum maximalen Wert V_{max} soll auch die entsprechende Teilmenge der Objekte, also Vektor x, berechnet werden.

Hinweis: Die Berechnung der enthaltenen Objekte kann aus der Matrix der Optimallösungen der Teilprobleme rekonstruiert werden.