Prof. Dr. Matthias Rarey
M. v. Behren, F. Heitmann, M. Hilbig,
F. Lauck, T. Otto, J. Röwekamp

Zentrum für Bioinformatik Hamburg (ZBH)
Abteilung für Algorithmisches Molekulares Design

Wintersemester 2012/2013

# Übung zur Vorlesung Algorithmen und Datenstrukturen
# Übung 7

Abgabe: 27.1.2013, 12 Uhr

## Aufgabe 1: Optimale Klammerung

Gegeben sind Matrizen mit den folgenden Größen

| Matrix | Größe |
|---|---|
| $A_1$ | $3 \times 8$ |
| $A_2$ | $8 \times 34$ |
| $A_3$ | $34 \times 12$ |
| $A_4$ | $12 \times 7$ |
| $A_5$ | $7 \times 11$ |

Berechnen Sie die minimale Anzahl der benötigten skalaren Rechenoperationen zur Berechnung des Produktes $A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5$ mithilfe von Matrix-Chain-Order und geben Sie die Matrizen $m$ und $s$ an. Geben Sie außerdem die optimale Klammerung an.
(2 Punkte)

## Aufgabe 2: Dynamische Programmierung

Eine Geldwechselmaschine hat Münzen mit den Werten 1, 4 und 5 Billionen ZWR. Um die Maschine möglichst selten mit neuen Münzen nachfüllen zu müssen, wird ein Algorithmus benötigt, der für eine beliebige Geldsumme die minimale Anzahl an Münzen herausgibt.

1. Geben Sie eine Tabelle mit der jeweils minimal benötigten Anzahl von Münzen für die Werte 1 bis 12 Billionen ZWR an. (1 Punkt)

2. Können in der Tabelle bereits berechnete Werte wieder benutzt werden? Geben Sie eine Rekursionsgleichung für die minimale Anzahl $A_W(x)$ von Münzen an, in die der Geldbetrag $x$ in Münzen mit den beliebigen aber festen Werten $W = \{w_1, \ldots, w_n\}$ gewechselt wird. Setzen Sie diese Rekusionsgleichung in einen dynamischen Programmier-Algorithmus in Pseudocode um, der für einen gegebenen Betrag $x$ die minimale Anzahl an Münzen berechnet und bestimmen Sie dessen Laufzeit. (3 Punkte)

## Aufgabe 3: NP

1. Zeigen Sie, dass das Entscheidungsproblem

$$\textsc{SubgraphIso} = \left\{ \langle G, H \rangle \;\middle|\; \begin{array}{l} G, H \text{ sind zwei ungerichtete Graphen, so dass } G \\ \text{einen Teilgraphen } G' \text{ besitzt, der zu } H \text{ isomorph ist} \end{array} \right\}$$

   in NP liegt. Geben Sie dazu ein geeignetes Zertifikat und einen Algorithmus zur Verifikation mit polynomieller Laufzeit an. Dabei heissen zwei ungerichtete Graphen $G = \{V_1, E_1\}$ und $H = \{V_2, E_2\}$ isomorph, wenn es eine bijektive Abbildung $\varphi : V_1 \to V_2$ gibt, so dass für alle $u, v \in V_1$ gilt:

$$\{u, v\} \in E_1 \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E_2$$

   (2 Punkte)

2. Zeigen Sie nun, dass SubgraphIso NP-vollständig ist indem Sie Clique auf SubgraphIso reduzieren. (2 Punkte)

## Aufgabe 4: NP Vollständigkeit

1. Zeigen Sie, dass das Entscheidungsproblem

$$\textsc{DistanceTwo} = \left\{ \langle G, k \rangle \;\middle|\; \begin{array}{l} \text{Der Graph } G = (V, E) \text{ enthält eine Knotenmenge } M \\ \text{der Größe } k \text{ mit } i, j \in M \Rightarrow \delta(i, j) \geq 2 \end{array} \right\}$$

   NP vollständig ist. Zeigen Sie dazu zunächst, dass DistanceTwo in NP liegt und reduzieren Sie dann ein geeignetes aus der Vorlesung bekanntes NP-vollständiges Problem auf DistanceTwo.
   (*Hinweis:* $\delta(i, j)$ ist die Länge des kürzesten Pfades von $i$ nach $j$ gemessen in der Anzahl Kanten) (4 Punkte)

2. Einige Spezialfälle sind jedoch nicht NP-vollständig: Geben Sie einen effizienten Algorithmus an, der das DistanceTwo Problem löst, wenn im Graphen $G$ der maximale Knotengrad 2 ist. (2 Punkte)